Big Data Platforms for the Internet of Things

Radu-Ioan Ciobanu, Valentin Cristea, Ciprian Dobre and Florin Pop

Abstract This chapter discusses the challenges, state of the art, and future trends in context-aware environments (infrastructure and services) for the Internet of Things, an open and dynamic environment where new Things can join in at any time, and offer new services or improvements of old services in terms of performance and quality of service. The dynamic behavior is supported by mechanisms for Things publishing, notification, search, and / or retrieval. Self-adaptation is important in this respect. For example, when things are unable to establish direct communication, or when communication should be offloaded to cope with large throughputs, mobile collaboration can be used to facilitate communication through opportunistic networks. These types of networks, formed when mobile devices communicate only using short-range transmission protocols, usually when users are close, can help applications still exchange data. Routes are built dynamically, since each mobile device is acting according to the store-carry-and-forward paradigm. Thus, contacts are seen as opportunities to move data towards the destination. In such networks data dissemination is usually based on a publish/subscribe model. We make a critical analysis of current opportunistic approaches using the elements of a newly defined taxonomy. We review current state-of-the-art work in this area, from an IoT perspective.

1 Introduction

Every day we create 2.5 quintillion bytes of data; so much that 90% of the data in the world today has been created in the last two years alone. This data comes from

Radu-Ioan Ciobanu
University Politehnica of Bucharest, Spl. Independentei 313, Bucharest, Romania, e-mail: radu.ciobanu@cti.pub.ro

Valentin Cristea
University Politehnica of Bucharest, Spl. Independentei 313, Bucharest, Romania, e-mail: valentin.cristea@cs.pub.ro

Ciprian Dobre
University Politehnica of Bucharest, Spl. Independentei 313, Bucharest, Romania, e-mail: ciprian.dobre@cs.pub.ro

Florin Pop
University Politehnica of Bucharest, Spl. Independentei 313, Bucharest, Romania, e-mail: florin.pop@cs.pub.ro

sensors used to gather climate information, from posts to social media sites, digital pictures and videos, purchase transaction records, or cell phone GPS signals, to name only a few. This data is *Big Data*. Analyzing large data sets already underpins new waves of productivity growth, innovation, and consumer surplus. Big data is more than simply a matter of size; it is an opportunity to find insights in new and emerging types of data and content, to make businesses more agile, and to answer questions that were previously considered beyond our reach. Until now, there was no practical way to harvest this opportunity. But today we are witnessing an exponential growth in the volume and detail of data captured by enterprises, the rise of multimedia, social media and Online Social Networks (OSN), and the Internet of Things (IoT).

Many of Big Data challenges are generated by future applications where users and machines will need to collaborate in intelligent ways together. In the near future information will be available all around us, and will be served in the most convenient way - we will be notified automatically when a congestion occurs and the car will be able to decide how to optimize our driving route, the fridge will notify us when we the milk supply is out, etc.. Technology becomes more and more part of our daily life. New technologies have finally reached a stage of development in which they can significantly improve our lives. For example, our cities are fast transforming into artificial ecosystems of interconnected, interdependent intelligent digital "organisms". They are transforming into *smart cities*, as they benefit more and more from intelligent applications designed to drive a sustainable economic development and an incubator of innovation and transformation that merges the virtual world of Mobile Services, IoT and OSN with the physical infrastructures of Smart Building, Smart Utilities (i.e., electricity, heating, water, waste, transportation, and unified communication & collaboration infrastructure). The transformation of the metropolitan landscape is driven by the opportunity to embed intelligence into any component of our towns and connect them in real-time, merging together physical world of objects, humans and virtual conversation and transactions. Town planners and administration bodies just need the right tools at their fingertips to consume all the data points that a town or city generate and then be able to turn that into actions that improve people's lives. Smart Cities of tomorrow will rely not only on sensors within the city infrastructure, but also on a large number of devices that will willingly sense and integrate their data into technological platforms used for introspection into the habits and situations of individuals and city-large communities. Predictions say that cities will generate over 4.1 terabytes per day per square kilometer of urbanized land area by 2016. Handling efficiently such amounts of data is already a challenge.

We have barely begun to get a sense of the dimensions of this kind of data, of the privacy implications, of ways in which we can code it with respect to meaningful attributes in space and time. As we move into an era of unprecedented volumes of data and computing power, the benefits are not for business alone. Data can help citizens' access to government, hold it accountable and build new services to help themselves. In one sense, all this is part of a world that is fast becoming digital in all its dimensions. People will develop more easily their understanding and design

ideas using digital representations and data. This will support the development of the new ideas for the future of urban and social life, weaved together under the umbrella of what is now called the future Internet of Things (IoT).

As part of the Future Internet, IoT aims to integrate, collect information from-, and offer services to a very diverse spectrum of physical things used in different domains. "Things" are everyday objects for which IoT offers a virtual presence on the Internet, allocates a specific identity and virtual address, and adds capabilities to self-organize and communicate with other things without human intervention. To ensure a high quality of services, additional capabilities can be included such as context awareness, autonomy, and reactivity. Things are very diverse. Very simple things, like books, can have Radio Frequency Identification - RFID tags that help tracking them without human intervention. For example, in an electronic commerce system, a RFID sensor network can detect when a thing left the warehouse and can trigger specific actions like inventory update or customer rewarding for buying a high end product [14]. In this simple case, RFIDs enable the automatic identification or things, the capture of their context (for example the location) and the execution of corresponding actions if necessary. Sensors and actuators are used to transform real things into *virtual objects* [43] with digital identities. In this way, things may communicate, interfere and collaborate with each other over the Internet [5]. Adding part of application logic to things transforms them into *smart objects*, which have additional capabilities to sense, log and understand the events occurring in the physical environment, autonomously react to context changes, and intercommunicate with other things and people. A tool endowed with such capabilities could register when and how the workers used it and produce a financial cost figure. Similarly, smart objects used in the e-health domain could continuously monitor the status of a patient and adapt the therapy according to the needs. Smart objects can also be general purpose portable devices like smart phones and tablets, that have processing and storage capabilities, and are endowed with different types of sensors for time, position, temperature, etc. Both specialized and general purpose smart objects have the capability to interact with people.

The IoT includes a hardware, software and services infrastructure for things networking. IoT infrastructure is event-driven and real-time, supporting the context sensing, processing, and exchange with other things and the environment. The infrastructure is very complex due to the huge number (50 to 100 trillion) of heterogeneous, (possibly) mobile things that dynamically join and leave the IoT, generate and consume billions of parallel and simultaneous events geographically distributed all over the world. The complexity is augmented by the difficulty to represent, interpret, process, and predict the diversity of possible contexts. The infrastructure must have important characteristics such as reliability, safety, survivability, security and fault tolerance. Also, it must manage the communication, storage and compute resources.

The main function of the IoT infrastructure is to *support communication* among things (and other entities such as people, applications, etc.). This function must be flexible and adapted to the large variety of things, from simple sensors to sophisticated smart objects. More specific, things need a communication infrastructure

that is low-data-rate, low-power, and low-complexity. Actual solutions are based on short-range radio frequency (RF) transmissions in ad-hoc wireless personal area networks (WPANs). A main concern of the IoT infrastructure developers is supporting heterogeneous things by adopting appropriate standards for the physical and media access control (MAC) layers, and for communication protocols. The protocol and compatible interconnection for the simple wireless connectivity with relaxed throughput (2 - 250 kb/s), low range (up to 100 m), moderate latency (10 - 50 ms) requirements and low cost, adapted to devices previously not connected to the Internet were defined in IEEE 802.15.4. Other similar efforts refer to industrial and vehicular applications - ZigBee, open standards for process control in industrial automation and related applications - ISA100.11a and WirelessHART, and encapsulating IPv6 datagrams in 802.15.4 frames, neighbor discovery and routing that allow sensors to communicate with Internet hosts - 6LoWPAN [6]. The main scope of IoT specialists is the world-wide network of interconnected virtual objects uniquely addressable and communicating through standard protocols. The challenge here is coping with a huge number of (heterogeneous) virtual objects.

The IoT architecture supports physical things' integration in Internet and the complex interaction flow of services triggered by events occurrence. The main concepts involved are the object-identification, sensing and connecting capabilities as the basis for the development of independent cooperative services and applications that address several key features for IoT architecture: Service Orientation, Web-base, distributed processing, easy integration via native XML and SOAP messaging, component-base, open access, N-tiered architecture, support for vertical and horizontal scalability [37]. These features allow "physical objects become active participants in business processes" [39]. As a consequence, the Web Services must be available to interact with the corresponding virtual objects over the Internet, query and change their state and any information associated with them. The new key features for the IoT architecture include persistent messaging for the highest availability, complete security and reliability for total control and compliance, platform independence and interoperability (more specific for middleware).

An IoT infrastructure considers *extended process functionality*, pathways and layered networks as main components. The IoT infrastructure should support object-connected technologies for both "Human-to-Objects" and "Objects-to-Objects" communications. Because of the wide heterogeneity of wireless devices involves, the communication platforms are heterogeneous, ad-hoc, and opportunistic. IoT is a large heterogeneous collection of things, which differ from each other. Even things that have the same nature, construction, and properties can differ from one another by their situation or context. Context means the conditions in which things exist in other words their surrounding world. Since virtual things in IoT are interconnected, the meaning of the data they exchange with other things and people is clearer only if they are interpreted in the thing's context. This is why the IoT infrastructure runs reliably and permanently to provide the context as a "public utility" to IoT services [13]. For human users, the context is the information that characterizes user's interaction with Internet applications plus the location where this interaction occurs, so that the service can be adapted easily to users' preferences, For things, we need an-

other approach. A very suggestive example is given in [24]. The authors explain the case of a plant that is the target of an automatic watering service. In order to control the watering dosages and frequency, the service has to sense the dryness status of the plant, to use the domain knowledge of plants and find their watering "preferences", and to ask the weather prediction service about the chances of rain in the next days. So, the context of a thing includes information about thing's environment and about the thing itself.

Several context modeling and reasoning techniques are known today, some of them being based on knowledge representation and description logics. Ontology-based models can describe complex context data, allow context integration among several sources, and can use reasoning tools to recognize more abstract contexts. Ontologies provide a formal specification of the semantics of context data that stay at the base of knowledge sharing among different things in IoT. In addition, new context can be derived by ontological reasoning. Ontology-based models can be used to organize IoT infrastructure context-aware services as a fabric structured into multiple levels of abstraction (see Figure 1) starting with collecting information from physical sensors (called low level context), which could be meaningless and consequently not useful to applications. Next, higher level context is derived by reasoning and interpretation. Finally, context is exploited by triggering specific actions [13].

The IoT infrastructure combines the context model with event-based organization of services that support the collection, transmission, processing, storage and delivery of context information. In the event-driven architecture vocabulary, events are generated by different sources, event producers, when for example a context change is significant and must be propagated to target applications, event consumers. Producers and consumers are loosely coupled by the asynchronous transmission and reception of events. They don't have to know and explicitly refer each other. In addition, the producers don't know if the transmitted events are consumed ever. A publish/subscribe mechanism is used to offer the events to the interested consumers. Other components are used such as event channels for communication, and event processing engines for complex event detection. To them, components for the event specification, event management, and for the integration of the event-driven system with the application must be added. It is worth noting that events are associated with changes of things' context [14]. There are also non-functional requirements

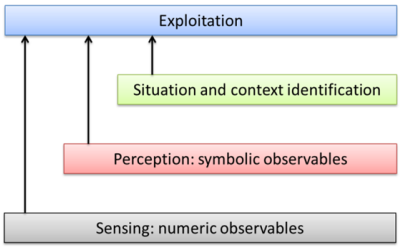

Fig. 1 Context-aware services.

associated with IoT infrastructure [14]: large scale integration, interoperability between heterogeneous things, fault tolerance and network disconnections, mobility, energy saving, reliability, safety, survivability, protection of users' personal information (e.g., location and preferences) against security attacks, QoS and overall performance, scalability, self-* properties and transparency.

In the remainder of this chapter we make a thorough review of methods and techniques designed to support the adaptation of modern context-aware platform towards Big Data infrastructures designed to support the IoT vision. The rest of this chapter is organized as follows. We first make an analysis of context aware infrastructures for IoT. As previously mentioned, the communication support is an important component of any IoT platform, and in Section 3 we make a comprehensive study of opportunistic data dissemination support for the Internet of Things. In Section 4 we present future trends and research directions in Big Data platforms for the Internet of Things. Section 5 presents the conclusions and final remarks.

2 Context-aware infrastructures for the Internet of Things

This section presents up to date solutions and research results regarding the structure, characteristics, and services for context aware Internet of Things infrastructure. A ubiquitous computing environment is characterized by a diverse range of hardware (sensors, user devices, computing infrastructure, etc.) and equally diverse set of applications which anticipate the need of users and act on their behalf in a proactive manner [35]. The vision of ubiquitous computing is only recently becoming a reality in a scale that can be practically distributed to end users.

A context-aware system is generally characterized by several functions. It generally gathers context information available from user interface, pre-specified data or sensors and add it to a repository (*Context Acquisition and Sensing*). Furthermore, the system converts the gathered raw context information into a meaningful context which can be used (*Context Filtering and Modeling*). Finally, the system uses the context to react and make the appropriate context available to the user (*Context Reasoning, Storage and Retrieval*).

Several cross-device context-aware application middleware systems have been developed previously. In their majority these were Web service-based context-aware systems, especially the most recent ones. However, there has been a big variety of middleware systems, developed mainly in the early 2000, that do not rely on Web service technologies and are not designed to work on Web service-based environments [40]. In our analyzed we began by studying several popular context-aware platforms, considering their provided functions and particular characteristics. From the category of non-based on web service context-aware platforms we mention the following. **RCSM** [45] is a middleware supporting context sensitive applications based on an object model: context-sensitive applications are modeled as objects. RCSM supports situation awareness by providing a special language for specifying situation awareness requirements. Based on these requirements, application-specific

object containers for runtime situation analysis will be generated. RCSM runtime system obtains context data from different sources and provides the data to object containers which conduct the situation analysis. The **JCAF** (Java Context Awareness Framework) [2] supports both the infrastructure and the programming framework for developing context-aware applications in Java. Contextual information is handled by separate services to which clients can publish and from which they can retrieve contextual. The communication is based on Java RMI (Remote Method Invocation). An example of application that use Java RMI is MultiCaR: Remote Invocation for Large Scale, Context-Aware Applications [15]. This application also address the issue of Big Data analytics.

The **PACE** middleware [25] provides context and preference management together with a programming toolkit and tools for assisting context-aware applications to store, access, and utilize contextual information managed by the middleware. PACE supports context-aware applications to make decisions based on user preferences. **CAMUS** is an infrastructure for context-aware network-based intelligent robots [30]. It supports various types of context information, such as user, place and environment, and context reasoning. However, this system is not based on Web services and it works in a close environment. **SOCAM** is a middleware for building context-aware services [22]. It supports context modeling and reasoning based on OWL. However, its implementation is based on RMI.

Web service-based context-aware platforms include the following. **CoWSAMI** is a middleware supporting context-awareness in pervasive environments [1]. The **ESCAPE** framework [40] is aWeb services-based context management system for teamwork and disaster management. ESCAPE services are designed for a front-end of mobile devices and the back-end of high end systems. The front-end part includes components support for context sensing and sharing that are based on Web services and are executed in an ad hoc network of mobile devices. The back-end includes a Web service for storing and sharing context information among different front-ends. The **inContext** project [40] provides various techniques for supporting context-awareness in emerging team collaboration. It is designed for Web services-based collaborative working environments. inContext provides techniques for modeling, storing, reasoning, and exchanging context information among Web services.

Being context-aware allows software not only to be able to deal with changes in the environment the software operates in, but also being able to improve the response to the use of the software. That means context-awareness techniques aim at supporting both functional and non-functional software requirements. Authors of [18] identified three important context-awareness behaviors:

1. The representation of available information and services to an end user.
2. The automatic execution of a service.
3. The tagging and storing of context information for later retrieval.

Such platforms are considered by many precursors of the Internet of Things vision, defined as a dynamic network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols [43]. In this vision,

physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network. In the IoT, *things* are expected to become active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information *sensed* about the environment, while reacting autonomously to the *real/physical world* events and influencing it by running processes that trigger actions and create services with or without direct human intervention. Interfaces in the form of services facilitate interactions with these *smart things* over the Internet, query and change their state and any information associated with them, taking into account security and privacy issues.

Sensors in IoT can run anywhere and on any objects. They are used to collect data such as biomedical information, environment temperature, humidity, ambient noise level. This is also a research issue close to Big Data science. The data provided by such sensors can be used by customized context-aware applications and services, capable to adapting their behavior to their running environment. However, sensor data exhibits high complexity (e.g., because of huge volumes and inter-dependency relationships between sources), dynamism (e.g., updates performed in real-time and data that can critical age until it becomes useless), accuracy, precision and timeliness. An IoT system should not concern itself with the individual pieces of sensor data: rather, the information should be interpreted into a higher, domain-relevant concept. For example, sensors might monitor temperature, humidity, while the information needed by a watering actuator might be that the environment is dry. This higher-level concept is called a situation, which is an abstract state of affairs interesting to applications [12].

Situations are generally representations (simple, human understandable) of sensor data. They shield the applications from the complexities of sensor readings, sensor data noise and inferences activities. However, in large-scale systems there may be tens or hundreds of situations that applications need to recognize and respond to. Underlying these situations will be an even greater number of sensors that are used in situation identification. A system has a significant task of defining and managing these situations. This includes capturing what and how situations are to be recognized from which pieces of contexts, and how different situations are related to each other. The system should know, for example, which situations can or cannot occur: a room hosting a "scientific event" and an "academic class" at the same time); otherwise, inappropriate adaptive behavior may occur. Temporal order between situations is also important, such as the inability of a car to go directly from a situation of 'parked' to 'driving on a highway'. Given the inherent inaccuracy of sensor data and the limitations of inference rules, the detection of situations is imperfect.

The research topics on situation identification for IoT involve several issues [47]. First, *representation* deals with how to define logic primitives used to construct a situation's logical specification. In representation, logical primitives should capture features in complex sensor data (e.g., acceleration data), domain knowledge (e.g., a spatial map or social network), and different relationships between situations. Also, an IoT system is assumed to be highly dynamic. New sensors can be introduced, that

introduce new types of context. Therefore, the logical primitives should be flexibly extensive, such as new primitives to not cause modifications or produce ambiguous meanings to existing ones. *Specification* deals with defining the logic behind a particular situation. This can be acquired by experts or learned from training data. It typically relies on a situation model with apriori expert knowledge, on which reasoning is applied based on the input sensor data. For example, in logic programming [8] the key underlying assumption is that knowledge about situations can be modularized or digitized.

Other logic theories, such as situation calculus [46], have also been used to infer situations in IoT systems. Kalyan et al. [28] introduce a multi-level situation theory, where an intermediate level micro situation is introduced between infons and situations. An infon embodies a discrete unit of information for a single entity (e.g., a customer or a product), while a situation makes certain infons factual and thus support facts. Micro situations are composed of these entity-specific infons which can be explicitly obtained from queries or implicitly derived from sensors and reasons. Situations are considered as a hierarchical aggregation of micro situations and situations. This work aims to assist information reuse and support ease of retrieving the right kind of information by providing appropriate abstraction of information.

Ontologies have also increasingly gained attention as a generic, formal and explicit way to capture and specify the domain knowledge with its intrinsic semantics through consensual terminology and formal axioms and constraints. They provide a formal way to represent sensor data, context, and situations into well-structured terminology. Based on the modeled concepts, developers can define logical specifications of situations in rules. An exemplar rule on an activity 'sleeping' is given in [22]:

```
(?user rdf:type socam:Person),
(?user, socam:locatedIn, socam:Bedroom),
(?user, socam:hasPosture, 'LIEDOWN'),
(socam:Bedroom, socam:lightLevel, 'LOW'),
(socam:Bedroom, socam:doorStatus, 'CLOSED')
-> (?user socam:status 'SLEEPING')
```

Ontologies, together with their support for representation formalisms, can support reasoning, including detecting inconsistency, or deriving new knowledge. An ontological reasoner can be used to check consistency in a class hierarchy and consistency between instances, e.g. whether a class is being a subclass of two classes that are declared as disjoint or whether two instances are contradictory to each other (such as a person being detected in two spatially disjoint locations at the same time). Given the current sensor data, the reasoner will derive a new set of statements. In the above 'sleeping' example, if the reasoner is based on a forward-chaining rule engine, it can match the conditions of this rule against the sensor input. If all the conditions are satisfied, the reasoner will infer the conclusion of the rule. The reasoning will terminate if the status of the user is inferred, when the status of the user is set to be the default inference goal in this reasoner.

Other solutions are based on the Dempster-Shafer theory (DST) [16], a mathematical theory of evidence, which propagates uncertainty values and consequently

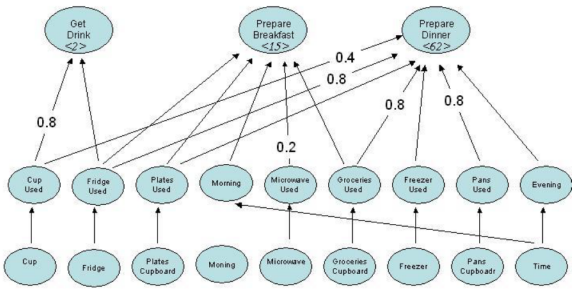

Fig. 2 An example of situation inferring using Dempster-Shafer theory (from [34]).

provides an indication of the certainty of inferences. The process of using DST is described as follows. First, developers apply expert knowledge to construct an evidential network that describes how sensors lead to activities. The left-hand side of Figure 2 describes that the sensors on the cup and fridge are connected to context information (e.g., 'cup used'). Such context information can be further inferred or composed to higher-level context. The composition of context information points to an activity (e.g., 'Get drink') at the top. Developers can use such an approach to determine the evidence space and degree of belief in an evidence. For example, in the figure the values on the arrows represent the belief in particular sensor (also called the uncertainty of sensor observations). Generally, in reasoning situations are inferred from a large amount of imperfect sensor data. In reasoning, one of the main processes is called situation identification - deriving a situation by interpreting or fusing several pieces of context in some way. Specifying and identifying situations can have a large variability depending on factors such as time, location, individual users, and working environments. This makes specification-based approaches relying on models of a priori knowledge impractical to use. Machine learning techniques have been widely applied to learning complex associations between situations and sensor data. However, the performance of reasoning is usually undermined by the complexity of the underlying sensor data.

Bayesian networks and Hidden Markov Models (HMMs) have been applied in many context-aware systems. In HMMs statistical models a system being modeled is assumed to be a Markov chain that is a sequence of events [41]. A HMM is composed of a finite set of hidden states and observations that are generated from states. For example, a HMM where each state represents a single activity (e.g., 'prepare dinner', 'go to bed', 'take shower', and 'leave house') is presented in [41]. They represent observations in three types of characterized sensor data that are generated in each activity, which are raw sensor data, the change of sensor data, the last observed sensor data, and the combination of them. The HMM is trained to obtain the three probability parameters, where the prior probability of an activity represents the likelihood of the user starting from this activity; the state transition probabilities represent the likelihood of the user changing from one activity to another; and the

observation emission probabilities represent the likelihood of the occurrence of a sensor observation when the user is conducting a certain activity.

Finally, Support Vector Machines (SVM) [20] is a relatively new method for classifying both linear and nonlinear data. An SVM uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyper-plane that separates the training data of one class from another. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated. SVMs are good at handling large feature spaces since they employ over fitting protection, which does not necessarily depend on the number of features. Kanda et al [29] use SVMs to categorise motion trajectories (such as 'fast', 'idle', and 'stop') based on the velocity, direction, and shape features extracted from various sensors (within a car for example). Different types of sensor data lead to different techniques to analyze them. Numerical data, for example, can be used to infer motions like 'walking' or 'running' from acceleration data. Situation identification at this level is usually performed in learning-based approaches, which uncover complicated associations (e.g., nonlinear) between continuous numerical data and situations by carving up ranges of numerical data (e.g., decision tree) or finding an appropriate algebraic function to satisfy or 'explain' data (e.g., neural networks or SVMs). Specification-based approaches can apply if the association between sensor data and situations are rather explicit and representable in logic rules. Situations can also be recognized from categorical features; for example, inferring a room's situation - 'meeting' or 'presentation' - from the number of persons co-located in the room and the applications running in the computer installed in the room. This higher-level of situation identification can be performed using either specification- or learning-based approaches.

Uncertainty can also exist in the use of oversimplified rules that are defined in an ad-hoc way. In representing uncertainty of rules, Web Ontology Language (OWL), a family of knowledge representation languages for authoring ontologies endorsed by W3C, can be extended with a conditional probabilistic class to encode the probability that an instance belongs to a class respectively given that it belongs to another class. Although good at expressing uncertainty, these qualitative approaches need to be combined with other machine-learning techniques to quantify the uncertainty to be used in situation identification. Learning-based approaches have a stronger capability to resolve uncertainty by training with the real-world data that involves noise. These approaches not only learn associations between sensor data and situations, but also the effect that the uncertainty of sensor data has on the associations. For example, the conditional probabilities learned in Bayes networks include the reliability of sensor data as well as the contribution of the characterized sensor data in identifying a situation. A popular architectural model for IoT is composed of autonomous physical/digital objects augmented with sensing, processing, and network capabilities. Unlike RFID tags, smart objects carry an application logic that let them sense their local situation and interact with the environment through actuators. They sense, log, and interpret what's occurring within themselves and the work, act on their own, intercommunicate with each other, and exchange data [31].

According to the scenarios illustrated in [31], the architectural differences in the way smart objects understand (sense, interpret or react to) events, and interact with their environment in terms of input, output, control and feedback, classify them as either activity-aware objects, policy-aware objects or process-aware objects. A process-aware object represents the most accomplished type, and characterizes: awareness (a process-aware object understands the operational process that is part of and can relate the occurrence of real-work activities and events to these processes), representation (its model consists of a context-aware workflow model that defines timing and ordering or work activities), and interaction (a process-aware object providers workers with context-aware guidance about tasks, deadlines, and decisions).

With an understanding of what context is and the different ways in which it can be used, application builders can now more easily determine what behaviors or features they want their applications to support and what context is required to achieve these behaviors. However, something is still missing. Application builders may still need help moving from the design to an actual implementation. This help can come from a combination of architectural services or features that designers can use to build their applications from. For example, the Webinos EU funded project aims to deliver a platform for web applications across mobile, PC, home media and in-car devices [42]. The webinos project has over twenty partners from across Europe spanning academic institutions, industry research firms, software firms, handset manufacturers and automotive manufacturers. Its vision is to build a multiplatform, applications platform based on web technology that is fit for purpose, across a wide range of connected devices, taking computing to its next logical step, that of ubiquity. In order to do so, knowing the state of the device and the user at any given time, and making decisions based on that context is crucial. Context-awareness and true cross-platform and cross-device applications cannot be achieved without the developer having access to an environment able to synchronize content and application state between devices, adapt to changes in user context, and provide standard JavaScript APIs to let web applications access device features. Therefore webinos is a good illustration of a cross platform level of abstraction for procedural calls, but at the same time, incorporate an additional data abstraction layer for use in third party context-aware and context-sharing applications that are webinos-enabled [42].

3 A study on opportunistic data dissemination support for the Internet of Things

Today, the way that people access information and communicate is radically changing, right before our eyes, in many ways that are not yet readily apparent. Wireless devices, such as mobile phones, connected devices and consumer electronics, are infiltrating all aspects of our lives. As the cost to mobilize these devices continues to drop, and as wireless networks become faster, ubiquitous and cheaper, it is easy to see a near future where almost everything and everyone are wirelessly online,

24x7. It is evident that the wireless universe of things is rapidly accelerating. This raises many questions as well as opportunities, especially for businesses that offer communications equipment and services, consumer electronics and other connected devices. People will increasingly expect to access a wide range of data and rich content on many devices. Up until recently, the most common mobile data types include voice, contacts (address books) and SMS (text messages). Today, people expect to wirelessly transparently access their email, social network messaging, calendars, photos, videos, files, music, games, apps and web pages. Behind we have a Big Data platforms and environments. Wireless applications need to easily manage and filter an avalanche of mobile data. This includes functions such as syncing, sharing, searching, transferring, archiving, deleting and caching. It means making it seamless to share mobile data and rich content with other devices, people, groups, businesses, schools, public agencies, systems, etc. But when things are unable to establish direct communication, or when communication should be offloaded to cope with large throughputs, mobile collaboration can be used to facilitate communication through opportunistic networks. These types of networks, formed when mobile devices communicate only using short-range transmission protocols, usually when users are close, can help applications still exchange data. Routes are built dynamically, since each mobile device is acting according to the store-carry-and-forward paradigm. Thus, contacts are seen as opportunities to move data towards the destination. In such networks data dissemination is usually based on a publish/subscribe model.

One type of such mobile networks that has been deeply researched in recent years is represented by opportunistic networks (ONs). They are dynamically built when mobile devices collaborate to form communication paths while users are in close proximity. Opportunistic networks are based on a store-carry-and-forward paradigm [36], which means that a node that wants to relay a message begins by storing it, then carries it around the network until the carrier encounters the destination or a node that is more likely to bring the data close to the destination, and then finally forwards it.

One of the main challenges of opportunistic networks is deciding which nodes should the data be relayed to in order for it to reach its destination, and do it as quickly and efficiently as possible. Various types of solutions have been proposed, ranging from disseminating the information to every encountered node in an epidemic fashion, to selecting the nodes with the highest social coefficient or centrality [26]. Prediction methods have also been employed [9], based on the knowledge that the mobile nodes from an opportunistic network are devices belonging to humans, which generally have the same movement and interaction patterns that they follow every day. The analysis of contact time (duration of an encounter between two nodes) and inter-contact time (duration between consecutive contacts of the same two nodes) has also been used in deciding a suitable relay node. Aside from selecting the node that the data will be forwarded to, research has also focused on congestion control, privacy, security, or incentive methods for convincing users to altruistically participate in the network.

An important topic in opportunistic networks is represented by *data dissemination*. In such networks, given the dynamic nature of the wireless devices in IoT, topologies are unstable. Various authors proposed different data-centric approaches for data dissemination, usually based on different publish/subscribe models, where data is pro-actively and cooperatively disseminated from sources towards possibly interested receivers, as sources and receivers might not be aware of each other and never get in touch directly. We analyze in the following lines existing work in the area of data dissemination in opportunistic networks. We analyze different collaboration-based communication solutions, emphasizing their capabilities to opportunistically disseminate data. We present the advantages and disadvantages of the analyzed solutions. Furthermore, we propose the categories of a taxonomy that captures the capabilities of data dissemination techniques used in opportunistic networks. Using the categories of the proposed taxonomy, we also present a critical analysis of four opportunistic data dissemination solutions. To our knowledge, a classification of data dissemination techniques for IoT platforms has never been previously proposed.

3.1 Opportunistic data dissemination and the Internet of Things

In recent years more and more IoT scenarios and use-cases based on delay-tolerant networks (DTNs) and opportunistic networks (ONs) have been proposed. These scenarios range from commercial (e.g. targeted advertising [23]) to information (e.g. floating content [17], context-aware mobile platforms [19]), smart cities [7] or even emergency situations (e.g. crowd location, disaster management [33], etc.). Such networks differ from traditional ones because they do not require a fixed infrastructure to function. Traditional networks generally assume that the nodes are fixed and the topology of the network is well known in advance. By analyzing the topology, a node that wishes to send a message will embed that message with the path and thus it will be sure it can reach the intended destination. With DTNs and ONs, the situation is different. Firstly, the nodes are mobile, so a node doesn't have the same neighbors at two different moments in time. Secondly, contact opportunities between two nodes only arise when they are in contact, so the contact duration is important in deciding whether data should be exchanged or not. Thirdly, nodes in ONs and DTNs are mobile devices, so they are restricted by buffer space. This means that, when two nodes encounter, simply exchanging all data between them is not feasible. ONs are based on a store-carry-and-forward paradigm, where nodes store messages and carry them until a suitable destination (in terms of probability of delivering a message to its intended target) is encountered, when they are forwarded. Such delay-tolerant and opportunistic networks have a real use in ensuring communication in IoT, when traditional methods are not available, are too costly, or alternatives are required.

Mobile online advertising nowadays has become ubiquitous, due to companies such as Google (on Android devices) or Facebook making their revenue by targeting advertisements to fit a user's preferences and tastes. However, users fear that, in

order to show suitable ads, big companies collect too much precious information and privacy is lost. However, ONs can help correct this and ensure privacy, and such a proposed solution is MobiAd [23]. It is an application that presents the user with local targeted advertisements, while still preserving the privacy. The ads are selected from an ad pool received from local hotspots or even broadcast on the local mobile base-station, but statistical information about ad views and user clicks is encrypted and sent to the advertisement channel opportunistically, via the other ON devices in the network (or using static WiFi hotspots). Thus, other nodes and even the ad provider cannot know which ads a given node has viewed or clicked on.

Since ONs and DTNs are used to carry information opportunistically, a suitable scenario is represented by information-sharing. An example of this is floating content in open city squares [17], where mobile nodes may enter a geographically-fixed area that specifies the physical boundaries of the network (an anchor zone), spend a certain amount of time there, and then leave. While located in the anchor zone, devices (or a static access point) produce content and replicate it opportunistically to other nodes, which may use the data for themselves, or carry it in order to forward it to interested nodes. When a node exits the anchor zone, the zone-specific data is deleted, so the floating content's availability is probabilistic and strictly connected to the anchor zone. A real-life use of floating content in open city squares is a touristic information centre, where users receive information about a certain building, place or statue when they are located in its vicinity.

Information can be spread using opportunistic networks in other environments, such as academics, where ONs can be employed as the backbone of a framework that facilitates interaction between the members of a faculty (students, professors, etc.) possessing smartphones or other mobile devices. Thus, communication inside the faculty (in a limited physical area, like in open city squares) may be performed opportunistically, instead of using a fixed infrastructure such as WiFi or 3G. The benefits are limiting the used bandwidth and the battery consumption, since Bluetooth (mainly used for opportunistic communication with smartphones) consumes less power than both 3G and WiFi. A platform for supporting generic context-aware mobile applications is CAPIM [19], and it can fully benefit from ON integration. CAPIM (Context-Aware Platform using Integrated Mobile services) is a solution designed to support the construction of next-generation applications. It integrates various services that collect context data (location, user's profile and characteristics, environment information, etc.). The smart services are dynamically loaded by mobile clients, which take advantage of the sensing capabilities provided by modern smartphones, possibly augmented with external sensors.

Moving towards the future, DTNs and ONs are a very good fit for smart cities [7]. These cities monitor and integrate the conditions of all their critical infrastructures (such as roads, bridges, tunnels, rails, subways, airports, seaports, communications, water, power, etc.) in order to better optimize their resources and plan their preventive maintenance activities. They react to changes in any of these infrastructures by analyzing the environment and deciding what the suitable decision is. Smart cities connect the physical, IT, social and business infrastructures, for the purpose of leveraging the collective intelligence of the cities. Delay-tolerant and opportunis-

tic networks can be employed to perform communication between various parts of a smart city. For example, the traffic lights system can be opportunistically connected to a service that offers information about traffic jams, crowded roads, accidents, etc., so it can adapt to the conditions of the environment. All data processing for smart cities applications require efficient Big Data platforms.

Another situation where ONs and DTNs may be used is in emergency situations, e.g. at crowded locations where groups of individuals get separated and need to locate each other. In such scenarios (e.g. an amusement park where a child gets separated from his parents, a concert where a group of friends split and they need to find each other, a football match, etc.), contacts between mobile devices happen often and for longer periods of time, but (due to the high number of mobile devices in a very small space), classic communication means such as 3G and mobile telephony may not work properly. In these cases, the mobile devices may be used to opportunistically broadcast the location of mobile device owners to interested receivers. For example, a child at an amusement park might carry a smartphone that opportunistically broadcasts the child's encrypted location, by leveraging the neighboring devices. The child's parents have a smartphone of their own which receives the broadcast from encountered nodes, thus being aware of the child's location at any point in time. Similarly, non-emergency scenarios may also benefit from this solution, where events regarding the bands that will be playing or any other announcements can be disseminated to participants of a music concert or any other type of social event with high participation.

Finally, opportunistic communication may also be used in disaster management scenarios [33], where regular communication might be down. Thus, mobile devices have the potential of forming a new opportunistic infrastructure to be used for communication, either by the rescue teams when searching for survivors, or even by the survivors to signal their presence and their location among the debris (since smartphones are generally equipped with a GPS). Rescue teams may even use the opportunistic infrastructure to communicate between each others, by placing temporary access points at key places in the disaster area.

These scenarios have shown that ONs and DTNs have the possibility of replacing traditional network communication in some situations, or even of working alongside it, in order to bring benefits for offering advance communication capabilities for future IoT platforms. Thus, we believe that research effort should be put into this area, since there are many issues that have yet to be explored.

3.2 A Taxonomy for Dissemination Techniques

We introduce the categories of a proposed taxonomy for data dissemination techniques in opportunistic networks (presented in Figure 3). This taxonomy was created by analyzing the existing methods in the literature and highlighting the common characteristics between various methods. Previous attempts to categorize different opportunistic dissemination techniques are focused on specific aspects. The

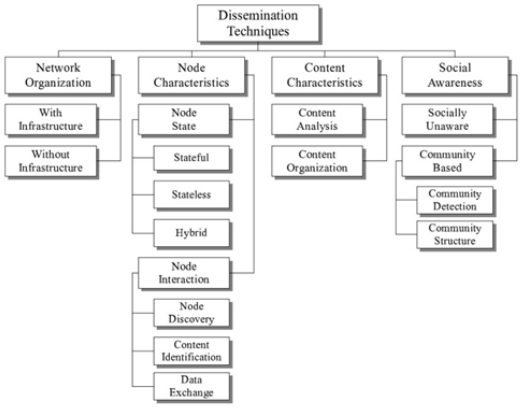

Fig. 3 A taxonomy for data dissemination techniques.

taxonomy proposed in [10] separates the forwarding methods according to mostly their knowledge about context, proposing a separation into three distinct classes: context-oblivious, partially context-aware and fully context-aware). Similarly, the taxonomy in [36] separates techniques based on their knowledge of the infrastructure, making a distinction between algorithms without infrastructure and algorithms where the ad-hoc networks exploit some form of infrastructure to forward messages. Algorithms without infrastructure can be further divided into algorithms based on dissemination (like Epidemic, Meeting and Visits and Networking Coding), and algorithms based on context (like CAR and MobySpace). Algorithms that exploit a form of infrastructure can also be divided into fixed infrastructure and mobile infrastructure algorithms. In case of fixed infrastructure algorithms (like Infostations and SWIM), special nodes are located at specific geographical points, whereas special nodes proposed in mobile infrastructure algorithms (like Ferries and DataMULEs) move around in the network randomly or follow predefined paths.

According to our proposed taxonomy, data dissemination algorithms can be categorized by the *organization of the network* on which they apply. In general, in an opportunistic network no assumption is made on the existence of a direct path between two nodes that wish to communicate. Nonetheless, some dissemination algorithms may exploit certain nodes called “hubs” and build an overlay network between them. The hubs are the nodes with the highest centrality in each community, where a node’s centrality is the degree of its participation in the network. There are several types of node centrality relevant to data dissemination in opportunistic networks (such as degree centrality, betweenness centrality or closeness centrality), that are detailed later on. The algorithms that build an overlay network based on hubs fall under the category of data dissemination algorithms *with infrastructure*. However, relying on an infrastructure might be costly to maintain (due to the large number of messages that have to be exchanged to keep it) and also highly unstable,

especially in case of networks that contain nodes with a high degree of mobility. Considering this aspect, many data dissemination algorithms assume that the opportunistic network is a network *without infrastructure*. The *network organization* is relevant for a data dissemination algorithm because it directly influences the data transfer policies.

The actual nodes that participate in an opportunistic network play an important part in the way a data dissemination algorithm works. Consequently, the proposed taxonomy also categorizes dissemination techniques according to *node characteristics*. A first characteristic of a node in an opportunistic network is the *node state*. Depending on its implementation, a dissemination technique can either follow a *stateful*, a *stateless* or a *hybrid* approach. An approach that maintains the state of a node requires control traffic (e.g. unsubscription messages in a publish/subscribe-based algorithm) that can prove to be expensive. Moreover, it suffers if frequent topology changes occur. On the other hand, a *stateless* approach does not require control traffic, but has unsatisfactory results if event flooding is used. The *hybrid* approach takes advantage of both the *stateful* and the *stateless* approaches.

Another important characteristic of a node in an opportunistic network is *node interaction*. As stated before, nodes in an opportunistic network generally have a high degree of mobility, so the interaction between them must be as fast and as efficient as possible. The reason for this is that contact duration (the time interval when two network devices are in range of each other) may be extremely low. According to the proposed taxonomy, there are three basic aspects of *node interaction*, the first one being *node discovery*. Depending on the type of mobile device being used, the discovery of nodes that are in the wireless communication range can be done in several ways, but it is usually accomplished by sending periodical broadcast messages to inform neighboring nodes about the device's presence. When two nodes come into wireless communication range and make contact, they each have to inform the other node of the data they store. Therefore the second aspect of *node interaction* is *content identification*, meaning the way in which nodes represent the data internally and how they "declare" it (usually using some form of meta-data descriptions). Nodes may also advertise the channels they have data from or they may present a hash of the data they store. The final subcategory of *node interaction* is *data exchange*, which is the way two nodes transfer data to and from each other. This refers not only to the actual data transferring method, but also to the way data is organized or split into units. The three *node interaction* steps presented here may also be done asynchronously for several neighboring nodes, and the way they are implemented affects the performance of a data dissemination algorithm.

As stated in [44], an interesting use case for opportunistic networks is the sharing of content available on mobile users' devices. In such a network, users themselves may generate content (e.g. photos, clips) on their mobile devices, which might be of interest to other users. However, content producers and consumers might not be connected to each other, so an opportunistic data dissemination method is necessary. Because there can be many types of content, each having different characteristics, the proposed taxonomy also classifies data dissemination algorithms according to the *content characteristics*.

An important aspect of the actual content is its *organization*. Most often, content is organized into channels, an approach used for publish/subscribe-based data dissemination. The publish/subscribe pattern is used mainly because communication is based on messages and can be anonymous, whilst participants are decoupled from time, space and flow. Time decoupling takes place because publishers and subscribers do not need to run at the same time, while space decoupling happens because a direct connection between nodes does not have to exist. Furthermore, no synchronized operations are needed for publishing and subscribing, so nodes are also decoupled from the communication flow. The approach allows the users to subscribe to a channel and automatically receive updates for the content they are interested in. Such an organization is taken further by structuring channels into episodes and enclosures.

Aside from the way content is organized at a node, the proposed taxonomy categorizes data dissemination techniques by *content analysis*. *Content analysis* represents the way in which the algorithm analyzes a certain content object and decides if it will fetch it or not. There are two reasons a node might download a content object from another encountered node: it is subscribed to a channel that the object belongs to, or the node has a higher probability of being or arriving in the proximity of another node that is subscribed to that channel than the node that originally had the information. Not all dissemination algorithms analyze the data from other nodes: some simply fetch as much data as they can, until their cache is full, like Epidemic routing, while others only verify if they do not already contain the data or if they have not contained it recently. More advanced *content analysis* can be accomplished by assigning priorities (or utilities) to each content object from a neighboring node. In this way, considering the amount of free cache memory, a node can decide what and how many content objects it can fetch from another node. A node can also calculate the priority for its own content objects, and advertise only the priorities. Thus, a neighboring node can choose the data that maximizes the local priority of its cache. One method of computing priorities is based on heuristics that compare two content objects. Heuristics can compare content objects by their age, by their hop count or by the number of subscribers to the channel the object belongs to. A more complex approach to computing the value of priorities is to use a mathematical formula that assigns weights to various parameters. This method is used especially in socially-aware dissemination algorithms, where users are split in communities, and each community is assigned an individual weight (more about socially-aware algorithms will be presented in the next paragraph).

The final category of the proposed taxonomy is the *social awareness*. Recently, the social aspect of opportunistic networking has been studied, because the nodes in an opportunistic network are represented by humans carrying wireless devices. The human factor can be an important dimension that is already considered by several data dissemination algorithms. When designing such an algorithm, it is important to know that user movements are conditioned by social relationships. The first subcategory of social awareness is represented by *socially-unaware* algorithms, which do not assume the existence of a social structure that governs the movement or interaction of the nodes in an opportunistic network. Data dissemination techniques of this

type may be as simple as spreading the content to all encountered nodes, but they can also take advantage of non-social context information such as geographical location. Most of the recent data dissemination techniques that are aware of the social aspect of an opportunistic network are *community-based*. Such dissemination algorithms assume that users can be grouped into communities, based on strong social relationships between users. Even though there are several proposed representations of social behavior, the caveman model [44] is by far the one mostly used, along with its variations. Users can belong to more communities (called "home" communities), but can also have social relationships outside of their home communities (in "acquainted" communities). Communities are usually bound to a geographical space (static social communities), but they may also be represented by a group of people who happen to be in the same place at the same time (e.g. at a conference - temporal communities). According to this model, users spend their time in the locations of their home communities, but also visit areas where acquainted communities are located. As previously stated, a utility function may be used to decide which content objects must be fetched when two nodes are in range of each other. In a *community-based* approach, each community would be assigned a weight, and the utility of a data object would be computed according to the community its owner comes from and the community of the (potentially) interested nodes.

One step that has to be executed before designing a *community-based* dissemination algorithm is the *community detection*. There are several methods used for organizing nodes from an opportunistic network into communities. One way is to simply classify nodes based on the number of contacts and contact duration of a node pair according to a threshold value, while another approach would be to define k-CLIQUE communities as unions of all k-CLIQUEs that can be reached from each other through a series of adjacent k-CLIQUEs [48]. The phase following the detection of existing communities is the design of a *community structure*. All nodes in a community can be identical (from the perspective of behavior), but there are also situations where certain nodes are more important in the dissemination scheme. As previously described, some data dissemination algorithms use network overlays constructed using hubs or brokers (e.g. nodes with the highest centrality in a community). The advantage of such an approach is that only nodes having a high centrality transfer messages to other communities. When a node wants to send a content object, it transfers it to the hub (or to a node with a higher centrality, which has a better chance of reaching the hub). The hub then transfers the object to the hub of the destination's community, where it eventually reaches the desired destination. The structure of a community has a high relevance in classifying data dissemination techniques, because a well-structured community can speed up the dissemination process significantly.

3.3 Critical analysis of dissemination algorithms

We now analyze the properties of four popular techniques for disseminating data in an opportunistic network, using the categories of the proposed taxonomy. The presented study evaluates the most relevant recent work in data dissemination algorithms. We also apply the proposed taxonomy to analyze and differentiate between the presented data dissemination techniques. The **Socio-Aware Overlay** algorithm [48] is a data dissemination technique that creates an overlay for an opportunistic network with publish/subscribe communication, composed of nodes having high centrality values that have the best visibility in a community. The data dissemination technique assumes the existence of a network *with infrastructure*, built by creating an overlay comprising of representative nodes from each community. The dissemination of subscriptions is done, together with the community detection, during the *node interaction* phase, through gossiping. The gossiping dissemination sends each message to a random group of nodes, so from a *node state* point of view, the Socio-Aware algorithm takes a *hybrid* approach. In order to choose an appropriate hub (or broker) in a network, the algorithm uses a measurement unit called node centrality.

Node discovery is performed through Bluetooth and WiFi devices, while there are two modes of *node interaction*, namely unicast and direct. The former is similar to Epidemic routing, while the latter provides a more direct communication mechanism like WiFi access points. From the standpoint of *content organization*, the Socio-Aware algorithm is based on a publish/subscribe approach. At the *data exchange* phase, subscriptions and un-subscriptions with the destination of community broker nodes are exchanged, as well as a list of centrality values with a time stamp. When a broker node changes upon calculation of its closeness centrality, the subscription list is transferred from the old one to the new one. Then, an update is sent to all the brokers. During the gossiping stage, subscriptions are propagated towards the community's broker. When a publication reaches the broker, it is propagated to all other brokers, and then the broker checks its own subscription list. If there are members in its community that must receive the publication, the broker floods the community with the information. The Socio-Aware algorithm is a socially-aware *community-based* algorithm, that has its own *community detection* method. This method assumes a *community structure* that is based on a classification of the nodes in an opportunistic network, from the standpoint of another node. A first type of node is one from the same community, having a high number of contacts of long/stable durations. Another type of node is called a familiar stranger and has a high number of contacts with the current node, but the contact durations are short. There are also stranger nodes, where the contact duration is short and the number of contacts is low, and finally friend nodes, with few contacts, but high contact durations.

In order to construct an overlay for publish/subscribe systems, *community detection* is performed in a decentralized fashion, because opportunistic networks do not have a fixed structure. Thus, each node must detect its own local community. The authors propose two algorithms for distributed community detection, named Simple and k-CLIQUE. In order to detect its own local community, a node interacts

with encountering devices and executes the detection algorithm. The detection algorithm is done in the *data exchange* phase of the interaction between nodes. Each node accomplishes the *content identification* by maintaining information about the encountered nodes and contact durations (represented as a map called the familiar set) and the local community detected so far. When two nodes meet, a data exchange is performed, with each node acquiring information about the other's familiar set and local community. Each node then updates its local community and familiar set values, according to the algorithm used. As more nodes are encountered over time, the shape of the local community may be modified. The **Wireless Ad Hoc Podcasting** system [32] extends podcasting to ad-hoc domains. The purpose is the wireless ad-hoc delivery of content among mobile nodes. Assuming a network without infrastructure, the wireless podcasting service enables the distribution of content using opportunistic contacts whenever podcasting devices are in wireless communication range. From the standpoint of content organization, the Ad Hoc Podcasting service employs a publish/subscribe approach. Thus, it organizes content into channels, which allows the users to subscribe and automatically receive updates for the content they are interested in. However, the channels themselves are divided into episodes and enclosures. Furthermore, enclosures are also divided into chunks, which are transport-level small data units of a size that can typically be downloaded in an individual node encounter. The reason for this division is the need for improving efficiency in the case of small duration contacts. The chunks can be downloaded opportunistically from multiple peers, and they are further divided into pieces, which are the atomic transport units of the network.

For *node interaction*, when two nodes are within communication range they associate and start soliciting episodes from the channels they are subscribed to. Since data is not being pushed, the nodes have complete control over the content they carry and forward. *Node discovery* is done by using broadcast beacons sent periodically by each node. *Content identification* is performed to identify channels and episodes at the remote peer that the current node is subscribed to. Two nodes in range exchange a Bloom filter hash index that contains all channel IDs that each node offers. Then each node checks the peer's hash index for channels it is subscribed to. The *data exchange* phase begins if one of the nodes has found a matching channel. In this case, it starts querying for episodes. In order to perform *content analysis*, the Wireless Ad Hoc Podcasting system proposes three different types of queries, employed according to the channel policy: a node requests any random episodes that a remote peer offers, a node requests episodes from the peer that are newer than a given date starting with the newest episode, or a node requests any episodes that are newer than a given date starting with the oldest episode.

When two nodes meet, and neither has content from a channel the other is subscribed to, several solicitation strategies are employed [32]. They are used to increase the probability of a node having content to share with other nodes in future encounters. The solicitation strategies proposed are Most Solicited, Least Solicited, Uniform, Inverse Proportional and No Caching. The Most Solicited strategy fetches entries from feeds that are the most popular. The Least Solicited strategy does the opposite, by favoring less popular feeds. The Uniform strategy treats all channels

equally, by soliciting entries in a random fashion, and has the advantage of being easy to implement. The Inverse Proportional strategy maintains a history list and solicits a feed with a probability which is inverse proportional to its popularity. Finally, No Caching is more of a benchmark for other strategies than a strategy itself, and assumes that a device has no public cache at all and that it stores or distributes only content from the fields it is subscribed to. Experiments show that the Uniform strategy has the best overall performance, while Inverse Proportional is the best one in regards to fairness.

Authors of [21] propose a probabilistic publish/subscribe-based multicast distribution infrastructure for opportunistic networks based on DTN (Delay Tolerant Networking). The protocol uses a push-based asynchronous distribution delivery model. The idea is that nodes in the opportunistic network replicate bundles to their neighbors in order to get the bundle delivered by multiple hops of store-carry-and-forward.

As its name states, **DPSP** has a *content organization* based on a channel subscription system, where users subscribe to channels and senders publish content. Although from the *network organization* standpoint, DPSP assumes *no infrastructure*, the nodes in the network are divided into three categories: sources, sinks and other nodes. Sources are the nodes that send content (in the form of bundles of data) to channels, while sinks subscribe to channels and receive information from them. The rest of the nodes are not interested in specific bundles, but they store, carry and forward bundles and subscriptions.

The *node interaction* phase has several steps. When two nodes meet, *content identification* is performed through the exchange of subscription lists. An entry in a subscription list contains the channel's URI, the subscription's creation time, its lifetime, the number of hops from the original subscriber to the current node, and an identifier for the subscription. Then, each node builds a queue of bundles to forward to the peer, and uses a set of filters to select the best. The selected bundles are subsequently sorted according to their priorities, and the *data exchange* stage is performed by sending the bundles one by one until the contact finishes or the queue becomes empty.

In this approach, a set of filters is used in order to select the best bundles in a queue. Because the DPSP protocol is *socially-unaware*, the filters used do not consider the organization of users into communities. There are three filters that handle the *content analysis* and that can be used in any combination: Known Subscription Filter, Hop Count Filter and Duplicate Filter. The Known Subscription Filter removes bundles nobody is interested in, the Hop Count Filter removes bundles that are too old, while the Duplicate Filter removes bundles that the peer has already received. *Content analysis* is also performed when the remaining bundles from a queue are sorted according to their priorities. Four heuristics are used to assign priorities to bundles: Short Delay, Long Delay, Subscription Hop Count and Popularity. Short Delay prefers newer bundles, Long Delay prefers older bundles, Subscription Hop Count sorts bundles according to hop count, and the Popularity heuristic sorts bundles by the number of nodes subscribed to the bundle's channel. The authors

noticed that the Short Delay heuristic performs better with respect to delivery rates than the other heuristics.

ContentPlace [4] deals with data dissemination in resource-constrained opportunistic networks, by making content available in regions where interested users are present, without overusing available resources. To optimize content availability, Content Place exploits learned information about users' social relationships to decide where to place user data. The design of ContentPlace is based on two assumptions: users can be grouped together logically, according to the type of content they are interested in, and their movement is driven by social relationships.

For performance issues, ContentPlace assumes a network *without infrastructure*. When a node encounters another node it decides what information seen on the other node should be replicated locally. When two nodes are in range, they have to discover each other. The *node discovery* is not specified, but since the nodes are mobile devices it is probably done by WiFi or Bluetooth periodic broadcasts. For *content identification*, nodes advertise the set of channels the local user is subscribed to upon encountering another node. ContentPlace defines a utility function by means of which each node can associate a utility value to any data object. When a node encounters another peer, it selects the set of data objects that maximizes the local utility of its cache. Due to performance issues, when two nodes meet, they do not advertise all information about their data objects, but instead they exchange a summary of data objects in their caches. Finally, the *data exchange* is accomplished when a user receives a data object it is subscribed to when it is found in an encountered node's cache. *Content organization* in ContentPlace is done through channels to which users can subscribe. Consequently, unsubscription messages are not necessary, so a *stateless* approach is used for the nodes. ContentPlace is a socially-aware, *community-based* data dissemination algorithm. To have a suitable representation of users' social behavior, an approach that is similar to the caveman model is used, that has a *community structure* which assumes that users are grouped into home communities, while at the same time having relationships in acquainted communities. For *content analysis* nodes compute a utility value for each data object. The utility is a weighted sum of one component for each community its user has relationships with. The utility component of a data object for a community is the product of the object's access probability from the community members, by its cost (which is a function of the object's availability in the community), divided by the object's size. *Community detection*, like at the Socio-Aware Overlay, uses the algorithms described in [27]. By using weights based on the social aspect of opportunistic networking, ContentPlace offers the possibility of defining different policies. There are five policies defined: Most Frequently Visited (MFV), Most Likely Next (MLN), Future (F), Present (P) and Uniform Social (US). MFV favors communities a user is most likely to get in touch with, while MLN favors communities a user will visit next. F is a combination between MLN and MFV, as it considers all the communities the user is in touch with. In the case of P, users do not favor other communities than the one they are in, while at US all the communities the users get in touch with have equal weights.

A critical analysis of the four described protocols, according to the proposed taxonomy, is presented in Figure 4.

Data Dissemination Technique	Network Organization	Node Characteristics				Content Characteristics		Social Awareness	
		Node State	Node Interaction			Content Analysis	Content Organization	Community Detection	Community Structure
			Node Discovery	Content Identification	Data Exchange				
Socio-Aware Overlay	Overlay Infrastructure	Hybrid	Bluetooth and WiFi	Encountered nodes and cont. duration	Subscriptions and list of centralities	N/A	Publish Subscribe	Simple and K-clique algorithms	Contact duration and no. of contacts
Ad Hoc Podcasting	No Infrastructure	N/A	Broadcast beacons	Bloom filter hash index	Episodes or chunks	Solicitation strategies	Publish Subscribe	N/A	N/A
DPSP	No Infrastructure	N/A	N/A	Subscription lists	Selection of bundles	Filters and priority heuristics	Publish Subscribe	N/A	N/A
ContentPlace	No Infrastructure	Stateless	Bluetooth and WiFi	Set of channels the node is subscribed to	Data objects	Utility function	Publish Subscribe	N/A	Caveman model

Fig. 4 Critical analysis of four dissemination techniques.

According to our analysis of the four solutions, only one assumes that the network over which data dissemination is performed has an infrastructure. The Socio-Aware Overlay algorithm builds an overlay infrastructure using the nodes with the highest centrality from each community. However, opportunistic networks generally contain nodes with a high degree of mobility, which make the task of creating and maintaining an infrastructure very hard to accomplish. The reason for this is that nodes may change communities very often (or they may not belong to a community at all), thus complicating the community detection phase. Furthermore, a device that is considered to be the central node (or hub) of a community may be turned off (due to different circumstances, like battery depletion), leaving the nodes in the hub's community without an opportunity to send messages to other communities, until a new hub is elected. Given these reasons, we believe that an approach that does not assume the existence of an infrastructure should be further considered.

The characteristics of a node from an opportunistic network play an important role in the structure of a data dissemination algorithm. Node characteristics refer to the way a node's state is represented and the way nodes interact when they are in contact. As stated in Section 4, the approach a data dissemination algorithm can take in regard to node state can be either stateless, stateful, or hybrid. Of the protocols we analyzed, ContentPlace chooses a stateless approach, while the Socio-Aware Overlay uses a hybrid representation of a node's state. The authors of the other two algorithms do not specify the node state, but we assume a stateful approach, because of the way the content is represented (for example, DPSP maintains subscription lists, for which node state is required). According to [48], a hybrid approach is the preferred solution because it takes advantage of both stateful and stateless approaches. Such an approach would not suffer under frequent topology changes, while at the same it would not require a large amount of control traffic.

The interaction between nodes has three steps that have been presented in detail in Section 4: node discovery, content identification and data exchange. Node discovery is usually done in the same way for all algorithms analyzed, but it may differ according to the type of devices that are present in the network. In case of the Socio-Aware Overlay and ContentPlace, the discovery is performed by using the Bluetooth or WiFi capabilities. The Ad Hoc Podcasting algorithm uses broadcast beacons, while the authors of DPSP do not mention a particular discovery method. It is a good approach to use the existing capabilities from the wireless protocols, but a data dissemination algorithm should try to extend the battery's life as much

as possible. For example, when the battery is low, the broadcast beacons should be sent at larger time intervals.

Content identification, meaning the way in which nodes represent the data internally, also has a big impact in the efficiency of a data dissemination technique. The Socio-Aware Overlay maintains information about the encountered nodes and the duration of contacts, Ad Hoc Podcasting uses a Bloom filter hash index that contains all channel IDs, DPSP exchanges subscription lists and ContentPlace advertises the set of channels a node is subscribed to. The most efficient method is using Bloom filters, because they are space efficient data structures of fixed size that avoid unnecessary transmissions of data that the receiver has already received [3].

Data exchange should also be performed in a manner that optimizes the duration of a transfer. The nodes from the Socio-Aware Overlay exchange subscriptions and lists of centrality values, Ad Hoc Podcasting exchanges episodes or chunks, DPSP uses bundles and ContentPlace nodes exchange data objects. The smaller the data unit, the bigger is the chance of a transmission to successfully finish, even in opportunistic networks where contact durations are very small. Therefore, one of the best approaches is the one employed by Ad Hoc Podcasting, where data is split into episodes and chunks.

The type of content organization that best suits opportunistic networks is the publish/subscribe pattern. The reason for this is that participants are decoupled from time, space and flow. Interested users simply subscribe to certain channels and receive data whenever the publishers post it. Publishers and subscribers do not have to be online at the same time, and it is not necessary that a direct connection exists between them. Consequently, all the analyzed data dissemination techniques organize their content according to a publish/subscribe approach. Content can also be analyzed in order for a node to decide what to download from an encountered peer. The Ad Hoc Podcasting technique uses five solicitation strategies that aim to increase the probability of a node having content to share with other nodes. DPSP has three filters used to select the best bundles in a queue and four heuristics that sort the remaining bundles. Finally, ContentPlace computes a utility function based on every community a node is in relationship with. The ContentPlace approach performs the best, because it takes advantage of the social aspect of opportunistic networking.

According to [11], human social structures are at the core of opportunistic networking. This is because humans carry the mobile devices, and it is the human mobility that generates communication opportunities when two or more devices come into contact. Social-based forwarding and dissemination algorithms reduce by about an order of magnitude the overhead, compared to algorithms such as Epidemic routing. Therefore, the social aspect has a very important role in the efficiency of a data dissemination technique in an opportunistic network. Social awareness is based on the division of users into communities, which are defined as groups of interacting individuals organized around common values within a shared geographical location. Thus, an important step for socially-aware dissemination algorithms is community detection. Of the techniques we studied, only the Socio-Aware Overlay proposes its own community detection algorithms, called Simple and k-CLIQUE. ContentPlace uses similar algorithms, while Ad Hoc Podcasting and DPSP are socially-unaware.

As far as community structure goes, the Socio-Aware Overlay splits the nodes in a community from the standpoint of another node, according to the contact duration and number of contacts, while ContentPlace adopts a model similar to the caveman model. We consider that the future of data dissemination algorithms should be based on a socially-aware approach to take advantage of the human aspect of opportunistic networking. After analyzing the four data dissemination techniques, we can conclude that there is no single best approach, but each algorithm provides certain aspects that offer advantages over the other implementations. In the next phase we plan to extend this work and propose a dissemination algorithm that uses the advantages of all analyzed solutions for maximum efficiency.

4 Future trends and research directions in Big Data platforms for the Internet of Things

The evolution of the Internet of Things towards connecting every Thing on the planet in a very complex and large environment gives raise to high demanding requirements, which are subject of actual research. The continuously increasing volume of data collected from and exchanged among Things will require highly scalable environments able to support the high resulting network traffic, and offer the necessary storage capacity and computing power for data preservation and transformation. Communication protocols are needed to enable not only the high capacity traffic but also maintain the connectivity between Things even in case of transient disconnection of wired or wireless links. Also, new solutions should be found for efficiently store, search and fetch the data manipulated in these environments.

IoT is at the base of developing many applications that include things, people, economy, mobility, and governance. They enrich the urban environment with situational information, which can be rapidly exploited by citizens in their professional, social, and individual activities to increase city competitiveness. In addition, outdoor computing and user-centric approaches to services can improve the core urban systems such as public safety, transportation, government, agency administration and social services, education, healthcare. Context-aware applications use large amounts of input data, in various formats, collected from sensors or mobile users, from public open data sources or from other applications.

Internet of Things is not yet a reality, “but rather a prospective vision of a number of technologies that, combined together, could in the coming 5 to 15 years drastically modify the way our societies function” [37]. The evolution of the IoT on medium and long term unleashed a huge interest and gave raise to many re-search projects, workshops, conferences, reports and survey papers. In this section we discuss the aspects related to the IoT infrastructure and services with emphasis on the main challenges.

It is estimated [38] that IoT will have to accommodate over 50,000 billion objects of very diverse types. Standardization and interoperability will be absolute necessities for interfacing them with the Internet. New media access techniques, commu-

nication protocols and sustainable standards shall be developed to make thing communicate with each other and people. One approach would be the en-capsulation of smart wireless identifiable devices and embedded devices in web services. We can also consider the importance of enhancing the quality of service aspects like response time, resource consumption, throughput, availability, and reliability. The discovery and use of knowledge about services availability and of publish/subscribe/notify mechanisms would also contribute to enhancing the management of complex thing structures.

Enhanced monitoring facilities are needed to support informed decisions cooperatively adopted in collections of things. Also, increasing the quality of information collected from things will be the use of distributed bio-inspired approaches.

The huge number of things will make their management a very difficult task. New solutions are needed to enable things' adaptation, autonomous behavior, intelligence, robustness, and reliability. They could be based on new general centralized or distributed organizational architectures. Another solution will be endowing things with self-* capabilities in various forms: self-organization, self-configuration, self-Healing, self-optimization, and self-protection.

New services shall be available for persistent distributed knowledge storing and sharing, and new computational resources shall be used for complicated tasks execution. Actual forecasts indicate that in 2015 more than 220 Exabytes of data will be stored [38]. At the same time, optimal distribution of tasks between smart objects with high capabilities and the IoT infrastructure shall be found.

New mechanisms and protocols will be needed for privacy and security issues at all IoT levels including the infrastructure. Solutions for stronger security could be based on models employing the context-aware capability of things.

New methods are required for energy saving and energy efficient and self-sustainable systems. Researchers will look for new power efficient platforms and technologies and will explore the ability of smart objects to harvest energy from their surroundings.

Obviously, new platform architectures and developing techniques are needed for the efficient storing, real-time processing, data placement and replication of Big Data in order to achieve higher real-time guarantee for data provisioning and increased data storage efficiency. One promising solution is Cloud computing for Big Data. Cloud technologies simplify building Big Data infrastructure, support massive growth of storage capacity, and guarantee the performance agreed with customers (SLA). New Cloud storage and compute components will be needed to allow global data availability and access over different types of networks, for various cooperative user communities, which exploit different facets of Big Data, in application with different service level requirements (batch, online, real-time, event-driven, collaborative, etc.), and with different security needs going up to highly trusted and secure environments.

The large variety of technologies and designs used in the production of Things is a main concern when considering the interoperability. One solution is the adoption of standards for Things inter-communication. Adding self-configuration and self-management properties could be necessary to allow Things inter-operate and, in

addition, integrate within the surrounding operational environment. This approach is superior to the centralized management, which cannot respond to difficulties induced by the dimensions, dynamicity and complexity of the Internet of Things. The autonomic behavior is important at the operational level as well. Letting autonomic Things react to events generated by context changes facilitates the construction and structuring of large environments that support the Internet of Things.

Special requirements come from the scarcity of Things' resources, and are concerned with power consumption. New methods of efficient management of power consumption are needed and could apply at different levels, from the architecture level of things to the level of the network routing. They could substantially contribute to lowering the cost of Things, which is essential for the rapid expansion of the internet of Things.

Some issues come from the distributed nature of the environment in which different operations and decisions are based on the collaboration of Things. One issue is how Things convergence on a solution and how the quality of the solution can be evaluated. Another issue is how to protect against faulty Things including those exhibiting malicious behavior. Finally, the way Things can cope with security issues to preserve confidentiality, privacy, integrity and availability are of high interest.

5 Conclusions and remarks

Actual evolution of the Internet of Things towards connecting every thing on the planet in a very complex and large environment gives raise to high demanding requirements, which challenge the actual and future research. The continuously increasing volume of data collected from and exchanged among things will require highly scalable environments able to support the high resulting network traffic, and offer the necessary storage capacity and computing power for data preservation and transformation. Communication protocols are needed to enable not only the high capacity traffic but also maintain the connectivity between things even in case of transient disconnection of wired or wireless links. Also, new solutions should be found for efficiently store, search and fetch the data manipulated in these environments.

The chapter addresses new research and scientific challenges in context-aware environments for IoT. They refer first to the identification, internal organization, provision of context information, intelligence, self-adaptation, and autonomic behavior of individual things. Then, actual research and main challenges related to IoT infrastructure are discussed, with emphasis on services for context awareness, inter-communication, interoperability, inter-cooperation, self-organization, fault tolerance, energy saving, compute and storage services, and management of things collections and structures.

We also analyzed existing relevant work in the area of data dissemination in opportunistic networks for IoT. We began by highlighting the use of ONs in real life and describing some potential scenarios where they can be applied. Then, we pre-

sented the categories of a proposed taxonomy that captures the capabilities of data dissemination techniques used in opportunistic networks. Moreover, we critically analyzed four relevant data dissemination techniques using the proposed taxonomy. The purpose of the taxonomy, aside from classifying dissemination methods, has been to analyze and compare the strengths and weaknesses of the analyzed data dissemination algorithms. Using this knowledge, we believe that an efficient data dissemination technique for opportunistic networks for future IoT platforms can be devised. We believe that the future of opportunistic networking lies in the social property of mobile networks, so a great importance should be given to this aspect.

Finally, future trends and research directions for the IoT infrastructure are discussed including performance, monitoring, reliability, safety, survivability, self-healing, transparency, availability, privacy, and others.

Acknowledgements The work was partially supported by the project "SideSTEP - Scheduling Methods for Dynamic Distributed Systems: a self-* approach", PN-II-CT-RO-FR-2012-1-0084.

References

1. Athanasopoulos, D., Zarras, A.V., Issarny, V., Pitoura, E., Vassiliadis, P.: Cowsami: Interface-aware context gathering in ambient intelligence environments. Pervasive and Mobile Computing **4**(3), 360–389 (2008)
2. Bardram, J.E.: The java context awareness framework (jcaf)–a service infrastructure and programming framework for context-aware applications. In: Pervasive Computing, pp. 98–115. Springer (2005)
3. Bjurefors, F., Gunningberg, P., Nordstrom, E., Rohner, C.: Interest dissemination in a searchable data-centric opportunistic network. In: Wireless Conference (EW), 2010 European, pp. 889–895. IEEE (2010)
4. Boldrini, C., Conti, M., Passarella, A.: Contentplace: social-aware data dissemination in opportunistic networks. In: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems, pp. 203–210. ACM (2008)
5. Chen, C., Helal, S.: A device-centric approach to a safer internet of things. In: Proceedings of the 2011 international workshop on Networking and object memories for the internet of things, pp. 1–6. ACM (2011)
6. Chen, L.J., Sun, T., Liang, N.C.: An evaluation study of mobility support in zigbee networks. Journal of Signal Processing Systems **59**(1), 111–122 (2010)
7. Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J.R., Mellouli, S., Nahon, K., Pardo, T.A., Scholl, H.J.: Understanding smart cities: An integrative framework. In: System Science (HICSS), 2012 45th Hawaii International Conference on, pp. 2289–2297. IEEE (2012)
8. Christensen, H.B.: Using logic programming to detect activities in pervasive healthcare. In: Logic Programming, pp. 421–436. Springer (2002)
9. Ciobanu, R.I., Dobre, C.: Predicting encounters in opportunistic networks. In: Proceedings of the 1st ACM workshop on High performance mobile opportunistic systems, pp. 9–14. ACM (2012)
10. Conti, M., Crowcroft, J., Giordano, S., Hui, P., Nguyen, H.A., Passarella, A.: Routing issues in opportunistic networks. In: Middleware for Network Eccentric and Mobile Applications, pp. 121–147. Springer (2009)
11. Conti, M., Giordano, S., May, M., Passarella, A.: From opportunistic networks to opportunistic computing. Communications Magazine, IEEE **48**(9), 126–139 (2010)

12. Costa, P.D., Guizzardi, G., Almeida, J.P.A., Pires, L.F., van Sinderen, M.: Situations in conceptual modeling of context. In: EDOC Workshops, p. 6 (2006)
13. Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D.: Context is key. Communications of the ACM **48**(3), 49–53 (2005)
14. Cristea, V., Dobre, C., Costan, A., Pop, F.: Middleware and architectures for space-based and situated computing. International Journal of Space-Based and Situated Computing **1**(1), 43–58 (2011)
15. Cugola, G., Migliavacca, M.: Multicar: Remote invocation for large scale, context-aware applications. In: Proceedings of the The IEEE symposium on Computers and Communications, ISCC '10, pp. 570–576. IEEE Computer Society, Washington, DC, USA (2010). DOI 10.1109/ISCC.2010.5546718. URL http://dx.doi.org/10.1109/ISCC.2010.5546718
16. Denœux, T.: A k-nearest neighbor classification rule based on dempster-shafer theory. In: Classic works of the Dempster-Shafer theory of belief functions, pp. 737–760. Springer (2008)
17. Desta, M.S., Hyytiä, E., Ott, J., Kangasharju, J.: Characterizing content sharing properties for mobile users in open city squares. In: 10th Annual IEEE/IFIP Conference on Wireless On-Demand Network Systems and Services (WONS), Banff, Canada (2013)
18. Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-computer interaction **16**(2), 97–166 (2001)
19. Dobre, C., Manea, F., Cristea, V.: Capim: A context-aware platform using integrated mobile services. In: Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on, pp. 533–540. IEEE (2011)
20. Doukas, C., Maglogiannis, I., Tragas, P., Liapis, D., Yovanof, G.: Patient fall detection using support vector machines. In: Artificial Intelligence and Innovations 2007: from Theory to Applications, pp. 147–156. Springer (2007)
21. Greifenberg, J., Kutscher, D.: Efficient publish/subscribe-based multicast for opportunistic networking with self-organized resource utilization. In: Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on, pp. 1708–1714. IEEE (2008)
22. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. Journal of Network and computer applications **28**(1), 1–18 (2005)
23. Haddadi, H., Hui, P., Henderson, T., Brown, I.: Targeted advertising on the handset: Privacy and security challenges. In: Pervasive Advertising, pp. 119–137. Springer (2011)
24. He, J., Zhang, Y., Huang, G., Cao, J.: A smart web service based on the context of things. ACM Transactions on Internet Technology (TOIT) **11**(3), 13 (2012)
25. Henricksen, K., Robinson, R.: A survey of middleware for sensor networks: state-of-the-art and future directions. In: Proceedings of the international workshop on Middleware for sensor networks, pp. 60–65. ACM (2006)
26. Hui, P., Crowcroft, J., Yoneki, E.: Bubble rap: Social-based forwarding in delay-tolerant networks. Mobile Computing, IEEE Transactions on **10**(11), 1576–1589 (2011)
27. Hui, P., Yoneki, E., Chan, S.Y., Crowcroft, J.: Distributed community detection in delay tolerant networks. In: Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture, p. 7. ACM (2007)
28. Kalyan, A., Gopalan, S., Sridhar, V.: Hybrid context model based on multilevel situation theory and ontology for contact centers. In: Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on, pp. 3–7. IEEE (2005)
29. Kanda, T., Glas, D.F., Shiomi, M., Ishiguro, H., Hagita, N.: Who will be the customer?: a social robot that anticipates people's behavior from their trajectories. In: Proceedings of the 10th international conference on Ubiquitous computing, pp. 380–389. ACM (2008)
30. Kim, H., Cho, Y.J., Oh, S.R.: Camus: A middleware supporting context-aware services for network-based robots. In: Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on, pp. 237–242. IEEE (2005)
31. Kortuem, G., Kawsar, F., Fitton, D., Sundramoorthy, V.: Smart objects as building blocks for the internet of things. Internet Computing, IEEE **14**(1), 44–51 (2010)

32. Lenders, V., Karlsson, G., May, M.: Wireless ad hoc podcasting. In: Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on, pp. 273–283. IEEE (2007)
33. Lilien, L., Gupta, A., Yang, Z.: Opportunistic networks for emergency applications and their standard implementation framework. In: Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa, pp. 588–593. IEEE (2007)
34. McKeever, S., Ye, J., Coyle, L., Dobson, S.: Using dempster-shafer theory of evidence for situation inference. In: Smart Sensing and Context, pp. 149–162. Springer (2009)
35. Ngo, H.Q., Shehzad, A., Liaquat, S., Riaz, M., Lee, S.: Developing context-aware ubiquitous computing systems with a unified middleware framework. In: Embedded and Ubiquitous Computing, pp. 672–681. Springer (2004)
36. Pelusi, L., Passarella, A., Conti, M.: Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. Communications Magazine, IEEE **44**(11), 134–141 (2006)
37. Project, C.E.F.: Rfid and the inclusive model for the internet of things. http://www.grifs-project.eu/data/File/CASAGRAS FinalReport.pdf (2012). [Accessed July 15th, 2013]
38. Society, E.C.I., DG, M.: Infso d.4 networked enterprise & rfid, working group rfid of the etp eposs. internet of things in 2020. roadmap for the future. http://www.iot-visitthefuture.eu/fileadmin/documents/researchforeurope/270808_IoT_in_2020_Workshop_Report_V1-1.pdf (2009). [Accessed August 15th, 2013]
39. Tan, L., Wang, N.: Future internet: The internet of things. In: Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, vol. 5, pp. V5–376. IEEE (2010)
40. Truong, H.L., Dustdar, S.: A survey on context-aware web service systems. International Journal of Web Information Systems **5**(1), 5–31 (2009)
41. Van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. In: Proceedings of the 10th international conference on Ubiquitous computing, pp. 1–9. ACM (2008)
42. Vergori, P., Ntanos, C., Gavelli, M., Askounis, D.: The webinos architecture: A developers point of view. In: Mobile Computing, Applications, and Services, pp. 391–399. Springer (2013)
43. Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I.S., Mazura, M., Harrison, M., Eisenhauer, M., et al.: Internet of things strategic research roadmap. O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, et al., Internet of Things: Global Technological and Societal Trends pp. 9–52 (2011)
44. Wu, J.: Small worlds: the dynamics of networks between order and randomness - book review. SIGMOD Record **31**(4), 74–75 (2002)
45. Yau, S.S., Karim, F.: A context-sensitive middleware for dynamic integration of mobile devices with network infrastructures. Journal of Parallel and Distributed Computing **64**(2), 301–317 (2004)
46. Yau, S.S., Liu, J.: Hierarchical situation modeling and reasoning for pervasive computing. In: Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop on, pp. 6–15. IEEE (2006)
47. Ye, J., Dobson, S., McKeever, S.: Situation identification techniques in pervasive computing: A review. Pervasive and Mobile Computing **8**(1), 36–66 (2012)
48. Yoneki, E., Hui, P., Chan, S., Crowcroft, J.: A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, pp. 225–234. ACM (2007)