Chapter 8

Context Awareness

Up to this point, this book has focused on two fundamental aspects of geospatial computing: (1) methods to compute one's position and (2) methods to store and visualize location-referenced information. These elements provide the basis for *location awareness* and for creating and consuming mobile LBSs, a topic that was covered in detail in Chapter 7.

The next topic, *context awareness*, may at first seem like a departure from the topics covered in the previous chapters, which focused mainly on *location*. In fact, this chapter is a broadening of our focus because, as we shall see, location awareness falls under the larger umbrella of *context awareness*. We will consider various definitions of this term, but a precursory (and open-ended) definition might simply be that context awareness is the superset of location awareness and many other types of awareness about a mobile user's environment, activities, and state of being.

Although context awareness has existed as a computer science research topic for nearly 20 years, the topic has in recent years received increasing interest, largely due to the proliferation of mobile devices in everyday life, as discussed in Chapter 1. It is simultaneously an extremely promising yet challenging topic, given the wide range of capabilities it encompasses.

This chapter presents context awareness at a conceptual level and provides historical perspective on the development of the field. In addition, we attempt to show at a practical level how different aspects of contextual awareness can be implemented in a mobile device. Due to space limitations, we have selected one popular mobile platform, the Android operating system, in order to show specific code examples of several concepts. We will also point readers toward methods of the Android API classes that they can use to implement their own context-aware applications. Similar capabilities are available in other major mobile platforms, such as iOS and Windows Phone 7/8, but unfortunately space does not allow us to demonstrate them here.

8.1 DEFINING CONTEXT AND CONTEXT AWARENESS

Context is one of those words, like system and information, that is difficult to define precisely. It is used prominently in many fields, including linguistics, psychology, neuroscience, law, and computer science, but ironically the definition of context depends heavily on its context of use. Loosely speaking, we can say that context is a category of information, but there is no general agreement on which types of information belong to this category.

Some researchers have studied the etymology of the word context and have even attempted to formalize and to build consensus concerning the definition [1–4]. Although these semantic topics are beyond the scope of this chapter, we require some working notional definition of context before we can define and examine context awareness (this chapter) or discuss *contextual reasoning* (Chapter 9).

Note that whatever definitions we adopt, they will surely reflect the particular focus of this book, namely mobile geospatial computing. In fact, in defining these concepts, we notice a definite trade-off between generality and concreteness. We have tried to be as general as possible without losing our focus on how context awareness can be implemented in mobile devices.

8.1.1 What Is Context?

In the *Merriam Webster Dictionary* [5], we find two definitions of the word context:

- The parts of a discourse that surround a word or passage and can throw light on its meaning;
- The interrelated conditions in which something exists or occurs: ENVIRONMENT, SETTING.

In this book, we adopt the second definition because we are not directly concerned with human discourse but rather with conditions of an environment or setting (i.e., geospatial information) that can be sensed by machines (i.e., computers and sensors). Clearly, these two definitions are interrelated in the sense that discourse can be (and most usually is) used as a representation of an environment or setting.

In other words, natural language is a common form in which contextual information is encoded. Our focus though is on techniques to sense and represent context automatically. Hence, when we refer to *context*, we refer directly to the conditions in the environment. When we use the term *contextual information*, we refer to representations of context, either in natural language or other form. Furthermore, when we want to explicitly distinguish between the two types of context defined above, we will use the terms *discourse context* and *conditions context*, respectively.[17]

In lieu of a formal and generally accepted definition of context, the above dictionary definition (second one) is broad enough to serve as a working definition for use in exploring context awareness. We will further elucidate this concept with a framework and example below.

There are a few general notes regarding context to cover before we dive deeper into the topic. First, note that our definition implies that the specific “conditions” relevant to context are dependent on the “something” to which the context applies. In other words, context is always specified from a particular viewpoint or *frame of reference* of an object or person. This is similar to the concept in physics, where a physical measurement like velocity is defined within a particular frame of reference. Within the domain of mobile geospatial computing, our frame of reference most often will be the mobile device itself, although oftentimes we take the frame of reference to be the same as that of the mobile user.

Second, we must choose some techniques for describing a particular context that help to build a framework for our contextual information. If our goal is to describe a particular context in natural language, then we might employ the classic technique of journalism (since journalism is an age-old craft for describing conditions and events), known as the *five Ws*: who, what, where, when, and why [6]. In fact, this technique dates back at least to the late second century BC when Hermagoras of Temnos defined seven elements of circumstance, which include (in addition to the five Ws) “in what manner” and “by what means” [7].

Using these questions as a starting point (with a slightly different order), we list possible elements of a particular context with a demonstrative example:

What: A small, impromptu gathering of colleagues.
Who: Mary, a smartphone user, as well as three of Mary’s coworkers who are nearby.
Where: 60.1609°N, 24.5460°E (WGS84); inside the main lobby of the FGI, specifically inside Mary’s pocket.
When: Friday, 20 April 2014 at 12:03 p.m.
Why: This gathering occurred because Mary and her colleagues are going out to lunch together. They are waiting for a fifth colleague, Steve, to arrive.

---

[17] The word context is also used in some image analysis literature to refer to pixels in the vicinity of a particular pixel (e.g., the neighbors of that pixel). See references [8–10] for examples. This choice of terms is, in fact, related to the *discourse* definition of context; just as the surrounding discourse can shed light on a particular word or phrase, the surroundings of a particular pixel can help in interpreting its value. In any case, this meaning of context is in some ways different from conditions context, and we mention this usage merely to caution the reader not to confuse the two.

In what manner: Mary’s smartphone is experiencing small, sporadic movements, consistent with the phone being in the pocket of someone who is standing and having a casual conversation.

By what means: All of the above information has been sensed or reasoned by the sensors and software existing in a smartphone, or acquired via a networked resource. In this case, the smartphone is a Samsung Galaxy S4 with Android 4.3 OS, which includes a GNSS receiver, WLAN-based positioning engine, Bluetooth module, microphone and audio analyzer, ambient light sensor, accelerometers, gyroscopes, and magnetometers.

This account of the situation is not likely to yield a Pulitzer prize in journalism, but it is enough to get a basic sense about what the mobile user is doing, as viewed from the perspective of her mobile device. We will see later how this contextual information can be used to create context-aware applications that record a user’s activities and even anticipate his or her needs.

There are of course many other contextual elements we could list, including details about the smartphone itself (e.g., battery level and mobile network that it is connected to) or about the immediate surrounding environment (e.g., the weather conditions outside). Like a good journalist, however, we only include the most relevant facts that are necessary to understand the situation at hand. If, for example, the weather outside were particularly noteworthy, we might have included it in the list, but we can assume from its omission that most probably it is nothing exceptional or affecting to the situation.

Last, we note that there are some elements omitted from this list that are certainly both contextual and relevant, such as details about the user’s state of mind or what the topic of conversation is. This illustrates the point that our contextual information will in some ways always be incomplete. In the domain of mobile geospatial computing, we are limited to the information that can be obtained or reasoned by a mobile device. As mobile technology rapidly advances, the breadth of information available to mobile devices will surely increase, but it will always face limitations, especially compared to the robust and subtle sensing ability of the human being.

8.1.2 What Is Context Awareness?

Now that we have a basic notion of context and a simple example to aid our understanding, we can define the term *context awareness*. Although this term inherits the same difficulties facing the term context, it is otherwise straight forward to define. Context awareness is the quality of having knowledge (being aware) of context. We say a device or application is *context-aware* if it possesses this quality, and we define *context-aware computing* as the set of computational techniques designed to obtain and use context.

Using the journalistic framework described above, we can describe one objective of context-aware computing as the building up of a description of a particular situation or context. As an ultimate goal, this description should be indistinguishable from the description that a perceptive (human) journalist would compose. Thus, the process can be roughly framed as 1) acquiring information from sensors and other sources and 2) translating this information into prose (or ordinary spoken language).

Just as a person can have different levels of awareness, a device or application can have different levels of context awareness. To a certain extent, nearly every mobile device is context-aware in the sense that it knows certain details about its state (e.g., battery level, networks it is connected to, and whether or not a phone call is taking place with it) and perhaps about its user or owner (name, phone number, etc). We arbitrarily distinguish, however, between context-aware and non-context-aware devices based on the amount and breadth of contextual information that the device can obtain and whether it possesses sensors and software that are employed specifically for the purpose of obtaining contextual information. Admittedly, this is a blurry definition, and this type of classification is often a matter of opinion.

Don't be surprised if some of this starts to sound a bit "sci-fi." The goal of creating a context-aware device is related to one of the primary prerequisites of artificial intelligence (AI). A machine exhibiting AI must have a general representation of the world, especially concerning the objects and beings in its immediate surroundings [11]. We shall see, however, that a device can achieve some elements of context awareness without even coming close to achieving AI. Hence, the two subjects are related but distinct.

Last, in order to aid the reader's understanding, we present briefly an example of a real-world, present-day context awareness product—Google Now. The web site for this product provides a succinct description of its context awareness capabilities [12]:

> *Google Now gets you just the right information at just the right time.*
>
> *It tells you today's weather before you start your day, how much traffic to expect before you leave for work, when the next train will arrive as you're standing on the platform, or your favorite team's score while they're playing. And the best part? All of this happens automatically. Cards appear throughout the day at the moment you need them.*

Of course, none of these capabilities would be possible without a source of contextual information, such as the user's position, preferences, or activity (e.g., leaving for work). It is precisely the acquisition of this type of information that we will explore further in this chapter; we will learn how to process it in Chapter 9.

8.1.2.1 Is Location Awareness Part of Context Awareness?

As stated above, location awareness is one element of context awareness, thus we could claim that any device that is location-aware is therefore, to a certain extent, context-aware. In fact, location awareness can be viewed as an important pioneer and forerunner in context awareness. There is no doubt that location is one of the most important elements of context, hence the focus of much of this book on positioning methods. This is one aspect of the "geospatial era," discussed in Chapter 1, where many forms of computing are adopting location as a central contextual element. The most recent advances in mobile technology, however, point to a future where mobile devices will be aware of much more than just location. In fact, positioning technologies are becoming so ubiquitous that a location-aware device may not seem noteworthy enough to merit the special distinction of being context-aware. Indeed, our view is that location awareness is a necessary precursor to context awareness (at least for the vast majority of applications) but does not in itself imply context awareness.

8.1.2.2 Relation Between Context Awareness and Situation Awareness

Some readers may note that a related term *situation awareness* (or *situational awareness*) is commonly used in certain disciplines. The distinction between context awareness and situation awareness is largely of a historical nature rather than conceptual. The former comes from mobile computing researchers, whereas the latter is rooted in military aviation.

Originally, situation awareness referred to the pilot's "awareness of conditions and threats in the immediate surroundings" [13], and its importance was known as early as World War I, especially by German flying ace Oswald Boelke [14]. Eventually situation awareness was also used to refer to machine awareness of such conditions and threats. Furthermore, the term became popular in a plethora of fields outside of aviation applications, including maritime[18] or ground-based operations, and even outside of military applications [15]. Hence, today the concepts are essentially the same, and the two terms can be used almost interchangeably. Because context awareness is more commonly used in mobile computing, whereas situation awareness is more commonly used in military, safety, and security applications, we will primarily use the former in this book.

8.2 WHY IS CONTEXT AWARENESS IMPORTANT?

Now that we have a basic understanding of the concept "context awareness," it is worth asking the question, "Why is context awareness important?" Given its

---

[18] Also known as maritime domain awareness.

prominence as a topic in computer science research, there should be compelling reasons for why it is studied.

As Anind Dey, one of the pioneers of context awareness, and others have pointed out, the goal of creating context-aware devices is largely about improving the richness and ease of human-computer interaction [16]. If a computer can be made more aware of the conditions and environment in which it and its users reside, then there will be less need for the users to explicitly tell the computer what it is they need the computer to do. To a certain degree, the computer will "know" what our needs are. This is because developers will have prescribed what is typically needed in a given situation into an application designed to serve such needs (or allowed the user to specify his or her needs in given situations).

A simple example could occur when a mobile device detects that you will be late for a meeting (because you are 30 minutes by car from the meeting place with only 15 minutes until the meeting starts), and could automatically send a message to a contact saved in your mobile device and identified in your mobile calendar as the organizer of the meeting. Therefore, to a large extent, context awareness is about increasing the automation capabilities of our computing devices, or *minimizing* the amount of human-computer interaction that is required to perform a given task.

Many other examples of such context-enabled capabilities could be listed here, but like many worthy scientific and technological endeavors, it is impossible to say what the most important reason for pursuing context awareness is. It is a bit like speculating in the early 1960s about the importance of the Internet (or Intergalactic Computer Network, as it was known in an early phase) that we see so clearly from our vantage point in the twenty-first century. Certainly at that time computer scientists could see the value of networking together four to eight mainframe computers to share resources and transfer data, but they could hardly envision at that time how such entities as Google, Facebook, Twitter, and Skype would change the way that billions of people operate on a daily basis; how businesses would operate; and even how revolutions would be organized! Similarly, although we can see some of the immediate promises of context awareness for mobile computing, we must rely on our intuition and curiosity to believe that making computers aware of context is an effort that will pay off in countless, unimaginable ways.

8.3 HISTORY OF CONTEXT-AWARE COMPUTING

It is worthwhile to review the history of context awareness from the literature and see how the concept has developed and been employed. Reviewing this history is challenging, however, due to the wide and varying use of the concept "context," as discussed above. Some argue that the topic of context can be traced back to

Frege [18, 19] or Russell [20], where the role of context in determining the meaning of symbols and words is discussed [17–20].[19] These philosophical issues regarding the word "context" are not, however, essential for a practical understanding of context awareness, so we will only mention them in passing during this brief review.

In a similar vein, we can find studies from the 1950s and 1960s, where context is used in various applications, such as machine recognition of handwriting, printed text, and verbal speech [21–23]. These works, however, are primarily concerned with discourse context rather than conditions context. Discourse context is fundamentally different from conditions context because in the former there is (by definition) a human in the loop, whereas representations of the latter could be generated automatically.[20] As we stated at the onset, we are interested in automatic methods; hence these studies are only of peripheral interest.

In some sense, however, discourse (i.e., natural language) is relevant to context awareness because it can be used to infer conditions context. Thus, *natural language processing* can help provide inputs to context-aware applications. A full survey of natural language processing techniques is beyond the scope of this book, so the reader is referred to Porzel's extensive coverage of the topic in [24].

As mentioned above, the subject of context awareness exhibits strong ties to the field of AI. As early as 1963, one of the "fathers of AI," John McCarthy, described the concept of a *fluent* (later called *situational fluent*), which we might also call a *contextual proposition* or *contextual statement*. A fluent is expressed as a function of a *situation*, which is described as "the complete state of affairs at some instant of time." One simple example of representing a fluent that McCarthy provides is `raining(s)`, which indicates that it is raining in situation `s`. From this example, we can clearly see the parallel between a set of *fluents* for a particular situation and a context [25].

It is curious to note, however, that McCarthy does not explicitly use the term context until a 1987 paper titled "Generality in Artificial Intelligence" [26]. Even then, he does not provide an explicit definition of context nor comment on its relation to the concepts of fluent or situation. Nonetheless, this paper seems to be the spark that set off an intense effort to study and even formalize the concept of context. By the early 1990s, we can find important works on formalizing context and on contextual reasoning [1, 2, 27, 28]. Although this literature does not

---

[19] Russell does not explicitly use the term "context," but one can recognize its implicit presence.

[20] Computers, of course, can simulate human discourse and can even generate discourse about the conditions context. Still, we feel justified in making this distinction because it seems that a computer capable of human-like discourse requires first some knowledge of at least some elements of conditions context. Hence, computer-generated discourse would constitute a particular use of conditions context, rather than a method of acquiring conditions context.

directly herald an era of context-aware devices, it constitutes important theoretical groundwork in the development of context awareness as a research topic.

Another important precursor to context awareness can be seen in the often cited article "The Computer for the 21st Century," published in *Scientific American* [29]. In this highly readable article, Weiser envisions a future state of technology where context awareness plays a crucial (albeit unspoken) role:

> *Sal awakens; she smells coffee. A few minutes ago her alarm clock, alerted by her restless rolling before waking, had quietly asked, "Coffee?" and she had mumbled, "Yes."*

This fictional account goes on to describe a number of automated and assisted functions that are performed by *ubiquitous computing* devices, which in many cases can be considered to be context-aware. When this article was published in 1991 it probably sounded like mostly science fiction. With the recent advent of wireless devices like Fitbit (which can monitor sleep periods), such scenarios are now on the verge of reality.

For the first reported implementation of a context-aware device, we look to the Active Badge Location System, developed at the Olivetti Research Laboratory in Cambridge, England, in the early 1990s [30]. Not surprisingly, this system was focused exclusively on creating *location awareness*, especially for employees in an office environment. A few years later, Schilit and Theimer were the first to explicitly use the term context-aware computing, which they defined as "the ability of a mobile user's applications to discover and react to changes in the environment they are situated in" [31].

By the mid 1990s, the study and development of "context-aware applications" becomes an active area (e.g., PARCTAB, stick-e notes [32], CyberGuide [33], CyberDesk [34]). The reader can find an excellent snapshot of the turn-of-the-century, state-of-the-art in context-aware computing in a special issue of the journal *Human-Computer Interaction*, published in 2001 [35]. What is clear from the literature of this period is that context awareness had become a strong research focus for several research groups, typified by prototype and architecture development. It is also clear, however, that an abundance of challenges still remained to achieving rich and robust context awareness.

During the past 10 years, context awareness research has continued to press forward, and it has begun to work its way into several commercial products. Several frameworks for context awareness have been developed by researchers. One challenge that remains, however, is that no widespread standard yet exists for representing contextual information.

As far as commercial products, the example of Google Now was given above, but there are plenty of other examples as well. Dexetra currently offers an application called "friday," which keeps track of a user's daily activities, such as commuting, activity on social networks, and various uses of the user's phone (calls, photo-taking, etc.), as well as the location where these features were used.

Qualcomm Retail Solutions offers a context awareness platform, called Gimbal, with SDKs available for iOS and Android. Already in 2012, Samsung's Galaxy S3 smartphone was marketed as a context-aware device with its ability to recognize when a face is looking at the screen and control whether the screen stays on base on this information. The Motorol Moto X, released in 2013, also exhibits many context-aware features. These include sensing when the phone is picked up (in order to turn on the screen and display notifications) and sensing certain hand gestures (in order to activate the camera application). An application called Motorola Assist[21] uses context to detect activities like driving and sleeping. It allows users to predetermine how to handle or respond to incoming text messages and phone calls, based on the current activity (e.g., reading text messages to the user while driving).

8.4 EXAMINING CONTEXT IN DETAIL

The remainder of this chapter examines in more detail the various elements of context first introduced in Section 8.1.1. Some of these elements are more straightforward to detect in a mobile device than others. For the more challenging elements, we will provide only precursory coverage in this section.

8.4.1 What: The Activity Context

The first element of context is a high-level description of what activity is taking place or what the overall current situation is.[22] Examples could include: driving a car, waiting at a bus stop, having lunch, and participating in a meeting. In practice, there can be many possible ways to formulate such high-level activity or situation descriptions, as well as many different techniques for classifying and sensing them. The overall objective of classifying and sensing activities is often called *activity recognition*, and a great deal of research has been conducted in this area alone.

The optimal set of techniques to detect high-level context is specific to the set of sensors or other data available, the scope of activities that one desires to detect, as well as the specific requirements of the application (such as whether or not the context must be detected in real time). Therefore, it is very difficult to provide general guidance on how to perform activity recognition. In broad terms, however, we can use one or more techniques from the fields of *machine learning* and

---

[21] At the time of this writing, Motorola Assist was compatible with four different Motorola smartphone models, including the Moto X.

[22] We recognize that "situation" is essentially a synonym for context, but it is used here to embody the highest level of contextual information that functions as a summary of the context. We also note that a context may sometimes be characterized by a great deal of *inactivity*, but even "sitting idly" can be one classification within a set of activity classifications.

pattern recognition. Specifically, the following signal processing techniques have been used for activity recognition:

- Decision tree;
- Hidden Markov model;
- Naïve Bayes classifier;
- Support vector machine;
- Kalman filter;
- Particle filter.

We will provide further details on some of these techniques in Chapter 9. Machine learning and pattern recognition are fields with a rich set of literature, and we can only scratch the surface in this book. Readers wishing to go deeper are encouraged to explore especially the references cited in Chapter 9.

8.4.2 Who: The User and Social Context

The second element of context is concerned with 1) who the user of the mobile device is and, possibly, 2) the social context in which the user is situated. Acquiring the identity of the mobile user is usually dependent on information the user has stored in the mobile device and/or on privacy permissions that the user has set to control access to various account information. For example, on Android devices a user can create a local profile (see code example below), or personal information can be imported via an account, such as Gmail, Facebook, or Twitter. Most major web services offer an SDK for Android that allows privacy-controlled access to profile information (e.g., Facebook SDK for Android [36]).

```
public HashMap<String, String> getProfileInfo() {
   HashMap<String,String> profile = new HashMap<String,String>();
   ContentResolver cr = getContentResolver();
   Cursor cursor = cr.query(
      // Specify the URI of the "Profile" contact.
      Uri.withAppendedPath(
      ContactsContract.Profile.CONTENT_URI,
      ContactsContract.Contacts.Data.CONTENT_DIRECTORY),
      Query.PROJECTION, null, null, null);
   cursor.moveToFirst();

   // Loop through each cursor row, appending key/value to HashMap
   while (!cursor.isAfterLast()) {
      String key = null;
      // Determine key for HashMap based on MIMETYPE column in
      // cursor row. Example shown for name field. Other fields
      // left for reader to implement as desired.
      if (cursor.getString(Query.MIMETYPE).contains("/name")){
```

```
         key = "email";
      }
      if (key != null){
         profile.put(key, cursor.getString(Query.DATA));
      }
      cursor.moveToNext();
   }
   return profile;
}
```

The amount and type of information that can be obtained via these methods varies greatly based on the particular information source. For each particular user, it also depends on what information that user has entered into his or her profile and what accounts are accessible on the device. Therefore, it is important to implement fallback methods, such as manual prompts, in case required user information is not available via automated methods.

8.4.3 Where: The Location Context

As discussed above, location is an important element of context. Since much of this book is concerned with describing mobile-based techniques to obtain a user's position, we will not repeat such descriptions here. In the Android platform, there are various techniques used to obtain geographic coordinates (i.e., latitude, longitude, and altitude) that hide much of the underlying complexity used to calculate them. Most importantly, the `LocationManager` class provides methods to obtain location (i.e., position) updates from either a GNSS[23] or a network-based provider. Table 8.1 below highlights the most important methods from this class.

It is important to note that location can be expressed in many different forms, such as in geodetic coordinates (see Section 2.1) or semantic descriptions like "at home." From the perspective of context awareness, it is often such semantic descriptions that provide the most readily useful information. In this form, the context is rather self-evident, whereas the contextual meaning of geodetic coordinates (e.g., latitude and longitude) must be interpreted.

This interpretation could be relatively straightforward, such as querying a database of points of interest (recall this topic from Chapter 7), or it could be complex, such as analyzing the location history of a particular user to infer the relevance of the location. In either case, it is clear that the relevant information is not the geodetic coordinates themselves but rather the significance of the location which the coordinates reference.

---

[23] As noted in Chapter 1, GNSS is the preferred term when referring to satellite-based navigation systems in general. In Android, there are classes and methods using the acronym GPS when most modern mobile devices have multi-GNSS enabled receivers.

There is one other location-related topic that has not been covered elsewhere in this book, that of *microlocation sensing*. This refers to the use of proximity-based positioning (recall Section 2.2.6) to detect that a mobile device is within a very small-scale region (e.g., centimeter and meter scale), such as near a user's desk or near a particular display in a store. This concept has also been referred to as "hyper local." As the size of the region becomes smaller, oftentimes the context becomes clearer. For example, if a mobile device has localized itself as lying on the nightstand next to the user's bed, then the context is likely related to sleep. Therefore, this topic is of particular relevance to context awareness.

Table 8.1

List of Useful Methods from `android.location.LocationManager` Class

Property	*Description*
addGpsStatusListener	Adds a GNSS status listener, which is powerful way to get low-level updates from the GNSS receiver.
addNmeaListener	Adds an NMEA status listener, which can receive NMEA sentences as they are output from the GNSS receiver.
addProximityAlert	Adds an event listener that is triggered when the device enters or exits a given circular region (supplied as latitude/longitude/radius).
getAllProviders	Returns a list of all available location providers.
getGpsStatus	Returns a GpsStatus object, containing the current state of the GNSS engine. Used in conjunction with `addGpsStatusListener`.
getLastKnownLocation	Returns a location object, containing the last known location from a specified provider (i.e., GNSS or network).
requestLocationUpdates	Used to set up events, which are triggered when a location update is available from the specified provider (i.e., GNSS or network).

There are several technologies related to microlocation sensing. For example, in late 2013 Apple starting using devices called iBeacons in its stores to detect at a fine scale where customers are located. These beacons use Bluetooth low-energy RF signals to detect when mobile devices are nearby and communicate with them accordingly. Qualcomm also offers devices called Gimbal Proximity Beacons, which operate in a similar fashion but are designed to interact with the Gimbal platform (mentioned in Section 8.3). Bluetooth low-energy is not the only signal that is used for microlocation sensing. Samsung also sells an add-on product for its mobile devices, called TecTiles, which are programmable NFC stickers. They can be programmed to interact with a mobile device whenever it is within NFC range (a few centimeters), allowing for extremely fine-scale localization. Imagine the potential applications if such NFC were deployed more widely. NFC tags could be integrated into such items as pockets in clothing and in handbags, laptops and desktop computers (e.g., for identification purposes), and cashier or payment systems. The possibilities are endless!

8.4.4 When: The Time and Date Context

The next element of context is trivial to obtain from a mobile device's OS. For example, in Android you can get the device's system time with two lines of code:

```
SimpleDateFormat sdf = new SimpleDateFormat(
   "MM/dd/yyyy HH:mm:ss");
String dateTime = sdf.format(System.currentTimeMillis());
```

A string representing the date and time (in local time zone) will be stored in the variable `dateTime` for use in your application. The difficult part can come in figuring out how to use this information.

A date and time coordinate must be interpreted in much the same way a location coordinate is interpreted. For example, 6:00 a.m. for one person might represent a normal weekday wake-up time, whereas for another person it might represent a precious "Do not disturb" sleep time. Similarly, 5:00 p.m. on December 21 might represent an entirely different context than 5:00 p.m. on June 21 (especially at higher latitudes). Last, significant dates, such as holidays and personal events (birthdays, anniversaries, etc.), must be inferred using some additional information source. These difficulties make inferring the context from a raw time and date measurement a nontrivial matter.

There are various techniques available for inferring the contextual relevance of time and date information. The most obvious choice is by utilizing the built-in calendar application with which most (if not all) mobile devices are equipped. The drawback of this approach is that it relies on information supplied by the user, which may not be complete or up-to-date. In most cases, however, it is the easiest of the available options to implement.

An alternative approach would be to analyze the activity history of a user to infer the likelihood that he or she would perform a certain activity, such as commuting to or from work, during certain times, days of the week, or dates of the calendar year. This approach, however, may not yield detailed enough information for some applications, and it is considerably harder to implement.

8.4.5 Why: The Motivational Context

The fifth element of context—the *motivational context*—is perhaps the most elusive of them all. It involves analyzing one or more other contextual elements in order to infer a user's intentions. The techniques used to perform such inferences are largely dependent on the specific contextual domain or application where the inferences will be utilized. We can already see several minor examples of motivational context being applied in commercial devices, although clearly we are in the early stages of development in this regard. For example, Motorola's Moto X smartphone uses its rich set of sensors and specialized processors, as well as the features of Google Now, to infer the needs and desires of its users. This includes sensing when the phone is picked up (in order to turn on the screen and display notifications) and sensing certain hand gestures (in order to activate the camera application).

In general, techniques from machine learning and pattern recognition could be used. It is clear, however, that further research and development is needed before motivational context can be inferred and utilized on a large scale. To build up accurate models of human behavior and human intent, it may be necessary to generate large amounts of training data, perhaps using a crowd-sourcing approach. This topic will be considered further in Chapter 10.

8.4.6 In What Manner: Motion Context and Other Details of Context

The sixth element of context can rightly be judged as somewhat of a catch-all of other contextual information. It is formed from the sixth component of Hermagoras's *circumstance* and can be used to groups together additional details of a situation that do not fit into the other elements. For example, if the answer to *what* is the activity "dancing," then *in what manner* might include such concerns as the type of dance and the tempo of the dance. In particular, motion attributes fit nicely into this category. For example, one can use motion recognition techniques to detect the particular mode of motion and to describe other motion attributes such as speed, heading, and acceleration.

8.4.7 By What Means: The Context-Aware Device and the Methods of Sensing Context

The last element of context includes descriptions of the context-aware device itself, as well as the methods employed to sense the context. Given the wide range of mobile devices available today (recall Chapter 1), it is important for an application to be aware of details about the device it is operating on. This will supply the application with information about the capabilities of the device and about what other contextual information might be available to it. Some details about the state of the device can even constitute a situation or context themselves (e.g., low battery context).

Basic information about the device for use in an application, such as device name, manufacturer, model, operating system, and build version, is generally available via the OS's APIs. For example, the Android APIs make available a set of system properties in the class `android.os.Build`. See Table 8.2 for a list of some of these properties, which may be useful for describing the mobile device.

In addition, it is generally straightforward to obtain contextual information about the data communication functionalities of the mobile device, such as whether or not a call is in progress and what cellular network or WLAN the mobile device is connected to. In Android, one can use the `TelephonyManager` class to obtain information related to the cellular radio. Examples of useful methods available from this class are shown in Table 8.3. Similar information about the WLAN radio of an Android device can be obtained using the `WLANManager` class. Alternatively, if one is interested in data connectivity regardless of whether it is through mobile networks, WLAN, Bluetooth, or other means, the `ConnectivityManager` provides basic information about the connected or available data networks.

Table 8.2

List of Useful System Properties (Fields) from `android.os.Build` Class

Property	*Description*
Device	Codename given to the device (e.g., maguro)
Brand	Brand-customized version of the operating system (e.g., Verizon)
CPU	The name of the instruction set (CPU type + ABI convention) of native code (e.g., armeabi-v7a)
Display	An ID representing the build version of the operating system (e.g., JRN84D)
Manufacturer	Manufacturer of the device (e.g., Samsung)
Model	Model name of the device (e.g., Galaxy Nexus)
Product	Codename given to the firmware version of device (e.g., takju)
Radio	Version of the cellular radio firmware (e.g., I9250XXKK6)
Serial	Serial number of the device (e.g., 014682070502301D)
Version.release	Version number of the device's operating system (e.g., 4.1)

Other information about the device's current state or condition, such as the current battery state, amount of memory available, and current usage of the CPU, is often also available. For example, in Android, detailed information about the battery state can be obtained from the BatteryManager class. For memory usage, use the ActivityManager. For CPU usage, there is no specialized method for obtaining overall usage, but since Android is based on Linux, one can use the `/proc/stat` file to obtain the desired information.

In addition to these classes and methods, Android offers a powerful technique to make applications aware of various actions or events triggered on the device, known as *intent messaging*. Intents are essentially intradevice messages, and by setting up *intent filters* in an application, the app can receive these messages, containing various types of information depending on the type of intent. A small sampling of intents that are triggered automatically by different system events includes: answering a phone call, pressing the "call" button, unlocking the keyguard, starting the camera application, plugging in a headset, switching airplane mode on or off, docking into a desktop or car dock, installing or removing an application, rebooting or shutting down the device, low battery warning, and low storage space warning.

Table 8.3

List of Useful Methods from `android.telephony.TelephonyManager` Class

Method	*Description*
getCallState()	Returns an integer representing the call state of device (e.g., CALL_STATE_RINGING = 1)
getDataState()	Returns an integer representing the data state of device (e.g., DATA_CONNECTED = 2)
getDataActivity ()	Returns an integer representing the current data activity of the device (e.g., DATA_ACTIVITY_OUT = 2)
getDeviceId()	Returns an integer that uniquely identifies the device (IMEI, MEID, or ESN) (e.g., 352563176148781)
getNetworkOperatorName()	Name of the cellular network operator (e.g., FI SONERA)
getNetworkType()	Returns an integer representing the type of cellular network currently in use (e.g., NETWORK_TYPE_UMTS = 3)
getPhoneType()	Returns an integer representing the type of radio used to transmit voice calls (e.g., PHONE_TYPE_GSM = 1)
getSimSerialNumber()	Serial number of the SIM card (e.g., 8893579030109339463)
getSubscriberId()	Unique subscriber ID for mobile customer (e.g., IMSI)
isNetworkRoaming()	Returns a Boolean value depending on network roaming state

Finally, custom intents can also be implemented, which in effect allows any software component in the device to be aware of any action of any other component, provided the appropriate intents and intent filters are specified. Indeed, intent messaging provides a powerful means to implement various types of context awareness in Android devices.

8.5 HOW TO USE CONTEXT

Now that we've seen an overview of the various elements of context, we will look at how context can actually be utilized in context-aware applications. In order to demonstrate this, we return to the fictional scenario outlined in Section 8.1.1, where Mary and her colleagues are waiting for their colleague Steve to arrive, in order to leave together for lunch.

It turns out that the colleagues had gathered for lunch after responding to an "*AreUin?*" request from Mary, a mobile app that they frequently use in their department to see who is interested in joining for lunch or for after-work social outings. Steve had answered the request with "*I'm in*," so it is a bit odd that he hadn't shown up in the lobby yet.

Timo says to Mary, “Did you try calling Steve?”

“Yes, but he didn’t answer. *Locator* says that he’s in the lab. It could be though that he left his phone there because it shows as completely static for the past hour and connected to a PC. I’ll run and check.”

Meanwhile, Timo starts the “*RestaurantFinder*” app on his phone to look for possible restaurants. The app sorts nearby restaurants according to the preferences of those who are joining for lunch, whose identities have been sent from the “*AreUin?*” app. *RestaurantFinder* shows at the top of the list that a new Mexican restaurant opened up this week—only a 10-minute drive from their work—and that Mexican food ranks highly among the group’s preferences.

“There’s a new Mexican place that opened up on Turuntie. You guys interested?” asks Timo.

“Sure,” says Anindya, as the others also nod approvingly. At that moment, Mary returns alongside Steve.

“Sorry guys, I had my headphones on and was working at the soldering table. Lost track of time,” explains Steve.

“No worries, Steve. But you do know that AreUIn has a built-in reminder feature, right?” replies Timo with a wink of the eye.

“Really? I didn’t know.”

“Yeah, it can even connect to your smartwatch, in case you don’t have your phone on you. So you can rock out to Zeppelin and forget about us…until it vibrates as a reminder,” jokes Timo. “Anyways, it’s good Mary found you because there’s a new Mexican place nearby called Pancho’s, and I already checked to see if there’s a table open. Not only was the answer yes, but they just sent me a coupon. 20% off for groups of five or more! That sound OK to you, Mary?”

“Sounds great. I have a big car, so I’ll drive. Can you send me the address?”

“Done. Ok, let’s go!”

As our characters walk to the parking lot, Mary’s smartphone has already started a navigation application to show the way to “Pancho’s.” Let’s turn away for a moment and take a look at how context has been utilized in this story.

First, Steve’s position—more precisely the position of Steve’s phone—was determined by the locator app. Furthermore, Mary had some contextual information that Steve’s phone had been idle for some time and connected to a

PC. This is an example of how motion patterns and information about the phone's state can be used to provide additional details beyond mere position.

In addition, this example illustrates an important point; when the smartphone is the sole sensing platform, the inference of *user context* is dependent on the smartphone being in close proximity to the user. If Steve had simply left his phone in the lab and was in another location, Mary would have been out of luck. This is one of the reasons why context awareness may shift in the future toward wearable devices, such as smartwatches. Such wearable devices can still communicate and cooperate with other mobile devices, but in addition they have considerable sensing and networking capabilities of their own. This technology is still in its infancy, but already nearly a dozen manufacturers are selling various incarnations of this idea. Wearable devices and sensors will be discussed further in Chapter 10.

Next, Timo used the *social context*, namely the identities supplied by the "*AreUIn?*" app and their preferences stored by the *RestaurantFinder* app, to select a suitable restaurant. *RestaurantFinder* also used the approximate position of Timo's phone to filter the choices, which is a typical feature in LBSs.

Last, Timo performed a context-dependent query to determine if the restaurant currently had a table available for five people. The restaurant replied in the affirmative and also appears to have engaged in what we might call *context-sensitive marketing*: based on the busyness of the newly opened restaurant and the opportunity to bring in five potentially regular customers, it offered a discount coupon. Let's return to the story to see if context was further employed during this lunch outing.

After their meals at Pancho's were served, the jovial colleagues got into a lively discussion about soccer. "Barcelona is going all the way this year. Messi is healthy now and playing incredibly well," argues Steve.

"They will go far, but my pick is Bayern Munich. The team is just as strong as last year's, and Pep Guardiola is doing a great job as manager," counters Timo.

"Hey guys, look." Mary holds up her phone to show a notification. The notification shows that the monthly department meeting starts in 15 minutes.

"Oh, man. I totally forgot. It's the third Friday of the month, isn't it?" notes Anindya.

"Yep. I'm glad I put it in my calendar. Let's get going. Waiter, check please!"

Here we see a somewhat more innovative use of context. Mary's phone had detected that she was located several kilometers from her workplace, in fact, a 10 minute drive. Since her calendar showed a meeting starting at 1:30 p.m., and it was already 1:15 p.m., it displayed a high-priority notification. Apparently the others had either not put this meeting into their calendars or had not set up such notifications. Luckily for them, Mary had done both, so they avoid being late this time.

This fictional example provides an example of how context can be utilized in context-aware devices, namely smart mobile devices. From our introductory examination of context awareness in this chapter, it should now be fairly straightforward to envision how these features could be implemented. Admittedly, this example is not as "futuristic" and impressive as the *ubiquitous computing* scenario that Weiser depicted in 1991. Our intention was to demonstrate the concepts in this chapter and to provide a bridge to more advanced context awareness capabilities researchers and developers can implement with the current generation of mobile devices.

This example also demonstrates that context awareness, at least at the present level of technology, does not completely replace human intelligence and initiative. It can, however, go a long way in unburdening humans from various tasks, both mental and physical (in a human-computer interaction sense), freeing our human capacities for other purposes. Thus, the role of the context awareness researcher and developer should be now clearer—we must minimize the burden required to perform various tasks and open up new possibilities that would not be possible without context awareness.

8.6 SUMMARY

This chapter introduced the concept of *context* in terms of its importance to mobile geospatial computing. Since context is notoriously difficult to define, we have presented an intuitive framework for understanding the various elements of context, centered around seven questions: *what, who, where, when, why, in what manner,* and *by what means*. This in turn was used to explore the concept of *context awareness*, which we defined broadly as the quality of having knowledge of context. We briefly examined context awareness from a historical perspective, where we could see it blossoming as a research topic in the 1990s.

Next, we examined each of the elements of context in more detail, providing some examples of how contextual information within particular elements can be obtained. For several of these elements, including *what*, *why*, and *in what manner*, the appropriate methods vary greatly depending on the desired contextual information. In addition, these are often the most challenging elements to implement. For these reasons, further coverage of these topics is reserved for Chapter 9 on *contextual reasoning.*

Last, we used a fictional story, depicting a common workspace scenario in order to demonstrate how context awareness can be used in mobile devices. This story also highlighted several new concepts, such as *social context* and *context-sensitive marketing*, which are increasingly important in modern mobile devices.

References

[1] McCarthy, J., "Notes on formalizing context," *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 555-562, San Mateo, California: Morgan Kaufmann, 1993. [Online]. Available: http://bit.ly/OEHAyO.

[2] Akman, V., and M. Surav, "Steps toward formalizing context," *AI Magazine*, Vol. 17, No. 3, 1996, pp. 55-72, available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.3474.

[3] Foltz, P. W., W. Kintsch, and T. K. Landauer, "The measurement of textual coherence with latent semantic analysis," *Discourse Processes*, Vol. 25, No. 2 & 3, pp. 285-307, 1998.

[4] Bazire, M., and P. Brézillon, "Understanding context before using it," *Modeling and Using Context: Lecture Notes in Computer Science*, pp. 113–192, A. Dey, B. Kokinov, D. Leake, and R. Turner (eds.), Berlin, Heidelberg: Springer Berlin/Heidelberg, 2005, available: http://dx.doi.org/10.1007/11508373_3.

[5] "Context," In *Merriam-Webster.com*, 2012. [Online]. Available: http://www.merriam-webster.com/dictionary/context.

[6] "Five Ws," In *Wikipedia.com*, 2012. [Online]. Available: http://en.wikipedia.org/wiki/Five_Ws.

[7] Bennett, B. S., "Hermagoras of Temnos," In *Classical Rhetorics And Rhetoricians: Critical Studies And Sources*, pp. 187-193, M. Ballif and M. G. Moran (eds.), Westport, CT: Praeger Publishers, 2005.

[8] Welch, J. R., and K. G. Salter, "A context algorithm for pattern recognition and image interpretation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-1 , No. 1, 1971, pp. 24-30.

[9] Toussaint, G. T., "The use of context in pattern recognition," *Pattern Recognition*, Vol. 10, 1978, pp. 189-204.

[10] Swain, P. H., S. B. Vardeman, and J. C. Tilton, "Contextual Classification of Multispectral Image Data," NASA Technical Report SR-PO-00443, 1980.

[11] McCarthy, J., and P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence," *Machine Intelligence*, Vol. 4, 1969, pp. 463-502.

[12] Google, Inc., Google Now web site, [Online]. Available: http://www.google.com/landing/now/.

[13] Morishige, R. I., and J. Retelle, "Air combat and artificial intelligence," *Air Force Magazine*, October 1985, pp. 91-93.

[14] Endsley, M. R., "Design and evaluation for situation awareness enhancement," *Proceedings of the Human Factors Society—32nd Annual Meeting*, 1988, pp. 97-101.

[15] Gilson, R. D., "Special Issue Preface," *Human Factors* (Special Issue on Situation Awareness), Vol. 37, No. 1, 1995, pp. 3-4.

[16] Dey, A. K., "Context-Aware Computing," In *Ubiquitous Computing Fundamentals*, Boca Raton, FL: CRC Press, 2010, pp. 321-352.

[17] Brézillon, P., "Context in human-machine problem solving: A survey," *The Knowledge Engineering Review*, Vol. 14, No. 1, 1999, pp. 1-34.

[18] Frege, G., *The Foundations of Arithmetic, a Logico-Mathematical Enquiry into the Concept of Number,* transl. J. L. Austin, Oxford: Basil Blackwell, 1974. Originally published in German as *Die Grundlagen der Arithmetik, eine logisch mathematische Untersuchung über den Begriff der Zahl,* Breslau: Verlag von Wilhem Koebner, 1894.

[19] Frege, G., "On sense and reference," *The Philosophical Review*, Vol. 57, No. 3, 1948, pp. 209-230. Originally published in German as "Über Sinn und Bedeutung," *Zeitschrift für Philosophie und philosophische Kritik*, Vol. 100, 1892, pp. 25-50.

[20] Russell, B., "On denoting," *Mind*, New Series, Vol. 14, No. 56, 1905, pp. 479-493.

[21] Bledsoe, W. W., and I. Browning, "Pattern recognition and reading by machine," *1959 Proceedings of the Eastern Joint Computer Conference*, 1959, pp. 225-232.

[22] Miller, G. A., G. A. Heise, and W. Lichten, "The intelligibility of speech as a function of the context of the test materials," *Journal of Experimental Psychology*, Vol. 41, No. 5, 1951, pp. 329-335.

[23] Duda, R. O., and P. E. Hart "Experiments in the recognition of hand-printed text: Part II—Context analysis," 1968 *Proceedings of the Fall Joint Computer Conference*, 1968, pp. 1139-1149.

[24] Porzel, R., *Contextual Computing: Models and Applications*, Heidelberg: Springer, 2011.

[25] McCarthy, J., "Situations, Actions, and Causal Laws," Stanford Artificial Intelligence Project Memo No. 2, 1963.

[26] McCarthy, J., "Generality in artificial intelligence," *Communications of the ACM*, Vol. 30, No. 12, 1987, pp. 1030-1035.

[27] Giunchiglia, F. "Contextual reasoning," *Epistemologia, Special Issue on "I Linguaggi e le Macchine*," Vol. XVI, pp. 345-364, 1993.

[28] Giunchiglia, F., and P. Bouquet "Introduction to contextual reasoning," *Perspectives on Cognitive Science*, Vol. 3, 1997, pp. 138-159.

[29] Weiser, M., "The computer for the 21st century," *Scientific American*, Vol. 265, No. 3, 1991, pp. 94-104.

[30] Want, R., et al., "The active badge location system," *ACM Transactions on Information Systems*, Vol. 10, No.1, 1992, pp. 91-102.

[31] Schilit, B. N., and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, Vol. 8, No. 5, 1994, pp. 22-32.

[32] Brown, P. J., "The stick-e document: a framework for creating context-aware applications," *Electronic Publishing*, Vol. 8, No. 2/3, 1995.

[33] Abowd, G. D., et al., "Cyberguide: a mobile context-aware tour guide," *Wireless Networks.*, Vol. 3, No. 5, 1997, pp. 421-433.

[34] Dey, A. K., G. D. Abowd, and A. Wood, "CyberDesk: a framework for providing self-integrating context-aware services," *Knowledge-Based Systems*, Vol. 11, 1998, pp. 3–13.

[35] Moran, T. P., and P. Dourish (eds.), "Special Issue on Context-Aware Computing," *Human-Computer Interaction*, Vol. 16, Nos. 2-3, 2001.

[36] Facebook, Inc., "Android SDK Reference," [Online]. Available: http://bit.ly/PoIpbX.