

Chapter 2

Application Scenarios

In this chapter, we present two application scenarios that illustrate two different types of *social context-aware applications* (in short *socially-aware applications*) and their associated requirements. The objective of this chapter is *twofold*. The *first* objective is to present application scenarios which introduce different kinds of social context information and their use in developing various socially-aware applications. These application scenarios will be used throughout the next chapters to explain the *framework* for social context-aware applications by providing suitable examples. Furthermore, these scenarios will be used in case studies to show how the framework supports the development of various applications. The *second* objective is to motivate and identify key requirements of developing socially-aware applications by analysing these application scenarios.

The chapter is organized as follows. Section 2.1 presents a scenario related to phone call interruptions illustrating the needs and requirements of a *data-centric* socially-aware application which reduces phone call interruptions by exploiting users' social context information, in particular, *connection-oriented social relationships* and *situations*. Section 2.2 presents a cooperative convoy scenario illustrating the needs and requirements of an *interaction-centric* socially-aware application which supports social interactions based on users' *interaction-oriented social relationships*. Section 2.3 generalises the *key requirements* for developing socially-aware applications. Finally, Sect. 2.4 summarises this chapter.

2.1 A Data-Centric Socially-Aware Application Scenario

In this section, we identify the requirements of developing *data-centric* socially-aware applications (where data drives the applications' behaviour) by analysing the phone call interruptions scenario. Section 2.1.1 discusses the phone call interruptions scenario, while Sect. 2.1.2 analyses the requirements.

2.1.1 The Phone Call Interruptions

Mobile phones have the obvious benefit of at the moment communication. Irrespective of time and place, we can expect a phone to ring. But a ringing phone at an inopportune moment can be very disruptive to the current task or social situation. In a survey of 100 senior executives, it was reported that undesirable interruptions constitute 28 % of the knowledge worker's day, which translates to 28 billion wasted hours for companies in the United States alone. It results in a loss of 700 billion dollars per year, considering an average labor rate of \$25 per hour for information workers [5]. Studies have also found that undesired disruption causes interrupted users to take up to 30 % longer to complete and commit up to twice the number of errors. Judge Robert Restaino jailed 46 people when a mobile phone rang in his New York courtroom and no one would admit responsibility [1].



Fig. 2.1: Illustrations of phone call interruptions

We witness how often people are interrupted by incoming phone calls which not only disturb the user but also his/her surrounding people (Fig. 2.1 illustrates two scenarios). Mobile phone interruptions have sparked problems that have already been studied extensively, and they are many in number and can be serious in nature (*e.g.*, they cause car accidents [14]). Meanwhile, the burden of managing social norms and patterns implied

by mobile phone usage is left to the user. For instance, when one is in a meeting, he/she must remember to switch the phone to silent and decide whether to pick up any incoming call. Many of these interruptions could be avoided or adequately managed, only if the caller could be informed of the current status of the intended callee [120]. For example, if the caller could be informed that the callee is in a meeting, she might decide to call at a better time, thus avoiding or reducing harmful interruptions and possible socially embarrassing situations. On the other hand, the static nature of mobile phone configuration (*e.g.*, reject all incoming phone calls while in a meeting) may lead to communication gaps and socially embarrassing situations [86]. For example, one may not want to miss any incoming call from his/her boss, even though he/she is in a meeting.

Let us consider that Alice, an university graduate student, wants to filter her incoming phone calls as follows: if my status/situation is *meeting* or *seminar*, and (1) a call comes from *friend*, the action is to reject, (2) a call comes from *family*, the action is to reject and forward my status as *Busy*, (3) a call comes from *colleague*, the action is to reject and forward status as *meeting* or *seminar*, (4) a call comes from the *supervisor*, the action is to vibrate, (5) if my status is a non-business situation, the action is ring. Alice also wants to make her situation accessible by her colleague when she is in a meeting or seminar.

2.1.2 Requirements Analysis

A socially-aware phone call application is a way to minimize mobile phone interruptions that are mostly caused by (1) *unawareness* of each other's (caller and callee) current situation, and (2) a lack of support to allow users to specify their *dynamic* filtering preferences over incoming phone calls. By observing how interruptions are currently handled by users, we might (at least partially) automate interruption management. Recently some studies (*e.g.*, [34, 64]) have tried to identify the context information (*e.g.*, users' situation) that is considered important in deciding whether to initiate phone call. In that regard, some other studies (*e.g.*, [120, 177]) have tried to identify factors that influence the users to share or disclose information about their situations. Some further studies also have focused on identifying actions that are performed currently by users to filter calls (*e.g.*, [86]). Users' social context, in particular, connection-oriented relationships (*e.g.*, father-child, mother-child, husband-wife) and situations (*e.g.*, meeting, seminar, watching a game in a stadium) have shown to have strong social underpinnings in both sharing current status information and filtering incoming phone calls [120, 176, 177].

A socially-aware phone call application that considers both the caller and callee perspectives, need to fulfill the following requirements:

First, the application should *acquire* its user's current situation information and allow the user to *share* some of this information with other users. In this regard, the application should also allow its user to specify her *privacy* preferences to retain control over who has access to her situation information under which conditions.

Second, the application should allow its user to specify her phone call filtering preferences based on different social context information including *social relationships* and *situations*. When a phone call comes, the application needs to execute these filtering preferences and behaves accordingly. To check the filtering preferences, application needs its user's social context information such as her current situations, her relationships with the caller, and so on. Studies have revealed that users want to define call filtering preferences that apply to all people with certain social relationships (e.g., family, best friends) or social roles rather than explicitly name each person [120, 177]. Thus, it is required to collect, classify, infer and manage different social context information.

2.2 An Interaction-Centric Socially-Aware Application Scenario

In this section, we identify the requirements of developing interaction-centric socially-aware applications (where interactions drive the applications' behaviour) by analysing the cooperative convoy scenario. Section 2.2.1 discusses the cooperative convoy scenario, while Sect. 2.2.2 analyses the requirements.

2.2.1 The Cooperative Convoy

Two groups of tourists in their two cars hired from two different rental companies want to drive together from Melbourne to Sydney. The car rental companies provide different levels of support to their customers based on the customers' insurance policies, payment, service availability, and so on.

Let us consider that the two cars, Car#1 and Car#2, as depicted in Fig. 2.2, are rented from Budget and AVIS respectively. The Budget Car Rental Company provides a car with a wet-sensor that can detect relevant road conditions and give an alert to reduce speed as necessary. The company also allows access to a Travel Guide Service (TGS) that suggests a possible route to a destination considering the shortest path. However, the service is unable to report real-time traffic information, e.g., road blocks due to road accidents. In contrast, the AVIS Company allows its car to access a Traffic Management Service (TMS) that provides information on traffic conditions ahead, e.g., the

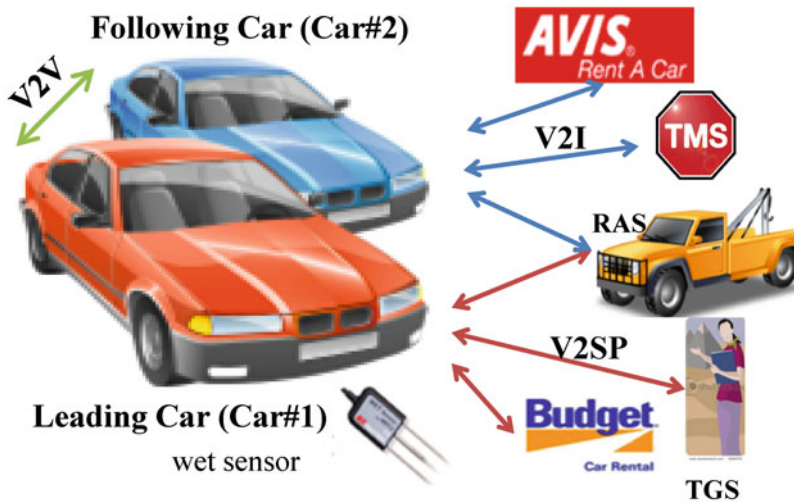


Fig. 2.2: An illustration of cooperative convoy

road is blocked due to accident or any emergency road works. However, both of these car rental companies provide a Roadside Assistance Service (RAS) to their customers. When a vehicle experiences a mechanical issue, its telematics system automatically informs the rental company and RAS, and requests a tow truck if the vehicle is no longer drivable. Information about the vehicle location and the mechanical issue is also sent with the request. Upon receiving the request, the road-side service validates it with the car rental company and takes the necessary steps as defined in the agreement with that company.

Drivers of Car#1 and Car#2 want to form a cooperative convoy to make their travel safe and convenient by collaborating and interacting with each other. In a *cooperative convoy*, a vehicle interacts with other vehicles, service providers and infrastructure systems to make the travel safe and convenient. Through these interactions a vehicle's driver can share information (acquired from the service providers and infrastructure systems) with other vehicles' drivers in performing their tasks. Such interactions are subject to defined agreements and constraints among the entities (*i.e.*, vehicle to vehicle (V2V), vehicle to service providers (V2SP), and vehicle to infrastructure (V2I)).

In the above scenario, we can see that Car#1 and Car#2 have access to different types of services: Car#1 has access to a Travel Guide Service (TGS), while Car#2 has access to a real time Traffic Management Service (TMS). In the cooperative convoy, they decide that Car#1 is the leading car (LC) whilst Car#2 is the following car (FC). The two cars follow the same route chosen by the leading car as it has access to a TGS. In addition, the drivers of both

cars agree on a number of issues. For instance, they will always keep the distance between them less than 1 km. Car#2 (the following car) will send road blocks information (obtained from its TMS) to Car#1 (the leading car) if there is any. Car#1 gets the updated route plan from its TGS by specifying its preferences (e.g., avoid the route with blocked road) and notifies that route information to Car#2. Both cars notify each other of their positions every 10 s. If either vehicle experiences mechanical problems (e.g., flat tyre, engine issue) it needs to notify the other vehicle.

2.2.2 Requirements Analysis

The socially-aware application facilitating the cooperative convoy needs to fulfill three major requirements:

First, the application should support interactions complying with the *agreed* interaction relationships (i.e., constraints and obligations). In the above scenario, these interaction relationships can be clustered into three groups by taking a collaborative view:

- *Interaction relationships between the driver of LC (Car#1) and the Budget Company with its associated services*: The interaction relationships such as getting route plan from the TGS, requesting road side assistance in case of any mechanical problem, and so on, fall into this category.
- *Interaction relationships between the driver of FC (Car#2) and the AVIS Company with its associated services*: The interaction relationships such as getting road blocks information from the TMS, and requesting road side assistance in case of any mechanical problem fall into this category.
- *Interaction relationship between the drivers of LC and FC*: The interaction-relationship between drivers of LC and FC (i.e., their mutual agreements) falls into this category. It includes that FC should send road blocks information to LC if there is any, LC should notify the updated route information to FC, FC follows the same route chosen by LC, both cars notify each other of their positions every 10 s, and in case of any mechanical problem both cars notify each other, and so on.

Thus, a model is required to capture the collaborative view of such interaction-oriented relationships.

Second, the application needs to provide a coordinated view of the interactions, and to allow drivers to specify their coordination preferences and to perform the coordination in an automated manner. According to the scenario, each driver is responsible for performing several tasks, for instance, the driver of LC is responsible for notifying route information, current positions, and mechanical problems (if any) to FC. Similarly, the driver of FC should send road blocks information, current positions, and mechanical problems to LC. For the drivers to perform such additional tasks

(e.g., forwarding information) may cause distraction and have undesirable consequences. Thus, to facilitate collaboration with less distraction, it is required to provide a coordinated view of the interactions and to allow drivers to specify their coordination preferences.

Third, the application needs to support runtime adaptation, as the interaction relationships evolve over time and thus need to *adapt* with the *changes* in requirements and environments. For instance, a third vehicle could join when the convoy is on the way; or the break-down of a following vehicle might result in its leaving the convoy before reaching the destination. A mechanical problem of the leading car may require it to handover the leading car role to one of the following cars. Because of heavy rain, the maximum distance may need to be reduced from 1 km to 600 m.

2.3 Requirements for Developing Socially-Aware Applications

The above discussions reveal that there are a number of key requirements for developing socially-aware applications:

- Req1 *Modelling Social Context Information*: There are different types of social context information such as interaction- and connection-oriented social relationships, social roles, social interactions and situations. Such social context information need to be modelled and represented consistently with its semantics to facilitate application use. Moreover, a pervasive social computing application assumed to be highly dynamic in the sense that it might require new types of social context information or might introduce new information sources that yield new types of social context, so the social context model should be flexible extensive.
- Req2 *Reasoning about Social Context Information*: Application may need social context information that is not directly available from context sources but can be derived from acquired information. For instance, users may want to filter phone calls based on relationship categories such as “family” and “best-friends”, and situations such as “busy” that may not be acquired directly from sources but can be inferred from other acquired basic information. Thus, there is the need to define and obtain inferred social context information based on the semantics, attributes and temporal correlations of the acquired basic information.
- Req3 *Preserving Privacy in Accessing Social Context Information*: A socially-aware application may require users to share their social context information for greater benefits but this also comprise their privacy. For example, in a socially-aware phone call application, allowing a caller to know the situation of the callee before calling might reduce interruptions, but may also raise serious concerns regarding the privacy and access control over

users' status and other data. Therefore, users should be able to retain control over who has access to their personal information under which conditions. In addition, a user may want to fine-tune the granularity of the answer. For instance, an employee may be happy that her boss knows her current status is "at desk". However, she may not be happy for her boss to know that her status is "chatting to friends via instant messaging". Thus, there is a need to provide a privacy policy that should allow users to control access to their information at different levels of abstraction.

Req4 *Acquiring and Managing Social Context Information*: Social media such as online social networks, blogs, and instant messaging, have radically changed the way people interact with each other and share information about their lives and works [184]. The popularity of online social networks such as Facebook, LinkedIn, Twitter and Google+ produce an unprecedented amount of social context information as people specify their relationships, update their status and share contents. Google Calendar also can be a possible source of situation information as people use it to maintain their schedule [126]. Furthermore, users' interaction events in different platforms, *e.g.*, social media, can be exploited to infer their situations. Thus, it is required to *acquire* users' such information from *different sources* and *manage* its changes for up-to-date, accurate and real time *provisioning* of social context information.

Req5 *Modelling Runtime Social Interactions*: Socially-aware applications that support social interactions based on the interaction-oriented relationships (*i.e.*, agreements and constraints) among collaborative actors, need runtime representation of such relationships, called social interaction model. Such runtime social interaction model mediates runtime social interactions between actors. For instance, in the cooperative convoy, to support social interactions between drivers of Car#1 and Car#2 according to their interaction-relationship, a runtime model of such interaction-relationship is required. However, an actor may involve in multiple interaction-relationships and it may have its own view of social interactions related to these relationships. Thus, it is also required to support an individual user to coordinate his/her social interactions by providing a coordinated view of runtime social interaction model.

Req6 *Managing Adaptation in Runtime Social Interaction Models*: Runtime social interaction models typically evolves and many aspects of runtime models such as topology, interaction constraints and non-functional quality properties need to *modify* frequently in response to *changes* in user requirements and environments. Thus, it is necessary to manage the *adaptation* in such runtime social interaction models. For instance, a third vehicle could join when the convoy is on the way; or the break-down of a following vehicle might result in its leaving the convoy before reaching the destination. As a consequence, the runtime social interaction model of convoy needs to be updated.

2.4 Summary

In this chapter, we have presented two application scenarios associated with two socially-aware applications of distinct focus such as *data-centric* and *interaction-centric*. These applications need different types of social context information and requires various supports in their development.

First, we have presented a phone call interruptions scenario that demands a *data-centric* socially-aware application. This application allows a person to filter incoming phone calls based on her situation and the relationships with the caller. Thus, the application needs *data*, e.g., connection-oriented social relationships and situations, to provide service to its users. The application also allows a caller to know the status of the intended callee before making the call. In this regard, the application requires to access the callee's situation information while preserving his/her privacy.

Second, we have presented a cooperative convoy scenario that demands an *interaction-centric* socially-aware application. This application enables a cooperative convoy by supporting social interactions among drivers and external services, based on their predefined *agreements* and *constraints*, which we refer as *interaction-oriented social relationships*. The application needs a technique to *model* such interaction-oriented relationships. It also requires support to *manage adaptation* in such runtime model to cope with the changes in requirements and environments.

Finally, we have identified the key *requirements* for developing socially-aware applications.

Pervasive Social Computing
Socially-Aware Pervasive Systems and Mobile
Applications

Kabir, M.A.; Han, J.; Colman, A.

2016, XXII, 282 p. 91 illus., 79 illus. in color., Hardcover

ISBN: 978-3-319-29949-5