From Ad Hoc Networks to Ad Hoc Applications

Position Paper

Benoît Garbinato, Philippe Rupp
Institut d'Informatique et Organisation (INFORGE)
Ecole des HEC - Université de Lausanne
1015 Lausanne - Switzerland
{benoit.garbinato, philippe.rupp}@hec.unil.ch

***Abstract* -- This paper presents the first result of an ongoing research on mobile peer-to-peer communication paradigm and on ad hoc networks. This result is threefold: (1) we propose a definition of ad hoc applications that is independent of the notion of ad hoc networks, (2) we advocate for the need of an adequate development framework for building ad hoc applications, (3) we introduce such a framework, which is independent of the underlying network technology but which relies on existing standards. To illustrate our approach, we also sketch how our framework can be used to build a real ad hoc application.**

Ad hoc network; Ad hoc application; Framework; Abstraction; Peer-to-Peer; Mobile; Collocation; Location-based; Java Phone.

I. Ad hoc networks as background

Several recent research projects aim at studying new generation mobile communication and information services, based on self-organisation (AODV [2], Terminodes [8], CarNet [15]). Such systems have become very topical lately, with the advent of the **peer-to-peer communication paradigm** and the emergence of **ad hoc network** technologies. Yet, many fundamental technical questions remain open, while real business applications still need to be rolled out.

One specific domain where the peer-to-peer phenomenon had a major impact is the data and media interchange on the internet. It had especially an immense influence on the music industry, to the point to question its foundations and existence [6].

However, this wave does not seem to be limited to desktop computers connected on the internet. Mobile devices will presumably be the next target, as storage and computing capacities raise, power consumption diminishes, and mobile communication tends to generalize, altogether guaranteeing the minimal requirements for multimedia applications (Apeera's [17] example shows this trend).

In the context of mobile peer-to-peer communication, ad hoc networks are expected to play a key role in the future, although nobody knows which emerging technology will win and when it will be ready for prime time. It seems nevertheless reasonable to forecast that when this will happen, we will need adequate development frameworks to leverage such ad hoc networks and turn them into mobile peer-to-peer applications.

A. A definition of ad hoc network

According to Murphy & al. [16], an ad hoc network is *"a transitory association of mobile nodes which do not depend upon any fixed support infrastructure. [...] Connection and disconnection is controlled by the distance among nodes and by willingness to collaborate in the formation of cohesive, albeit transitory community"*.

Distance among nodes. With ad hoc networks, no additional infrastructure is required besides the network nodes themselves. It is the distance among nodes, or rather their *proximity*, that defines the boundaries of the network. That is, the mere collocation of two or more mobile nodes within a certain perimeter defines a new network in an *ad hoc* manner. Now, if the nodes were not mobile, an ad hoc network would not be different from LAN (Local Area Network). So, it is also the mobility of nodes, causing variations in their distance, that gives such networks their ad hoc nature.

The actual perimeter defining an ad hoc network really depends on the technology being used. Some ad hoc network solutions are limited to so-called PAN (Personal Area Network). The Bluetooth technology, for instance, allows only for the definition of a PAN (up to about 10 meters [12]). This is also the case of some proprietary solutions like Spotme by Shockfish [18] which is limited to approximately 30 meters.

Willingness to collaborate. The collocation of several nodes within a certain distance is a *necessary* but *not sufficient* condition to form an ad hoc network. In addition, collocated nodes need to be willing to collaborate. By definition of ad hoc networks, this willingness is expressed at the *network level*: roughly speaking, the decision to collaborate or not is expressed by going online or offline.[1]

Transitory peer-to-peer communities. Intuitively, the above features of ad hoc networks characterize their some-

1. The way a node goes on-line or off-line then depends on the actual ad hoc network technology being used.

how *"here and now"* nature. That is, at any point in time (*now*), the network is defined by all nodes that are both within a certain distance and online (*here*). As consequence, nodes tend to appear and disappear in an ad hoc network much more often than in other types of networks, leading to so-called *transitory community.*

Interestingly, transitory community is also a common aspect of peer-to-peer applications. People can join and withdraw from the community at any time, making it an ever changing map. The absence of a network infrastructure besides the nodes themselves accentuates the peer-to-peer nature of ad hoc networks even more. Because of this absence, indeed, nodes must interact via network-level peer-to-peer communication.

B. What applications for ad hoc networks?

As a first approximation, we could simply define ad hoc applications as pieces of software that run on ad hoc networks. We do not believe however that this definition adequately characterizes the essence of ad hoc applications. Defining a class of applications (ad hoc applications) by coupling it to one (but only one) possible implementation technology (ad hoc networks) reduces the generality and effectiveness of the definition. This is particularly true in such a changing arena as mobile network technology.

Indeed, no widely adopted ad hoc network exists nowadays. However, many researchers [2, 8, 15] work on the subject in order to build what will probably be the next generation networks. As no general consensus about future ad hoc network technology seems to emerge, a wise approach is to abstract ourself from the network level as will be seen later in this paper.

In the remainder of the paper, we thus propose a more general definition of ad hoc applications and we describe an application-level framework for building such applications. More precisely, we start by providing the business background that we consider in our research, following a typical *top-down* approach. Then, we define ad hoc applications without mentioning the underlying network technology, and we illustrate our definition with some examples. Finally, we provide an architecture overview of our framework solution and we compare it with some existing related work.

II. TOWARDS AD HOC APPLICATIONS

For this research, we are following a fairly classical top-down approach, starting with an analysis of the business aspects of ad hoc applications.[2] The aim is to identify existing ad hoc applications in order to see the evolution of this sector in the actual mobile commerce (*m-commerce*, for short) world. This important part of the research is still ongoing and helps to draw a possible scenario in the case of a wide adoption of ad hoc networks [3]. At the same time, it provides useful and viable examples that will help to validate our working process. Some of those examples are discussed later in this paper, but as it is not the main goal of this paper, they are not further developed. They nevertheless constitute the business basis for the definition of ad hoc applications that follows.

A. A definition of ad hoc application

We define an *ad hoc application* as a self-organizing application composed of mobile and autonomous devices, interacting as peers and which relationships are made possible because of relative physical distance (collocation). In addition, this dynamic community defined by a geographical proximity needs to have a common (application-level) interest. More formally, three features must be present in an application for it to deserve the *ad hoc* label.

Mobility. In order to be able to use the application everywhere, the user should not be limited by the range available. Thus, a PAN is not enough. The range limit is set by the business logic of the application.

Peer-to-Peer. Direct communication between peers is mandatory. This means that the pieces of software forming an ad hoc application have to interact directly, without using a central server. As said earlier, the underlying physical infrastructure has no impact on the fulfillment of this aspect.

Collocation. All logical interactions between applications have to result in a physical interaction between users. This is called collocation. It means that in order to be called an ad hoc application, the service has to be location-based.

B. Abstracting the network level

The above definition allows us to abstract the network completely, and focus on the application aspects. This way, there is a clear decoupling between the application and the network, and any type of network - GSM, WLAN or ad hoc - can be used.

This decoupling is fundamental in understanding the key issues underlying ad hoc applications and in providing reusable solutions to solve these issues (in order to ease the development of ad hoc applications). By abstracting the network level, we are able to build ad hoc applications in absence of any ad hoc network, as long as the underlying infrastructure can provide support for the three basic aspects that we described earlier.

C. Existing ad hoc applications

Some possible applications have already been cited in the literature. For example, Kortuem discusses an ad hoc task trading system [13]. The idea is that two individuals use their

2. More precisely, we are following an *iterative* top-down approach, as proposed by the Unified Process [9].

mobile devices to negotiate about and to exchange real-world tasks: dropping off someone's dry cleaning, buying a book of stamps at the post office, or returning a book to the local library. The idea is that you do a favor for others because you know that one day they will do it for you.

Some mobile operators, like Swisscom with their Friend-Zone service [7], are also active in this domain. The Friend-Zone service for instance lets you meet new people close to you, even flirt with them or locate your friends. More recently, Swatch released a mobile dating device in the form of a watch, under the name *synchro.beat* [19]. This system uses a sound made by a watch to transfer information to another one. Then, a matching algorithm tells you the level of compatibility you have with the other watch's owner, based on information both parties entered in their respective synchro.beat watches.

D. An example: the ubiquitous flea market

In order to illustrate our approach, we define an ad hoc application referred to as the *ubiquitous flea-market*.[3] An important characteristic of traditional flea markets is that they are only available on a certain day and time. Then, walking around takes time and energy, as one has to carefully scrutinize what is available. And finally, the different roles (buyer and seller) are clearly separated.

The ubiquitous flea market is available wherever you are and at all time. It is available on many mobile devices and matches buyers and sellers present within a certain range, the latter being previously defined by the user. While walking, driving or flying, this ad hoc application scans its surroundings for possible peer sellers or buyers. It has to be noted that any user can be buyer and/or seller. When the application finds another mobile device that runs the same piece of software, it scans the shared items in order to find a matching one. If there is a match, the user is alerted and can then ask the peer to get in touch and make the physical transaction. All this, in a matter of simplification, is based on a shared and known taxonomy describing the items that can be bought and sold.[4] A use-case diagram helps to have a better idea of the fundamental features of our ubiquitous flea market.

As can be seen in Figure 1, the ubiquitous flea market ad hoc application has three actors and six different use-cases. The first one, *Start*, launches the application and registers, through the *Register Location Interest* use-case, in the Ad Hoc Framework through the location service. In *Manage Item List*, the user (a buyer and/or a seller depending on the wished configuration) can add, update or remove items to be bought or sold, whereas the *Update Proximity Preferences* is where the trading range, among other possible preferences, is set. Finally, *Find Matching Item* is where most of the work is done. This use-case gets a notification of a peer's presence by the Ad Hoc Framework and then notifies the user of the possibility to make a transaction with the peer. In a last move, both users get physically in touch in order to seal the deal.

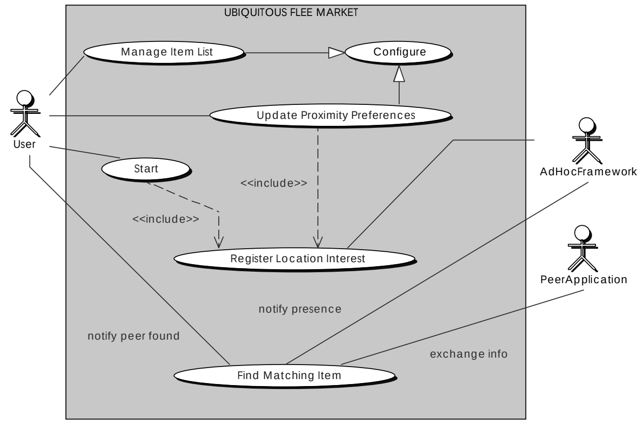

Figure 1. use-case of the ubiquitous flea market

III. A FRAMEWORK FOR AD HOC APPLICATIONS

Based on our definition of ad hoc application, we advocate for the need of a development framework dedicated to the creation of ad hoc applications. A key requirement of such a framework is that it should be independent of any underlying ad hoc network technology, just as our definition of ad hoc application. If we fulfill this requirement, it is then relatively easy to adapt the framework to upcoming ad hoc technologies, with no impact on previously developed applications.

A. A lightweight framework

In our ad hoc application framework, we do not reinvent the wheel. Rather, it is based on existing standards, thus making it lightweight, in order to drive its adoption by developers. Some frameworks close to ours actually redefine for instance a transport protocol, as discussed in Section IV. This is not necessary as with the advent of Java enabled mobile devices, most of these basic services are already available and standardized.

In Figure 2, we present the different layers for an ad hoc application when based on our framework. The P2P Low level corresponds to the physical infrastructure needed to communicate. As said earlier, this layer does not need to be an ad hoc network. It can be for instance a regular GSM network. But the communication between two peers has to be *logically* direct. In a GSM network [20], the location information has to be given by a mobile operator. With the generalization of GPS and advances in the ad hoc network research field (as can be seen in [4]), this information will be directly

3. As we shall see, this application also serves for testing and validating the framework we propose in Section A. of Part III.

4. Describing this taxonomy is beyond scope here.

available in a near future. Our framework uses standard HTTP in order to guarantee a general interoperability, but more important to have the widest adoption possible.

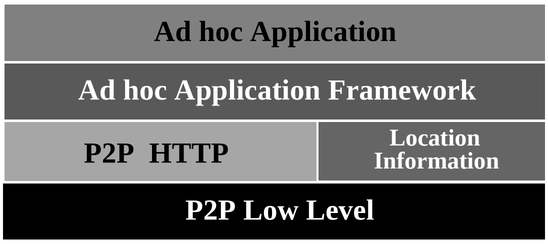

Figure 2. The different layers of an ad hoc application

As said earlier, the adoption of our framework will be driven by the use of existing standards. Java 2 Micro Edition (J2ME) [10], and eventually project J2ME over JXTA [1], will be the runtime environment used, but this is beyond the scope of this paper.

Table I shows a short excerpt of our APIs, specifically one concerning the collocation or proximity functionality. The `LocationService` is where an ad hoc application registers its interest in the proximity of other peers. Whenener adequate, the `ProximityListener` is then notified of the proximity of peers via a callback. Finally, `PeerGroup` is simply reused from the JXTA framework to perform the peer-to-peer communication.

TABLE I. PROXIMITY API

```
interface LocationService {
   /** Register interest in proximity of peers
       within a certain range, given in meter.
   */
   void registerInterest
        (ProximityListener listener, int range);
   // Unregister interest in peer proximity
   void unregisterInterest
        (ProximityListener listener);
}

interface ProximityListener
   extends EventListener {
   // Notify of the proximity of one or more peer
   void notifyProximity(PeerGroup peers);
}
```

B. *Implementing the ubiquitous flea market*

As already said, the ubiquitous flea market is a good illustration of the three basic aspects of ad hoc applications discussed in Section A. of Part II., and it is thus also a good illustration of the use of our framework. Table II sketches how the API of Table 1 is used in this application. Note that the actual implementation of the proximity service is completely hidden (via a factory method) and could initially rely on the services of a telco operator, but later rely on an emerging ad hoc network technology. Today already, several telco operators provide such a service in so-called location-based services [7].

TABLE II. THE UBIQUITOUS FLEA MARKET EXAMPLE

```
ProximityListener myListener= new
       UbiquitousFleaMarketProximityListener();

LocationService theService=
       LocationService.getUniqueInstance();

theService.registerLocationInterest(myListener,
       ubiquitousFleaMarketPreferences.range);
```

C. *Status and next steps*

The work presented here is still in progress and the first version of the framework will be available early 2003. It will be illustrated by the first implementation of an ad hoc application, the ubiquitous flea market described earlier in this paper.

IV. CONCLUSION & RELATED WORK

In this paper, we proposed a technology-independent definition of ad hoc applications, together with a framework easing the development of such ad hoc applications. Key features of our framework are its independence of any underlying network technology and its systematic adoption of existing standards, which makes it lightweight. Similar frameworks have already been described in the literature.

One such framework is *PeerWare* [5], which is *“a middleware whose design is geared towards peer-to-peer and mobile systems. [...] It provides abstractions that appropriately encompasses the kind of reconfigurability required by distributed and mobile teamwork applications”*. What differentiates this framework from ours is threefold. First of all, PeerWare only takes into account the mobility and peer-to-peer aspects. As explained earlier in this paper, collocation is a crucial aspect of our framework to target ad hoc applications. Then, our approach is not to provide any low level communication protocol as PeerWare, but rather to base the implementation of our framework on standard technology and APIs. Finally, our framework is not limited to teamwork applications. It is possible to build any software requiring the three aspects described above.

Proem [14], another example, is an open computing platform targeted at mobile ad hoc information systems. Built by the University of Oregon, it provides a complete solution for developing and deploying collaborative peer-to-peer applications for mobile ad hoc networks and personal area networks (PAN). It is a collection of tools, API and runtime for developing and deploying applications and works on a really low level. A specific transport protocol has been specified. It uses

XML for representation of messages and can then be implemented on top of TCP/IP, UDP or HTTP for instance.

V. Acknowledgement

The work presented in this paper was supported by the *National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS)*, a center supported by the *Swiss National Science Foundation* under grant number 5005-67322

VI. References

[1] A. Arora, C. Haywood, K. Singh Pabla, “JXTA for JTME - Extending the Reach of Wireless With JXTA Technology”, Sun Microsystems, March 2002

[2] E.M. Belding-Royer, “Report on the AODV Interop”, *UCSB Tech Report 2002-18*, June 2002

[3] G. Camponovo, Y. Pigneur, Analyzing the m-Business Landscape, Annals of Telecommunications, Hermes, Novembre-Decembre 2002.

[4] S. Capkun, M. Hamdi, J.-P. Hubaux, “GPS-free positioning in Mobile Ad-Hoc Networks”, *Proceedings of the 34th Annual Hawaii International Conference on System Science*, January 2001.

[5] G. Cugola, G. P. Picco, “PeerWare: Core Middleware Support for Peer-to-Peer and Mobile Systems”, *ESEC/FSE 2001*, Vienna, Austria, September 10-14, 2001.

[6] T. Estier, Ph. Rupp, “A Model for a Better Understanding of the Digital Distribution of Music in a Peer-to-Peer Environment”, Proc. 36th Hawaii International Conference on System Sciences, January 2003.

[7] V. Finney, D. Williams, “Switzerland: Learning form Europe's most Advanced MLS Nation”, *Mobile Internet*, July 2001.

[8] J.-P. Hubaux, “Terminodes: Toward Self-Organized Mobile Ad Hoc Wide Area Networks”, *Technical Report No. SSC/1999/022*, June 3, 1999.

[9] Jacobson, I., Booch, G., Rumbaugh, J., *The Unified Software Development Process*, Addison-Wesley, February 1999.

[10] “Java 2 Micro Edition (J2ME) Technology for Creating Mobile Devices, White Paper”, Sun Microsystems, May 19 2000.

[11] G. Kan, “Gnutella” in *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, pp. 94-122, Edited by Andy Oram, O'Reily & Associates, 1st Edition, 2001.

[12] J. Kardash, “Bluetooth Architecture Overview”, *Intel Techno-logy Journal Q2*, Mobile Computing Group, Intel Corporation, 2000.

[13] G. Kortuem, Z. Segall, T.G.C. Thompson, “Close Encouters: Supporting Mobile Collaboration through Interchange of User Profiles”, *First International Conference on Handheld and Ubiquitous Computing (HUC)*, Karlsruhe, Germany, September 1999.

[14] G. Kortuem, J. Schneider, D. Preuitt, T. G. C. Thompson, S. Fickas, Z. Segall, “When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad Hoc Networks”, *International Conference on Peer-to-Peer Computing*, Linköping, Sweden, August 27-29, 2001.

[15] R. Morris, J. Jannoti, F. Kaashoek, J. Li, D. Decouto, “CarNet: A Scalable Ad Hoc Wireless Network System”, *The 9th ACM SIGOPS European Workshop: Beyond the PC, New Challenges for the Operating System*, September 2000.

[16] A.L. Murphy, G.-C. Roman, G. Varghese, “An Exercise in Formal Reasoning about Mobile Communications”, *Proceedings of the Ninth International Workshop on Software Specifications and Design*, IEEE Computer Society Technical Council on Software Engineering, IEEE Computer Society, Ise-Shima, Japan, April 1998, pp.25-33.

[17] “Phones Join File-Sharing Revolution”, *BBC News World Edition*, Technology Section, September 12, 2002.

[18] “Spotme - The Know-How for Meeting People”, *BRO - 003 - 010*, Shockfish SA, 2002.

[19] “Synchro.beat Online User Manual”, Swatch, 2002.

[20] Tisal, J., *GSM: Réseaux et Services*, Masson, Paris, 2nd Edition, 1997.