Models and Infrastructures for Pervasive Computing

Wang-Chien Lee
Dept of Computer Science & Engineering
Penn State University
University Park, PA 16802-6106
wlee@ieee.org

Dik Lun Lee
Department of Computer Science
University of Science and Technology
Clear Water Bay, Hong Kong
dlee@cs.ust.hk

1 Pervasive Computing

The field of Pervasive Computing has emerged as an active research area in the past few years [8]. Pervasive computing refers to computing environments which contain millions of computing devices or information appliances [7], embodied in the environment we live in (e.g., our cars, watches and clothes, mobile phones, microwave ovens, refrigerators. etc.). These devices interact with us using our own languages and among themselves transparently and adaptively to provide the services we need - often without explicit requests from us.

Pervasive computing represents a vision and grand challenges integrating research in areas from user interface, web and database technologies, and all the way down to hardware and communication network. The applications of pervasive computing are only limited by our imagination. We can think of refrigerators that can place an order to the supermarket with the best price when an item runs out. Likewise, terrible stories could be told on privacy issues, human beings being out-smarted by computers, etc. However, we believe there is a large area in between these extremes that even the most conservative would feel comfortable with. For example, systems that automatically display the pollution index on our watches are important for health care; systems that can automatically deliver the right kind of information to us, depending on when and where we are, will boost our productivity. In this paper, we focus on the technology issues instead of dealing the privacy and social issues.

Since pervasive computing is massive in terms of the number of devices and services involved and mobile because devices are likely attached to moving physical objects, we place special attention on the temporal and spatial aspects of modeling, dynamic and complex composition of services, and mechanisms that match requests to services in a scalable manner. We point out two important aspects of pervasive computing, namely, model and infrastructure. A logical model for computing devices and services captures features that are essential for devices to connect to services and the recursive composition of services. A basic requirement of pervasive computing is that these devices may exhibit different behaviors depending on the location and time. As such, the notion of temporal and spatial properties, including their inter-dependence and impacts on the services provided by the system needs further study. The infrastructure supports devices transparent access to services. This includes the notion of data-centric routing and service community to allow the pervasive computing environment to scale up to potentially billions of devices.

2 Models

Since smart applications must be able to run on different types of information appliances, a model is needed to provide a high-level abstraction of an information appliance and its services. A primary challenge is to identify the requirements of the model for pervasive computing. The following are some examples that can guide the development of the model:

Example 1. Smart home: A household service that would turn on the microwave oven at 7am; raise an exception when the oven is empty, and notify the owner when such exception occurs. Event and exception handling are required.

Example 2. Personalization: An automobile enters an area. The navigation system on the car communicates with a nearby base station. Based on the car's identity, the base station knows that the car has not been in the area before. It volunteers a street map to the car (along with a few advertisements of nearby restaurants). The car, knowing that its driver speak French, asks the based station if it has maps in French. If it does, it will furnish the maps in French (together with advertisement translated to French).

Some of the features that we anticipate include, but not limited to,

- New object classes: for example, those for moving objects [2, 6], personalization objects and resource objects, each of which could be semantically very rich.
- Logical model of a device: including, its capabilities, services it can offer, services it is willing to accept from other services, its location, and temporal/spatial constraints (see below)
- Context awareness modeling: temporal and location properties of a device (e.g., the location of a device and other devices in its neighborhood), and the communities it has established relationships with.

The model will integrate essential features that could be derived from traditional temporal and spatial models, e.g., the modeling of time and location [3, 5]. In addition, there are strong relationships as well as various types of dependencies between time and location in a pervasive computing environment. For example, actions may or may not be allowed to take place at a certain time or location (e.g., targeted advertisement). On the other hand, that an object must be at a certain location at a certain time is a direct dependence of location on time. While object-oriented, temporal, and location models have been studied extensively, they are treated as independently.

The notion of location in pervasive computing environments needs to be explored further. Location information has to be delivered at the level of abstraction suitable to the application. For example, location can be universally defined as a point in the 3-D space (e.g., lat/long/alt). However, it is unlikely to be useful to an application. A home application may specify location in terms of floor, unit, street address, etc., whereas an application may specify in terms of highway number, landmark buildings (next to the Empire Building) or cities (3 miles South of Chicago).

3 Infrastructures

Like the previous section, we use an example to illustrate the need of infrastructural support for pervasive computing applications.

Example 3. Resource discovery: Suppose you have a microwave oven that can be controlled by downloadable recipes. You want to cook a chicken for dinner. You could (i) browse a directory to find the recipe you want (traditional exploratory navigation), (ii) find out the most frequently downloaded chicken recipe, or (iii) find out the nearest download recipe. The example requires an infrastructure that supports dynamic resource discovery.

An infrastructure must be provided to "link" up hundreds of millions of information appliances and services on the network. In the traditional network, a client must specify a destination (e.g., a URL), and a message (e.g., a HTTP request) is routed through the network to arrive at the destination (e.g., a web server). This method will not be able to scale up to billions of nodes in the network. For one thing, it will be difficult for a client to know the address of a service that would meet her needs. For another, the implementation cost (routing tables, etc.) will be very high. As a matter of fact, clients would like to be serviced transparently by any services that could meet her needs. As such, we envisage a service model where

- A request is made when a device can not meet the request by itself
- The device doesn't (need to) know the address of a service provider which could satisfy her request
- The device doesn't care who provides the service, as long as her request can be met within a specific time limit at a specific level of satisfaction (which can be measured in terms of responsiveness, costs, etc.)
- The device doesn't expect an exact/perfect answer as long as the answer meets certain minimum criteria

This above entails a new set of requirements on the infrastructure supporting this service model. Assuming that all information appliances and services are connected physically, the infrastructure has to maintain certain facilities to make the model efficient. For example, the fact that a request may be satisfied approximately means that the infrastructure must support lookup services based on "best" matches [1]. Moreover, the notion of community can be introduced [4]. It will lead to the notion of authority and hub as exemplified in the web. To illustrate this point, imagine a request for recipe (so that the smart microwave oven can prepare a delicious meal for us) originated in Boston may well be satisfied locally by a local master cookery without the need of going all the way to someone in Hong Kong. Furthermore, the notion of "master" can be established by a variation of "authority" in the web. For example, the authority is established by how often a service is invoked to service a request. This notion can be extended in two aspects. I.e., a dynamic registry, created in a community, records how many services are routed into the community and who is invoked to satisfy the requests. In addition, since services can be composed recursively, a somewhat static relationship will also exist among services (e.g., services in E-speak[1]). The relationship can be used to establish a service community. As such, the infrastructure will contain active directory which will detect (by querying the services) the connection between services and the request/service relationship, and establish service communities with authorities identified. This is analog to search engines on the web. The active directories can periodically consult, update, exchange, and synchronize their knowledge.

4 Summary

Pervasive computing refers to computing environments consisting of a large number of computing devices or information appliances embedded everywhere in our living environment. While pervasive computing may exist in different forms, a general theme is that it is massive in terms of the number of devices involved but at the same time invisible to the users in terms of user-friendliness. In this paper, we point out two important aspects of pervasive computing, namely, model and infrastructure. Because of the mobile and transient services in a pervasive computing environment, special attention on the spatial and temporal aspects of modeling, dynamic and recursive composition of services, and infrastructures that match requests to services in a scalable manner are needed.

Pervasive computing is a new field filled with challenging research problems. In addition to the model and infrastructure aspects of pervasive computing, many crucial technical issues, such as power management, mobility management, data synchronization, needs more attention. To realize the vision of pervasive computing, sustained efforts from researchers and supports from government will be essential. With the efforts and supports, we shall see a lot of exciting research progress and technology advances in the coming decade.

References

[1] S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz. An architecture for a secure service discovery service. In *Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99)*, pages 256–262, Seattle, WA, 1999.

[2] L. Forlizzi, R. H. Guting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *Proceedings of ACM SIGMOD Conference 2000*, pages 319–330, 2000.

[3] O. Gunther, T. Sellis, and B. Theodoulidis. Integrating spatial and temporal databases. In *Dagstuhl Seminars 98*, http://www.dagstuhl.de/DATA/Seminars/98/, 1998.

[4] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *WWW9 Proceedings*, 1999.

[5] W. A. O'Neill and E. A. Harper. Linear location translation within GIS. In *Proceedings of ESRI User Conference*, http://www.esri.com/library/userconf/proc97/proc97/to150/pap103/p103.htm, 1997.

[6] E. Pitoura and G. Samaras. Locating objects in mobile computing. In *IEEE Transactions on Knowledge and Data Engineering*, to appear, 2000.

[7] R. Want and G. Borriello. Survey on information appliances. *IEEE Computer Graphics and Applications*, May/June 2000.

[8] M. Weiser. Some computer science issues in ubiquitous computing. *CACM*, 37(7), July 1993.

[1] http://www.e-speak.net, an open source e-service platform from HP, are composed and queried.