Proximity is in the Eye of the Beholder

Peter Barron[1] Jessie Dedecker[2] Éric Tanter[3]

[1] Distributed Systems Group,Trinity College, Dublin, Ireland
[2] Programming Technology Lab,Vrije Universiteit Brussel, Belgium
[3] DCC/CWR, University of Chile, Santiago, Chile
Peter.Barron@cs.tcd.ie - jededeck@vub.ac.be - etanter@dcc.uchile.cl

Abstract. The notion of proximity is a key to scalable interactions in distributed systems of any kind, both natural and artificial, and in particular in pervasive computing environments. However, proximity as such is a vague notion that can be considered both in a very factual manner (spatial distance) and in a very subjective manner (user affinity). We claim that an adequate system or programming language for ambient intelligence applications ought to support an open notion of proximity, making it possible to rely on different, possibly subjective, understandings of proximity, as well as their combinations.

1 Proximity: A Key to Scalability

Proximity can be defined as a *state of nearest*, the perception of *being close* to something or someone. Proximity naturally plays a significant role in how, as humans, we interact with our environment. This can be seen in the relationships we maintain with others, or in the manner in which we interact with everyday objects around us. Think about the closeness of a friend or relative, or about how books of a particular topic are considered to be close to each other.

The concept of proximity is interesting in the field of pervasive computing and ambient intelligence, where the focus is on unobtrusively managing and assisting in the tasks of users. Introducing proximity in these systems allow for better scalability both at the level of interactions [8] and at the level of demarking content of interest [6]. The scoping strategies enabled by proximity make it possible to tailor system behavior to better match the situations of users, going a step further in the direction of the non-intrusiveness requirement identified by Weiser [9].

As an example, consider how service discovery in a pervasive computing environment can take advantage of proximity. Instead of trying to discover *any* accessible service provider that matches the required service type, embedding a proximity criteria within the discovery process can drastically reduce the set of answers to process at the client side. Service providers that are able to determine that they do not meet the proximity requirement will simply skip the service request, thereby alleviating the burden of the client.

Also, the concept of proximity is indeed prevalent in biological systems, such as shoals of fish and social insects like ants and termites, where interactions are

limited to the local environment [2, 4]. To unleash the potential of biologically-inspired communication models in pervasive computing environments, it is hence required to have at hand a notion of locality [1].

From this point of motivation, that *proximity should be supported by pervasive computing environment*, an analysis of possible useful notions of proximity is presented in this paper. The proliferation of potential notions in turn suggests that *proximity should not be hardwired into the infrastructure*, but rather supported in an *open* manner so that application-specific notions can be used and propagated among participants.

2 What Do You Mean, "Proximity"?

> *"Proximity is defined as the state, quality, sense, or fact of being near or next"* – The American Heritage

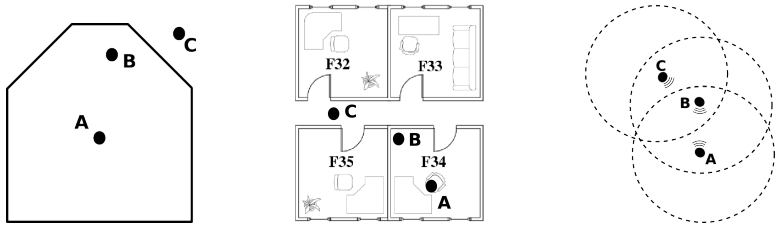
The above definition of proximity leads us to considering two orthogonal dimensions when it comes to analyzing proximity. First, being near or next depends on the notion of distance used; that is, one entity is closed to another *with respect to* a given metric. Such a metric can be based on *physical* properties of the entities (*e.g.* physical location), or on a more *abstract* criteria, not related to the material world (*e.g.* nearness of relatives). We discuss physical vs. abstract proximity in Section 2.1, and then consider the interests of being able to *compose* several proximity metrics in Section 2.2.

Second, the definition mentions the word "sense" in addition to "state" or "fact", which tends to suggest a *subjective* notion of proximity, that depends on the actual perception of the subject entity. This is in contrast to *objective* criteria or metrics, for which all entities share the same understanding of what it means to be close. We elaborate on this dimension in Section 2.3

2.1 Physical vs. Abstract Proximity

Physical Proximity. In current pervasive computing and ambient intelligence systems, the proximity of entities is primarily determined by physical considerations. For example, in YABS [1] interactions are limited to the local environment, where "local" is defined by a geometric parameter (Fig. 1(a)). In Gaia [7], proximity is administratively bound to a physical location which, in this case, is a meeting room (Fig. 1(b)). Taking a different approach, systems such as AmbientTalk [3] implicitly define proximity based on the signal strength of wireless communications (Fig. 1(c)): interactions can only take place when entities are in range of communication.

Abstract Proximity. Physical notions of proximity are very useful in developing pervasive computing systems [6, 8], but it is also possible to extend the benefits of proximity considerations by examining abstract notions of proximity:

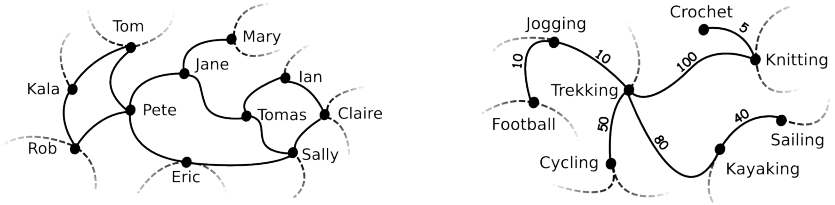


(a) Defining proximity in the form a polygon.

(b) Defining proximity using a physical boundary of a room.

(c) Defining proximity using the signal propagation of a wireless network.

Fig. 1. Different notions of physical proximity (B is "near" A, but C is not).



(a) Defining proximity using the relationships of users. Distance is determine by the degree of separation between two users.

(b) Defining proximity base on the interests or hobbies of users. Weightings on links indicate similarity of hobbies.

Fig. 2. Different notions of abstract proximity

an abstract proximity does not directly map to physical characteristics of the considered entities, but rather relies on logical, domain specific criteria.

First of all, one may consider a virtual rather than physical concept of place: *e.g.* although videoconference participants are geographically at distant places, they all share the same virtual meeting room. On another line, one can define proximity based on the *relationships* of users - friends, acquaintances, or friends of friends. The distance between two users (or entities owned by users) is the degree of separation between them, *i.e.* the length of the path relating them on a relationship graph (Fig. 2(a)). This metric can be used for instance to allow access to your personal devices to yourself, your friends, and friends of friends (that is, a friendship distance of at most 2). One can consider that present instant messenger applications consider the buddy relationship, restricting interactions to a distance of 1. In a different vain, it is possible to define proximity based on the interests or hobbies of users. The distance in this case can be described in terms of the similarity of one hobby or interest to another. For instance, jogging is arguably much more similar to trekking than to knitting (Fig. 2(b)). It is likewise possible to devise a wide number of abstract proximities, related to particular domains or applications.

```
proximity(5);  // circle of radius 5
proximity(-5,-5,-10,5,-10,20,10,20,10,5,5,-5); // polygon as in Fig.1(a)
proximity(F34); // symbolic location as in Fig.1(b)
```

Fig. 3. Proximity definitions in YABS.

2.2 Composite Proximity

Most pervasive computing systems consider proximity as a singular concept: the idea of *composing* different proximities to refine the overall scope of interactions is generally not considered. This is a strong limitation, because considering the potentially wide variety of proximity notions we have discussed above, it is clear that there is a lot to gain in being able to combine different types of proximity to express a more subtle requirement.

For example, composing a proximity base on geometric distance (Fig. 1(a)) and user hobbies (Fig. 2(b)) would first, aid scalability through the scoping of interactions within the local environment, and secondly, highlight content in the local environment that may be of interest. One could also consider spatio-temporal proximity, relating entities that are or have been, within a given time frame, in the same local environment. Another example is to combine spatial locality with network link quality, *e.g.* to aid in the development of an application disseminating multimedia content to local participants

2.3 Objective vs. Subjective Proximity

We now turn to a crucial issue when it comes to considering different notions of proximity in the context of open networks.

Objective Proximity. Existing pervasive computing systems support a notion of proximity that can be defined as *objective* in the sense that the semantics of the proximity function are hardwired in the middleware layer. That is, all entities in the system share the same notion(s). In a system like AmbientTalk, where network connectivity is the only proximity factor, this shared understanding is obvious. In Gaia as well, proximity is defined by physical presence in an active space, *i.e.* a meeting room. In a system like YABS, each entity can define its own proximity requirement using the `proximity` function (Fig. 3). Although the actual parameters of the proximity functions are specific to each entity, the *interpretation* of the proximity function is defined in the infrastructure, and cannot be changed.

Subjective Proximity. The way systems reliant on objective proximity work implies that the different shared interpretations of proximity are installed or configured upfront in the infrastructure. Although this approach is feasible if we consider a limited and fixed number of interpretations (like in YABS), it does

not fit our claim that many proximity notions are of interest, both physical and abstract, and that these notions are potentially specific to certain applications or domains. It is necessary that clients are able to define, compose and use new notions of proximity.

In other words, if a new entity joins a certain environment and looks for services of a certain type that are "close" to it, this entity ought to be able to use its *own notion* of what it means to be nearby. This means that the proximity function should be defined by the client itself, not pre-defined by the underlying infrastructure. In this case there is no globally shared understanding of the proximity, rather a *subjective view* of the client, that reflects the particular requirements of the application.

3 Perspective: A Proximity Metaobject Protocol for AmbientTalk

Pervasive computing systems typically fail to support many abstract and composable notions of proximity, as well as to allow subjectivity in proximity definitions. It is clear that such flexibility raises important challenges at the implementation level. It is however, to our understanding, a very important and valuable approach to enable better scalability and usability in open pervasive computing systems.

We are currently exploring a proximity metaobject protocol for the Ambient-Oriented Programming language AmbientTalk [3]. Metaobject protocols are well-defined interfaces to the language implementation that allow the semantics of the language to be customized by programs [5]. In our case, this extends to a distributed language with proximity support.

We plan to first provide this proximity metaobject protocol at the service discovery level. In a second phase, it is necessary to go further, considering that since proximity can change dynamically, a service that was near at the time of discovery may "move away" while interactions are in progress.

Finally, although subjective notions of proximity imply that one client perceives its surrounding in a particular manner, it is important to distribute the *evaluation* of proximity functions among nodes, to limit network traffic. It can also be interesting to actually propagate proximity functions so as to dynamically upgrade the knowledge of involved participants with new proximity notions.

4 Summary

In this position paper, we have drawn attention to the important notion of proximity for building scalable and relevant pervasive computing and ambient intelligence applications. Starting from the different possible notions of proximity that can be useful, both physical and abstract, as well as their user-defined composition, we have argued that proximity should not be hardwired into the infrastructure, but rather supported in a way that makes it possible to use application-specific notions in a subjective manner.

References

1. Peter Barron and Vinny Cahill. YABS: a domain-specific language for pervasive computing based on stigmergy. In *GPCE '06: Proceedings of the 5th international conference on Generative Programming and Component Engineering*, pages 285–294, New York, NY, USA, 2006. ACM Press.
2. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence From Natural to Artificial Systems*. Oxford University Press, 1999.
3. Jessie Dedecker, Tom Van Cutsem, Stijn Mostinckx, Theo D'Hondt, and Wolfgang De Meuter. Ambient-oriented programming in AmbientTalk. In Dave Thomas, editor, *Proceedings of the 20th European Conference on Object-oriented Programming (ECOOP 2006)*, Lecture Notes in Computer Science, pages 230–254, Nantes, France, July 2006. Springer-Verlag.
4. P.-P. Grass. Le reconstruction du nid et les coordinations inter-individuelles chez bellicositermes natalensis et cubitermes sp. la theorie de la stigmergie: essai d'interpretation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–81, 1959.
5. Gregor Kiczales, Jim des Rivières, and Daniel G. Bobrow. *The Art of the Metaobject Protocol*. MIT Press, 1991.
6. Tim Kindberg and Armando Fox. System software for ubiquitous computing. *IEEE Pervasive Computing*, 1(1), 2002.
7. Manuel Roman, Christopher Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt. Gaia: a middleware platform for active spaces. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(4):65–67, 2002.
8. M. Satyanarayanan. Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4):10 –17, Aug 2001.
9. M. Weiser and J. Brown. The coming age of calm technology. *PowerGrid Journal*, 1.01, July 1996.