K-Trek: A Peer-to-Peer Infrastructure for Distributing and Using Knowledge in Large Environments

Paolo Bouquet[∗†], Paolo Busetta[†], Giordano Adami[†], Matteo Bonifacio[∗†], Francesco Palmieri[‡]
[∗]Department of Information and Communication Technology – University of Trento
Via Sommarive, 10 – 38050 Trento (Italy)
Email: {bouquet}@dit.unitn.it
[†]ITC-IRST – Istituto per la Ricerca Scientifica e Tecnologica
Via Sommarive, 14 – 38050 Trento (Italy)
Email: {busetta,gioadami,bonifacio}@itc.it
[‡]Generation3 s.r.l.
Via Valpetrosa, 10 - 20123 Milano (Italy)
Email: francesco.palmieri@generation3.it

***Abstract*— In this paper, we explore an architecture, called *K-Trek*, that enables mobile users to *travel across knowledge* distributed over a large geographical area (ranging from large public buildings to a national park). Our aim is providing, distributing, and enriching the environment with location-sensitive information for use by agents on board of mobile and static devices. Local interactions among K-Trek devices and the distribution of information in the larger environment adopt some typical peer-to-peer patterns and techniques. We introduce the architecture, discuss some of its potential knowledge management applications, and present a few experimental results obtained with simulation.**

I. Introduction

In several recent papers, the idea of *Distributed Knowledge Management* (DKM) was proposed as a new and promising approach to the design and implementation of systems for managing knowledge within complex organizations and, in general, in scenarios in which there is a multiplicity of autonomous knowledge sources [1]. The main idea was that, over time, different people (or groups of people) produce heterogeneous and partial views (called *contexts*[1]) on the information available within an organization, each from their own perspective (*principle of autonomy*), and that these views – far from being an obstacle to management and coordinated action – are a potential source of innovation and knowledge creation, if suitably managed (*principle of coordination*).

Current work on DKM focuses on issues of semantic autonomy and coordination; for example, an experimental testbed has been developed in a peer-to-peer system called KEx (*Knowledge Exchange*) [5]. Here we explore a different direction of DKM, closely related to what is called *ambient intelligence*. We imagine a scenario in which knowledge is context-dependent not only because it embodies the (semantic) perspectives of different people (as for KEx), but also because (i) it says something about a specific location of a given environment, and (ii) is physically stored in that location. To understand the underlying intuition, we suggest an analogy with signs in the physical world, which provide the intended information only if they are placed in the location in which they were designed to stay.

We present an architecture, called *K-Trek*, that supports this new form of *context-awareness*. K-Trek enables mobile users to *travel across knowledge* distributed over a large geographical area (ranging from large public buildings to a national park). This is obtained by providing, distributing, and enriching the environment with location-sensitive information for use by agents on board of mobile and static devices.

Context-aware computing is an area of active research at the very heart of *pervasive computing* and *ambient intelligence* [6], even if a clear focus has yet to emerge (see for instance the recent [7]). Context-awareness is usually defined as sensitivity to the user's state, the environment where she currently is, and the current physical environment [8]. Distinguishing features of our approach with respect to the known literature are:

- our definition of *context*, derived by applying the formal framework described in [4] to knowledge management issues, is based on data accumulated and categorized by each user during an extended period of time. An explicit negotiation phase (which subsumes traditional feature-based selections based on user preferences or profiling as particular cases) is used to filter or annotate information given to and left by users during their movements;
- no long-range, permanent wireless networks or sensors of any kind are involved. Instead, we "augment" the environment, as well as mobile devices, with very low cost, easily available hardware for wireless, short range communication. Bluetooth [9] is our reference technol-

[1]This definition of context is a direct derivation of the work on the contextual reasoning by Giunchiglia and his group [2], [3], [4].

ogy, but the architecture can be easily adapted to future standards as they will emerge;
- agents on board of static as well as mobile devices can exploit the users they get in contact with for transporting information to agents they cannot directly reach.

K-Trek adopts some typical peer-to-peer patterns and techniques. Small peer-to-peer networks are formed on-the-fly and enable localized, context-aware interactions among agents. Users movements are exploited to provide message transport in the larger environment, in a way that reminds query propagation on some well-known peer-to-peer networks. This mechanism is effectively a particular form of *ad hoc* wide area networking that does not need any permanent long-distance communication infrastructure.

The paper is organized as follows. First, in Section II, we describe the architecture of K-Trek. Context awareness in K-Trek is discussed in more detail in Section III. Section IV proposes some application scenarios. Finally, Section V illustrates some experimental results collected by simulating a few different scenarios.

II. K-Trek: an overview

K-Trek is an infrastructure based on three main types of device, called K-Trek devices (see Figure 1):

K-Beacon: static device (such as an embedded system with integrated Bluetooth board) which stores contextual information about a specific location (i.e., the location where it is placed), and can interact in various forms with other K-Trek devices;

K-Voyager: mobile device – such as a PDA or a last generation mobile phone – with any number of K-Trek applications on board;

K-Plug: any device with a standard network interface that acts as a gateway between K-Trek devices and back-end servers.

K-Trek devices can be connected to each others in two main types of networks: *K-Trek micro networks*, i.e. on-the-fly networks that connect a limited number of K-Trek devices in a very small geographical area; and K-Trek Wide Area Network (K-wan), i.e. a wide-area, message-based, asynchronous network, where mobile K-Trek devices may be used as temporary bridges between disconnected devices on the same K-wan. In the next two sections, we describe how the two types of networks work.

A. Micro networking with localized resources

The main feature of K-Trek is the ability of setting up "micro networks" on-the-fly, i.e. networks that cover a very small geographical area (no more than a few ten meters) with a limited number of devices and limited bandwidth, without the need for dedicated, static equipment (wires, routers, access points, or other paraphernalia)[2].

Communication among K-Trek devices in a micro network is in charge of special light-weight message handling agents. Their tasks include the most basic peer-to-peer interaction, i.e. discovery. To this end, they periodically broadcast *announcements*. For instance, a K-Beacon announcement contains the K-Beacon's contact information and a set of short messages, sent by local application agents and directed to the agents on passing K-Voyagers. The processing of this announcement is discussed later, in Section III. This discovery-based approach, inspired by peer-to-peer systems, contrasts with location-aware systems based on geographical coordinates, commonly adopted with mobile phones and other wireless networks, for various reasons. First, no location sensor such as a Global Positioning System (GPS) is needed. Second, since there are no coordinates, there is no need for geo-referencing information to be delivered to users, as it is commonly required when central services are involved (typically with mobile phones), or when local applications need to retrieve data already on board of the user's mobile device or to access a centralized directory.

B. K-wan: a wide-area asynchronous network

The second type of network is what we call K-Trek Wide Area Network (K-wan). A K-wan is a wide-area, message-based, asynchronous network, where messages may be delivered long after being posted, and only stochastic guarantees are given concerning their actual delivery, latency, and the geographical area of distribution. As discussed later, a K-wan exploits the users' movements for message transport, thus no special equipment is required other than what is required to set up micro-networks (e.g., Bluetooth boards).

Some micro networking mechanisms are implemented by the message handling agents to support transport within a K-wan. One of them follows the K-Beacon discovery by K-Voyagers mentioned in Section II-A above, and consists of two complementary actions. The first is downloading any message for the K-Beacon contained in a dedicated K-wan buffer on board of the K-Voyager; in other words, a K-Voyager delivers, to the K-Beacons it gets in contact with, anything for them that was picked up during its trip. Conversely, the second action is uploading on the K-Voyager messages from the K-Beacon directed to agents running remotely.

The second mechanism needed by K-Wan is applied between K-Voyagers. The announcement mechanism of Section II-A enables K-Voyagers to discover each others; this is followed by the exchange of the contents of their K-wan buffers, in a truly peer-to-peer fashion. At the end of this process, any message addressed to either of the two K-Voyagers is delivered to the appropriate agent and discarded from both buffers (since it reached its destination), while all

[2]Our reference technology is Bluetooth [9], because it is suitable to very low-cost, low-power, wireless devices, and it is commonly built into many last-generation mobile phones and PDAs. However, the definition of K-Trek devices is independent from Bluetooth.

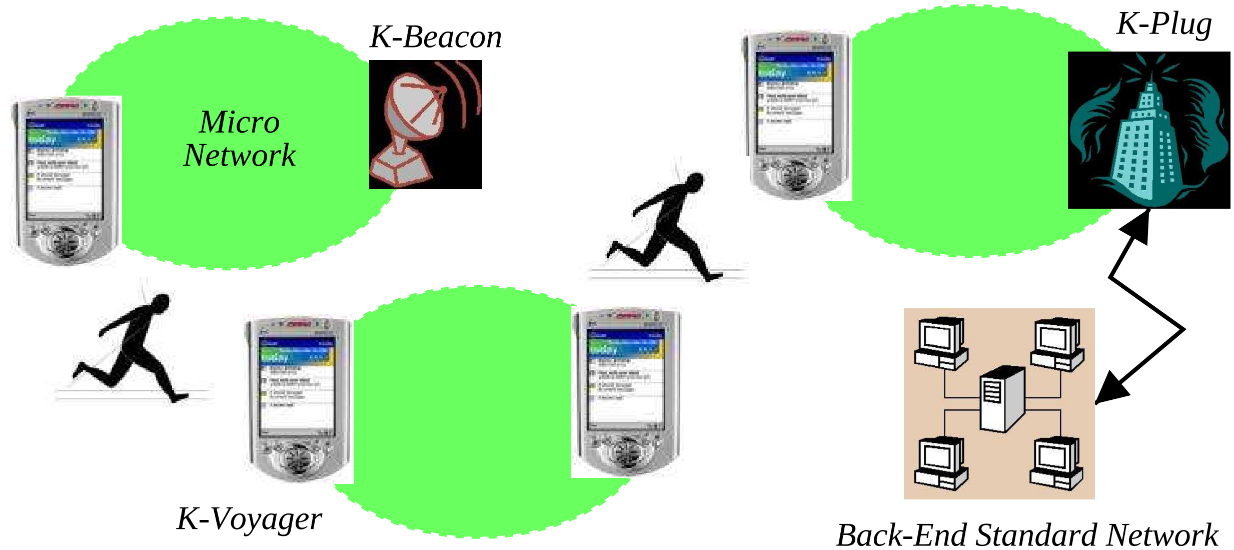

Fig. 1. K-Trek: main components

others are duplicated[3].

The last major micro networking mechanism used by a K-wan involves the third type of K-Trek device, K-Plugs. A K-Plug can be any device (e.g., a personal computer or a Bluetooth *access point*) with a standard network interface that acts as a gateway between devices on a K-wan and back-end servers. To this end, all K-Plugs provide access to a single, centralized mailbox service. When a K-Voyager gets within the range of a K-Plug, a set of peer-to-peer protocols similar to those presented above are used to deposit messages for agents on back-end systems, and to pick up messages addressed to the K-Voyager (or its user) and for other K-Trek devices; the first are immediately delivered to their destination agents, while the others are deposited in the K-wan buffer.

We expect that more than one K-Plug are part of a K-wan. Ideally, they should be located in places where, sooner or later, most if not all users pass by[4]. In situations where the paths followed by users can be predicted, messages for a K-Beacon K are distributed only by the K-Plugs along the paths that touch K. Since message duplications are likely while delivery cannot be guaranteed, care is taken in the mailbox administration, for instance by making sure that messages for K-Beacons are not removed until expired or requested by their senders (possibly after an application-level handshake).

A K-wan is particularly suited to cases where low-power embedded systems distributed on a large territory need to perform occasional exchanges of non-critical data (e.g., collecting data from sensors detecting animal or tourist movements in a national park). These scenarios currently require either expensive links (such as microwaves or satellite), or people physically going to each device for uploading and downloading data via floppy disks or other media. As shown in the examples in the concluding section of this paper, a careful analysis can predict the performance of a K-wan with some precision. To this end, we have developed analysis tools that can be used to set up a K-wan so that any required level of performance (e.g., maximum time for delivery) is achieved, thus making a K-wan appealing for a large number of application scenarios.

III. Context-sensitive mobile applications in K-Trek

Our first objective is to enable the exchange of contextually relevant information among the K-Trek devices temporarily connected in a micro-network. Context here is used in two distinct senses:

1) *context as location*: this is the more traditional sense of context in context-aware applications. However, K-Trek supports a particular form of location-awareness, where the "location" is determined not by geographic coordinates but by the co-presence of other K-Trek devices (e.g., a meeting can happen anywhere as long as all the required participants are present);
2) *context as perspective*: context here is used in the same sense of standard DKM, and refers to structures that encode a semantic perspective on a collection of "objects" (as in KEx). Whenever a micro-network is established, K-Voyagers discover whatever resources are

[3]This buffer content exchange happens whenever two users carrying K-Trek devices get close by, without any human involvement. This effectively implies that messages spread around the geographical area covered by moving K-Trek users as a sort of benign – but highly infectious – virus. A number of mechanisms – such as setting expiration dates on messages, maintaining lists of those already delivered, managing buffer overflows – are used to keep things under control. However, a number of questions arise about this transport technique, e.g. what buffer size is required, what is the probability of reaching the destination, which geographical area is covered; the answers are affected by many factors, the most important being the pattern of movement of users. The last section of this paper shows some studies on the suitability of a K-wan to specific scenarios.

[4]For this reason, and to reduce the amount of circulating messages, a K-Trek administrator may configure K-Plugs so that K-Voyagers can pick up messages for themselves and for K-Beacons, but not for other K-Voyagers.

available on other K-Trek devices, attempt to perform mappings between the contexts they have on board and those on board of the others[5], and act consequently (e.g., they may report on the findings to their users). Context-sensitivity is achieved by "augmenting" the environment with K-Beacons, with their own contexts on board, representing or annotating local information such as data generated by local sources (typically on embedded systems) or information left by other mobile devices.

Application agents running on a K-Voyager are associated to one or more contexts. By operating on the K-Voyager's GUI, the user decides which applications, and which contexts, to keep active. This means that user gets *only information relevant to her at that particular time at that particular location* – which is to say, a K-Voyager is context-aware as commonly meant [8]. Since the interaction is two-way, also data flowing from K-Voyagers to K-Beacons can be annotated with contextual information, so agents on the static device can get additional information on mobile users and possibly select only that information that is of their interest.

User contexts can be edited by users; this is a typical off-line process, better performed on a more convenient platform than a mobile device, e.g. a PC. Similarly, contexts on board of K-Beacons are typically edited off-line and downloaded by a system configurator. In the future, it is foreseeable that contexts may be acquired semi-automatically by K-Trek devices themselves, e.g. in a mixed-initiative process where some of the information collected by a K-Voyager during a trip is suggested to the user for addition to her contexts.

The interactions between K-Trek devices follow a common pattern; we illustrate here the case of a K-Voyager in the range of a K-Beacon. When the message handling agent on board of the K-Voyager receives a K-Beacon announcement, it performs a discrimination of its content, then a first type of context-sensitive processing. Application messages addressed to a remote system or to a different K-Voyager are stored in the K-wan buffer; their processing has been discussed above. The others (i.e., those addressed to either anybody or specifically to this K-Voyager) are filtered against the user contexts (using context mapping techniques).

Eventually, the messages left after filtering are delivered to their destination agents. Typically, these messages are further application-specific announcements or local information to be shown to the user. Apart from those described in Sec. II-B, further interactions between K-Beacon and K-Voyager are driven by the application agents, for instance to retrieve or deposit data or obtain services from K-Beacon agents. Since a K-Voyager may fall within reach of multiple K-Beacons, application agents must be able to handle simultaneous interactions.

[5]For a description of context mapping techniques, see [10].

IV. Distributed Knowledge Management Applications on K-Trek

Most things that one can imagine doing in the physical world by putting a sign, leaving a mark, depositing a form in a mailbox, attaching a "post-it" card, and so on, can be done electronically with K-Trek, with the exception of those actions that require knowledge of the exact location and direction of the user (e.g., direction-giving relative to the user position, such as "move for 20 meters on your left and you will see the Colosseum", cannot be supported without additional sensors).

Looking at K-trek from a broader knowledge management perspective, its architecture is suitable to situations in which:

- the physical environment is populated by objects whose value can be increased by either delivering to, or collecting information from, other objects or users;
- linking these "informative" objects by means of an information network based on long-distance wireless connections is not feasible, because of costs or environmental constraints;
- mobile actors in the environment need to locally exchange information either with informative objects or with other actors;
- mobile actors move across the environment along paths that, statistically, connect all the informative objects;
- an environment administrator has an interest in enhancing the environment through the provision of infrastructural services;
- there may be external actors that have an interest in "owning" the informative processes related to one or more objects.

A first example of potential K-Trek enabled environment is natural parks and, in general, geographically dispersed entertainment environments such as archaeological sites. Parks are populated by objects such as natural attractions, routes or historical sites whose value can be enhanced if able to exchange information with users, other objects, the administrator, or the "owner" of the site (an entity that has an interest in updating and collecting the information that belong to the site). For example, a historical site may receive information: from a school of architecture in order to update its description; from a visitor that wants to leave a message to those that will visit the site in future ("virtual post-it"); and, from a member of the maintenance staff that has periodically to asses its status. Conversely, the site can provide: architectural information to a visitor whose context shows an interest in architecture; maintenance information to inspectors, previously deposited by members of the maintenance staff; and, information about number of visits, type of users and the kind of information they deposit on the site to the park administrator. Visitors and maintainers unintentionally provide the "lazy" communication channel needed to ensure information delivery, update, and collection by K-wan.

Another scenario involves field management activities of geographically distributed industrial settings. Relevant objects are industrial sites or components (power stations, junction boxes, and so on) that generate information about their status and collect information about those maintenance activities that must be performed and assessed in site. Here, since the certainty of information delivery and collection is more critical, maintenance visits are intentionally scheduled not just as a function of each maintenance task, but also for enabling the circulation of information across the overall system. For example, maintainer A that has to visit and asses the status of site 1, has a route that passes in front of site 2 whose maintenance is under the responsibility of the maintainer B. In such case, A deposits his visit report on site 1 and automatically collects the visit report of B done on site 2. The latter will be delivered to the environment administrator whose task is to monitor the overall system[6].

It is worth to stress again that the annotation of messages with information taken from the originating agent's contexts helps in performing typical knowledge management tasks, varying from the ability to support communities of mobile users to classical data mining processes, such as understanding tourists' interests, identifying patterns of visit per user category, and so on.

V. Quantitative studies on K-wan

Before deploying a knowledge management solution, even before developing any software for K-Trek, we deemed necessary to assess the characteristics of a K-wan and to define a set of criteria for network design. This is a very complex task, because a large number of factors influence the network behavior: for instance, the number of mobile users, their patterns of movements, the number and location of K-Beacons and K-Plugs, the size of the K-wan buffers, the lifetime of messages. The general question to be answered can be formulated as follows: given a certain configuration, what is the probability that a message reaches its destination within a given timeframe? Or, equivalently, which factors should a network designer focus on, so that messages are delivered on time with a given probability (possibly 100%)?

The most effective way to answer this question is through simulation. For our initial studies, we adopted a multi-agent simulation tool, called NetLogo [11] – easy to use, ideally suited to classroom experiments but not adequate to complex scenarios analysis; nonetheless, it revealed to be enough for our objectives. Ultimately, our aim is to build a library of models that cover a reasonable large number of situations, and use it as a design tool for a K-wan. In the following, we discuss two simple models and present some of the collected results.

Objective of our first model was to understand if we could identify some relationships among a selected set of parameters on a relatively small scale scenario. The model has not been thought with reference to any specific domain. A grid of roads, whose overall size and density was controlled via parameters, was randomly generated and a set of travelers with K-Voyagers scattered over them. Travelers followed random walks at a fixed speed, and bounced back when reaching the border of the grid. A set of K-Beacons and K-Plugs were casually scattered over the grid. A constant number of messages (5) were generated by K-Beacons with random destinations, which could be either specific K-Voyagers or generic back-end applications (that is, any K-Plug). We ran a large batch of simulations, varying road density, number of K-Voyagers, K-Beacons, and K-Plugs, lifetime of messages, and size of the K-wan buffers.

[6]The scenario above provides an example on how K-wan can handle certain levels of information criticality when the administrator is able to exploit the value of predictable "visit paths" in terms of connections that will happen with a known frequency and with a known level of reliability. Another good example is represented, in a urban environment, by mailmen that, in addition to their usual task of mail delivery, might deliver to and collects updates from those K-Beacons that are positioned on their typical routes.

K-voyagers	Buffer size %	% delivered
20	30	30
20	60	31
20	90	31
40	30	49
40	60	50
40	90	50
60	30	60
60	60	66
60	90	69
80	30	76
80	60	81
80	90	81
100	30	78
100	60	87
100	90	88

TABLE I
EXPERIMENTAL RESULTS: GENERIC MODEL

The table I contains an extract from one of the many statistics we elaborated, the most interesting in our opinion. The first column is the number of K-Voyagers; the second, the size of the buffer as a percentage of the total number of circulating messages (i.e. 5 times the the number of K-Beacons); finally, the average percentage of messages that reached their destination, which revealed to be quite independent of other parameters. What the table shows is, in summary, that in all the configurations we simulated the buffer size is relatively unimportant, while the most important factor is the density of K-Voyagers. This is not surprising – as any doctor would tell, the highest the density of the population, the highest the chance for a virus to spread. No matter how good this result looks like, we refrain from jumping to definitive conclusions, since there is too a large number of configuration choices (e.g., the way we distributed roads), policies (e.g., concerning buffer overflow management), and behaviors (e.g., paths followed by travelers) to consider this model of general applicability.

Differently from the previous one, the second model was built by analyzing a realistic scenario, which is also a potential target domain: tourism in a historical town. We recreated a partial and slightly simplified map of the town center of Trento, Italy, roughly corresponding to a square with a 600

K-Voyagers	Lifetime (min)	% delivered
5	15	24
5	30	37
5	60	48
5	120	60
10	15	29
10	30	41
10	60	57
10	120	60
15	15	34
15	30	53
15	60	58
15	120	61
20	15	43
20	30	57
20	60	66
20	120	69
25	15	41
25	30	58
25	60	62
25	120	68
50	15	42
50	30	61
50	60	61
50	120	67
100	15	43
100	30	55
100	60	64
100	120	67

TABLE II

EXPERIMENTAL RESULTS: TOURISM IN TOWN

mt long side. This historical center features a thick network of roads, fairly typical of medieval towns, open to pedestrians only. We assumed that Bluetooth devices can communicate at a distance of up to 30 mt, which experiments show to be a conservative estimate in open spaces. Mobile users crossed the mapped area following a random walk at a speed of 5 km/h; also, they could stop anywhere for a while, or leave and come back later. On average, a mobile user stayed within the area for an hour. We put 4 K-Plugs at the corner of busy streets. Twenty food outlets (restaurants and cafes) advertised their presence with K-Beacons. Similarly to the previous model, these K-Beacons periodically sent messages to K-Voyagers or to back-end applications (thus, delivered to any K-Plug).

In our reference application, a message contains the address of the outlet owning the sending K-Beacon and a note left by a passing tourist with a K-Voyager; examples of notes include remarks on the outlet, suggested meeting location, satisfaction forms for the tourist office. A note can be sent either to another K-Trek user (that is, to a K-Voyager), or to an Internet email account (by means of an e-mail server, i.e. via a K-Plug).

For our simple model, we assumed that every K-Beacon had always two messages to deliver. A new message was generated when one expired. The number of mobile users was constant over time. Message destinations were chosen randomly in a set formed by the K-Voyagers plus 4 e-mail addresses; for instance, given 96 users, there was a 4% probability that a message had to be delivered to the e-mail server via a K-Plug. We set the K-wan buffer size to 50% of the number of circulating messages, i.e. 20. Our goal was to determine the probability that a message reached its destination, as a function of its lifetime and the number of mobile users.

The table II summarizes the results we obtained after simulating a 12 hours period by discrete cycles corresponding to a simulated period of 10 seconds each. It can easily be seen that the message lifetime, not surprisingly, has an important impact. After analysis, we found out that undelivered messages were for K-Voyager users that left the area too soon to be reached, while e-mails were always delivered (apart from unrealistic cases of very short lifetime, not shown in the table). The number of mobile users has an important influence, too, in a slightly surprising way. Indeed, with high density, messages lifetime decreases its importance, indicating that messages spread around more quickly than with lower densities; still, the best case is with a relatively low number of K-Voyagers. The reason seems to be the buffer size - indeed, the quicker messages spread around, the higher the chance of buffer overflows (our management policy is FIFO). For our reference application, we consider these results satisfactory.

REFERENCES

[1] M. Bonifacio, P. Bouquet, and P. Traverso, "Enabling distributed knowledge management. managerial and technolo gical implications," *Novatica and Informatik/Informatique*, vol. III, no. 1, 2002.

[2] F. Giunchiglia, "Contextual reasoning," *Epistemologia, special issue on I Linguaggi e le Macchine*, vol. XVI, pp. 345–364, 1993, short version in Proceedings IJCAI'93 Workshop on Using Knowledge in its Context, Chambery, France, 1993, pp. 39–49. Also IRST-Technical Report 9211-20, IRST, Trento, Italy.

[3] M. Benerecetti, P. Bouquet, and C. Ghidini, "Contextual Reasoning Distilled," *Journal of Theoretical and Experimental Artificial Intelligence*, vol. 12, no. 3, pp. 279–305, July 2000.

[4] C. Ghidini and F. Giunchiglia, "Local Models Semantics, or Contextual Reasoning = Locality + Compatibility," *Artificial Intelligence*, vol. 127, no. 2, pp. 221–259, April 2001.

[5] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori, "Kex: a peer-to-peer solution for distributed knowledge management," in *Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-2002)*, D. Karagiannis and U. Reimer, Eds., Vienna (Austria), 2002.

[6] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman, "Scenarios for ambient intelligence in 2010," Information Society Technologies Programme of the European Union Commission (IST), Tech. Rep., Feb. 2001, `http://www.cordis.lu/ist/`.

[7] G. D. Abowd, M. Ebling, G. Hunt, H. Lei, and H. Gellersen, "Context-aware computing," *IEEE Pervasive Computing*, vol. 1, no. 3, 2002.

[8] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, 1994. [Online]. Available: citeseer.nj.nec.com/schilit94contextaware.html

[9] I. Bluetooth SIG, "The Official Bluetooth Wireless Info Site," 2002, `http://www.bluetooth.com/`.

[10] P. B. B. Magnini, L. Serafini, and S. Zanobini, "A SAT–based algorithm for context matching," in *Proceedings of the 4th International and Interdisciplinary COnference on Modeling and Using Context (CONTEXT-03). Stanford University (CA), June 23-25, 2003*, ser. Lecture Notes in Artificial Intelligence, P. Blackburne, C. Ghidini, R. Turner, and F. Giunchiglia, Eds., vol. 2680. Springer Verlag, 2003.

[11] U. Wilensky, "NetLogo," Evanston, IL, 1999, `http://ccl.northwestern.edu/netlogo/`.