Smart Clustering – Networking Smart Objects Based On Their Physical Relationships

M. Strohbach
Computing Department
Lancaster University
Bailrigg
Lancaster LA1 4YR
United Kingdom
Email: strohbach@comp.lancs.ac.uk

H. Gellersen
Computing Department
Lancaster University
Bailrigg
Lancaster LA1 4YR
United Kingdom
Email: hwg@comp.lancs.ac.uk

***Abstract*-We propose a new concept that facilitates networking of smart objects by creating topologies that reflect the physical environment. As the objects are clustered according to their physical relationships we obtain a self-organizing, scalable and fault-tolerant peer-to-peer like network. This paper presents the ideas and benefits of this concept and our ongoing work that implements the detection of the relationships between a load-sensing table and sensor-augmented objects.**

I. Introduction

In the near future even arbitrary, non-technical objects will be networked. Thus, the challenge of structuring the increased number of networked nodes in a scalable and adaptable way becomes even more difficult. In this paper we present a possible solution to this problem by augmenting objects with sensing and communication technology that enables them to detect physical relationships and organize themselves accordingly in a scalable and fault-tolerant peer-to-peer (P2P) like network. We present our idea along the example of a load sensing table and sensor-augmented objects.

A. *The concept*

The network topology is usually a result of a device discovery protocol and depends on the characteristics of the used communication technology like radiofrequency (RF) or infrared (IR). Devices that are in communication range automatically join the network and are clustered in such a way that communication traffic is most between nodes within a cluster. This minimizes communication costs as communication across cluster boundaries is usually more expensive. The result is a topology that does not take information about the physical world into account. Thus, scalability can only be maintained by considering network parameters like the number of nodes which leads to splitting and rejoining clusters.

Our goal is to optimize the clustering process by creating topologies that match the physical environment. As the communication technology itself is not sufficient for this task, we use sensors of augmented objects. Thus, we are able to detect relationships like “object put on table”. This information is then used to build a topology that consists of clusters that reflect these relationships, e.g. a cluster consisting of the table and the objects on its surface. Scalability is maintained as the number of objects that are in a physical relationship like “on top of a table” are limited. No network reorganizing operations like splitting and joining of clusters have to be performed as clusters adopt instantly according to changes in their physical environment, e.g. removing an object from a table.

We also include static knowledge about the objects’ physical properties like identity and type. This facilitates organizing the communication infrastructure within the cluster and facilitates the choice of a communication master based on three main characteristics:

1. memory and computing power
2. abilities to detect changes in the physical relationship used for clustering
3. Perpetuity: how long has the object been or will it be part of the physical relationship compared to other objects.

The master monitors changes in its relationships, connects the objects within its cluster by answering queries and structures the network by offering access to its objects from outside the cluster.

By taking these three characteristics into account our approach is not dependent on a dedicated central node as long as the three characteristics can largely be satisfied. In section IV we will show how due to the fact that our objects are equipped with a set of generic sensors they are still able to detect changes in the environment and negotiate a new master.

In section II we will describe related work. Section III will describe the design including the used technology and ongoing implementation work. Section IV will discuss the implications of our concept. Finally, section V concludes the paper.

II. Related work

Although the RF signal is influenced by the physical environment like walls and objects, it cannot easily be used to discover physical relationships like objects on a table. Even mere positioning with RF seems to be a hard problem. However, infrared (IR) has been used in the Active Badge location system [3] in order to gain room level accuracy. Here, the property that infrared light is not able to penetrate walls has been used to structure the network. In order to increase the accuracy dedicated objects like tables have been equipped with low range radio transmitters. However, accuracy was too low to detect if a object was on or just near a table.

The Active Bat [1] positioning system achieved centimeter level accuracy in 95% of the cases. With such accuracy relationships like “on top of the table” or “close to the table” could theoretically be detected. However, it would be more prone to errors due to accuracy limitations and its dependency of the infrastructure that must be able to update the model of the complete environment in real-time including the calculation of the relationships. Furthermore, only physical relationships can be used whereas we want to include every information about the environment that might be useful. This requires additional sensors.

The idea of linking devices according to user's interaction has been introduced in the Smart-its friends application [4]. However, no general concept for connecting multiple objects according to their context or their relationships has been used.

Several other objects pursue the idea of networked surfaces that all have in common that networking is enabled by objects if they are attached to a surface. The surface plays an active part in the communication as it provides power, the network medium itself or both functionalities. The pushpin [11] and Networked Surfaces [7] focus on the technical implementation of networked surfaces. They do not exploit the concept of relationships of smart objects in general. The Pin&Play project [6] explores surfaces like walls and notice boards as communication medium to provide connectivity to objects like light controls, pictures and notes that are already attached to walls.

III. Design

In this section we will describe the ongoing implementation of our general concept. We use a load sensing table that acts as master and sensor-augmented bottles and glasses. The relationship by which the objects are clustered is "on the table".

A. The Technology

The table as well as the augmented bottle and the augmented glass are built on the smart-its hardware platform. Smart-its are small computing devices with sensing and communication capabilities. The vision of the Smart-its project [13] are devices that can be attached post-hoc to arbitrary objects just as unobtrusive as bar codes on virtually every object that can be bought in a supermarket. Objects augmented with this technology are able to sense their environment, perform tasks collectively and are independent of fixed infrastructures.

Two of the three available sub-platforms are being used for this application. For the augmentation of the bottle and glasses the smart-its P hardware platform is used. It consists of a PIC-based RF core board sending on the 800MHz band. It uses a slotted time division multiple access (TDMA) carrier sense/collision avoidance (CS/CA) based MAC protocol. Features include the adjustment of the signal strength achieving a range of up to 15m indoors. The core board is connected via I^{2}C to a PIC-based sensor board offering five standard sensors: light, pressure, acceleration, microphone, and actuators including LEDs and a buzzer. It has further connectors for adding own sensors and for interfacing via RS232 to a standard PC. The application code is running on the sensor board. This platform has been optimized for size, power consumption and reducing communication latencies. This is achieved by choosing a broadcast based protocol generally avoiding device discovery for simple applications.

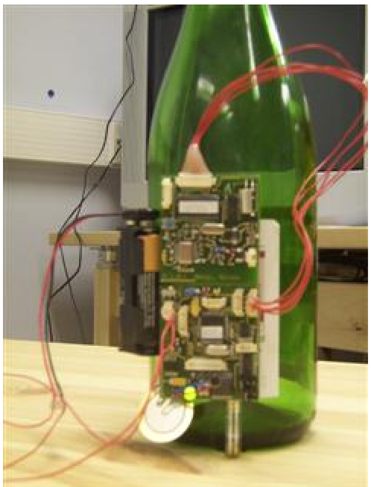

Fig 1: Smart-its enhanced bottle

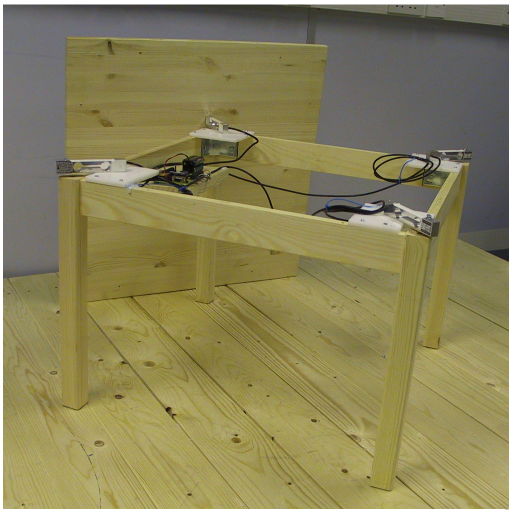

Fig. 2: The Load Table

Bottles (Fig. 1) and glasses have been augmented with the Smart-its P devices. The pressure sensor has been attached to the bottom of the objects in order to detect if the object is on the table.

The Rapid Prototyping platform was used as the basis for the table (Fig. 2). This platform is modular as well but based only on one PIC. A wide range of sensor board add-ons are available including a load-sensor board that is being used to interface the load cells of the table. This platform has been optimized for fast and easy assembly and rapid application development [14]. As the

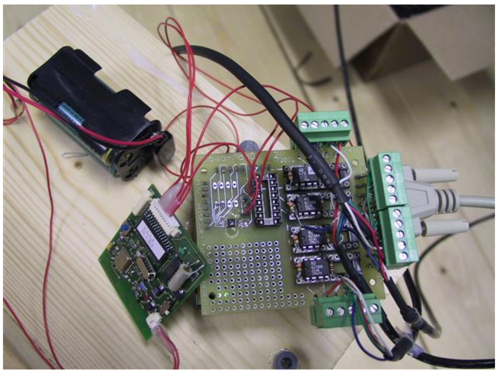

Fig. 3: Load sensing board with Smart-its P core board

prototyping platform uses a different RF encoding the RF-Chip has been replaced by a smart-its P core board in order to achieve interoperability (Fig.3).

By using the four load cells the table is able to detect events like “object put on table”, “knocked over”, or “removed from table”. In addition, for every object put on the table – augmented or not – the table knows its weight and position. For further information on the load table and its applications see [9,10].

B. *Implementation*

When a user puts the augmented bottle on the table the object is automatically joined to the “objects on the table” cluster by correlating the events from the object and the table. The object’s event is generated by the pressure sensor that is attached to the bottom of the bottle. The event is then broadcasted together with a short description of the identity and type of the object.

At the same time the table senses a new object at position *(x,y)* with weight *w*. As the table receives the event of the bottle it can link the object’s identity with position and weight by comparing the time stamps of the events. The table is the ideal candidate for the communication master as it fulfills all three requirements:

1. As it is the biggest object in the cluster its memory and computing capabilities are likely to be the most appropriate.
2. By its load sensing technology it is able to detect changes on its surface even providing additional information like weight and position.
3. It is physically constantly present. Without the table there would be no table cluster.

Thus, the table implicitly takes over the role of the communication master and sends an acknowledgement to the object indicating that it is now part of the table cluster. The message also contains information about the weight and position of the bottle. So all knowledge about the object is stored on the object themselves, being independent of any infrastructure. Furthermore, the object uses the information to infer further context about itself. For example, the filling state that can be derived from the weight information more accurately than from the pressure sensor. In contrast, the table only stores the data about the objects it could infer itself. That is weight, position, identity and type of object.

The information stored by the table also defines the queries the table is able to answer. Initially the objects do not know about the other objects on the table and do not listen to the communication between table and other objects. However, they can query the table to find objects of given type, weight and position. The table returns the corresponding information including the identity of the object. Thus the querying object can directly communicate with the objects matching the query.

The reverse process, removing objects from the table, works respectively: In most cases, i.e. when stacked objects differ in weight, the table can detect the removal of objects autonomously. However, to detect this more reliably, the table verifies its perception by checking with the according event sent by the removed object. If an appropriate event was sent, the object can be safely removed from the table’s list of objects on its surface. Finally, the table sends a message to the object indicating that it no longer belongs to the cluster.

We are currently implementing an application where empty wine glasses send queries to the table in order to find the nearest wine bottle of the same type of wine.

IV. Discussion

In this section we will discuss the implications of our concept. We first describe how our approach differs from traditional device discovery. Then we describe the implications of our choice of the master. Finally, we discuss the fault-tolerance of our cluster as we do not rely on the table as master.

A. *Event-based clustering*

Usually communication protocol stacks are structured in layers where each layer provides services to an upper layer. This enables to abstract from low level communication details like physical encoding, medium access, etc. Also flexibility is improved as lower layers do not use services of upper layers making it possible to exchange lower layers. The Rendezvous layer in Bluetooth (BT) [2] for example, is responsible for the device discovery.

In our implementation we do not provide a dedicated device discovery layer but rather pursue an event-based approach that connects the devices in a semantic way. We avoid the overhead of building a multi-layered network infrastructure such as BT does. We rather use simple broadcast communication and obtain our topology by registering the objects on a central node based on sensor events.

When the event-based device discovery cycle between object and table is finished both the object and the table have already a lot of information about the objects which would normally be exchanged in a later stage. And as the radio signal covers the physical space of interest (area around the table), our communication technology itself does not imply a certain topology of the devices in any way. Thus, we obtain a cluster including knowledge about the physical environment which could not easily be obtained by performing traditional device discovery.

B. *Choosing the master*

As explained in the introduction we choose the master on the basis of three properties that require knowledge about the objects. In a certain way Bluetooth uses such information during its inquiry procedure which implements the device discovery process. Usually the master is obvious, because Bluetooth was intended to connect peripheral devices to a central processing unit like wireless keyboards and mice to a PC or headsets to a mobile phone. However, Bluetooth revealed some major difficulties in mobile ad-hoc network-like environments [5]. Some improvements have been suggested [12].

In the described implementation we similarly chose the master implicitly. However, we provide a general framework that enables to choose the master dynamically without having the problems of fixed masters and slow device discovery.

Compared to the centralized master-slave protocol in BT, our structure resembles more a P2P network where the master is comparable to central indexing servers like in the Napster file sharing network [8]. However, as explained in the next section our approach is much more dynamic as we do not completely rely on a dedicated master.

C. Fault tolerance

Although the table takes a central role the table cluster is still fault tolerant. As each object stores completely its own context, the table cluster can be seen as a partly and temporal replicated database. Objects on the table can be *replicators* of other object's context as well as they are likely to store information about other objects in the cluster as long as they maintain a communication relationship. Other forms of replicators may include *observers*, objects that by using their sensors are able to support context inference of the observed object by sending – and possibly storing – additional sensor data or even higher level context. A camera that detects color changes of an object could be an example for such an observer.

The existence of replicators is also guided by the fact that objects may need to outsource computational tasks or store their data externally. As smart-its are generally independent of an infrastructure, objects in the environment starting in the same cluster must be found that meet computational and storage requirements of that object. Again, the physical properties, e.g. the type of the object, could be used as a parameter for an algorithm finding such objects.

At the moment the master fails, no information is lost, as every object keeps its own data. However, the table cannot be used by the objects to learn about their physical space and cluster accordingly. Furthermore, the cluster information is not available and the table cannot be used as a master any more. But the system is still able to recover. For short term failures, the table can learn the last known state from the objects and by further interaction eventually completely recover the cluster information. If the table fails permanently, the objects can take over themselves: As communication is based on a shared medium theoretically every object could take over and answer and forward requests from outside and from within the cluster. This could be done cooperatively in a true P2P way. However, this would enforce a lot of cooperation overhead. More practical is the solution of choosing once a new master that meets the three requirements in a similar way as the table:

1. A new master should provide according memory and computing power. Again, the object type can help satisfying this requirement.
2. The new master should be able to detect the relationship "new object on the table" and "object removed" on its own. A vase is a good example, because it might be possible to detect changes in the acceleration of objects on the table when other objects are put on the table.
3. Ideally the new master should be as static as the table itself so that over a longer period of time no new master must be chosen. E.g. a chair or the vase would fulfill this requirement. The criteria for choosing the objects would be their object type and their history indicating how long they have been in the proximity to the table.

This case shows that an appropriate protocol will be necessary because the new master can now not be chosen as easily. In order to satisfy the first requirement of finding a node with appropriate memory and computing power, potential algorithms will probably be similar to those used in mobile ad-hoc networks. However, our approach provides the benefit that object type parameters can be included which may help to perform the search more efficiently.

After the negotiation, the new master would try to retrieve the cluster information by querying the objects. So it is still feasible to choose a new communication master that can cluster objects according to their physical relationships. However, it is quite likely that additional information provided by the table like the weight and position cannot be used any more. The only thing the new master could do would be sending a maintenance request to get the table fixed …

V. Conclusion

We have shown an alternative way of networking devices clustering them according to their physical relationship. Instead of introducing overhead by using a dedicated device discovery layer, we link objects semantically by clustering based on their sensor-events.

We have described the ongoing implementation using a load-sensing table for clustering smart objects on its surface. We have shown that we are able to obtain P2P like fault-tolerant and scalable clusters that reflect the real-world relationships achieving better manageability in a large scaled network of smart objects.

Acknowledgment

The Smart-Its project is funded in part by the Commission of the European Union (contract IST-2000-25428). More information on the project is available at [13].

References

[1] M. Addlesee, R Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a Sentient Computing System," *IEEE Computer Magazine*, Vol. 34, No. 8, August 2001, pp. 50-56

[2] Bluetooth: http://www.bluetooth.org/

[3] A. Harter and A. Hopper, “A distributed location system for the active office,” *IEEE Network*, pp. 62-70, January/February 1994.

[4] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl and H. Gellersen, “Smart-Its friends: a technique for users to easily establish connections between smart artefacts”, *In Proceedings of Ubicomp 2001*, Springer-Verlag LNCS 2201, pp. 116-122, 2001

[5] O. Kasten and M. Langheinrich, “First experiences with Bluetooth in the Smart-Its distributed sensor network,” *Workshop on Ubiquitous Computing and Communications, In: Proc. PACT 2001*, Oct. 2001

[6] K. Van Laerhoven, A. Schmidt and H.-W. Gellersen. „Pin&Play: Networking Objects through Pins,” *In Proceedings of Ubicomp 2002*, G. Boriello and L.E. Holmquist (Eds), Lecture Notes in Computer Science; Vol. 2498, ISBN 3-540-44267-7; Springer Verlag, Göteborg, Sweden. September 2002.

[7] J. Lifton, D. Seetharam, M. Broxton, and J. Paradiso, “Pushpin Computing System Overview: a Platform for Distributed, Embedded, Ubiquitous Sensor Networks,” *Proc. of the Int. Conf. on Pervasive Computing*, Zurich (2002), pp.139-151

[8] Napster: http://www.napster.com, http://www.howstuffworks.com/napster5.htm

[9] A. Schmidt, M. Strohbach, K. Van Laerhoven, A. Friday and H.-W. Gellersen. “Context Acquisition based on Load Sensing,” *In Proceedings of Ubicomp 2002*, G. Boriello and L.E. Holmquist (Eds). Lecture Notes in Computer Science, Vol 2498, ISBN 3-540-44267-7; Springer Verlag, Göteborg, Sweden. September 2002.

[10] A. Schmidt, M. Strohbach, K. Van Laerhoven and H.-W. Gellersen, “Ubiquitous interaction - Using surfaces in everyday environments as pointing devices, ”. *In Lecture Notes in Computer Science (LNCS)*, N. Carbonell & C. Stephanidis (Eds.). Springer Verlag, 2002

[11] J. Scott, F. Hoffmann, G. Mapp, M. D. Addlesee and Andy Hopper, “Networked surfaces: a new concept in Mobile Networking, *WMCSA 2000*, December 2000, Monterey, California

[12] F. Siegemund, M. Rohs, “Rendezvous layer protocols for bluetooth-enabled smart devices,” in: *Proc. 1st International Conference on Architecture of Computing Systems - Trends in Network and Pervasive Computing, ARCS*, Hartmut Schmeck; Theo Ungerer; Lars Wolf (Eds): pp. 256-273, April 2002, Springer-Verlag

[13] Smart-its project: http://www.smart-its.org

[14] Smart-its prototyping platform: http://www.comp.lancs.ac.uk/~albrecht/smart-its/platform/