IEEE International Conference on Data Engineering

AURA: Enabling Attribute-based Spatial Search in RFID Rich Environments

Tejas.A.Bapat, K. Selçuk Candan, V. Snehith Cherukuri, Hari Sundaram

Computer Science Department, Arizona State University, Tempe, Arizona, USA
{tbapat,candan,venkata.cherukuri,hari.sundaram}@asu.edu

***Abstract*— In this paper, we introduce *AURA*, a novel framework for enriching the physical environment with information about objects and activities in order to support searches in the physical world[1]. The goal is to enable individuals to use the environment in which they function as a living (short-term) memory of their activities and of the objects with which they interact in this environment. In order to act as a memory, the physical environment must be transparently embedded with relevant information and made accessible by in-situ search mechanisms. We achieve this embedding through innovative algorithms that leverage a collection of parasitic RFID tags distributed in the environment to act as a distributed storage cloud. Information about the activities of the users and objects with which they interact are encoded and stored, in a decentralized way, on these RFID tags to support attribute-based search. A novel *auraProp* algorithm disseminates information in the environment and a complementary *auraSearch* algorithm implements spatial searches for physical objects in the environment. Parasitic RFID tags are not self-powered and thus cannot communicate among each other. *AURA* leverages human movement in the environment to propagate information: as they move in the environment, users not only leave traces (or *auras*) of their own activities, but also help further disseminate *auras* of prior activities in the same space. *AURA* relies on a novel signature based information dissemination mechanism and a randomized information erasure scheme to ensure that the extremely limited storage spaces available on the RFID tags are used effectively. The erasure scheme also helps create an *information gradient* in the physical environment, which the *auraSearch* algorithm uses to direct the user towards the object of interest.**

I. INTRODUCTION

Human beings typically live and function in environments that are shared by multiple individuals. This is true in our personal spaces, such as our homes and offices, as well as in more public environments, such as shopping malls and book stores. These environments are designed to support and, in fact encourage, synchronous social interactions (such as two friends shopping for a shoe together in a department store or a couple enjoying a candle-lit dinner in their dining room). However, asynchronous social interactions in physical environments are facilitated through secondary means, such as bulletin boards at conferences or post-it notes on refrigerator doors. While these means are somewhat effective, especially in instances when the asynchronous social interaction is expected (a conference attendee expecting a message from a colleague or a wife expecting that her husband will check the refrigerator door before leaving for work to see what to buy for dinner), they fall short when the interaction is serendipitous.

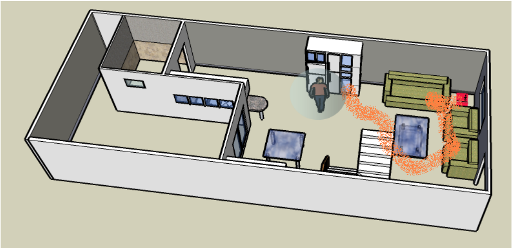


Fig. 1. Ben is looking for a book which her wife must have recently taken from the bookshelf to read. If the environment could serve as a memory of the interactions of the individuals interacting with daily objects, then Ben would not have to search the entire house for the book

Example 1.1: Figure 1 shows Ben looking for a book which his wife had mentioned to him last week. The book does not appear to be in the bookshelf, so his wife must have taken it to read. At this point, Ben has three choices: to look for all the places his wife may potentially have taken the book to read, call her at the office where she works, or pick another book to read for now. If only the environment could have highlighted for him where his wife took the book, then he would not need to worry about calling her to ask about the book or having to choose another book to read.

In this paper, we are introducing *AURA*, which supports information search in the physical world. The goal of AURA is to let the individuals to use the environment in which they function as a (short-term) memory of their activities and of the objects with which they interact. AURA achieves this by making use of RFID technology, involving small, cheap, and passive (not self-powered) radio tags which are attached to objects in the environment. In particular, AURA re-imagines the collection of RFID tags as a de-centralized, distributed storage cloud which can store information about the objects and activities within the environment. AURA is made possible by the recently developed EPC GEN 2 RFID protocol [1], which lets the so called *interrogator* devices to write user supplied data on RFID tags. While such storage-enabled RFID tags are becoming increasingly cheap and are being incorporated into everyday objects, the following limitations of these devices make their use as a distributed storage space challenging and interesting for further investigation:

- The amount of memory available for storage on each RFID tag for user data storage is exceedingly small (128

---

[1] This project is supported by the NSF grant “NSF-0725910, Design of Dense RFID Systems for Indexing in the Physical World across Space, Time, and Human Experience”

1084-4627/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDE.2009.203

bytes). This makes it essential that the encoding of the data onto the tags be done in a most efficient manner.

- The passive nature of the RFID tags (i.e., lack of transmitters on the RFID tags) makes it impossible for these devices to exchange information, without the help of an interrogator device that the users carry with them. In other words, information dissemination process needs to leverage (and thus mirror) users' activities in the environment.
- Storage is only part of the story. For the users to be able to search for objects in the environment, they need to be able to specify queries using RFID-enabled cell phones or PDAs. But since the range of these devices is limited and since the tags cannot communicate with each other without the help of an interrogator device (carried by a user) acting as intermediary, the user herself has to be part of the search process. This implies a truly interesting cost model for query processing: *the number of steps the user has to walk before locating the object of interest.*

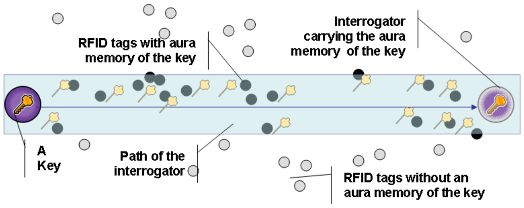
Not so surprisingly, there is limited amount of prior work on the use of RFID tags as distributed storage systems. Most existing work, such as Drishti [2] (which encodes information about nearby landmarks on RFID tags to help navigate user who are blind), assume that the information is static. Others, such as Rfind [3], store information about the environment in a central database and given a query contacts this central database to get the position of the object in the environment. Unlike these solutions, AURA is neither read-only nor assumes that there is a central database. Instead, it embeds the relevant information into the environment and disseminates information along the popular routes (with the help of those very individuals using those routes) to ensure that search queries can be answered with low walk overheads.

II. DISSEMINATION AND SEARCH

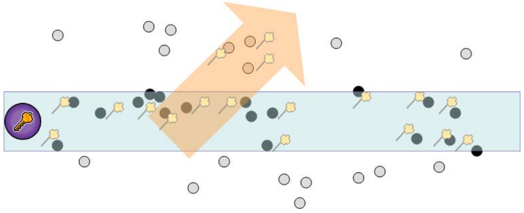
AURA uses the EPC GEN2 protocol to implement information dissemination and search algorithms. In this protocol, the interrogator device manages the tag population using three basic operations:

- *Select*: The device selects the tag population on which further operations will take place, based on a user specified criterion. At this phase, those tags that match the criterion are all activated (since RFID tags are not self-powered, being activated simply means that a specific bit on these tags is set to "1").
- *Inventory*: This operation identifies all the tags that have been activated in the *select* phase. At the end of this phase, the device knows about all the matching tags.
- *Access*: Once the tags have been identified, the device is ready to issue data management commands, *READ* and *WRITE*, to exchange data with the tags.

As the user moves in the environment, the device goes through these three phases repeatedly to discover and interact with the new tags in the vicinity. The *select* operation is very fast since it involves a single broadcast command. On the other hand, the *inventory* and *access* operations are implementation dependent and are relatively slower: inventory is on the order of 100s to 1000s of tags per second [4], whereas access operations are one the order of 1000s of bytes per reads and writes [5] [2].



(a) Interrogator-based information dissemination along the user's path



(b) Information is further disseminated when other users' paths cross the original path

Fig. 2. *AURA* disseminates information along paths used by the individuals in the environment

A. Data Model

In *AURA*, each object has a set of descriptive attribute-value pairs. For example, the book that Ben was looking for in the previous example may have the following set of attribute-value pairs: $\{type ='' \ book'', keyword ='' \ economy'', usertag ='' \ favorite''\}$. To enable searches for objects which are beyond the interrogators' immediate vicinities, AURA uses a novel *auraProp* algorithm to disseminate these attribute-value pairs in the environment.

B. auraProp: Information Dissemination

AURA piggy-backs the information dissemination task onto the interrogator devices that the users carry with themselves and leverages people's mobility in the environment to *transparently* spread information into the environment: as the user moves from one region to the other, the interrogator device that she carries, picks up information about objects in the current vicinity and spreads this information along the path the user follows (Figure 2).

Thus, each RFID tag is split into two memory banks: *local aura bank* and the *aura memory bank*. The local aura bank contains only auras of the attribute-value pairs specific to that tag (or the object on which the tag is attached). The aura memory bank, however, may contain propagated auras of objects that are physically elsewhere.

---

[2] While in this paper, we do not discuss the impact of these on information dissemination and search algorithms, these are important constraints that impact algorithm and system design.

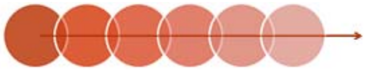


Fig. 3. Effect of erasure on information density (the number of tags that know about a particular object) along the path of dissemination

1) Signature-based Data Encoding: *auraProp* relies on a signature based encoding scheme to reduce the footprint of the attribute-value pairs on the extremely limited storage space available on the RFID tags [6]: each attribute-value pair is hashed to a unique bit string (i.e., a signature); if a tag is to store more than one attribute-value pair, all these hashed bit-strings are logically *OR*ed to obtain a combined signature. The lookup is then performed by creating a search signature for the attribute-value pairs given by the user and then matching the bits in the search signature against the bits in the data signature encoded in the given tag. If any of the search bits is not found in a given tag, then that tag can be eliminated from further consideration.

2) Local Object Signatures: As the user moves in the environment, the interrogator device reads the signatures of all the objects in the environment and creates a set of combined local object signatures to act as "*aura memories*". Each combined local object signature is created by selecting a subset of the objects randomly and merging their signatures. These local object signatures are written into the aura memory banks of the tags in the environment redundantly.

3) Propagation-Combined Object Signatures: As the user continues her movement, the interrogator repeatedly reads *local object signatures* written in the environment and combines them with previously seen signatures into the so-called *propagation-combined object signatures* which are both carried on the interrogator device and also distributed onto the RFID tags that the interrogator meets in the environment as *propagated aura memories*.

4) Random Bit-Erasure for Preventing Information Overflow due to Propagation: A naive superimposition of propagated signatures with the local signatures would soon make the combined signature unusable because most bits will eventually be set to 1. To address this, AURA uses a novel erasure scheme, which turns a number of 1's in the combined *signature* to '0'. The *erasure rate*, ϵ, determines the probability with which a '1' in a given signature will be turned to a '0'. The signatures created by this process are embedded into the local tags and also carried along with the interrogator to the next region along the walk.

Note that, if performed naively, the *erasure* scheme will degrade with high number of walks along a given path. If different walks along the same path carry different erasure-variants of the same signature, then superimposition of these variants will eventually overload the signature. Hence *auraProp* ensures that the same bits are turned from 1 to 0 for different walks by using the *tag Id* on each tag as the seed for the consistent hash function [7].

A side effect of the random bit-erasure process is that after the erasure some data on the tag will become unintelligible. However, since each signature is written redundantly and since the bit-erasure process is randomized, there will be other tags in the environment which will still have that particular data. We define *query match rate* as the number of tags which answer a given query to the total number of tags in a given region. Note that, as the distance of propagation increases (i.e., as user moves away from the original object), the probability that the 1's corresponding to this object signature are flipped to 0 increases. Thus, the number of tags that can answer to queries about this object decreases until it reaches a region where none of the tags can identify the object (Figure 3). As discussed next, AURA leverages the correlation between distance and *query match rate* to implement *auraSearch*.

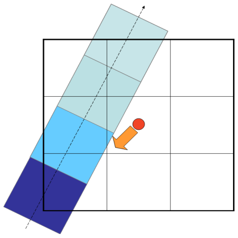


Fig. 4. A search region overlapping with an information dissemination pathway. Information gradient along the path, due to erasure-based density changes, enables search in physical space, back to the information source – the grid representation is only a simplifying approximation; in reality the regions are not grid shaped

C. *auraSearch: Information Search*

Search starts with the user specifying the attribute-value pairs for the object which she is interested in searching. *auraSearch* uses the same hashing mechanism as used in *auraProp* to generate a query signature.

1) Erasure based Information Gradient: As we mentioned earlier, the interrogator can only access tags in its current spacial vicinity. In order to help the user locate a given object in the environment (but not in the immediate vicinity), during search, *AURA* uses directional antennas at high power levels to read from the current region as well as an adjoining region in a particular direction (Figure 4). This asymmetric design enables AURA to leverage the differences in the *query match rate* to guess the direction of information propagation and to direct the user towards the source of information, essentially walking against the "erasure". We refer to this as the *information gradient* based search. In other words, *auraSearch* is exploiting the *erasure* process (which is also needed for preventing information overload) for creating reverse pointers in space that can be followed by the user (Figure 4). The search algorithm selects the direction in the ratio of tags that answer the given query is maximum and recommends the user to walk in that direction. Since the search makes use of only the *select* and *inventory* phases of GEN2, the data exchange involved is minimal and this renders the communication overhead low.

2) auraSearch Algorithm: The system picks the best direction to go to as the direction with highest match rate. In order to recover from local errors, at each step, the user also stores the other candidate walk direction in a stack with the worst

TABLE I

THE SYSTEM PARAMETERS

Symbol	Definition	Value
N	Number of tags in the region	10
r	Redundancy in storing local information	100%
L	Length of the signature stored on each tag	2000
m	Number of bits set to 1 in each object signature	4
ϵ	Probability of turning a bit from 1 to 0 (erasure rate)	0.05-0.3

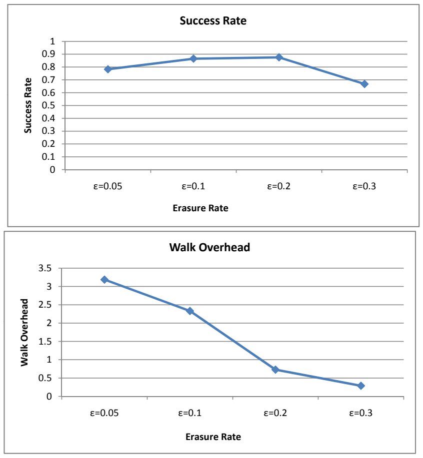


Fig. 5. Effects of erasure on *auraSearch*

possible direction to follow placed at the bottom. When the user reaches a point where none of the neighboring regions (except for the one from where he previously came) answers the query, a path termination condition is reached. On path termination, the searcher pops the next best closest region from the stack and the interrogator will guide the user back to this new location. This "depth-first" like search ends when the user has reached the region where the object is located (i.e., search terminated successfully) or when the user failed to locate the object (i.e., search terminated with failure). In case, there are multiple objects in the environment having the same attribute-value pairs, the above *information gradient*-based ensures that the user is guided to the nearest object of interest from the position where the query was originated.

III. SAMPLE RESULTS

The performance of AURA can be analyzed by considering two measures: the *success rate* and the *walk overhead*. The *success rate* is the ratio of the queries for which the user is able to find the object successfully using *auraSearch*. The *walk overhead* is defined as the ratio of the difference between the number of steps that the user walks using the *auraSearch* scheme and the optimal walk length (if she knew where the object is) to the optimal walk length.

A. Setup

The simulation results reported here are for a home based setup, which we model as a 20x20 meters space (10x10 cell grid). Each path along which the information is disseminated corresponds to a single cell in width and runs from one end of the home to the other end. The locations of the 2 horizontal and 2 vertical paths in the home are chosen randomly for different setups. Table I shows the values for the various setup parameters. The results are averages of 2000 runs.

B. Sample Results

Figure 5 shows the effect of the *erasure rate* ϵ on the success rate and the walk overhead. As can be seen in this figure, for low values of ϵ the information gradient is not well formed and thus the resulting walk overhead is high. On the other hand, for high values of ϵ since the information is forgotten too quickly, the success rates also drops. With a properly selected value of ϵ, *auraSearch* can guide the user towards the object of interest with a high success rate while also ensuring that the walk overhead is low.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have described an RFID based environment, *AURA*, which supports spatial searches for physical objects. The underlying *auraProp* and *auraSearch* algorithms provide *interrogator-based* information-dissemination and search for attribute-based access to physical objects. Using signature based techniques, *auraProp* distributes the information along the paths commonly followed by the users. An *erasure* scheme prevents overload and creates an *information gradient* that enables users to locate objects in the physical space given attribute based searches. Initial results show that AURA guides the user to the object of interest with a low walk overhead and high success rate.

One future research direction is to extend auraSearch to address the issue of tag mobility. Since tags are attached to objects which can be moved around in an environment, the memory data that was encoded in a given region may be lost because of tag mobility. Another challenge is to associate a validity period for each attribute-value pair that is encoded in the system. After the validity period has expired, the system can forget the attribute-value pair. We are working on randomized algorithms to overcome these challenges. Enabling the privacy of the users in the shared environment is also a future research direction.

REFERENCES

[1] EPCGlobal, "Specification for rfid air interface: Epc radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz 960mhz," Tech. Rep., Jan. 2005.

[2] S. Willis and S. Helal, "Rfid information grid for blind navigation and wayfinding," *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, pp. 34–37, Oct. 2005.

[3] A. Saxena, S. Ganguly, S. Bhatnagar, and R. Izmailov, "Rfind: An rfid-based system to manage virtual spaces," in *PERCOMW '07*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 382–387.

[4] Wikipedia, "Singulation — wikipedia, the free encyclopedia," 2008, [Online; accessed 2-November-2008]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Singulation&oldid=246382575

[5] "Fasttrack series rfid tags singulation." [Online]. Available: http://www.datascansystems.com/upload/Data_Sheets/RFID/lrp125-250tags_.pdf

[6] C. Faloutsos and S. Christodoulakis, "Signature files: an access method for documents and its analytical performance evaluation," *ACM Trans. Inf. Syst.*, vol. 2, no. 4, pp. 267–288, 1984.

[7] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi, "Web caching with consistent hashing," in *WWW '99: Proceedings of the eighth international conference on World Wide Web*. New York, NY, USA: Elsevier North-Holland, Inc., 1999, pp. 1203–1213.