Context Cube: Flexible and Effective Manipulation of Sensed Context Data

Lonnie Harvel[1, 2], Ling Liu[2], Gregory D. Abowd[2], Yu-Xi Lim[1],
Chris Scheibe[1,2], Chris Chatham[1,2]

[1]School of Electrical and Computer Engineering
[2]College of Computing & GVU Center
Georgia Institute of Technology
Atlanta, Georgia 30332-0280, USA
{ldh, lingliu, abowd}@cc.gatech.edu

Abstract. In an effort to support the development of context-aware applications that use archived sensor data, we introduce the concept of the Context Cube based on techniques of data warehousing and data mining. Our implementation of the Context Cube provides a system that supports a multi-dimensional model of context data and with tools for accessing, interpreting and aggregating that data by using concept relationships defined within the real context of the application. We define the Context Cube information model, demonstrate functional applications that we have developed with the system, and explore possible applications that may be developed more easily using our tool.

1 Introduction

Humans are good at making use of implicit context information when interacting with other people and the surrounding environment, whereas computers usually require the interpretation of the context information to be explicitly provided in order to use it appropriately. One of the challenges for ubiquitous computing is to bridge the gap between captured context data and the different levels of interpretation of captured context information needed by applications. Working with archives of context data provides opportunities for understanding user behavior by analysis and assessment of that behavior over time [17, 18, 19]. In this paper, we focus on accessing and manipulating an archive of sensed context data to support applications in context-aware and everyday computing, using concept hierarchies to relate context data to real-world definitions.

Currently, most context-aware applications focus on the immediate state. In contrast, there are many applications, envisioned as supportive of our everyday lives, which require more complicated context that we cannot sense directly. For example, providing a summary of a person's daily movement, determining when and how often a family gathers for meal time, assessing an elder relatives level of social interaction, or understanding space usage within a building in order to optimize HVAC resources all require processing of the lower-level context data and the interpretation or analysis of that data over time.

There are tools in the field of data warehousing and mining that we can use to manipulate low-level context data into higher-level information. However, to apply these techniques: context must be effectively modeled, structured to allow for interpretation and aggregation, and made easily available to applications and mining engines. In this paper, we describe the Context Cube, a system that supports a multi-dimensional model of context data and provides tools for accessing, interpreting and aggregating that data.

1.1 Motivating Example

Of the possible application domains that would benefit from accessing historical context, much of this work focuses on one of our larger research agendas: aiding senior adults who remain in their homes. Many adult children live far away from their aging parents. One of the biggest concerns for these children is the everyday well being of the parent. In fact, many times the decision to move a parent out of their own home and into some assisted living facility is the desire to maintain the peace of mind of the adult children. Mynatt *et al.* proposed the Digital Family Portrait (DFP) as an enhanced information appliance in the home of the adult child [1]. The DFP portrays an individual, the parent, whom we assume lives in a sensor-rich home that tracks certain critical parameters of everyday life.

Figure 1: the primary Digital Family Portrait display, the image is a substitute

The center of the portrait is a picture of the parent, and the border is a visualization of the activity of the parent over the last 28 days (see Figure 1). In the prototype proposed by Mynatt et al., the visualization consisted of butterfly icons to reflect physical activity of the parent. A small (medium, large) icon represents little (medium, much) activity for that parent for that day. The number of detected room transitions for the parent on a given day determined the size of the icon. This version of the Digital Family Portrait demonstrates a simple example of analyzing contextual data, in this case room-level position data, over some period of time to produce an indicator of trends. In this case, the visualization leaves up to the adult child any decision on whether a significant trend has emerged. More automated techniques might automatically signal when one day's movement trend is out of the ordinary.

The Digital Family Portrait serves as a useful motivation for storing, manipulating and visualizing a history of contextual data. There are many other examples, particularly in the case of monitoring the well being of an aging individual, in which trends of contextual data, specifically where an individual is and what they are doing,

are important indicators of potential problems. Once we have gone to the trouble of instrumenting a space to detect human behaviors, we need to provide better ways to access and manipulate large collections of data. This paper proposes a technique for simplifying the development of this important class of context-monitoring applications.

1.2 Overview

We introduce the concept of the Context Cube Model in Section 2. Section 2.1 describes the structure for modeling the relationships between context data and with associated context derived from world knowledge. Section 2.2 describes how we use datacube operations with this model, and section 2.3 describes our actual implementation of this system. In Section 3 we present four applications that have been built on context cubes. The first application is a new version of the Digital Family Portrait that is working on live data from an instrumented home. The second application shows the co-location history of two individuals and demonstrates how new applications can share existing context cubes and associated context. The third application is a visualization of activity levels in the different areas of a home. It is also built from the same context cube structures as the previous two, but focuses on the location aspect instead of identity. Unlike the previous applications, the fourth example is a hypothetical design, given in greater detail, to demonstrate how a set of context cubes and associated context would be developed. It also shows how existing context cubes, in this case the ones from the previous application examples, can be augmented with new cubes to provide for new applications. We give a brief overview of the related work in the context-aware computing and database communities in Section 4, and our concluding discussion in Section 5.

2. Bridging the Context Divide - The Context Cube Model

As the amount of stored context grows, we are reaching the point where we are data rich, but information poor. There is growing effort in the context-aware community to extract interesting knowledge and information (rules, regularities, constraints, patterns) from large collections of context data. Without this level of analysis, it will be hard to build context-aware applications that need sophisticated context knowledge and historical context information. Some forms of this data analysis are difficult, even impossible, to compute with standard SQL constructs. In particular, aggregations over computed categories, roll-up and drill-down of aggregation hierarchies, and symmetric aggregation require a more powerful data representation. To meet this need, we have adapted a technique from data warehousing, namely the data cube, and provided a system for creating and managing the resulting context cubes. In this section, we present a brief overview of how this adaptation applies to context-aware applications. For more detail on data cubes in general, we refer the reader to the work of Gray et al. [2].

2.1 Context Dimensions and the Context Cube

Our idea behind using a warehouse is to represent relationships within the actual structure of the data. The benefits are threefold: easier representation and processing of queries [3], the inclusion of expanded definitions and relationships, and the creation of new context constructed from analysis of the existing data. We represent the relationships between context using dimensions. As an example, take location, a single dimension of context. Many applications, most notably tour guide systems, have been built with a simple knowledge of immediate location. However, other applications, like reminder services, need both location and time. We now have a two dimensional relationship as shown in Figure 2. If there is a need, as in the Digital Family Portrait, to perform an analysis or calculation over the data contained within this relationship, a materialized representation may simplify queries and improve performance. After adding a third dimension, identity, we are able to represent the three dimensional relationship (location, time, identity) as well as the two dimensional relationships of (location, time), (location identity), and (time, identity).

Thus, a *Context Cube* is a model for representing and manipulating multi-dimensional context, based on the principles of data cubes from the traditional database and data mining community [2, 3]. Each dimension of the cube represents a context dimension of interest. A relation of N attributes produces an N-dimensional

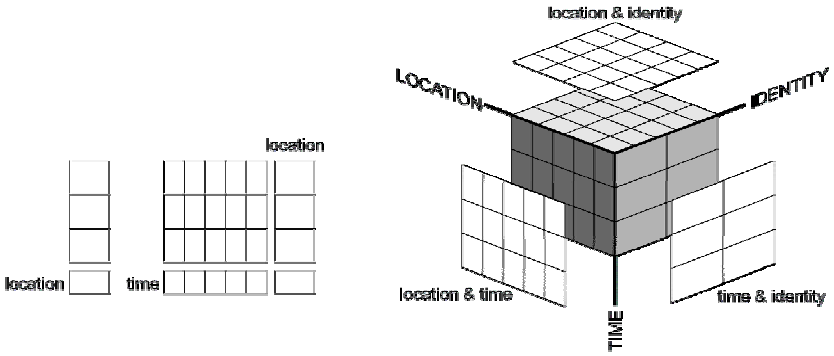


Figure 2: Examples of data cubes. From left to right, we depict an array of locations, a matrix showing all the possible relationships between location and time, and finally a cube showing all the possible relationships between location, time and identity

context cube. To structure these context relationships and capture the quantitative measures of these relationships, we construct a *context fact table*. This table contains both the factual context data on each of the dimensions, and the *derived context* that are of importance to the domain analysts and are obtained by analysis and summarization over the captured context relationships. The derived context may be viewed as residing in the boxes of the context cube. We access the derived context by using the dimensions as filter conditions. Figure 3 shows the design of a context fact table as a star schema. The fact table represents the context that we will gather to form

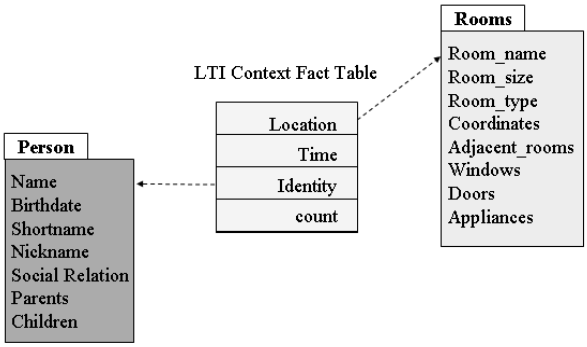


Figure 3: An example star schema showing a context fact table for a Location, Time, Identity cube with context attribute tables for Person associated with the Identity dimension and Rooms associated with the Location Dimension.

the dimensions (Location, Time, Identity) as well as the derived context "count", a measure of room transitions.

The star schema also shows the connection of the context fact table to the *context attribute tables*. The context attribute tables allow for an expanded definition of an individual point of context data. They can be used by domain specialists who are interested in using higher levels of context to interpret and reason about the captured context fact data. One advantage of the star schema approach to context modeling is the reusability of context knowledge both within a domain and across domains. In our example design, used to form a simple context cube to support the Digital Family Portrait, we connect the Location dimension to the Rooms context attribute table, and the Identity dimension to the Person context attribute table. We can use the context attribute tables to provide alternative interpretations and associated information. Another important purpose of the context attribute tables is to enable the use of the *context dimension hierarchy* to introduce different levels of abstractions to the captured context data and context relationships.

A context dimension hierarchy is a concept hierarchy on the given context dimension. It is a mechanism that enables higher-level abstractions and context interpretations to the captured context data. Context dimension hierarchies allow us to manipulate a cube, along one of the context dimensions, using a combination of the data in the context fact table and the associated context in the context attribute table. Figure 4 shows two possible hierarchies constructed from the Location dimension and the Rooms attributes. In each of the examples, the leaf nodes are room names from the [removed for anonymity]. The first hierarchy classifies the locations by room type. The second hierarchy classifies the locations by region. If we only wanted information about Bedrooms, we could use the first hierarchy to make that selection, if however, we wanted information about Bedrooms on Floor 1, we would need to use both hierarchies to make the selection.

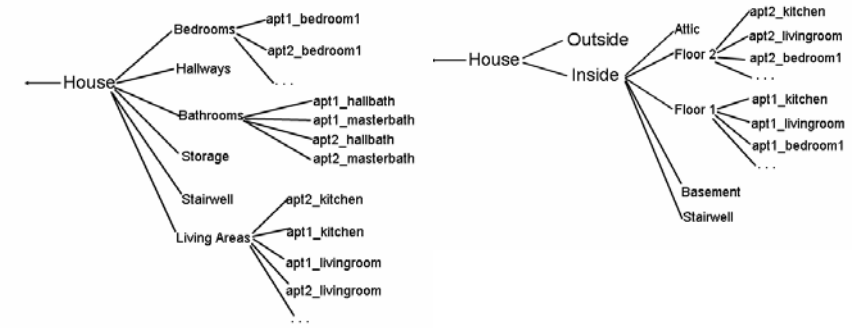


Figure 4: two context dimension hierarchies, the first depicts a classification based on room types and the second is based on regions

2.2 Context Cube Operations

We have shown how context cubes can be used for answering context queries at different levels of context abstraction. In this section, we present the four context operators used to manipulate the context cube: *slice, dice, roll-up,* and *drill-down.* These operators enrich the context cube model to support a much broader collection of context queries with efficient implementation methods. As evidenced in Gray's et al. seminal work on data cubes, cubes and cube operators provide an easier and more effective way to analyze multidimensional data sets. For example, using calls to cubes, a few lines of code can accomplish what would otherwise require a convoluted query with dozens of lines [2]. The result of a cube operation is another cube allowing us to perform operations in combination. To provide an intuitive understanding of these four cube operators, we provide an informal description of each operation with some illustrative examples.

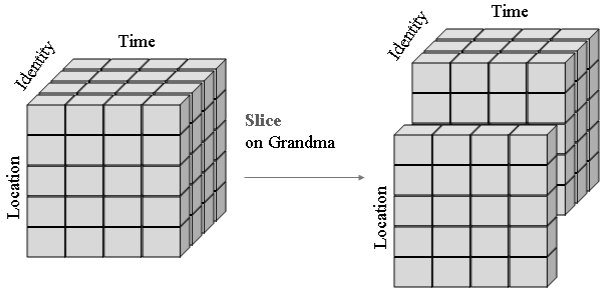


Figure 5: A slice on Grandma returns Location and Days information about Grandma alone with count showing the number of times Grandma was in a given location on a given day

A *slice* is a selection on one dimension of an N dimension cube. For example, in the LTI (Location, Time, Identity) cube, used in the Digital Family Portrait, a slice could be on one user, or a group of users, selecting all conditions from the other two dimensions. We can express a slice using a representation from the context fact table, the context attribute table, or from the context dimension hierarchy. Figure 5 shows a slice on a single user, Grandma, over the three-dimensional context cube LTI. We can view a slice as an operator that cuts the cube by slicing it over the identity of Grandma.

The *dice* operator is a selection to all dimensions of an N dimension cube. Again using the LTI cube, we can dice for Grandma, in the Kitchen, Today. A dice is an easy way of doing a sequence of slices, Figure 6.

Roll-up generates a new cube by applying an aggregate function on one dimension. The roll-up operator uses the structure defined in a context dimension hierarchy and information in a context attribute table to determine how the roll-up should be performed. The values of the derived context are updated, when necessary, to reflect the change in resolution or scale of the filter conditions applied to the dimension.

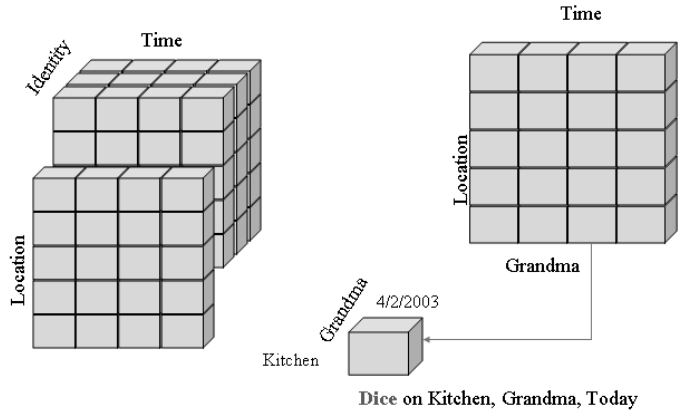


Figure 6: A dice on Kitchen, Grandma, Today results in a cube that matches only those filter conditions with a count showing the number of times Grandma was in the Kitchen on that day

Again using the LTI context cube, a roll-up on Identity might generate a cube that uses the type of Social Relationship such as Family, Friend or Business instead of individual names. Another roll-up on identity might generate a cube that uses age group or faculty/grads/undergrads as the identity dimension unit instead of person names. A rollup on Location might generate a cube that uses Room Type as the scale instead of Room Name.

Drilldown is the reverse operation of rollup. It generates a context cube with finer granularity on one of the n dimensions. A drilldown on Years generates a cube with time in Months. Drilldown on the location region Bedroom would generate a cube with location in each of the bedrooms in the house. Drilldown on identity in a senior age group (suppose it is age from 65 above) would generate a cube with seniors above 65.

Another operation over a cube, though not used for manipulations, is the ability to construct context dimension aggregate lattices. Given a context dimension table and a

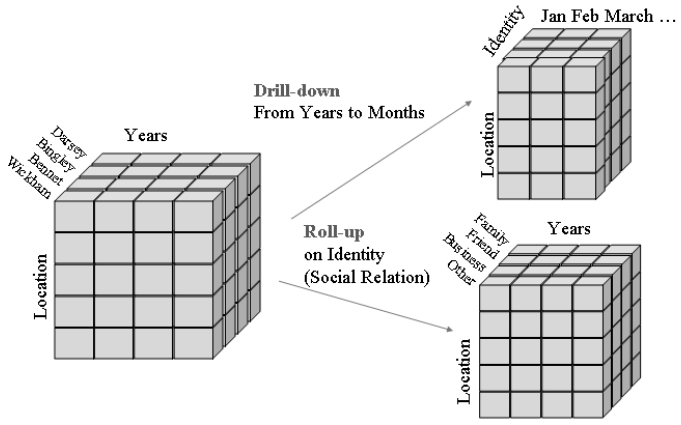


Figure 7: A roll-up on Identity based on Social Relation, a drill-down on time going from Years to Months

corresponding cube, one can generate the aggregate lattice that contains all possible combinations of aggregates one can apply to the context dimension table data. It would require a great deal of space to store a set of context cubes that represented the whole context dimension aggregate lattice for a given cube and set of context dimension hierarchies. One strategy for deciding what to construct and maintain is to use a greedy algorithm that retains the context cubes most commonly referenced. For our system, we have designed a registration-based system that allows applications to inform the Context Cube system of its data needs and an appropriate context cube will then be generated and maintained. The details of the implementation follow.

2.3 The Context Cube System

The Context Cube System (CCS) implements the context cube model and provides management and interface functionality. The first prototype of the CCS consists of a context warehouse, a context cube constructor, and context cube manipulation operations. The CCS is a package of Java applications constructed over a MySQL database running on a Solaris server. There are currently over 1,000,000 context data points stored in the context warehouse.

At the heart of the CCS is the Cube Manager. A Cube Manager creates a context cube, maintains the cube, handles data requests and cube operations. Within the Cube Manager, a class call sets up a data store and accesses the specified tables and fields to get the raw data from the context warehouse. Also in the Cube Manager, you can specify at what base scale/resolution you want the system to construct the context cube. We cannot perform a drill-down operation on a cube below the base granularity

of the data. Specifying DAY as the default scale sets the base data resolution to Day along the Time dimension. A similar definition can be made on each of the dimensions of a context cube as long as the context dimension hierarchy information exists. Once you bring in the raw data, you call the `Cube` class to process the raw data into a new cube. There are I/O classes to write out a cube, as a context fact table, to the database and for accessing an existing cube. The `Cube` class also provides the structure for N-dimensional arrays. A Cube Manager only requires a small amount of programming; the CCS does the rest of the work.

The average size of a Cube Manager for our current context cubes is about 25 lines of code. In future work, we hope to provide a GUI interface for constructing Cube Managers as well as providing for a construction language to allow for application control. Application interaction with a context cube is done either with ODBC/JDBC calls or through a PHP and XML interface. We use the latter method to provide data to our Flash based interactive displays, like the Digital Family Portrait.

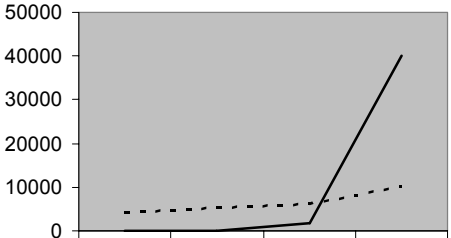
Above the Cube Manager is the Cube Register. This maintains the information about all the context cubes in the system. It knows how to construct the cubes, where the cube data originates, and the status of the cubes. The Cube Register also serves to maintain cubes that update periodically (like every night) as well as launching, or re-launching, Cube Managers as needed. The Cube Register contains the data table dependency information needed by dynamic cubes that update based on *context triggers*. A context trigger signals a Cube Manager that an event has occurred in the context environment that necessitates an update to its context cube. We constructed our context triggers as Context Toolkit widgets [8]. The Cube Register also maintains a qualitative description of the cubes.

From this registry, application developers can determine what kinds of cubes and context dimensions are already available. Information on available context dimension hierarchies is kept in the Dimension Register. The Dimension Register functions like the Cube Register, but knows about available context dimension hierarchies in the system.

The time to construct a cube is related primarily to the size of the cube and secondly to the amount of data in the data source tables. However, as Figure 8 shows, the time it takes to compute a cube grows slower than the size of the cube. Making the initial connection to the database is a fundamental time cost for all cubes. In designing dynamic cubes updated by context triggers, we must consider the time necessary to construct the cube. In most cases, only small cubes are appropriate. Though larger cubes can be set to update with context triggers, it will probably be more viable to have them update at regular, predetermined intervals.

3. Building Applications using the Context Cube System

In this section, we provide four examples of the context cube in use. The first three examples demonstrate how the cube supports data related, query-based context applications by performing cube operations on a single context cube. The final example is hypothetical and shows a more complex use of context dimension hierarchies, and demonstrates a technique for aggregating cubes.



	Month	Day	Hour	Millisec
cube size (rows)	42	112	1800	39970
construction time (millisec)	4012	5121	6171	10181

Figure 8: Construction time of context cubes, in milliseconds, compared to the size of the cube. Month, Day, Hour, and Millisec refer to the base resolution of the data in the Time dimension. All cubes are for 28 days of data.

3.1 Supporting the Digital Family Portrait

The Digital Family Portrait, described earlier, provides a visual representation of an elderly individual's daily activity level. It also provides information on temperature and other environmental data. Direct reporting of the single-dimensional context can provide the environmental data. The LTI cube provides rapid querying and basic analysis of the multi-dimensional context data.

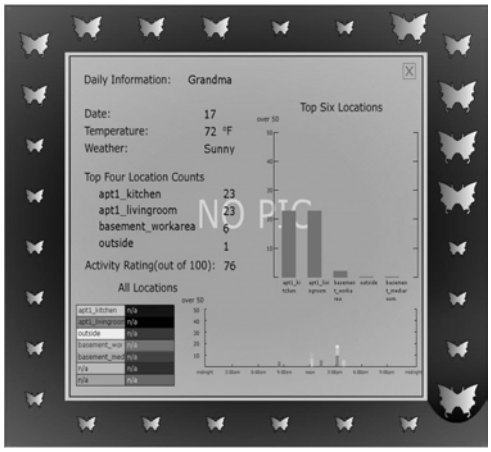


Figure 9: a drill-down on an individual day

In the cube operations shown earlier in Figure 6, the first slice is on Grandma, interpreted by the Person context attribute table. This creates a Grandma cube containing Time and Location data. The slice along the Time dimension is a simple date, provided by the application. This can be constructed by a single dice operation. Currently, the Digital Family Portrait defines activity as moving from hall-to-room or room-to-room. The context dimension

hierarchy associated with the Location dimension provides the hall and room generalization. Using this information, the correct position changes can be easily derived and stored as the value of the remaining cell. This calculation is stored as the derived context "count", and the context cube maintains the value.

Instead of slice on the time dimension as the second step, we could have diced the cube with a range of dates to include the past 28 days. The result would be a cube that holds the defined activity value for those days. This is the information needed to support the DFP, so it will be able to request that information via a simple query. This is an example of how we can glean information from basic position data, by using warehousing techniques and the Context Cube.

Other manipulations of the LTI cube would produce the information for an example of family gatherings. In that case, we would need a context dimension hierarchy that encoded information about relationships so that we could distinguish between family and other visitors to the home. We would use the same raw data, but this application would request a different set of operations to create the context cube it needed. It could also use the same context dimension hierarchies as those created to support the DFP.

3.2 Visualizing Co-location

This application is similar to the DFP but instead of tracking the activity level of one individual, it shows a representation of two people's co-location history. The figure shows a drill-down on a single day. Colored sections in the schedule show the individual's presence in the home. Times of co-location are highlighted in the space between the schedules. The distances between the figures in the border reflect the level of co-location for that day. Days with only one person's data have one figure; days with no data for either individual are grayed out.

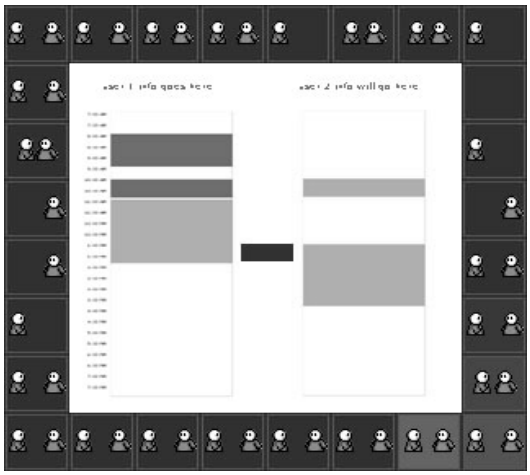


Figure 10: A drilldown in the co-location application.

This is again derived from the LTI cube. We perform a slice on two individuals instead of just one. Next, we perform a roll-up on the Time dimension to provide Location information at the MINUTE resolution. Finally, we need to perform a roll-up on Location. In our original implementation, we provided information at the room level. However, the results

Fi

were counterintuitive. In this Home, the kitchen and living room are open and adjacent spaces. Two individuals, located in the kitchen and living room, would be able to see and talk with each other easily, but our system would not acknowledge that as co-located. We had to construct a new interpretation of the Location roll-up to account for this inaccuracy in our results.

3.3 Space Activity

We have also built this application over the LTI cube. However, instead of focusing on the Identity dimension, the purpose of this application is to represent activity in a specific location over time. The Location data is at the base room level and we have done a roll-up on Identity to ALL, a way to represent a total aggregation of that dimension. The application shows a visualization of the activity levels in individual rooms for selected time periods.

Future versions of the application will allow users to manipulate Location and Identity as well as time. The purpose is to show how easy it is to manipulate interpretations and generate aggregates with context cubes.

3.4 Tracking Social Interaction

Has Grandma been getting enough social interaction lately? In this hypothetical example, we are looking at the level of interaction for an elderly individual living alone. This is an example of the kind of qualitative information that can be gleaned from context data using the context cube. The power of the cube is to provide

organization and structure to a growing archive of information. By classifying and grouping data points, based on concepts derived from the real world, we make it easier for applications to analyze that data quickly.

The first step is to determine what constitutes a social interaction. For the purpose of this example, we will use a somewhat simplistic definition: a social interaction is an email, incoming or outgoing phone call, or a personal visit, from a friend or a family member. To track this, we will need to pull from three separate data sources: an email log, phone log, as well as the identity of individuals in the home from the LTI context cube. The email log would look something like the excerpt in Table 1.

Subject	Sender	Date	Size
breakfast?	ivxlcdm@aol.com	9:42 PM	2k
Grow more zinnias	ntwt@buyme.net	8:34 AM	24k
heard from Zoe	phwerty@blah.com	11:43 PM Saturday	3k
missed you	curio@carlosm.org	8:22 PM Saturday	1k

Table 1: part of an email log showing likely fields

The phone log would have a similar structure. There are several ways of capturing phone call information. Many local phone companies now will provide an online service; there are standalone devices that interface with caller-id systems as well as computer based systems. Instead of size, the phone log provides the duration of the call (Table 2).

Number	Caller	Time	Duration
(404)xxx-1234	Elizabeth Bennet	9:42 PM	43 min
(404)xxx-5678	Fitzwilliam Darcy	8:34 AM	6 min
(770)xxx-9123	Charles Bingley	11:43 PM Saturday	22 min
1-800-xxx-7734	Out of Area	8:22 PM Saturday	1 min

Table 2: part of a phone log showing likely fields

The existing LTI context cube can provide information about people located in the home. Assuming that we would possess identity information for every visitor, email, or phone call is unreasonable. Instead, we assume that appropriate identifying information will be available for those individuals that are socially close to the target subject.

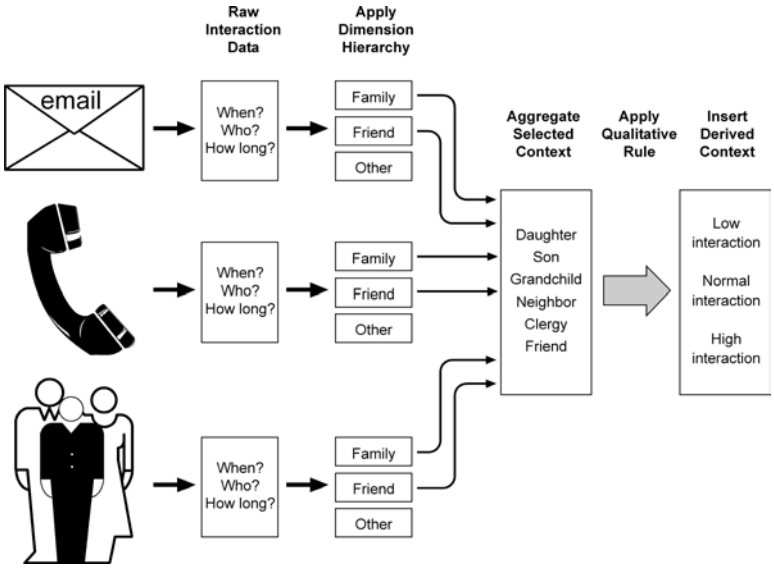


Figure 12: Creating a social interaction cube from raw interaction data.

The next step is to process or clean the data: email addresses, phone numbers, and caller Ids need to be interpreted as names. For each of the interaction types, we will create a cube with two dimensions (Identity, Initiation Time) and one derived context, the duration of the interaction. For visits and phone calls, the duration is a measure of time, for email the duration is a measure of size. We then perform a roll-up on each of the cubes using an Interaction context dimension hierarchy that classifies they type of interactions based on Identity. This is best done in conjunction with the Person context attribute table shown in Figure 2.

In the next step, we perform a slice on Social on each of the three context cubes and aggregate the results into a new context cube. This cube now contains only the interactions that we are interested in to track social interaction. If we perform a roll-up on the Initiation Time dimension to DAY, we will have a listing of social interactions by day. The data is now ready for analysis. A simple clustering algorithm can be used to classify a set of daily interactions as low, normal, or high. We could also use other systems, dedicated to trending analysis, to construct the qualitative rule, such as the Coordinates system at Microsoft [17]. Once we have a classification to use as a qualitative rule, we can apply that rule to each day and add a new derived context, Social Interaction Level, to the cube.

4. Related Work

This works lies at an intersection between the research domains of context-aware computing and that of data warehousing and data mining. In the area of context-aware

computing, our work relates to the area of context infrastructures. Context cubes are not an independent infrastructure, instead they provide a data warehousing component needed to leverage the historical context. As referenced earlier, context cubes are an adaptation of data cubes from the data warehousing community. In addition to these, the context dimension hierarchies are related to work done with ontologies [1] and inference engines like Cyc [4] as well as lattices used in machine learning [5, 6]

4.1 Context-Aware Computing

In many cases, developers create *ad hoc* solutions for individual context-aware applications. However, as exploration in the field of context-aware computing has broadened, a desire to move away from single application development has emerged. The works of Schilit [7], Dey [8], and Brown [9], all provide a framework for developing multiple applications using shared context resources. The Context Fabric is a more recent infrastructure approach from Hong and Landay [10]. The Context Toolkit [8], Context Fabric, and the Event Heap [11] all provide the ability to store context data. The Context Fabric extends those capabilities by providing a logical context data model, separate from the physical storage model, which they represent using four concepts: entities, attributes, relationships, and aggregates. Hong et. al. created a prototype application, the Communication Information Agent (CIA) [12] that uses stored document access information to predict future document needs. This application is similar to the Remembrance Agent [13] that also recommends documents based on past and current context.

In using an infrastructure framework, middleware transports, manipulates, and manages context data, acquired from the context sources and on behalf of the applications receiving the information. Aggregation, higher-level interpretation, synthesis and fusion, and analysis are all activities performed by middleware applications. Middleware that processes lower-level context are relatively new in context computing. The location management work in QoSDREAM [14] from Cambridge that models and combines location information from multiple heterogeneous sources is an examples of an infrastructure that support immediate context applications. At this time, there are few models or systems that support applications that need context beyond the current state or that require sophisticated data manipulation to produce the necessary information. Of those that exist, the Context Cube is closest in philosophy to the "database" approach of the Context Fabric model. That infrastructure, however, strives for a homogenous data model for capturing, storing and accessing context data. The Context Cube model is more of a "data warehouse" model, expecting context data to have a heterogeneous representation. We designed the Context Cube to be an extension of the previous architectures, agnostic as to the underlying infrastructure used to capture data. Theoretically, a Context Cube system could function simultaneously over all of the current context infrastructures that store context data. At present, the Context Cube is using data provided by the Context Toolkit and from several UPNP devices.

The structured data model of the Context Cube also supports trending and other forms of data analysis. The previously referenced Coordinates system from Microsoft [17, 19] and the work of Hudson et. al. [18] are examples of systems that use analysis in order to interpret or predict human behavior.

4.2 Datawarehousing

Inmon describes a data warehouse as a "subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions." [15] Traditional databases are designed to be transactional (relational, object-oriented, network, or hierarchical). In contrast, data warehouses are mainly intended for decision-support applications. Traditional databases support *on-line transaction processing* (OLTP), which includes insertion, updates, and deletions. They generally function over a small sub-set of the available data, and that data is usually small, with historic data moved to external archives. Data warehouses, on the other hand, generally contain large amounts of data from multiple sources and they are inherently multidimensional. They are optimized to support on-line analytical process (OLAP), decision support systems (DSS), and data mining operations that need large collections of data. This may include data constructed under different data models and acquired from a collection of external, independent sources [16]. Since context data is also multi-dimensional, with few strong relations among data within context archives, (time being one of those), and since the access to the system will be similar to DSS and OLAP style interactions, storing the archived data as a *context warehouse* is a natural choice.

Unlike traditional data, context data is unidirectional with respect to time. As a result, context storage systems do not require update or delete functions. In time, we will need a "correction" function to replace inaccurate context information, but such a function would still preserve the forward advance of time. This means that context data, once captured, is static and only incoming context is effected by changes in the data environment. This allows us to adapt traditionally static techniques into dynamic ones.

In constructing a context warehouse, the raw data will be stored in *context tables*. Our context tables are constructed directly by the widgets of the Context Toolkit, using the store functions provided by default, and also from UPNP devices in the Aware Home environment. A context table may be a single source of context data or an integration of the related data from the context environment.

5. Conclusions

There is a gap between the context that is being provided by sensors and that needed by more sophisticated, context-aware applications. Adapting data warehousing techniques to create a context warehouse and the Context Cube is one step in bridging that gap. By using the raw data stored in the context tables, referencing that data through information provided by the context dimension hierarchies and with the cube operations, we are able to provide information about individuals (like Grandma) or collections of people (like the family), provide summary information, perform aggregations based on relationships among the context, determine collocation over some period of time, or provide information suitable for higher-level analysis.

Storage space is an important issue when dealing with cube technologies. There are three major implementation alternatives when implementing a cube [3]. Physically materializing the whole cube takes the most space but provides the fastest response to queries. The storage space consumed by a cube has a direct impact on the time necessary to construct a cube. At the other end of the spectrum, a representation may be generated without materializing any of the cubes. In this case, every cell is computed from raw data on request. It requires no extra space, but has extremely slow query response time. The final method materializes a small subset of cells in a cube and then generates the remaining "dependent" cells on request. Dependent cells are either calculated or approximated based on the values of the materialized cells. In our initial prototype, we chose to materialize the entire cube in order to optimize query response time, and to simplify implementation. However, we will need to consider more efficient cube materialization as the amount of data grows. A full LTI cube over all 300,000+ data points takes several minutes to calculate, 1GB of system memory, and produces a materialized representation of over 500,000 lines.

A future goal of our work is to use data mining and knowledge discovery techniques to generate new context through synthesis and analysis. The system developed by Elite Care [20] to determine quality of sleep through an instrumented bed is an example of work that requires context analysis. Though aggregating and interpreting context data can provide a lot of information, more context may be acquired by appropriately analyzing the stored data. For example, though the simple measure of activity used in the current implementation of the Digital Family Portrait is effective, it is not very informative. The system represents a day in which Grandma moves repeatedly from the bedroom to the bathroom and back in the same way as a more normal day, which includes the same number of room changes. If the DFP had more qualitative information available, like the social interaction example, it could in turn provide more information to the family. We can produce this kind of information through analysis of the stored context. In the case of Grandma's movement patterns, a day in which she only moved between two rooms would be identified as different from a day in which her movements covered several rooms. Having the ability to recognize significant changes in physical behavior provides important information for a variety of applications. For our initial work, we are using k-clustering to divide a collection of context data into groups that have strong similarity. Other clustering techniques as well as associative techniques will expand the nature and sophistication of the generated context.

Acknowledgements: This work is sponsored in part by the Aware Home Research Initiative, the Digital Media Lab at Georgia Tech, and the National Science Foundation (grant 0121661). Our thanks to Kris Nagel for her efforts in securing our IRB approval for gathering and storing the context data and to Thomas O'Connell and his team for supporting the Aware Home Living Laboratory. IRB Protocol Title: Aware Home Context-Aware Services Experiment, PI: Abowd, Gregory D., Protocol Number: H02047

6. References

1 E. D. Mynatt, J. Rowan, A. Jacobs, S. Craighill, *Digital Family Portrait: Supporting peace of mind for extended family members.* In *CHI2001*. 2001, Seattle, WA.

2 J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, H. Pirahesh, *Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals*, In *Data Mining and Knowledge Discovery,* 1, 29-53, 1997, Kluwer Academic Publishers, The Netherlands

3 V. Harinarayan, A. Rajaraman, D. Ullman, *Implementing data cubes efficiently,* in *ACM SIGMOD International Conference on Management of Data.* 1996

4 D. B. Lenant, *Cyc: A Large-Scale Investment in Knowledge Infrastructure,* In *Communications of the ACM,* 1995, 38(11)

5 R. Poell, *Notion Systems,* http://www.notionsystem.com

6 S. Newcomb, M. Biezunski, *Topic Maps for the Web,* IEEE Multimedia, 2001.

7 B. Schilit, *System architecture for context-aware computing,* Ph.D. Thesis, 1995, Columbia, New York

8 A. K. Dey, G. D. Abowd, D. Salber, *A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications,* In *HCI Journal,* 2001, 16(2-4), pp. 97-166

9 P. J. Brown, J. D. Bovey, X. Chen, *Context-aware applications: From the laboratory to the marketplace,* In *IEEE Personal Communications,* 1997, 4(5), pp. 58-64

10 J. I. Hong, J. A. Landay, *An Infrastructure Approach to Context-Aware Computing*, In *Human-Computer Interaction,* 2001, Vol. 16,

11 B. Johanson, A. Fox, *The Event Heap: A Coordination Infrastructure for Interactive Workspaces,* In *Fourth IEEE Workshop o Mobile Computing Systems and Applications (WMCSA 02),* Callicoon, New York, June 2002

12 J. I. Hong, J. A. Landay, *A Context/Communication Information Agent,* In *Personal and Ubiquitous Computing: Special Issue on Situated Interaction and Context-Aware Computing,* 2001, 5(1), pp. 78-81

13 B. Rhodes, T. Starner, *Remembrance Agent: A continuously running automated information retrieval system,* In *Proceedings of Practical Applications of Intelligent Agents and Multi-Agent Tech (PAAM)*, 1996, London, UK

14 H. Naguib, G. Coulouris, *Location Information Management,* In *UBICOMP 2001,* Atlanta, GA, 2001

15 W. H. Inmon, *Building the Data Warehouse*, 1992, Wiley

16 R. Elmasari, S. B. Navathe, *Data Warehousing and Data Mining,* In *Fundamentals of Database Systems,* 2000, Addison-Wesley, pp. 841-872

17 E. Horvitz, P. Koch, C. Kadie, A. Jacobs, "Coordinate: Probabilistic Forecasting of Presence and Availability", Proccedings of the 2002 Conference on Uncertainty and Artificial Intelligence, July 2002, AAAI Press, pp. 224-233

18 S. Hudson, J. Fogarty, C. Atkeson, J. Forlizzi, S. Kiesler, J. Lee, J. Yang, "Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study", *Proceedings of CHI 2003*, ACM Press, 2003

19 N. Oliver, E. Horvitz, A. Garg. "Layered representations for human activity recognition." In Fourth IEEE Int. Conf. on Multimodal Interfaces, pages 3--8, 2002

20 Elite Care, http://www.elite-care.com

21 Y. Ding, S. Foo, "Ontology research and development. Part 1: A review of ontology generation", Journal of Information Sciences, 2002 28 (2)123-136