Object Lessons Learned from a Distributed System for Remote Building Monitoring and Operation *

Frank Olken **Hans-Arno Jacobsen**† **Chuck McParland** **Mary Ann Piette** **Mary F. Anderson**

Lawrence Berkeley National Laboratory
Berkeley, CA 94720

Abstract

In this paper we describe our experiences with the design, the deployment, and the initial operation of a distributed system for the remote monitoring and operation of multiple heterogeneous commercial buildings across the Internet from a single control center. Such systems can significantly reduce building energy usage.

Our system is distinguished by its ability to interface to multiple heterogeneous legacy building Energy Management Control Systems (EMCSs), its use of the Common Object Request Broker Architecture (CORBA) standard communication protocols for the former task, development of a standardized naming system for monitoring points in buildings, the use of a relational DBMS to store and process time series data, automatic time and unit conversion, and a scripted time series visualization system.

We describe our design choices and our experiences in development and operation. We note requirements for future distributed systems software for interoperability of heterogeneous real–time data acquisition and control systems.

*This work was supported by the Office of Energy Research, Office of Computational and Technology Sciences, Mathematical, Information, and Computational Sciences Division. of the U.S. Department of Energy under Contract DE-AC03-76SF00098. Contact: Frank Olken, Lawrence Berkeley National Laboratory, Mailstop 50B-3238, 1 Cyclotron Road, Berkeley, CA 94720. Tel: (510) 486-5891, Fax: (510) 486-4004, Email: olken@lbl.gov

†Current affiliation is Humboldt University, Berlin, Germany, Email: jacobsen@wiwi.hu-berlin.de

1 Introduction

Buildings consume one-third of all energy in the United States at a cost of $200 billion per year, with $85 billion per year for commercial buildings. A large amount, perhaps half of this energy is wasted compared to what is cost-effectively achievable. Much of this waste is related to our inability to optimally control and maintain today's complex building systems. Recent building performance case studies suggests that typical savings of about 15%, and as much as 40% of annual energy use can be gained by compiling, analyzing, and acting upon energy end-use data. The largest source of savings arises from turning off equipment when not needed. Other sources of savings include better equipment scheduling and optimizaton of temperature set point controls. [17], [7]. For an example of the building energy efficiency analyses which detailed building monitoring enables see the paper by Piette, et al. [28].

The applications under development for the Remote Building Monitoring and Operations (RBMO) project are based on utilizing the capabilities of, and expanding upon information available through legacy Energy Management Control Systems (EMCSs) which control the HVAC (heating ventilating and air conditioning equipment) in commercial buildings.

Several studies have explored EMCS monitoring capabilities (Heinemeier, Akbari, and Tseng). EMCSs for commercial buildings are readily available. There are over 150 EMCS manufacturers [14]. About 50% of all buildings over 45,000 meter^2 (500,000 sq. ft.) have an EMCS, and they are present in nearly all large office buildings [12]. For buildings built since 1992, almost 50% of the floor area is in buildings with EMCSs.

This project is developing software systems to support remote monitoring and control of multiple buildings, i.e., HVAC, lighting, etc., across the Internet, using CORBA - Common Object Request Broker Architecture protocols. It is intended to work with heterogeneous EMCS and HVAC systems

We are also developing a remote building monitoring and control center which provides data visualization, database management, building energy simulation, and energy usage analysis tools (cf. Section 4 for a more detailed presentation of the application.) The remote monitoring facility is comprised of two components:

- A data acquisition and database management server which is built on a Unix system.
- Several analysis (Unix) workstations, which are used for data visualization, statistical analysis of the data, and simulation.

The ultimate users of the system will be owner/operators of multiple buildings e.g., U.S. GSA (federal buildings), school districts, universities, retail chains, banks, property management firms, ESCOs, utilities, et al.

The research project has three primary goals: demonstration of technical feasibility of the proposed design, evaluation and experimentation with component software technology and their interoperability across heterogeneous environments, and demonstration of the utility of the system.

Close attention and careful analysis of the data from building EMCS's typically affords many opportunities to improve building operations, resulting in improvements to occupant comfort and reductions in energy use.

However, for most buildings, such close attention and analysis are not undertaken due to lack of appropriate software, operator training, staff, and/or inaccurate/malfunctioning sensors. Remote building monitoring offers the possibility to reduce the labor costs involved in monitoring/analyzing EMCS's by spreading such labor costs over a large number of buildings. This will also make it economically feasible to employ expert HVAC engineers.

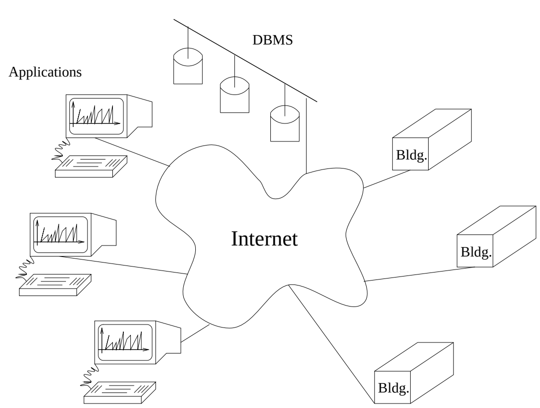

Figure 1: The RBMO system setup.

Use of the Internet, as underlying communication infrastructure, has two aspects: use of the Internet communications protocols, and use of the actual Internet network.

Use of Internet *networking protocols* permits access to a wide variety of affordable interoperable hardware and software from many different vendors over many different media. Internet markets are huge, in the millions of users, hence vendors can readily amortize development costs. In contrast, proprietary protocols typically entail more expense, because of lack of competition and because development costs are being amortized over smaller markets. Industry specific protocols, such as BACnet, fall in between these two extremes. Internet protocols are clearly the most popular for wide area networking applications.

Use of the Internet *network* permits users to share the costs of a common communications infrastructure. When building automation applications can share existing Internet infrastructure, substantial savings compared to telephone dial-up systems are possible [16]. The Internet also offers the prospect of inexpensive access to high bandwidths when needed - e.g., for remote video surveillance for security, fire, or remote diagnostics. Finally, the Internet offers a reliable, redundantly connected, backbone. The downside of using the Internet is the necessity of more stringent security precautions, discussed further below.

2 Background

2.1 Energy Management and Control Systems

Today's EMCS have been built primarily for control. EMCSs are special-purpose computerized control systems, programmed to operate building equipment such as chillers, fans, pumps, dampers, valves, motors, and lighting systems. EMCSs are installed for many different purposes such as: controlling equipment, alerting operators to possible equipment malfunction, maintaining comfortable conditions, and permitting modification of control specifications.

EMCSs typically monitor building operational data such as damper positions, set points, state variables of the working fluids, and equipment status. They are not generally used to monitor electrical demand. In some cases, more accurate sensors may be needed for diagnostics than are typically included in an EMCS. For example, to track chiller efficiency, a very accurate measurement of the temperature drop across and water flow rate through the chiller is needed. The sensors in a typical EMCS may not be sufficient for this calculation. We have added additional flow sensors and better temperature sensors to the cooling plant at Soda Hall (but not at Oakland Federal Building) where we are conducting some of our research.

2.2 Building Data Model Standards

ASHRAE in 1995 approved a new communications standard, for building automation systems – The Building Automation and Control Network (BACnet) Standard [1], [3]. This standard views the HVAC (Heating, Ventilating and Air Conditioning) system as a collection of objects. However, the standard does not (yet) include standard applications–level objects such as chillers or cooling towers. We expect that BACnet will shortly be used to connect building gateways to EMCSs.

The International Alliance for Interoperability [19], [6] is a consortium of architectural/ engineering/ construction software vendors (e.g., Autodesk and Bentley Systems) who are working on developing a standard data model for description of buildings. The data model is intended to be used by architectural and engineering design tools, construction cost estimation software, construction scheduling software, etc. We are tracking these efforts for use in our naming conventions for monitoring points.

2.3 The Common Object Request Broker Architecture

The Common Object Request Broker Architecture (CORBA) is a standard for distributed computing which has been developed by the Object Management Group (OMG) [27], a consortium of independent companies. CORBA aims at providing a uniform communication infrastructure for building distributed applications. It supplies unifying mechanisms for interoperating software components, operating on various hardware platforms, and running under different operating systems implemented in different programming languages.

Interoperability is gained by specifying all component interfaces in a universal interface definition language (IDL). IDL is a declarative 'lingua–franca' for specifying interfaces, following a C++–like syntax, [22].

3 RBMO system design

In this section we describe the architecture of the RBMO system in further detail, motivate our primary design decisions, and describe the following components of the system:

- Gateway systems in each building
- Data acquisition sub–system
- Applications–level object specifications and unit conversion
- Time series database

3.1 System architecture

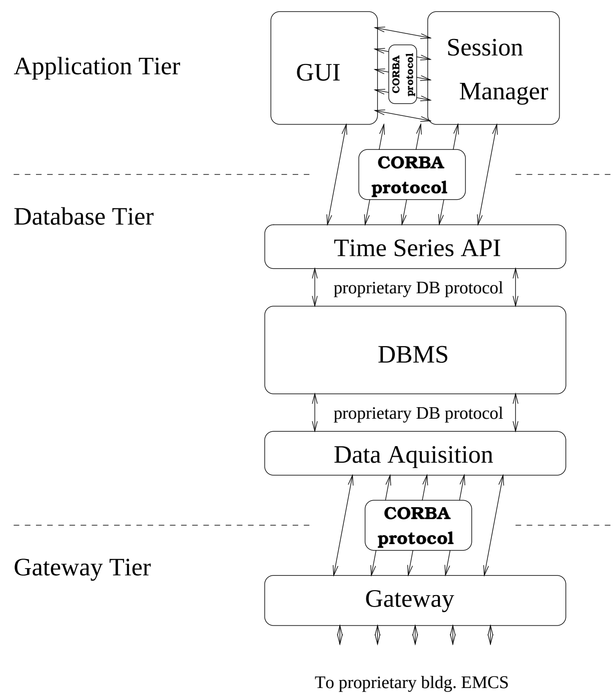

Figure 2: RBMO system design

The RBMO system architecture is shown in Figure 2. It constitutes a three tier architecture with the individual components being: the applications, the database management system, and the building gateway system. The individual tiers are entire sub–systems inter–connected through the Internet via CORBA adaptors.

The system has been designed in a modular manner to ease the evaluation of alternative technologies for database, user interfaces and visualization components (cf. Section 3.5). The following mysubsection will describe the individual components and their design in further detail.

The RBMO system constitutes a highly heterogeneous system incorporating several different hardware platforms and operating systems, ranging from extremely proprietary systems (building EMCSs) to common platforms (Sun workstations/UNIX and PC/NT). This large heterogeneity was one primary motivation for using CORBA instead alternate protocols, such as OLE/Active X/DCOM, SNMP, MMS, and netDDE. protocols. For a more detailed exposition see either our earlier papers [24], [25] or our web site [23].

For additional information on MMS see [4] (including a bibliography) and [9], or [13] for utility applications. For information on SNMP see [8], [33], [20], [32], or [35]. For information on netDDE see [34].

3.2 Gateway

The system architecture requires that each participating building have an internet point-of-presence, also referred to as a building ”gateway”. These building gateways are addressed in the usual internet fashion (e.g. northTowerBuilding.myCompany.com). Within the gateway itself, further addressing of building-specific objects (more below) is accomplished by a CORBA-compliant object request broker (ORB). The gateway ORB acts as a mediator between incoming CORBA object references and the actual objects resident in the gateway process. Currently, building gateways are Pentium-class PCs running the Windows NT operating system.

One of our goals was to present a network view of a building's monitoring and control data that was consistent across multiple buildings - regardless of EMCS system differences. In order to provide this homogeneous view at the building gateway level, it was necessary to have building-to-building (i.e. EMCS to EMCS) differences resolved within the gateway process itself. This has led to a three part gateway design consisting of an EMCS adaptation layer, an internal building data point database, and an external CORBA interface layer.

The building data point database portion of the gateway contains entries for all building monitoring and control data points accessible through the building's EMCS. Each entry contains attributes for units, point name, and description, as well as methods for reading and modifying values. Since this database has to mirror that of the EMCS itself, methods have been developed for deciphering and re-creating data point lists for each vendor-specific EMCS encountered. Specific attention has been given to mapping data point units information from vendor or installation-specific descriptions to standard representations. Standard units representations are required for automatic unit system transformation facilities located further downstream. Since the logical place to resolve any mapping discrepancies is at the building level, these issues are dealt with by the internal gateway database.

The EMCS adaptation layer is responsible for interacting with the building EMCS via the specific communications formats and protocols required by the EMCS vendor. In some cases, e.g., the interfacing to the Barrington EMCS in Soda Hall, this involves use of proprietary message protocols using standard serial communications media. In others, e.g., interfacing to the Johnson Controls EMCS in the Oakland Federal Building, the netDDE protocol over LAN-based media is used. In all cases, specific message formatting and handling algorithms are encapsulated in methods associated with each element of the internal data points database (see above) and vendor-specific differences are not visible above the EMCS adaptation layer.

The external CORBA interface layer provides a common layer of C++ objects that are accessible to the gateway's ORB. These objects have full access to the internal data points database and are able, through appropriate overloaded methods, to query the EMCS system for current monitor and set point values. Since these objects execute as part of the gateway's ORB process and not as a part of the gateway program itself, synchronization mutexes are provided to assure safe and exclusive access to building data object methods and attributes.

3.3 Data Acquisition System

Our telemetry data on the building state are initially available as cross–sectional data, i.e., we sample the entire building state periodically, typically at 1–2 minute intervals. Such high resolution data permits us to study oscillations in the control system. However, typical analyses use only a few state variables, but will typically want to look at long time series of each such state variable. Thus it is expedient for analysis purposes to have the stored data in a separate time series for each state variable.

The question then arises as to when and where the data are to be transposed from cross–sectional to time series storage formats. This can be done at collection time, analysis time, or some intermediate time.

Transposition at collection time slows the collection process, but speeds the analysis phase. Transposition at analysis time permits very fast data collection, but slows the analysis phase. Transposition at intermediate times permits both very fast data collection and fast analysis. However, the software architecture becomes more complex and costly to implement and maintain. Transposition can also occur either at the gateway machines, the database server, or the analysis workstations.

We chose to transpose the data at the database server as it is collected. This will generate at least one disk I/O operation per data point monitored for each sampling period. This appears to be the simplest architecture to implement, and should have adequate performance for our prototype system.

The data acquisition system (DAQ) presently is a polling system residing on the central server together with the time series DBMS. Placing the DAQ on the TS DB server is useful for performance reasons as the DAQ does a lot of I/O to the DB server.

Polling (pull architecture) was chosen initially over a push architecture from the gateways due to perceived simplicity of implementation. We anticipate moving to a push architecture, which we expect will offer better reliability against single point failures of the central system (assuming it has been replicated), easier support for replicated or partitioned central DAQ/DBMS systems.

Our data acquisition system is multi-threaded, with one DAQ process serving multiple buildings. This offers performance (and software licensing) advantages over a single threaded, one process per building design, at the expense of reliability. Any failure in the DAQ process now causes data acquisition from all buildings to fail. Replication or partitioning of the DAQ would be desirable for large scale applications.

3.4 Measurement Units and Time Zones

In dealing with multiple buildings we were forced to confront the diversity of the measurement units employed for various monitoring points and EMCSs.

There are 3 approaches to dealing with this issue:

1. record all measurements as pairs (value, units),
2. record the units for each monitoring point as part of the metadata for the monitoring point,
3. record all measurements in canonical set of units.

Conventional practice in EMCS systems is method 2. The IAI data model adopts method 1. However, the variation in measurement units across monitoring points and EMCSs render comparisons and computations in the DBMS extremely awkward. We have adopted the convention of storing all measurements in standard SI (metric) units in the DBMS (the conversion is done by the data acquisition system). This expedites the construction of indices and comparisons or computations in query processing. We plan to permit users to specify a preferred systems of units for display of query results.

Similar difficulties arise with time zones. There are two problems - a collection of monitored buildings may span multiple time zones (e.g., a large retail chain), and the use of daylight savings time introduces anomalies. We address both problems by converting all timestamps to UTC (Universal Coordinated Time (a.k.a. Greenwich Mean Time (GMT)). This is done in the data acquisition system prior to storing data.

3.5 Time Series Database

The bulk of the data we are collecting will be time series, e.g., of temperatures, power usage, etc. Although the raw data may be irregular time series, we will transform these to regular time series for analysis. Regular time series assume periodic temporal sampling. Hence, our most important criterion for selection of the database management system (DBMS) is its suitability for time series data and queries.

It is possible to support time series data on any of several DBMS: relational, object-oriented (OO), or hybrid object-relational. Use of relational DBMS for regular time-series data typically offers mediocre performance, weak support for operations such as smoothing and calendars. Object-oriented DBMSs permit the implementation of time-series data collections with better support for calendars, smoothing, etc. To minimize development costs we initially chose an early object-relational DBMS, Illustra (since merged with Informix) with built-in time series support. When this DBMS proved unreliable we switched to a relational DBMS, Oracle. See discussion below. Recently, relational DBMS vendors (e.g., Oracle) have announced built-in time series support.

Most of the data which are not time series are building description data - i.e., CAD (Computer Aided Design) data describing the physical configuration of the building, HVAC system, and the EMCS system. Object-oriented DBMSs (OODBMSs) are generally much better suited to CAD data than relational DBMSs (RDBMSs).

CORBA interfaces, for at least some of the commercial CORBA implementations, are available for several object-oriented DBMSs, but not yet for most of the relational or object-relational DBMSs. However, Oracle has announced it intends to support CORBA interfaces. Third party CORBA interfaces are available for relational DBMSs [18].

Support for client-server DBMS configurations is typically available for all of the DBMSs. This will allow us to move the analysis and visualization codes to the PC-workstations while maintaining the database on a robust database server.

3.6 System partitioning

We have introduced CORBA interfaces at several points in the system. This partitioning serves two key purposes:

- distribution of components
- decoupling for reengineering

The CORBA protocols between both the time series database, the graphical user interface, and the session management facility (cf. Figure 2) permit us to vary the distribution of these components depending on the communications bandwidth and client machine performance. Moreover, they allow us to better decouple the individual components for reengineering purposes, e.g., Internet browser integration of GUI.

4 The RBMO applications software

The RBMO applications are designed around providing the following three capabilities:

- Archiving historical time series data in a database

- Providing visualization means for building energy performance analysis
- Performing a series of regressions and statistical analysis techniques to (1) define “baseline” conditions and (2) evaluate energy performance after a retrofit, operations and maintenance changes, occupancy changes, or for historical tracking.

The archiving and visualization applications presently run on Sun workstations in C++. We envision, however, that a future deployment of the visualization and statistical analysis applications will primarily be PC-based, running Java under Windows NT or a web browser.

The whole–building and cooling plant monitoring at Soda Hall consists of a few dozen sensor points (out of 1500 total sensor points in the building) that cover whole–building electricity use, weather variables, plus chiller power and cooling tower status, and several temperatures and water flow rates. We added several power and flow meters, and temperature sensors to the cooling plant to compare them with the original EMCS sensors and augment information that was not available from the EMCS. At the Oakland Federal Building we collect 120 monitoring points concerned with the chiller plant, from a total of 8,000 points monitored by the EMCS.

We are developing a standard set of performance tracking plots to allow users to view archived data with a user–friendly menu. An example of the central control menu of the application is depicted in Figure 3, the time series visualization capabilities showing some of the tracking plots are also shown. This includes daily (24-hour) load shapes of 5-minute operating data for all values; weekday, weekend, and weekly load shapes (hourly data); and monthly average load shapes. The chiller and cooling plant analysis will consist of plots such as power versus cooling load (kW versus tons) and efficiency versus load (kW/ton versus tons).

The applications are intended for use by a remote expert. Most of the focus on the graphics is on energy use. However, the capability exists for the user to look at detailed data such as time series of cooling plant water temperatures or cooling tower fan start–stop intervals. The concept is to provide a tool for a remote user who might be responsible for tracking the performance of building systems at multiple locations. The menu of standard graphs are designed to allow the user to easily view the most critical, macro–level performance data, while providing additional means for more detailed analysis of micro–level data. In other words, the scripted graphs should allow the user to answer the question, “is the building and cooling plant operating well?”

To keep the application extensible we developed a scripting language which drives the visualization and GUI software. Certain extensions and additions may, however, either be performed online, or by preparing a script. The scripting language developed is simple, we therefore omit a detailed discussion. In retrospect, we should have adopted a conventional scripting language.

For further information on applications see our earlier papers [24], [26].

5 Discussion

This section presents a detailed account of our experiences in the design, development, and operation of a prototype of the RBMO system.

5.1 Interoperability platform concerns

5.1.1 Event notification capabilities

Central to any remote monitoring and control system is some mechanism for the remote sites to notify the monitoring sites of “interesting” events. Such event notifications need to be asynchronous, persistent, and (often) multicast. Ideally, events should be typed (so that certain processes can listen for specific types of events).

Event notification services address the need to propagate notification of significant occurrences to relevant system components. Depending on system functional requirements, this notification may be considered unreliable (e.g. periodic monitoring value updates) or may be required to be reliable (e.g. equipment status changes). In a distributed environment where system components can fail or become unreachable, the technical requirements for implementing these services can be substantial.

Although low level event delivery protocols are defined in CORBA, the semantics of describing and specifying higher level event service behavior remains (thus far) vendor-specific. Event service reliability and delivery persistence capabilities are not yet consistently defined and implemented. It is therefore important to identify and describe a significant subset of high level event service functionality. These event service functions should be sufficiently well described to allow multiple implementations that exhibit interoperable behavior. Some of these issues are being addressed in recent OMG proposals on improved Event Notification Services.

5.1.2 Security and privacy

The principal security requirement for RBMO is to preclude unauthorized persons from altering the programming of the HVAC equipment. Aside from disrupting the activities of building occupants, it is possible to damage HVAC equipment with inappropriate (or malicious) commands. Hence, some scheme for secure authentication and access control is required. The absence of such facilities in CORBA implementations at the beginning of the project caused us to defer

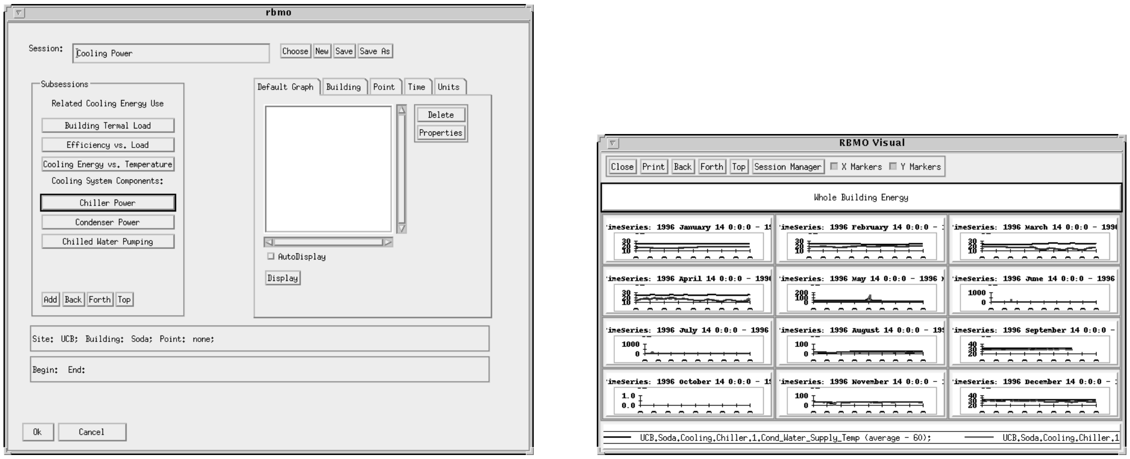

Figure 3: Graphical user interface and time series plots of RBMO application.

control applications.

In addition, some sites have indicated a desire for privacy of HVAC utilization data (e.g., occupancy data), and hence asked for encryption of all communications.

One of our major concerns has been the interoperability of the security services. We have been unable as yet to identify interoperable CORBA security services software. Furthermore, we would have liked to be able to construct interoperable software composed of different services (e.g., event services, security services, naming services, et cetera) from different vendors. We were unable to do so.

5.1.3 Performance

Our experience with CORBA was that the RPC facilities were too slow to permit the modeling of individual measurement points as separate objects. It proved necessary to aggregate many of these measurement points together into named collections (called “scan lists”) and to query the groups of points. In database terms these constituted pre-compiled queries. This dramatically reduced the number of CORBA invocations (messages) and improved performance. Such aggregation of messages appears to be a commonplace strategy for improving the performance of distributed systems. However, it can be tedious and error prone to implement and maintain. It forces implementors to mix logical and physical design issues. While faster ORBs would certainly help, we believe that network latency in wide area networks will ultimately constratin performance of fine grained monitoring systems. Many distributed systems have found it efficient to aggregate messages.

Automatic (or convenient semi-automatic) tools for aggregating individual monitoring points into larger scan objects are very desirable. Such tools are also needed to support multicast publish/subscribe communication protocols. Such tools would permit a clean separation of logical and physical design considerations in the design and implementation of distributed systems.

For latency and throughput measurements of several CORBA ORB implementations (albeit over much higher speed networks) see the work of Schmidt, et al. in [15] and [29]. Brose [2] also reports recent performance numbers across a high speed local area network, indicating that commercial ORBs require about 2 milliseconds for a roundtrip no-op. Note that many remote building monitoring applications require operation across wide area networks (with longer latency and lower bandwith than local area networks).

Part of our performance problems arose from a desire to permit run-time binding of the schema of the monitoring points. Such late binding was perceived as necessary to accommodate changes in the HVAC configuration (e.g., due to maintenance, rennovation, etc.) without requiring recompilation of the RBMO code. We could have used CORBA’s dynamic invocation capabilities - however these were perceived as too slow for large scale RBMO performance requirements.

Finally, in this application and other related applications such as electric power grid monitoring there is often a need to deal with a great many objects/points being monitored. It is our understanding that current ORBs maintain in-memory object tables (several tens of bytes per object) for all objects in the system. Such an approach has obvious problems when scaling to very large collections of objects. It is our view that some sort of automatic migration to secondary storage (and related pointer swizzling) similar to that in object-oriented databases will be needed to properly support very large applications.

5.2 Legacy system integration issues

In the integration of heterogeneous building EMCS into a seamless global distributed system we have made the following observations.

Wrapping an legacy system with objects can hide essential system features that are critical to both robustness and operation. Even if one is able to hide most of the system's distributed nature by use of distributed object technologies, the underlying system may still consist of seperate components that were designed as stand–alone systems with no expectation of co–operating with, or being controlled from, outside processes.

Centralized input and maintenance of configuration and description data is critical. This feature was mostly discovered by its absence. Although EMCS configuration information may be naturally maintained by that sub-system, these systems are designed as single, stand–alone entities. While it may be possible to create automatic mechanisms for incorporating configuration change information from one component (e.g., EMCS) into other systems (e.g., building simulation), there are typically no mechanisms to effectively signal participating components that such changes have been made. The result is that, although the collection of individual systems into a single larger one has been accomplished via distributed OO techniques (see above), the new, ”larger” system depends on a higher, heuristically–driven mechanism (e.g. system manager) for both co-ordination and correctness of operation.

The ability to seamlessly interact with objects within this new ”larger” system can lead one to believe that the underlying components are co–operating in a manner that yields synchronized, correct operation. Since each of the respective components (EMCS, automated data loggers, etc.) are typically designed as stand–alone systems, they often lack any mechanism for synchronization or exclusive control path lockout. Therefore, although it may be possible for our system to have a component that tracks chiller performance, factors in near–term weather effects, and changes cooling plant operating setpoints, there is no way to ensure that such a component has exclusive control of all relevent system components. A control layer can be inserted into the architecture; but, it's connection to all components may prove problematic.

We believe that part of these issues may be addressed by having stronger support for event specification, event notification, and a system design based on this. We envision a lose coupling of stand–alone components sending and receving events. For a more complete discussion on event service support see our discussion above.

5.3 Semantic heterogeneity and data management

5.3.1 Naming and directory services

Effective configuration description and control is the bane of distributed systems. Distributed systems are constructed, in part, by binding together objects distributed over many platforms. Without a mechanism for describing these bindings, the resulting system lacks flexibility and it's complexity remains hidden to all but the original implementors. Naming services provide a way to bind meaningful human names to object instances.

These problems take on greater importance in real-time control environments where systems must accommodate equipment replacement and/or upgrade cycles as well as ”live” system reconfiguration. Stable and ubiquitous naming and directory services are a key tool in the design of systems that will accommodate these changes. Examples include the CORBA naming services and LDAP Directory services.

The function of these services goes well beyond the nuts and bolts issues of publishing specific object names and attributes. An effective distributed naming and directory service can expose, through the use of multiple hierarchical views, objects aggregated by function, location, control subsystem, etc. In addition, these services can provide a single, ubiquitous repository for object meta information. In the case of control systems, this can include information about sensors and instrumentation used in generating object method values, resolution of object name aliases due to differing local and supervisory object namespaces, and provide automated mechanisms for locating physical and/or logical control system components.

5.3.2 Mediation Services

While standards efforts offer some hope in the long run for reconciling semantic heterogeneity among various network devices and controllers (HVAC or electric power grid), the existence of many incompatible legacy systems requires that some facilities for mapping between the global schema and local schemas be provided. At present, e.g., in RBMO (and other systems) this is often done in an ad hoc fashion with hand-crafted mapping tables. Such facilities are variously referred to as mediators (partial mappings, late binding) or schema integration tools (nearly complete mappings, early binding). Such tools will require access to metadata (e.g. DB or object schemas). Examples of related work include OMG Object Interface Repository, and Meta Object Facilities.

5.3.3 Static vs. dynamic schemas

Traditional database designs specify complete detailed schemas statically. Such static schemas facilitate systematic design, query optimization, etc. However, for many monitoring and control applications (shipped as shrink wrap software), it is desirable to be able to alter the schema (e.g., to accommodate new types of equipment) without rebuilding the DB or recompiling the code. Examples of such approaches include: frame-based knowledge representation systems, and the dynamic invocation interface of CORBA, together with interface repositories. Because such dynamic schema capabilities are typically significantly slower than static interfaces, some groups have built hybrid schemas - examples include the International Alliance for Interoperability (building data model) which provide both static schemas (e.g., for geometry) and dynamic schemas (frames) for extensibility. Conceivably, on-the-fly compilation techniques might provide a reasonable performance.

5.3.4 Units conversion services

We found it necessary to provide measurement unit conversion capabilities for our RBMO project.

Unit conversions have been a chronic source of errors in data acquisition/management systems. The ubiquity of these issues (in both industrial measurement and commercial trading) suggests their provision as standard services. Ideally, one would like an extensible type systems which incorporated dimensionality and automatic units conversions. However, impure unit conversions (e.g., between mass and volume) are commonplace and difficult to deal with systematically.

Our choice was to store all measurements in canonical units in the DBMS. While this facilitated indexing and querying, it made it very difficult to diagnose errors in the data acquisition system.

5.3.5 Time Zones and Synchronization

For large scale monitoring applications we believe that recording all times in UTC (Universal Coordinated Time) is essential. Conversion routines to local time zones, including daylight savings time are essential.

Some protocol for synchronizing the clocks is needed for monitoring systems. Such protocols, e.g., NTP [5], have been developed For other distributed applications. We note that synchronization requirements for monitoring of electric power grids are sufficiently stringent (to measure phase differences among geographically distributed generators) as to lead to the use of GPS timing signals for synchronization.

5.3.6 Time series DBMS

Our first implementation used the early release of the Illustra DBMS product, chosen because it offered facilities for storing and querying time series data, including calendars. We encountered reliability problems with the DB server. We therefore switched to storing our time series data in a more mature conventional relational DBMS, i.e., Oracle Version 7.3 (at present). This has solved our server reliability problems.

Whereas Illustra provided direct support for storing and querying time series data, in Oracle we have had to construct these facilities atop a relational system. We have not (yet) replicated the calendar facilities, etc. Note that Oracle has announced time series facilities similar to Illustra/Informix.

Whereas Illustra storage usage was asymptotically 4 bytes/data value (for real numbers), Oracle is presently consuming approximately 25 bytes/data value, because each time series data value has become a full tuple in the DBMS.

For additional research papers concerned with time series database management systems and data models see the works [10], [31], [30], [11].

5.3.7 Data Quality Assurance

We greatly underestimated the difficulty and effort required to assure proper data quality. We have encoountered a variety of problems with respect to data quality - some due to errors in units conversion, some due to missing or erroneous documentation of system configuration files, some due to overflow problems with small (16 bit) counters.

We should have paid much more attention to the development of filtering software in the data acquisition system. The presence of dirty data in the time series database has been a barrier to the use of conventional database query facilities for aggregation. Dirty data is a ubiquitous problem in all sorts of data warehousing applications. Obviously, the preferred approach is to clean up the data at the source. If this is not feasbile, concise, powerful methods are needed to specify constraints on data validity.

5.4 System operation

A prototype of the system was operational for over one year. Data was acquired from two remote sites, Soda Hall and Oakland Federal Building. The central data storage and management site is located at Lawrence Berkeley National Laboratory.

We have collected approximately one year of data from Soda Hall (plus one year of historical data). We have at present approximately 2 months of data from Oakland Federal Building. We are presently collecting several dozen monitoring points for both Soda Hall and Oakland Federal Building. Sampling rates for both buildings are once per minute. Basic data rate is presently approximately 1 KB per minute.

Soda Hall

Soda Hall, a 109,000 ft^2 Computer Science building was selected as our case-study site for several reasons. The building was equipped with an Energy Management Control System that the research team had worked with before. Second, the building was located near LBNL, permitting easy access. Finally, the building has two, 220 ton screw chillers, for a total of 440 tons of cooling plant capacity (or 248 sqft/ton). Early in the project we decided to focus on chillers, which are the largest single energy-using component in buildings with central cooling plants. The building has served as the subject of several research tasks that involved commissioning and performance.

Results from this work are reported in [28], and [21] A total of 46 points were monitored for about two years, starting in September, 1995 during the first few months of inital occupancy.

Oakland Federal Building

A total of 107 points were monitored at the Oakland Federal Building. (Note, not all of these points are sampled, many are computationally derived from sampled points.) This building is a 1.1 M sqft twin-tower, 18-story office building complex. The building was constructed in 1989 and houses Federal agencies such as the IRS, US District Courts, the US DOE, and Department of Veteran Affairs. The building is served by a central plant with five chillers, three at 1000 tons and two at 450 tons totaling 3900 tons of capacity. This is 280 sqft/ton.

The 107 points cover systems similar to those at Soda Hall, including whole-building power, chiller power and tons, plus cooling tower, and pump data. Additional points include heating loop flow, a series of outside temperatures from 5 air handlers, and several internal zone temperatures.

One of the interesting questions we are examining at both sites is chiller capacities and over sizing. The operators report that the last 1000 tons of cooling is not needed, even in the warmest weather.

6 Related work

An extensive discussion of related work can be found in our earlier paper, [24].

Honeywell recently announced a new building monitoring system, Atrium, (Honeywell, 1998) quite similar to our project. Atrium employs a different operating system and window system (Windows NT), different relational database management system (SQLserver). However, it has a very similar overall architecture. They have also chosen to use a (popular) proprietary distributed object management system, Active X/DCOM, in contrast to our use of the standard CORBA distributed object system. Atrium also employs an standard building automation communications protocol, BACnet, to communicate with the building EMCS systems. BACnet is a recently deployed standard. BACnet equipped EMCSs were not available to us when we developed RBMO.

Other major differences between RBMO and Atrium lie in the time series database. Honeywell has chosen to only store changed data values (discarded repeated measures within dead bands). This results in a smaller, but irregular time series. Honeywell has also implemented a simple time series query language and temporal join operator to query and extract align multiple (irregular) time series.

The Enflex System from CTI Ltd. (CTI, 1998) employs proprietary applications protocols atop TCP/IP.

Finally, readers should be aware that remote building management presents many of the same issues addressed in the management of communications networks. Network management systems have employed SNMP and RMON as the principal (standard) communications protocols. See discussion above under alternative protocols and citations there.

7 Future work

We plan to investigate the use of the CORBA Security Service and Secure IIOP for remote control the HVAC operations, e.g., authentication.

Remote monitoring of large heterogeneous collections of buildings poses serious problems of software distribution to the buildings. We plan to study static and dynamic software distribution over the Internet onto heterogeneous target platforms.

8 Conclusions

We have described the design and development of the remote building monitoring and operation system, which employs the CORBA protocol across the Internet for: communications between remote buildings and a timeseries database, and to distribute individual *application* components *locally* across different machines. We have discussed design tradeoffs, our development and operational experiences, and related work.

We have argued for the use of CORBA within our project context. We have discussed the design of a distributed system that has to monitor events across large distances, across multiple time zones, heterogeneous sources and diverse measurement units. The development of the time series database atop different commercial DBMSs has been described.

We found it necessary (for performance reasons) to aggregate individual monitoring points into larger objects for

CORBA-based retrieval. Such aggregation is typical of many distributed applications.

None of the commercial DBMS available at the time of the project proved entirely satisfactory for storing and retrieving time series. Recently announced products may remedy this.

The most serious issues were dealing with the heterogeniety of the various building EMCSs, i.e., units and naming, and problems with dirty and missing data. Developers of distributed monitoring systems for legacy systems would do well to address such issues thoroughly early in their projects.

This CORBA-based approach is both feasible and competitive with other systems based on proprietary communications protocols. The attraction of our approach lies in the extensive availability of commercial software products which support distributed computing applications based on the CORBA model and underlying Internet communications protocols.

Acknowledgements

Our collaborators include: Kristin Heinemeier (Honeywell), Sarah Shirazi, Romeo Cruz, and James Patterson (UC Berkeley Facilities Management), Giles Adi and William Curran (Barrington Systems), Sue Williams, Diana Williams, and Ed Grey (at Oakland Federal Bldg). Prof. Carlo Sequin (UCB) arranged access to Soda Hall. Eleanor Lee (LBNL) assisted us in gaining access to Oakland Federal Building.

Vladimir Bezjanac (LBNL) has kept us abreast of developments with the IAI data model. Ed Wong wrote the time series DB code. Brian Smith wrote the first version of the graphics code. Alex Gitelman wrote the second version of the graphics application. Brian O'Clair worked on gateway code. David Gould wrote the unit conversion code. Graham Carter and Kris Kinney worked on data analysis and monitoring point selection.

For further information see the web page: http://www.lbl.gov/∼olken/RBO/rbo.html

References

[1] ASHRAE. Bacnet - a data communication protocol for building automation and control networks. Technical Report ANSI/ASHRAE Standard 135-1995, American Society of Heating, Refrigerating and Air-conditioning Engineers, 1995. URL: http://www.ashrae.org/-BOOK/bacnetcd.htm.

[2] Gerald Brose. Jacorb performance. Technical report, Free Unviversity, 1998. http://www.inf.fu-berlin.de/-∼brose/jacorb/performance/results_09.html.

[3] J.F. Butler. Bacnet: An object-oriented network protocol for distributed control and monitoring. Technical report, Cimetrics Technology, 1996. URL: http://www.cimetrics.com/tipcic96.htm.

[4] Pierre Castori. Manufacturing messaging specification. Technical report, Ecole polytechnique federale de Lausanne (EPFL), Institute for Computer Communications and Applications, 1997. URL: http://litpc13.epfl.ch/-MMS/mms_main.htm.

[5] Norman Chang et al. Time synch - time synchronization server. Technical report, Univ. of Delaware, 1997. URL: http://www.eecis.udel.edu/∼ntp/.

[6] IAI UK Chapter. United kingdom chaper of industry alliance for interoperability home page. Technical report, UK Chapter of Industry Alliance for Interoperability, 1996. URL: http://helios.bre.co.uk/iai/.

[7] D. Claridge, J. Haberl, M. Liu, J. Houcek, and A. Athar. Can you achieve 150% of predicted retrofit savings? is time for recommissioning? In *Proceedings from the ACEEE 1994 Summer Study on Energy Efficiency in Buildings*, volume 5. American Council for an Energy-Efficient Economy, 1994.

[8] Yoram Cohen. SNMP - the simple network management protocol. Technical report, RAD Data Communications, 1995. URL: http://www.rad.co.il/networks/-1995/snmp/snmp.html.

[9] ESPRIT CCE-CNMA Consortium. *MMS: A Communication Language for Manufacturing*. Springer-Verlag, 1995. ISBN 3-540-59061-7.

[10] Werner Dreyer, Angelika Kotz Dittrich, and Duri Schmidt. Research perspectives for time series management systems. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 23(1):10–15, March 1994.

[11] Werner Dreyer, Angelika Kotz Dittrich, and Duri Schmidt. Using the calanda time series management system. In Michael J. Carey and Donovan A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD Internatioinal Conference on the Management of Data*, volume 24, no. 2 of *SIGMOD Record*, page 489, 1995.

[12] EIA. *Annual Energy Review for 1991*. Energy Information Administration, U.S. DOE., 1992.

[13] EPRI. Utility communications architecture. Technical report, Electric Power Research Institute, 1997. URL: http://www.epri.com/uca/index.html.

[14] EUN. Eun product guides: Building automation systems. *Energy User News Magazine*, pages 50–51, October 1994.

[15] A. Gokhale and D.C. Schmidt. Evaluating the performance of demultiplexing strategies for real-time corba. In *Proceedings of the GLOBECOM '97 Conference*, November 1997.

[16] Jeff Haberl, 1996. Personal Communication.

[17] P. Herzog and L. LaVine. Identification and quantification of the impact of improper operation of midsize minnesota office buildings on energy use, a seven building case study. In *Proceedings from the ACEEE 1992 Summer Study on Energy Efficiency in Buildings*, volume 3. American Council for an Energy-Efficient Economy, 1992.

[18] I-Kinetics. I-kinetics home page. Technical report, I-Kinetics Corp., 1997. URL: http://www.i-kinetics.com/.

[19] IAI. Industry alliance for interoperability home page. Technical report, Industry Alliance for Interoperability, 1996. URL: http://iaiweb.lbl.gov/.

[20] IETF SNMPv3 Working Group. SNMP Version 3 Working Group Home Page. Technical report, Internet Engineering Task Force, 1997. URL: http://www.ietf.org/html.charters/snmpv3-charter.html.

[21] S.R. Meyers. Chiller modeling error analysis: Implications for energy-saving retrofits and control strategies. Technical report, Mechanical Engineering Department, U. C. Berkeley, 1996. Master's Thesis.

[22] Thomas J. Mowbray and R Zahavi. *The Essential CORBA, Systems Integration using Distributed Objects*. John Wiley & Sons, 1995.

[23] Frank Olken. Remote building monitoring and operations project home page. Technical report, Lawrence Berkeley National Laboratory, 1997. http://www.lbl.gov/∼olken/RBO/rbo.html.

[24] Frank Olken et al. Remote building monitoring and control. In *Proceedings of the ACEEE Summer Study Conference*, volume 4, pages 285–296, August 25-31 1996. URL: http://www.lbl.gov/∼olken/mendel/RBO/-pubs/aceee96.ps.

[25] Frank Olken, H.-Arno Jacobsen, and Chuck McParland. Middleware requirements for remote monitoring and control. In *Proceedings of the OMG-DARPA-MCC Workshop on Compositional Software Architectures*, January 6-8 1998. http://www.objs.com/workshops/-ws9801/index.html.

[26] Frank Olken, Hans-Arno Jacobsen, Chuck McParland, Mary Ann Piette, and Mary F. Anderson. Development of a remote building monitoring system. In *Proceedings of the 1998 ACEEE Summer Study on Energy Efficiency in Buildings*. American Council for an Energy Efficient Economy, August 1998.

[27] OMG. Object management group home page. Technical report, Object Management Group, 1996. URL: http://www.omg.org/.

[28] Mary Ann Piette, Graham Carter, Steve Meyers, Osman Sezgen, and Steve Selkowitz. Model-based chiller energy tracking for performance assurance at a university building. In *Proccedings of the Cool Sense National Forum on Integrated Chiller Retrofits*, Sept. 23 - 24 1997. LBNL Report-40781.

[29] D.C. Schmidt, A. Gokhale, T.H. Harrison, and G. Parulkar. A high-performance endsystem architecture for real-time corba. *IEEE Communications Magazine*, 14(2), February 1997.

[30] Duri Schmidt, Werner Dreyer, Angelika Kotz Dittrich, and Robert Marti. Time series, a neglected topic in temporal database research? In James Clifford and Alexander Tuzhilin, editors, *Recent Advances in Temporal Databases, Proceedings of the International Workshop on Temporal Databases*, Workshops in Computing, pages 214–232. Springer Verlag, September 1995.

[31] A. Segev and R. Chandra. A data model for time-series analysis. In Nabil R. Adam and Bharat K. Bhargava, editors, *Advanced Database Systems*, volume 759 of *Lecture Notes in Computer Science*, pages 191–212. Springer Verlag, 1993.

[32] SimpleGroup. The simpleweb. Technical report, Centre for Telematics and Information Technology (CTIT) of the University of Twente (UT), 1997. URL: http://wwwsnmp.cs.utwente.nl/.

[33] SNMP Research International, Inc. SNMP Research International Web Page. Technical report, SNMP Research International, Inc, 1997. URL: http://www.int.snmp.com/.

[34] Wonderware. Common questions about netdde. Technical report, Wonderware, Corp., 1996. URL: http://www.wonderware.com/products/netddeq2.html.

[35] Yahoo. Snmp (simple network management protocol). Technical report, Yahoo, 1997. URL: http://pbm.yahoo.com/Computers_and_Internet/-Information_and_Documentation/Protocols/SNMP_-Simple_Network_-Management_Protocol_/.