

## The NIST Calibration Check Standards Database (CCSD)

Kevin Brady<sup>1</sup>, Rand Elmquist<sup>2</sup>, Frederic de Vault<sup>1</sup>

(1) Semiconductor Electronics Division

(2) Quantum Electrical Metrology Division

Electronics and Electrical Engineering Laboratory

*National Institute of Standards and Technology (NIST)*

### Introduction

For several years now, NIST has been a leader in the development and use of Information Technology to process calibration information. Beginning in 1996, researchers designed and developed a calibration database, the NIST Information System to Support Calibrations (ISSC), to track and maintain all calibration information from process benchmarking to customer calibration reports. Through the use of the ISSC, NIST has proved that the calibration process is more efficient when aided by electronic data handling and web-enabled clients, as one might expect. What is not as obvious is that modern automated calibration techniques generate too much data for researchers to handle manually, even in spreadsheets, or to exchange electronically with other researchers through reports, e-mails, and voice. The Metrology of the Ohm Project within the Electronics and Electrical Engineering Laboratory (EEEL) at NIST developed and now relies on a Calibration Check Standard Database (CCSD), a complement to the ISSC, for data storage, backup, and electronic retrieval of metrology data.

Metrologists and IT researchers are now testing a CCSD that is designed to be expanded as a general Calibration Database to handle check standards and customer standards data from every calibration area at NIST. This is the exact route the successful ISSC database took, first showing success in a single calibration area then expanding to a division, to EEEL, and finally to all calibration stakeholders at NIST. The CCSD is integrated with the ISSC to capitalize on data the ISSC already contains. The CCSD provides enhanced search features that are customizable and easy to use, have better security features, and provide more control of data by researchers. Like the ISSC, the system is configured to be useful to metrologists, based on two observations:

- 1) Measurement automation increasingly allows rapid accumulation of data. As more and more systems are automated, no longer will a single researcher be able to process the data manually. Improved technology will be required to process the overwhelming amount of data that will be generated and build detailed records of the trends and characteristics associated with precision reference standards and measurement systems.

- 2) Unprocessed reference data in a standard format has a useful lifetime of many years. Data maintained in the format proprietary to an instrument manufacturer are not easily transferred after the fact, and thus immediate transfer to database storage is preferable. An ideal metrology information system is not dependent on proprietary methods, since

---

measurement methods and data formats change frequently and are driven by new technology.

This paper will explain the design and implementation of the CCSD, along with the reasoning and rationale for the design decisions made during the development process. Also, we will explain the impedes for the creation of the database and the future directions we see further development headed.

## **1 Calibration Process**

First we look at a very simplified calibration to show where in the process the overload of data can occur:

- A scientist performs measurements on an instrument.
- The instrument is connected to an automated calibration system, it is configured, and the measurements are started.
- The environment is configured using a computer to choose which parameters should be measured.
- At the end of the process, a file containing hundreds of test points is generated reflecting the settings, the multiple readings and other information coming from the calibration system (This file is often written in a proprietary format).
- The scientist then extracts the data he needs and performs an analysis, using other data from the archives also, to check the history, and then he finishes the calibration of the instrument.

## **2 Observations**

Given this calibration process, three observations on data handling can be highlighted:

Accumulation of data: Measurement automation increasingly allows rapid accumulation of data. As more and more systems are automated, no longer will a single researcher be able to process the data manually. Improved technology will be required to process the overwhelming amount of data that will be generated and build detailed records of the trends and characteristics associated with precision reference standards and measurement systems.

Data in a standard format: Unprocessed reference data in a standard format have a useful lifetime of many years. Data maintained in the format proprietary to an instrument manufacturer are not easily transferred after the fact, and thus immediate transfer to database storage in a common format is preferable. An ideal metrology information system is not dependent on proprietary methods, since measurement methods and data formats change frequently and are driven by new technology.

A single storage location: A single place for data storage allows users to look up, manage, retrieve, and analyze data that came from different calibration processes. Also, a

central location makes emergency restoration simpler if a system is lost due to hardware malfunction.

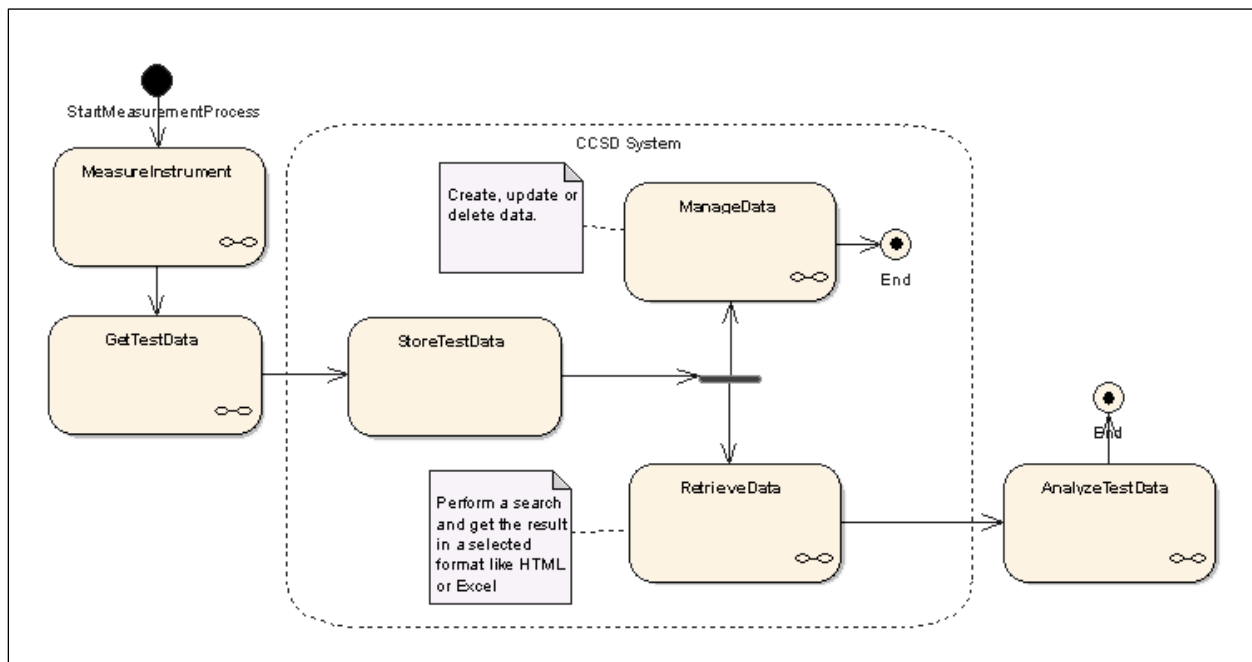
In summary, having a place where the data can be stored regardless of its previous format and retrieved in an efficient manner drove the development of the CCSD Application.

### 3 Why Use The CCSD Application

The CCSD Application is based on the design requirements of metrologists and is designed to:

- Store valuable data with minimal formatting.
- Allow the scientists to modify, delete, and add data.
- Provide an optimized search tool to select only the data needed for analysis.
- Download the data resulting from searches using different formats (Excel, HTML, Text)
- Allow the display of multiple searches in separate tabs of a browser at once for comparison.
- Sort the results of a search using any of the data columns.
- Restrict the access to the data only to people with the right to access it.

The CCSD can augment the calibration process as shown on figure 1:



**Figure 1: The CCSD Application in the calibration process**

After the test is done and all the measurements have been taken, the scientist collects the data he wants and stores them in the database. Later, he can select only the data he needs to analyze using the search parameters specific to his Calibration Area (usually the controls used during the calibration process). Since the data are stored in the same place,

the scientist will be able to retrieve data for the same instrument spanning several years and be able to monitor the behavior of the instrument over time.

Another important aspect of the CCSD Application is that it is web-based. The application is available from any laboratory, office, or even home (with the proper security access) as long as an internet connection is available.

Having the right credentials to store, retrieve, and delete calibration data required the utmost attention. Calibration data security had the highest priority in the design of this system. We have made sure only those authorized to access the data are allowed to do so. Also, we have further broken down the access to roles; people accessing the data are only allowed to perform actions they need to, most needing only “read” access. We provide this security to the CCSD Application via an access control and security layer:

- A user needs to be authenticated and authorized (administrator, user, and guest) in order to access any page of the application.
- All requests to the application are filtered to remove any possible malicious data.

#### **4 A Single Application For All NIST Calibration Areas**

The other aspect of the CCSD Application is to make it common enough to satisfy the requirements of all Calibration Areas at NIST. This allows for a single application that needs to be maintained; it is easier to protect, store, backup, and retrieve the data for all calibration areas.

The following assumptions were used in the design:

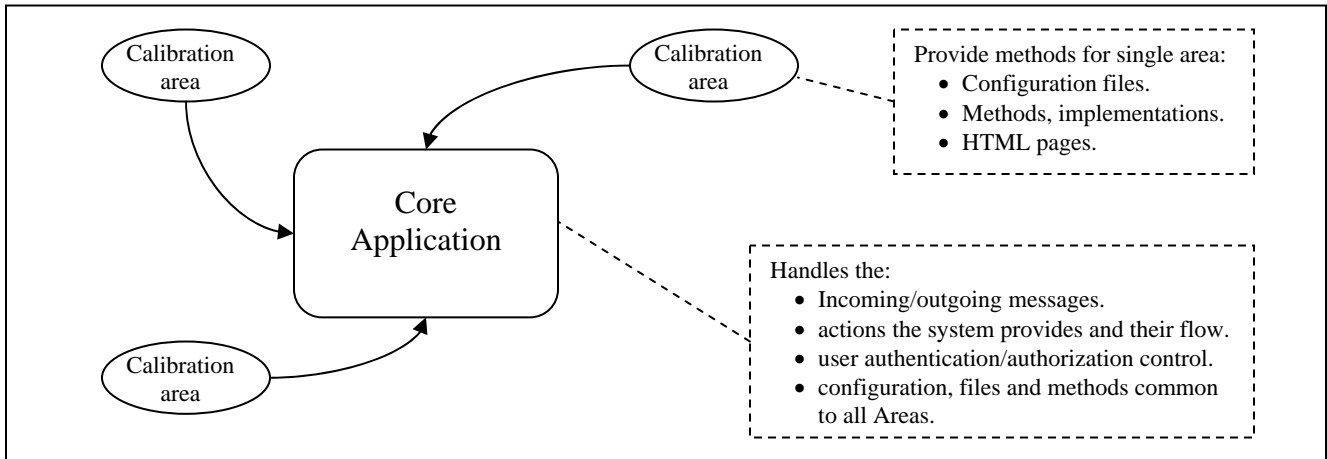
- Each Calibration Area has common information (instrument name, test data, instrument characteristics, and measurement system) that is stored in the database.
- Each Calibration Area needs a set of actions to retrieve data to perform actions (modify or analyze them).
- Each action should be performed in a secure environment so that only the authorized users can perform the action for a given Calibration Area.

Other considerations:

- A set of actions shared by every Calibration Area is most desirable, however each Calibration Area must be able to customize input and output parameters (e.g., which parameters are used for a search or what columns are displayed for the result).
- Each Calibration Area may want a different appearance for their web pages (design of the pages).
- Each Calibration Area doesn’t necessarily want to store the same information in the database.

To allow for each of these considerations, we designed a system that was composed of a core application and modules added to the core. When the core application needs to handle an action requiring a specific Calibration Area, it calls an Area Interface (façade describing what that Calibration Area can handle). A module or collection of files

specific to each Calibration Area can then be added to the application core providing the necessary specific Area Interface and views with the custom information for that Calibration Area. This provides the unique abilities required by each different Calibration Area along with a core set of common functionality, eliminating the possibility of a different design for each Calibration Area.



**Figure 2: System based on a core application and modules**

## 5 Current Implementation

Originally, the CCSD was a skeletal implementation that had only a minimum of attributes for data storage and search and was written in the PERL programming language. The application was converted to the newer Java technology to increase speed and allow more options for the client-driven search and retrieval technology. We started with two Calibration Areas in EEEL and analyzed, compiled, and implemented their data needs. After generalizing the approach, the CCSD is now capable of storing data from any of the calibration areas within EEEL. More generalization and expansion will occur in partnership with other metrology areas, after the proven success with the Resistance and Power Calibration Areas.

In order to store the results of a measurement process, the data-taking apparatus is required to generate files containing appropriate identification data (Calibration Area and process, time and date, owner) as well as specific results. NIST uses non-commercial software for many processes, but even commercial data-taking packages can usually be modified to provide these files. If this is not practical, spreadsheet methods (macros) can reformat data and generate text files that satisfy the input scripts used by database programs, see figure 1.

The web interfaces are developed and customized based on the requirements from each area and again generalized to support a common data format. There should be no long-term economic barriers to using this method, since the cost to expand to other areas would be much less than the original development costs. The software only requires the use of an internet browser on the client machine along with a spreadsheet package for analysis.

The CCSD brings database and data communication methods to bear directly on the problems from the project areas (voltage, phase, impedance, and other NIST calibration areas) and works with these users to produce custom interfaces for their diverse and complex measurement systems. This database keeps the NIST calibration services at the forefront of technology to help maintain them as “World Class.” Since these data are used to generate calibration reports, decreasing the time spent on analysis will free the researcher to better provide the services which improve the calibration process. Customers also benefit by the reduced calibration time and in the future may be provided secure, on-line access to the data and analysis tools used by NIST.

### **5.1 Persistent Data Storage**

The application still relies on a persistent data storage that was developed at the very beginning, but has been modified to allow for new calibration areas. The data are stored and managed by an RDBMS (Relational Database Management System) which is queried by the CCSD Application. An RDBMS is a powerful tool used to store and retrieve database information. It is optimized for search and allows the CCSD Application to retrieve specific data from among thousands of candidate data very quickly.

The information stored contains:

- Information about the calibrations.
- Information about the user running the application.

Part of the design phase was to describe the information each calibration area needed to store and retrieve. Since the application can be integrated in the calibration process, it needs information like:

- The instrument that is being measured.
- The set of properties (or characteristics) the instrument has.
- The measurement system used and its properties.
- The test data that represent each data point measured during the calibration process.

An important part of the information being stored are the data points measured during the calibration process. The values of these data points are the ones that will be stored, retrieved, and analyzed later on. The challenge was to predict how many measurement values will be taken for each data point, especially when trying to deal with more than one Calibration Area. We tried to contact other areas initially to help gauge the storage requirements and design of the database schema to handle these data.

The final design resulted in a compromise where the values from a data point would be divided in specific known fields (e.g., like model number, serial number, etc.) and the other part would allow for a variable number of input elements. The first part would hold the information about the data point and the controls settings used during the calibration process. The second part would hold the information about the readings.

Figures 3 and 4 present the database model and schema used by the CCSD Application. They describe the data model used to store the information about the calibration and the information about each user of the application (for security, to allow access only to authorized people).

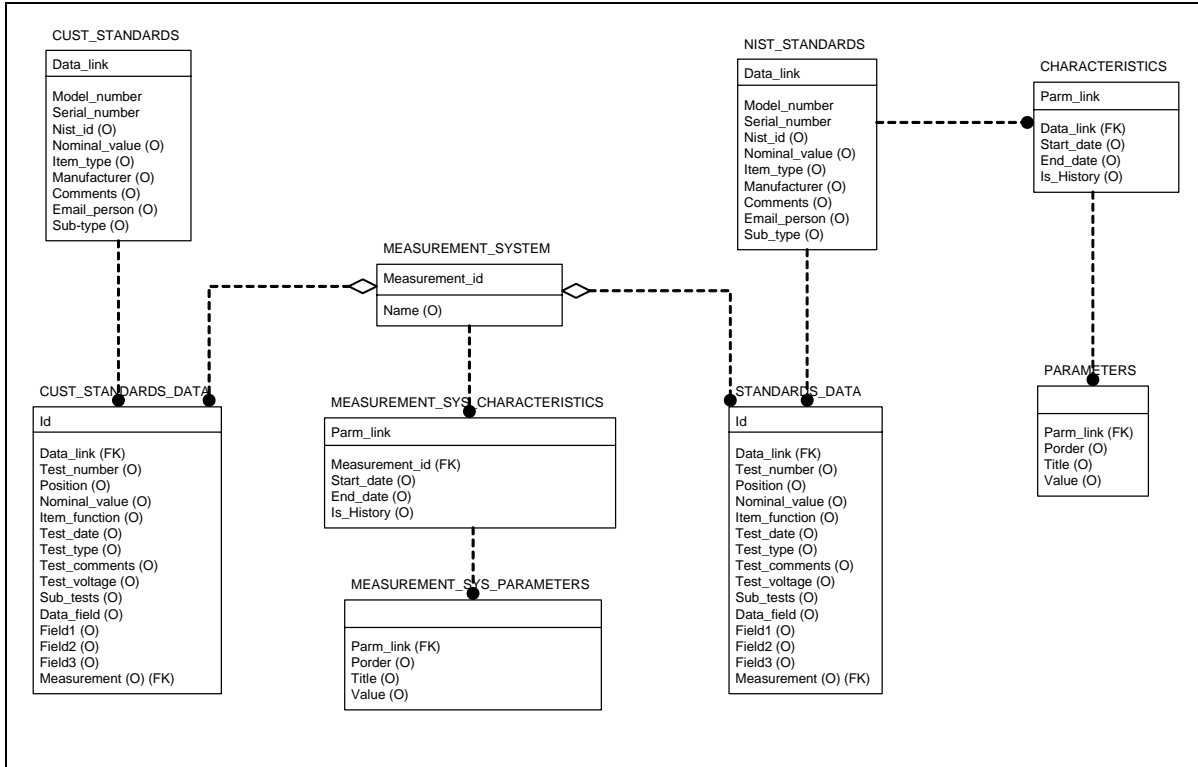
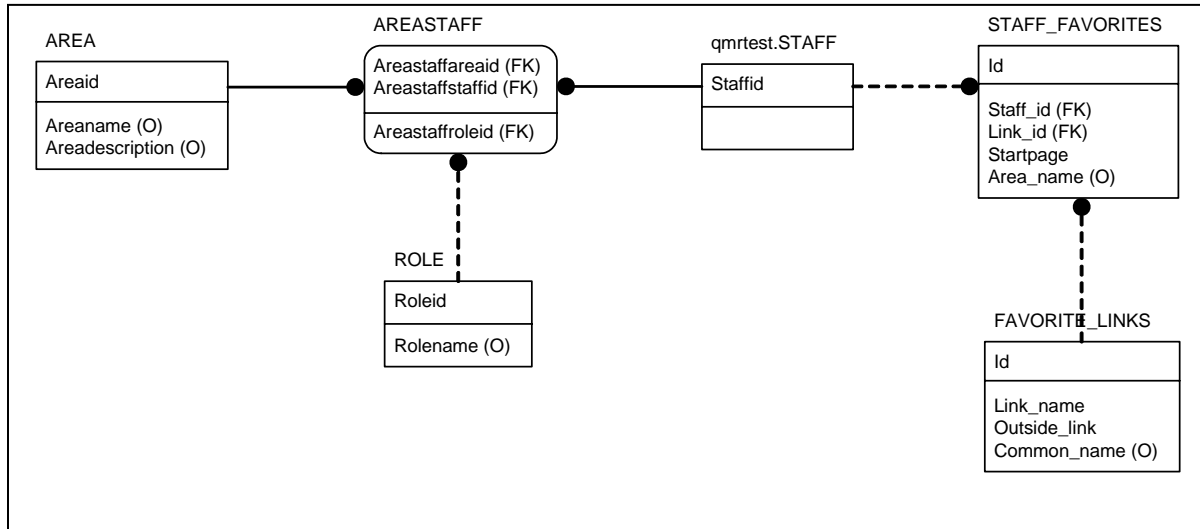


Figure 3: Data model and schema of the CCSD used to store calibration data.



**Figure 4: Data model describing Users and their preferences**

## 5.2 Getting Data into the Database

Data can be inserted into the database by creating a formatted file and simply placing it into a directory. The directory is networked and mounted from the server running the CCSD application. Every five minutes, a prescheduled job scans the directory looking for files. As files detected in the directory, they are opened and all data read in from them and stored into the database, and the files deleted. This process allows for automated calibration programs to take measurements twenty-four hours a day, seven days a week and have the data stored in the database with no user intervention.

Prior to storing data in the database, the administrator must manually add the information about the standard (instrument) to the table, containing the model number, serial number and other information. The data from the file described above are stored with a specific instrument by looking up the serial number and model number (which are contained in the file) and storing the data in the database with that specific item.

As mentioned above, the first three lines of the data file to be added to the database are common to each calibration area. They include the Calibration Area, test number, email address of the person responsible for the calibration, the local filename of the file, and the date and time of the measurement. The rest of the lines contain the actual data points and each start with the line number, the serial number, model number and then a variable number of fields with the accrual measurement. Figure 5 shows a sample file used by the Calibration Area “Power and Energy.” The information in the file will be used to locate the instrument (with serial number 160, type for Wathour, and model number MSB001A. The rest of the lines are the actual data points read and are stored as a string to allow for variable number of measurements. We used a “string” data type to allow varying lengths, since the actual analysis is not done in the database, but in the spreadsheets to which the data are retrieved.



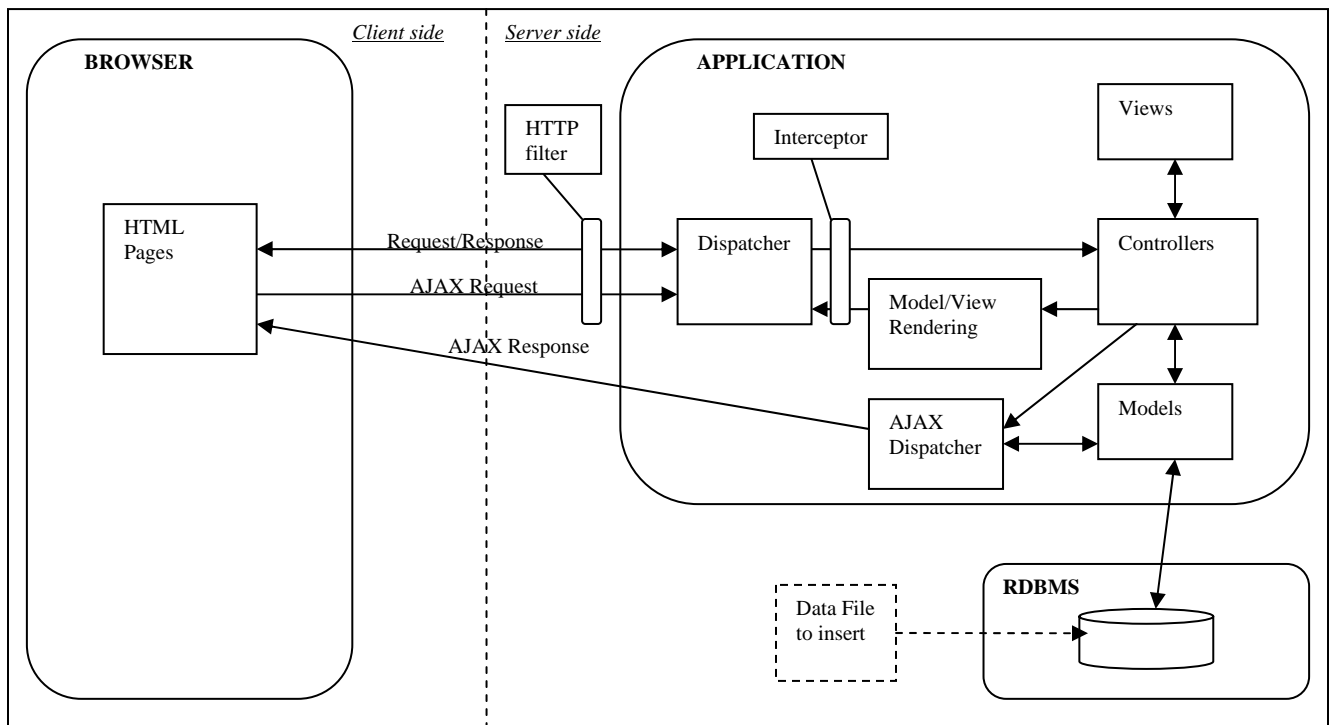
```
Power and Energy|56200C|NPB|
thomas.nelson@nist.gov|File Name: c:\DATA\file|
1|4/24/2005|13:00:00|65|115|6|0|24|
1|160|WATTHOURS|MSB001A|115|9|80.9850125|5.0103|115.001|5.1|80.9849|99.851|98.7753|65.9891|78.4854|98.9136|48.9741|98.1453|
```

**Figure 5: Sample of a file used by “Power and Energy” to insert data to the database**

Using a file to insert the data allows the user to populate the database with hundreds of data points at once and allows use by automated measurement systems. The sample above contains only a single measurement, but could contain thousands of lines of data.

### 5.3 Application Architecture

During the design it was obvious the application must be web based to allow for easy access to the data, with the only requirement a web browser. We have added no other requirements and use a “vanilla” version of the Mozilla and Internet Explorer<sup>3</sup> browsers. The application is running on a server, and Figure 6 shows the logic of the CCSD Application, the key components of its architecture, and the flow of messages passed from component to component.



**Figure 6: Implementation Schema**

The system is composed of the browser which is on the client side (user’s computer), the CCSD Application, and the RDBMS which are in the server side.

<sup>3</sup> NIST does not endorse the use of any particular internet browser

During a usual execution, the user will access a web page and send a request to the application which will use its dispatcher to send the request to the right controller. The controller will create a model using data from the database by sending queries to the RDBMS, get the appropriate view, and return the result of the merge back to the dispatcher that will send it back to the user.

The Model-View-Controller is an architectural pattern used to design the application. It decouples the business logic from the user-interface that allows a better flexibility, maintainability, and scalability of the application.

The interceptors located in between the dispatcher and the controllers are used to control things like:

- Is the user logged in?
- Has the user the right credentials?
- Has the user selected a Calibration Area before performing a search?

The AJAX (Asynchronous JavaScript And Xml) technology is used to send a request that will refresh only a section of the web page.

#### **5.4 Limitations Of the Current Implementation**

The main limitations of the current implementation are coming from the data model. The way the data model is designed plays a role in the capacity of the application to handle the information since it is directly linked to the business logic. Right now, all information that is valuable to the user is stored in the database; however where pieces of this information are stored and how is the issue.

Data in a pipe delimited string: This design solution has been chosen to allow a data point to have as many readings as needed without having to extend the number of fields in the table. The problem is that it is very difficult to make a search through this kind of information. It is also difficult to sort it and modify it. Even though at the beginning of the project, only data without modification were to be in this field, it hasn't been adhered to with the integration of other Calibration Areas (too many control fields for instance).

Data not stored in specific fields: Even though the design of the data model has been taken into account for multiple Calibration Areas, some of the fields remains too "Calibration Area specific." It results in using the fields only for storage but not for their name or data type. This means that a single field of a table could hold information about voltage for one Calibration Area and about time for another one. It becomes difficult to know what is what.

Non specific data types : Since almost everything can be stored everywhere, we need to have generic data types which give some limitations, like sorting through a particular field. Even if for instance it would seem that all the data from a selection were numbers,

if the data type was actually a string the sort would be alpha-numerical instead of being numerical.

These limitations do not stop the CCSD Application from running and serving its users. Two Calibration Areas have been integrated with success so far. The concerns are more about future features that would ask more treatments from the application and also make the data model even more generic.

## **6 Future Directions**

This section presents the new analysis that is being pursued in parallel with the development of the current implementation.

### **6.1 New Concepts In Analysis**

The first analysis was mostly based on the data model; in other words we first looked how to store the information we need and then tried to integrate this data model into the application.

The new analysis is more based on an object representation of the system where the first goal is to describe the environment of a Calibration Area like:

- The actions it performs.
- The data that are produced.

The way the data are stored is secondary. Some of the data could be represented in an XML file or in the database. However we have to keep in mind that we need an efficient way to search through the data.

We also can use metadata to describe an object. During the initialization of the application, each object would be configured using the metadata and we would know dynamically what are the type a data needed for each Calibration Area.

### **6.2 Another Approach To Describe A Data Point**

Instead of trying to know what fields a data point has according to a Calibration Area, we could look at what creates a data point.

For instance when a scientist wants to measure an instrument, he is going to set up a calibration process. This process has condition values (or controls) that will be used during the measurement and generate readings. The scientist knows exactly what the controls and the readings are for a specific calibration process. In order to analyze the data produced, the scientist needs to keep the controls values and the readings values. One set of controls and readings will represent one data point.

### **6.3 Conclusion**

This database will have a significant impact on the ability of NIST Calibration researchers to efficiently perform their work. As mentioned above, eventually they will be “drowning” in data and will take many different approaches in developing single-function database applications. This database will solve that problem now, before it becomes serious, and allow all calibration areas to more efficiently process their data. This database also “takes the data out of the file cabinet” and puts it on-line. Currently much of the calibration service data at NIST are not electronically accessible, or are stored in spreadsheets or files without a cataloging system. The CCSD will allow spreadsheets to be stored online and also provide continuing backup of raw data for disaster recovery. Reducing time for calibrations, making the process more efficient, and providing backup and electronic access clearly benefits NIST’s calibration customers. Reducing calibration times while not increasing the cost to customers is also a benefit which can be passed on to customers of calibration services.