CORMS AI: Decision Support System for Monitoring US Maritime Environment

Haleh Vafaie, Ph.D.
Northrop Grumman Corporation
Chantilly, VA 20151 USA
Haleh.vafaie@ngc.com

CDR Carl Cecere, USPHS
NOAA Ocean Service
Silver Spring, MD 20910 USA
carl.cecere@noaa.gov

Abstract

Rule based reasoning and case based reasoning have emerged as two important and complementary reasoning methodologies in artificial intelligence (Al). This paper describes the approach for the development of CORMS AI, a decision support system which employs rule-based and case-based reasoning to assist NOAA's Center for Operational Oceanographic Products and Services watch standing personnel in monitoring the quality of marine environmental data and information.

CORMS AI has been in operation since July 2003. The system accurately and reliably identifies suspect data and network disruptions, and has decreased the amount of time it takes to identify and troubleshoot sensor, network, and server failures. CORMS AI has proven to be robust, extendable, and cost effective. It is estimated that CORMS AI will save government over one million dollars per year when its full range of quality control monitoring capabilities is implemented.

1. Introduction

The National Oceanic and Atmospheric Administration's (NOAA) National Ocean Service (NOS) has implemented a manned Continuous Operational Real-Time Monitoring System (CORMS) to provide 24/7 quality control monitoring of marine environmental information, data acquisition and ingestion networks, and data dissemination servers. The primary function of CORMS is to ensure the availability and accuracy of real-time data provided by the Center for Operational Oceanographic Products and Services (CO-OPS) in support of NOAA's mission goal of ensuring safe, efficient and environmentally sound maritime commerce. This mission is especially important as 98% of all cargo entering the United States passes through our Nation's ports and harbors. Nearly half of this cargo is classified as hazardous.

The first version of CORMS (CORMS Classic) was designed to receive automated sensor data in real-time and near real-time intervals from the Physical Oceanographic Real-Time System (PORTS®) [9] and the National Water Level Observation Network. Quality control checks are performed to flag data that appear suspicious. The CORMS system receives these sensor data and presents this information in graphical and textual format to the watch stander. At this point, the effectiveness of the CORMS system to monitor data quality is dependent on the relative capacity of the watch stander to consistently recognize suspect sensor data from the presented information, and to choose the correct series of remedial actions.

The ability of watch standers to consistently make "quality decisions" (i.e., accurately access suspect senor data and follow approved methods and procedures) is directly tied to their knowledge and experience. Thus, a distinct variation in the quality of decisions being made between novice and expert watch standers has been observed. Other factors such as fatigue, distractions, and workload tend to detrimentally impact the ability of all watch standers to consistently assess data quality and choose the most appropriate form of remediation. The volume of data currently being assessed by CORMS is relatively small compared to what is projected for the future. CO-OPS is pursuing an aggressive research and development effort that will result in the deployment of both increased numbers and new types of sensors to support new applications. To ensure the continued viability of the national PORTS® program, a robust, efficient, and flexible quality control system must be in place to ensure the availability of high quality data and information [10].

2. Background

In order to maintain and improve the current CORMS process, a more robust decision support system for monitoring sensor, data ingestion, and data dissemination system failures must be developed. This system must extend the capability of the existing system by enhancing the capacity of watch standers to use established policies and procedures as well as the acquired knowledge of oceanographers, field engineers and computer specialists to consistently and accurately identify suspect data and system failure scenarios.

To develop this system CO-OPS desired to acquire a commercial off-the-self (COTS) tool to meet the above-mentioned requirements. Of the tools in the market, two categories best satisfied these requirements, rule-based tools and case-based reasoning (CBR) tools. Rule-based systems are commonly used to depict policies, procedures, and best practices and apply them to real-world problems. Case-based reasoning captures the past experiences and reasons on the new situation based on the past experiences. The next section briefly describes tool selection process, please refer to Vafaie et. al [11] for detailed description.

3. Tool Selection

Extensive tool evaluation is a very time and resource intensive process. Hence, the tool selection process was segmented into two stages. During the first stage over 70 tools were investigated. These tools were listed at various web sites for commercial expert systems tools [4, 5], Case-based reasoning tools [2,3], books and articles [1,6,7]. After some initial research six tools were selected for detail investigation. These tools are shown in table1.

Vendor Name	Products	Approach
NRL [1]	(NaCoDAE)	Case
Kaidara	Kaidara Advisor	Case
Empolis	Knowledge Builder	Case
Mindbox	ART-Enterprise	Hybrid
Haley Enterprise[2]	Easy Reasoner	Hybrid
Brokat Technology	Brokat Advisor	Rules
Gensym	G2 Classic	Rules

Table 1: Initial candidate tools

A set of criteria using eight dimensions were developed to aid in the process of down-selecting tools. A subsequent task was to determine the relative impact or weighting of these eight dimensions of criteria to ensure those criteria most important to a given project are given precedence.

The first four dimensions of down-selection criteria, User Presentation, Data Connectivity, Architecture, and Functionality mapped to the project requirements directly to ensure adequate coverage of project requirements with criteria. The Technical Support, Pricing, Installed Customer Base, and Company Profile dimensions are included to indicate opportunities to include factors into the COTS evaluation process that are tangentially related to the relative strength of a COTS product to meet ones criteria. The scores within each category were averaged, and the category scores weighted and summed to create a single score for each product. The product with the highest score is selected as the vendor and product of choice.

MindBox's ART Enterprise was the finalist COTS package. Key to this decision was that MindBox is a "hybrid" application, in which it incorporates both a rules and a case-based approach to reasoning. MindBox provides the minimum satisfaction of project requirements for the CORMS AI solution, and provides a capacity for additional add-on tools to extend their software to accommodate a variety of potential future applications. Because this product employs a mature case-base set of functionality, this COTS is positioned to enable rapid deployment into a prototype CORMS AI solution. Finally, because the product is a "hybrid", we estimate this tool will provide us with additional flexibility in terms of how it is deployed to lessen the overall impact of the system to the culture, methods and existing processes.

[1] Not included in the evaluation because it was still in development phase.
[2] This company did not participate in the evaluation process.

4. System Architecture

The purpose of the CORMS AI system is to enable quality control oversight of network and oceanographic sensor data being used for real-time marine navigation applications. To aid the safe navigation in the ports and harbors across the United States, water level (WL), current (CU), meteorological (MT), and conductivity-temperature (CT) sensors have been deployed at strategic locations. In combination, these sensors provide vessel pilots with a real-time and comprehensive view of conditions potentially affecting his capacity to safely navigate the local waterway and shipping channels.

Every six minutes, each oceanographic sensor generates a static measurement. This data is downloaded to a data acquisition system (DAS), and transmitted to CO-OPS. When received, the data are loaded into a database. The CORMS AI application polls the database for new data and reads it into memory. Reasoning logic is applied consistent with the stored rules and cases. Results from the reasoning process are written back to the database. These results include a flag indicating the quality status of the measurement for real-time use.

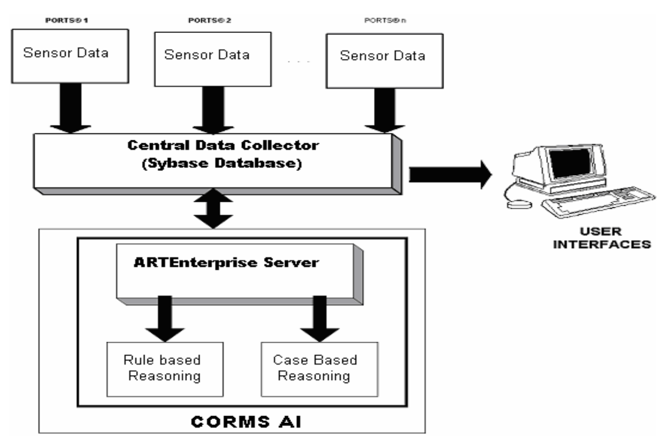

Figure 1: CORMS AI System Architecture

Applications that enable the presentation of real-time sensor measurement information to maritime users read the results tables in the database. Applications, which deliver real-time oceanographic data and information, include wireless applications, an interactive voice response telephone system, and web-based products on the Internet. Figure 1 shows high-level CORMS AI system architecture.

5. Application Architecture

In this section we describe the approach for the development of CORMS AI application. As described previously, CORMS AI monitors marine environmental information, data acquisition and ingestion networks, and data dissemination servers. For the data acquisition and ingestion networks as well as data dissemination servers there are well-defined set of rules that could be used for problem detection. Hence, it was determined that implementation of logic for identifying network interruptions incorporating rules is the optimal method.

Marine environmental data are collected using strategically located sensors at various estuaries across the United States. There are well-defined guidelines for interpreting the flags and the quality of the data reported by these sensors. Therefore a rule-based reasoning approach is also optimal for monitoring the quality of sensor data.

When a failure is identified, the watch stander must interpret the nature of the failure, find the cause of the failure, decide what course of action (or non-action) should follow. The evaluation of the potential sources of the problem and prescribed methods for mitigating the situation fully depends on an expert's experience and decision. Therefore, a case-based reasoning approach was used to aid personnel with this process.

Rule-based Reasoning

A rule-based system represents domain knowledge in terms of a set of rules that indicates the actions and conclusions in different situations. A rule-based system consists of a set of rules, a set of facts, and an inference engine for controlling the application of the rules, given the facts or conditions. Whenever the conditions in a rule change, the inference engine reevaluates all rules that contain that condition. If the actions taken by those rules impact other rules' conditions, then those rules are reevaluated. The inference engine offers the pattern matching capability that specifies when a rule should fire. This eliminates the need for the complex navigational programming required by non-rule based systems. All of the rules used by the system are stored in a *rule base* [12]. Rule-based systems have many advantages over the traditional systems. Rule-bases can be easily updated since rules are independent of each other.

For initial implementation of CORMS AI, incoming environmental data has already been quality controlled and the appropriate data flags have been set for each sample. Given this scenario it would appear that traditional data constraints encoded on the RDBMS could be used to detect and report suspect data conditions. Established rules for identifying failure conditions, however, rarely correspond to the data quality of an individual sample.

Rules for identifying suspect data conditions generally require a condition to persist across multiple contiguous samples or for a certain percentage of samples over a given time frame. Additionally, some of the rules for determining potential system failures aren't based on sample data quality, but on the absence of data over some period of time. These types of anomalies can't be detected via conventional data constraints. In many cases, suspect data conditions occurring simultaneously on samples from different sensor or from different locations may be indicative of a larger problem. MindBox rule-based reasoning engine is specifically designed to support rule salience – i.e. the application of a rule hierarchy – to detect these kinds of conditions. Lastly, the design of the MindBox rule-based reasoning engine is well suited to making backwards looking checks against previous sample data, doing forward looking checks against predictive data, and for making side by side comparisons with data which should share similar characteristics. Traditional data constraints encoded on the RDBMS are not particularly well suited to efficiently perform these kinds of checks in a highly transactional, real-time environment.

Rule-based Component- Identify Data Flow Interruptions

Again, since there are well-defined procedures to detect failures we have represented them in terms of rules. CORMS AI currently incorporates 16 rules to identify suspect data conditions and network problems, and provides 38 cases to help diagnose the most probable cause of the identified conditions. Figure 2 and 3 shows the pseudo code of a rule developed for CORMS AI and a screen shot of CORMS AI output.

```
If (measurement instance no-data-failure =1 )
    Identify Instrument instance
    Identify station instance
    set variable station-failure-no-data = 1
     For  all instruments in the station
       if ( inst object  no-data-failure != nil)
             station-failure-no-data = 0; end
       if  station-failure-no-data =1
      Set sys-status to 1
        Add a ticket ; end; end
end
```

Figure 2: FNDF001 - PORTS® Station Failure No Data Flag Condition Failure pseudo code

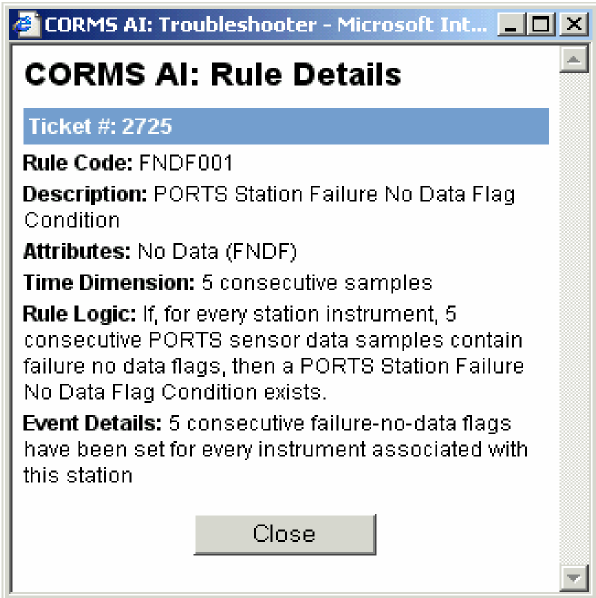

Figure 3: sample screen shot of a rule output

Case-based Reasoning

Case-based reasoning is based on the notion that human expertise is not merely comprised of formal structures like rules, but also of experience. An expert reasons by relating a new problem to previous ones [8]. Case-based reasoning makes decisions based on the actions taken in similar problems previously encountered. The decision is based on comparing current situation to circumstances of past problems, and looking for close matches to determine the best action to take. The past experiences are stored in case-bases, which are a database of cases. Each case describes the problem and its specific features and values, as well as an appropriate action for that problem.

Case-based reasoning has several advantages over reasoning with rules. The main advantage is that it is relatively easy to set up a knowledge base. Experience has shown that it is commonly very difficult to capture knowledge on a problem domain in a set of rules if there are no well-defined standard operating procedures. However, common examples of problems in a domain with their solution are either available or could be acquired. Because adding new cases to the case base can accommodate changing circumstances, a CBR system can learn from experience. Domain experts can easily maintain the case base, because there's minimal programming involved, and the CBR system automatically handles contradicting cases. Expanding a rule-based system on the other hand is much more difficult: adding one rule often means rewriting a large part of the rules [1,8]. Case-based reasoning systems can be built without knowledge acquisition bottleneck, since you can have a case-base by only having a single past experience or case.

Case-based Component- Evaluate Data Quality Failures

In developing CORMS AI we first had to create the many cases for the different case-bases. Five cases-bases were created, one for resolving network interruptions and one for each different type of sensor. The reasoning for this approach was that the sensors are quite different in themselves and also the problems reported by them are quite different in nature and type.

In developing the many scenario case-types we interviewed watch standers, oceanographer and field engineer staff deemed "expert" at interpreting the actual data generated directly by sensors and the quality flags generated by the DAS system that are used to determine the existence of failures. The examples and the scenarios were then used to define the case-bases and to convert the examples into formal cases. Figure 4 shows a sample case-base output.

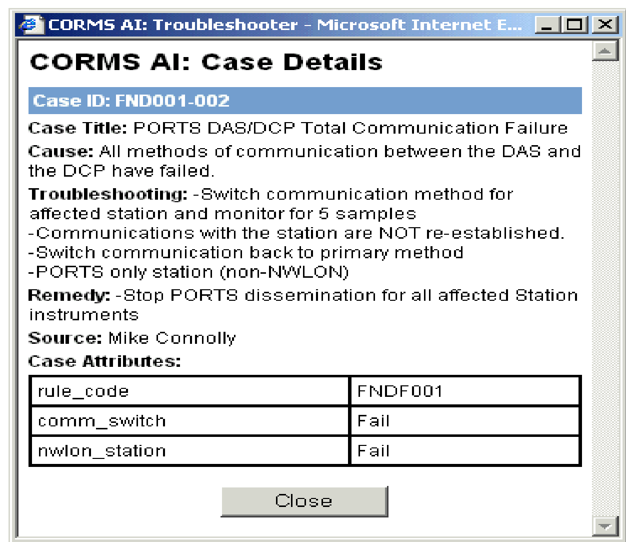

Figure 4: A screen shot of an example case output.

6. Benefits and Payoffs

CORMS AI has been in operation for 24 months and is currently running in parallel with CORMS Classic. CORMS AI is operational 24/7, supporting CORMS watch standing personnel, managers and supervisors, as well as field engineers and oceanographers.

When CORMS operations were first brought online, there were only 4 PORTS®, 34 stations, and a total of 51 sensors to monitor. The interface developed to assist the CORMS watch standing personnel to monitor sensor data and network performance data was easily presented on a single display. Today, CORMS watch standing personnel are responsible for monitoring environmental samples from 12 PORTS®, 150 stations and approximately 275 sensors distributed throughout coastal and inland waterways on the United States Atlantic, Pacific and Gulf coasts, Hawaii, Alaska and the Great Lakes. Each sensor reports every six minutes. This amounts to approximately 65,000 discrete samples per day being collected and quality controlled. Prior to the implementation of CORMS AI these samples were monitored manually. Because of the rapid expansion of the PORTS® program in recent years, one of the greatest challenges facing the watch standers today under CORMS Classic is simply navigating through the multitude of textual and graphical displays required to organize and present the current volume of information to be monitored.

Quality control issues require a suspect condition to persist for as few as one to as many as ten samples. Therefore, watch standing personnel using CORMS Classic are responsible for navigating through all of the interfaces described above in order to proactively identify and monitor suspect data quality conditions and potential network failure scenarios as they are emerging over time. They must accomplish this task while they are taking mitigating actions with respect to previously identified quality control issues and notifying the appropriate scientific and field staff of ongoing quality control issues.

At any given time there are, on average, thirty or so potential emerging data quality issues that must be identified and monitored. This is where CORMS AI has perhaps its greatest impact on CORMS operations. CORMS AI is able to identify and monitor all of the emerging data quality issues. The watch standing staff no longer has to proactively monitor the incoming data, but can simply take action when CORMS AI notifies them that a data quality or network issue has persisted long enough to require some mitigating action to be taken, based on formally established rules. The thirty or so emerging data quality or network issues that are continuously in progress result in an average of about twenty true quality control issues per day being identified and reported to the CORMS watch standing staff.

Under CORMS Classic it is possible for a condition to persist for an indefinite period of time before it is recognized and acted upon by the watch stander. Data quality or network issues may also be allowed to persist undetected for some period of time and then would correct themselves before the condition was ever discovered and reported by the watch stander. Either of these cases results in the dissemination of data of suspect quality to a community that relies on the data to make decisions having potential public health and safety consequences. Under CORMS AI, each quality control issue is identified the instant some specific, predetermined criteria are met. This guarantees that CORMS watch standing personnel are alerted to quality control issues as soon as they occur. This also guarantees that each and every time these specific, predetermined criteria are met the CORMS watch stander will be alerted. This allows CORMS watch standing personnel to consistently and expeditiously stop the dissemination of data of suspect quality to the navigational community and the public at large.

Once CORMS AI identifies and reports a suspect condition, it automatically initiates standard operating procedures associated with the identified issue. For example, CORMS AI will automatically stop the dissemination of suspect data, or will initiate network diagnostics if there appear to be network outages. This ensures a consistent response to the identified condition, and removes the burden of initiating these standard procedures from the watch stander, freeing him or her to do other tasks. While CORMS AI intervenes on behalf of the watch stander, it does not remove him or her from the process. The watch stander has the option to continue to monitor the identified situation in place, elevate the issue to the next level, or simply acknowledge the issue and reinitialize the condition. Should the watch stander choose to elevate the issue, CORMS AI ensures that a consistent reporting format is used to notify the appropriate headquarters and field personnel based on the nature of the problem identified, the type of sensor experiencing the problem, and the geographic placement of the sensor.

The number of PORTS® and sensors in the PORTS® program continues to increase at an accelerated pace. CO-OPS is projecting that as many as 150 PORTS® and upwards of 1,000 sensors could be brought on line in the future. To keep up with this demand under CORMS Classic it has been projected that the current staff of seven watch standers, one supervisor and one manager would have to be increased to 27 people at an increased cost of approximately $1.2 million per year to the government. This rapid expansion of the PORTS® program has been one of the main considerations driving the continuing development of CORMS AI.

As CO-OPS continues to expand the number of PORTS® and sensors being monitored, CORMS AI has proven capable of meeting the increased workload. To meet anticipated demands CORMS AI must be capable of processing data from all of the sensors in the network before the next sample from each sensor arrives, which will be six minutes later. Presently, CORMS AI is capable of processing through data from the current load of about 250 sensors in approximately 10 seconds. CO-OPS believes that CORMS AI will be able to handle the full anticipated load of PORTS® and sensors allowing CORMS to continue to operate while adding only one additional watch stander to the staff.

7. Application Development and Deployment

The development of CORMS AI took approximately 21 months to move from project initiation to the implementation of the first operational prototype. Of these 21 months, the first twelve months were spent analyzing business requirements, selecting COTS tools to best support the business, designing the overall system architecture, and gathering the specific domain knowledge from CO-OPS experts to allow for the development of the rule and case base. The subsequent nine months were spent developing the three major components of the CORMS AI system. These components include the central database, the Art Server component (rule- and case-based reasoning system), and the user interface, as demonstrated in Figure 1. These three components were designed in close concert and integrated with one another. The entire CORMS AI system was then tied into CO-OPS' existing data ingestion infrastructure. The database and user interface components were developed using existing CO-OPS software and development methodologies. These components were also developed and staged on existing hardware platforms according to established CO-OPS software development procedures. The rule- and case-based reasoning component was developed on a dedicated server purchased specifically for this purpose. All source code, including database DDL and DML, Art*Script, HTML, JavaScript, as well as project documentation is maintained in a software version control system.

The database component was developed using Sybase version 12.5. The database was initially hosted on a development Windows NT platform but was subsequently ported to an existing database server running on one of CO-OPS Sun systems. This move was made to ensure the database was running in CO-OPS standard data environment under the scrutiny of the CO-OPS database administration function. The database serves three main functions: It initializes the rule-based system with all of the information it needs to know about PORTS®, the stations associated with the various PORTS®, and the array of sensors deployed at each of the stations. This information is read into the rule-based system's memory upon system initialization. The database also serves to ingest and present incoming data samples to the rule-based reasoning system. Finally, the database records the details of any rules that are fired and takes steps to initiate an appropriate course of action on behalf of the watch stander. While it is the rule-based reasoning system that identifies a particular quality control issue has occurred, it is the database that facilitates the course of action that will follow. This is accomplished by storing in the database data which represents the business logic for each of the rules that are implemented in the rule-based reasoning system.

The user interface is a web-based CGI application developed using Perl and JavaScript. This browser based application displays information about data quality issues (tickets) that have been identified by the rule-based reasoning system to the CORMS watch standers, supervisors and managers, as well as the scientific staff and field crews. The information available via the user interface depends on the role of the user who is logged in. Use Case modeling was used to discover, document and implement the various user roles, interactions and dataflow associated with CORMS AI.

The user interface alerts watch standers to an identified data quality issue and provides all necessary supporting information, such as the specific nature of the data quality issue, when it occurred, and where it occurred. If the data quality issue is related to a communications failure the user interface will include information indicating that communications have been disrupted for an entire PORTS®, or for a particular station within a PORTS®. If the data quality issue is related to a specific sensor the user interface will identify the specific sensor which caused the data quality issue. The user interface will allow the watch stander to gain additional information about each data quality issue identified by providing information about exactly what condition(s) had to be met in order for that issue to be identified and reported via CORMS AI. The user interface allows the watch stander to search the case base to determine the most likely cause of the problem that was identified. The user interface will report on all actions that were taken on behalf of the watch stander as a result of the data quality issue being identified. It is via the CORMS AI interface that the watch stander can either elevate the issue by notifying CO-OPS scientific staff and field personnel, or to close out the issue and reset the data quality condition. When a data quality issue is identified, depending on the nature of the problem, the appropriate personnel (scientific or field) are notified. Each group has access to their case-base through the user interface. The case-bases are searched to determine the best course of action to resolve the issue at the minimum time and cost.

The user interface interacts only with the central database. It queries and updates data via a series of database stored procedures. This ensures a more secure application environment by limiting interaction with the database to stored procedures written and stored on the database. It also allows for the behavior of the user interface to be modified without having to make changes to the interface code. The user interface uses native Sybase drivers to communicate with the database.

Operational Evaluation

CORMS AI underwent operational evaluation for several months. During that time it ran in parallel with the legacy

CORMS application. The initial plan called for watch standing personnel to monitor the legacy system to identify problems and then check to see that CORMS AI had also correctly identified the problem. The evaluation procedure was modified, however, because CORMS AI was able to identify suspect data and system failures more quickly and more consistently than the legacy application. Additionally, CORMS AI was identifying conditions that were not being caught by the legacy application.

System Development Challenges

In developing the rule and case base component of CORMS AI the major difficulty was the knowledge extraction bottleneck. After deciding on the rules and cases to be implemented, the experts needed to agree on a single behavior for a rule or a standard course of action for a case. Rules and policies were developed, to accommodate conflict resolution, or overcome deadlocks to achieve a consensus on rules behavior, or a case's course of action.

Some other challenges faced by the CORMS AI development staff were creating the data structures and database objects (triggers, stored procedures, and views) to execute the business processes linked to the identification of a particular data quality issue. To accomplish this required an analysis of not only the conditions that constituted a specific data quality condition to be reported to the watch stander, but the precise set of actions to be taken as part of the problem mitigation. This information had to then be translated into a system of data structures, stored procedure and triggers which worked synergistically to drive the behavior of CORMS AI following the identification of a specific data quality issue. Other challenges related to the implementation of the database centered on the fact that CORMS AI was designed to ingest and monitor data from multiple sources in multiple formats. This required the data design to be flexible enough to accommodate the variances between the formats of the data received from different systems, and to accommodate the variances in how these systems were physically represented in other corporate databases. This challenge was recognized early in the design phase of CORMS AI.

While the final data design in CORMS AI supported incoming data from multiple sources in multiple formats, the project team deemed it too difficult a task to integrate CORMS AI into multiple existing data systems. This was perhaps one of the biggest lessons learned. The CORMS AI development team had identified early in the development process the risks associated with trying to integrate CORMS AI with more than one existing system. As a practical matter, the scope of the project should have been limited to integration with a single system. This is not in any way a shortcoming of the CORMS AI development effort, but an information architecture deficiency that should have been addressed within the organization prior to attempting the desired level of integration.

Another lesson learned was that of keeping the entire CORMS AI staff engaged in the project, even when their input was not directly required. For example, the amount of work associated with developing the central database and user interface portion of CORMS AI was significantly under estimated. To further complicate matters, once the project team was ready to start engaging the staff to build the database and user interface, the data architect and application programmer had to be brought up to speed on the entire project. Once engaged on the project the data architect and user interface developer, because of their specialized skill sets, were able to identify areas of the overall system design that either had to be reviewed and revised, or expanded upon to ensure the desired specifications could be delivered.

8. Maintenance and Enhancements

Figure 1: System Architecture demonstrates that CORMS AI is comprised of functionally partitioned components, which may be enhanced and maintained independently. Overall, CORMS AI is a stable system and has required very little maintenance over the course of time it has been operational. The most frequent reason for maintenance of the CORMS AI system is related to keeping the database component synchronized with the data in the existing organizational infrastructure. This maintenance is relatively simple to perform and can be done by CO-OPS staff. Keeping the database in sync with the data in the organization's existing infrastructure requires the CORMS AI project team to monitor the data in the existing infrastructure for change. This was initially a manual effort, which required up to 5 staff-hours per week, depending on how many changes were to be made. There was always a danger that changes made to the data in the organization's existing infrastructure would not be noticed and CORMS AI would be making decisions on an obsolete or incomplete set of sample data. This risk was identified before CORMS AI was put into operation and CO-OPS has taken steps to automate the process of discovering and reporting changes in the data in the existing infrastructure which require an update to the CORMS AI database. CO-OPS is working to consolidate data from all operational systems, which will eliminate this problem altogether.

There have been relatively few modifications made to the user interface component of CORMS AI. One modification that was made was done so to provide the watch standers with some way of evaluating whether or not the entire CORMS AI system was up and running. The watch stander's only interaction with CORMS AI is via the web-based user interface. This interface displays information about the data quality issues (tickets) that have been previously identified by the rule-based reasoning system. It is not uncommon for there to be long periods of time where no data quality issues are identified. During this time the user interface will simply appear static. The same behavior would occur if the database was no longer able to ingest data samples, or if the rule-based reasoning system went offline. For this reason an indictor of the systems overall operational status was included so that watch standers would not have to wonder if one or more of the system components had failed.

Technical modifications and enhancements to the rule-based reasoning component of the CORMS AI system are relatively infrequent. The enhancements made to this component was to make it more stable, and to develop a robust error handling and logging capability to aid the CORMS AI project team in trouble shooting any problems.

Based on the successes of CORMS AI during the time it has been operational, CO-OPS plans to incorporate many enhancements into the system. Most notably, more rules will be developed to ensure CORMS AI is performing a full range of quality checks by looking both back over time and forward against predictive values to determine that relative rates of change for specific environmental phenomena occur within expected ranges for a given geographic area. CO-OPS also plans to use CORMS AI to monitor samples received from co-located redundant/backup sensors to determine in real-time which, if either sample should be ingested and disseminated to the public. CO-OPS also plans to expand the rule base so that the samples will be evaluated differently based perhaps on the model of sensor making the measurement, the geographic location of the sensor, or expected seasonal affects. As more rules are implemented, additional cases and perhaps different case-bases need to be developed to assist in the trouble shooting process.

Modifications to the rule and case base present the most challenging maintenance issues for CORMS AI. Policies and procedures have been written to facilitate the updating of the rules and cases. In both situations a committee of experts is to agree on the proposed changes.

9. Summary and Conclusions

CORMS AI was implemented using rule-based and case-based technologies. The rules are used to identify failure within the monitoring system and the case bases are used to aid the experts in identifying the source of the problem and remedying the situation.

CORMS AI performs well enough to be used as the sole method of data quality control for the National Ocean Service in the future. The system accurately and reliably identifies suspect data and network disruptions, and has decreased the amount of time it takes to identify and troubleshoot sensor, network, and server failures. CORMS AI has proven to be robust, extendable, and cost effective.

Based on the successes of CORMS AI during the time it has been operational, CO-OPS plans to incorporate many enhancements into the system. Most notably, more rules will be developed to ensure CORMS AI is performing a full range of quality checks by looking both back over time and forward against predictive values. As more rules are implemented, additional cases and perhaps different case-bases need to be developed to assist in the trouble shooting process. With the full expansion of PORTS® it is anticipated that CORMS AI will substantially increase CO-OPS' ability to effectively support NOAA's mission goal of ensuring safe, efficient and environmentally sound maritime commerce while saving the United States government over one million dollars per year.

References

[1] Watson, I, "Case-Based Reasoning: Techniques for Enterprise Systems," Morgan Kauffman Publishers, July 1997.

[2] www.ai-cbr.org

[3] www.cbr-web.org

[4] PC-AI Expert Systems http://www.pcai.com/pcai/New_Home_Page/ai_info/expert_systems.html#Expert_System_Sites

[5] Commercial Expert System Shells http://www.landfield.com/faqs/ai-faq/expert/part1/section-7.html

[6] Stylianou, A, G. Madey, and R. Smith, "Selection Criteria for Expert System Shells: A socio-Technical Framework," *Journal of Communications of ACM*, vol. 35, Oct. 1992.

[7] Case-Based Reasoning www.aic.nrl.navy.mil/~aha/cbr

[8] Appell, G., T. Mero, T. Bethem and G. French, "The Development of a Real-Time Port Information System, *IEEE Journal of Oceanic Engineering*," Vol. 19, No. 2, 1994.

[9] Tenback, R.H., "A comparison of Similarity Measures for Case-Based Reasoning," Master's thesis, *the Utrecht University*, 1994.

[10] Real Time Tide and Current Data Systems in United States; Implementation Status Safety and Efficiency Needs Expansion Plan - *A Report to Congress*. July.

[11] Vafaie, H., M. Shaughnessy, T. Bethem, and J. Burton, "Selection Process of Intelligent Decision Support Tool for Real-time Monitoring Systems". In proceedings of the twelfth International Conference on Tools with Artificial Intelligence, Washington D.C., *IEEE computer Society*, Nov. 2002.

[12] zeichick, A., and Rice, A., "Improving Decisions and Increasing Customer Satisfaction with Intelligent Systems," *MindBox Internal Report*, July 2000.

[13] Schank R., "Explanation: A first pass", *Experience, Memory, and Reasoning*, J. Kolodner and C. Riesbeck (eds), pp139-165, 1986.