



CLARUS WEATHER SYSTEM DESIGN

ARCHITECTURAL

ALTERNATIVES ANALYSIS

January 2006

Prepared By:



Mixon/Hill, Inc.

12980 Metcalf Ave, Suite 470

Overland Park, Kansas 66213

913-239-8400

All documentation, software, and data related to this project are proprietary and copyrighted. Use is governed by the contract requirements as defined in the U. S. Department of Transportation Federal Highway Administration Contract No. DTFH61-05-C-00022. Unauthorized use of this documentation is a violation of law except as provided for in said contract.

Copyright © 2006 Mixon/Hill, Inc. All rights reserved.

Table of Contents

1	INTRODUCTION.....	1
1.1	Purpose.....	1
1.2	Scope.....	2
1.3	Definitions, Acronyms, and Abbreviations	2
1.4	References.....	2
1.5	Overview	2
2	OPERATIONAL CONCEPT	4
3	ASSESSMENT CRITERIA	6
4	ARCHITECTURAL ALTERNATIVES ASSESSMENT.....	8
4.1	Service Topology	8
4.1.1	Centralized Service.....	8
4.1.2	Distributed Service.....	9
4.2	Data Interchange Methods	10
4.2.1	Polling Process	10
4.2.2	Publish/Subscribe Process.....	11
4.2.3	Notify/Retrieve Process.....	11
4.2.4	Hybrid Push/Pull	11
4.3	Contributor Assignment.....	12
4.4	Georeferencing Methods.....	13
4.5	Station Identification	14
4.6	Data Cache and Repository.....	15
4.7	Message Formats.....	16
4.8	Quality Checking Methods.....	17
	APPENDIX A - DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	19

Table of Figures

FIGURE 1 – CONTEXT FOR ANALYSIS OF ARCHITECTURAL ALTERNATIVES 1
FIGURE 2 – CLARUS SYSTEM PROCESS CONTEXT 4
FIGURE 3 – HIGH LEVEL CLARUS SYSTEM PROCESSES 5

Revision History

Revision	Issue Date	Status	Authority	Comments
01.00	2005.09.29	Review	DTFH61-05-C-00022	Initial release for comment.
02.00	2006.01.26	Final	DTFH61-05-C-00022	Resolved comments.

Electronic File

Saved As: 04037-ak401arc0200.doc

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to provide an assessment of architectural alternatives for key features relative to the needs and design of the *Clarus* system. Alternatives for each desired capability are discussed in terms of their relative strengths, limitations, opportunities, and challenges. This assessment will contribute to an analysis of the gaps between existing surface transportation meteorological system capabilities and the requirements specified for the *Clarus* system.

This document is intended to be read and used by the U.S. Department of Transportation (USDOT) and system development team members. As indicated in Figure 1, the Analysis of Architectural Alternatives is an intermediate deliverable in the larger context of the *Clarus* Weather System Design project, using criteria documented in the High-Level Requirements Specification and architectural component descriptions from the System Architectural Description to identify more specific components and attributes (“potential features”) as input to the Design Gaps Analysis.

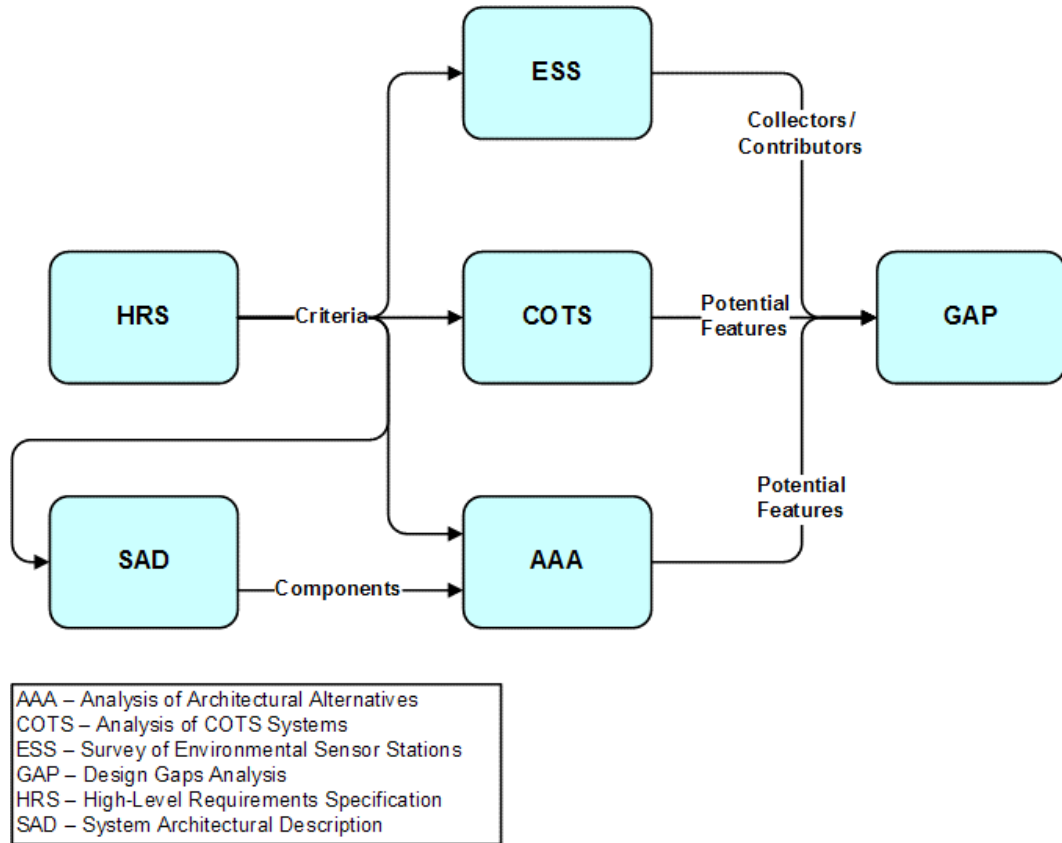


Figure 1 – Context for Analysis of Architectural Alternatives

1.2 *Scope*

This document provides an assessment of architectural alternatives for key features enabling *Clarus* system objectives. *Clarus* will collect weather and pavement condition information from environmental data sensors and mesonets. The system will qualify the environmental information using appropriate quality assessment methods to provide a relative indication of confidence in the information. *Clarus* will then format the qualified environmental information for dissemination to weather (or, more broadly speaking, environmental) service providers and for quality feedback to the environmental data contributors.

Clarus will provide benefits to a diverse group of stakeholders. Observing system owners and environmental equipment manufacturers will use *Clarus* information to improve the reliability and accuracy of their products. Transportation agencies will use qualified *Clarus* information to enhance their decision making in system operations and maintenance. Weather service providers, including the National Oceanic and Atmospheric Administration (NOAA), will use the qualified environmental information to enhance products distributed to the research community and the public.

1.3 *Definitions, Acronyms, and Abbreviations*

This document may contain terms, acronyms, and abbreviations that are unfamiliar to the reader. A glossary of these terms, acronyms, and abbreviations is provided in Appendix A.

1.4 *References*

1. *Clarus Final Draft Concept of Operations*; Iteris and Meridian Environmental Technology, Inc.; May 16, 2005.
2. *Clarus Weather System Design – High Level System Requirements Specification*; Mixon/Hill, Inc.; July 2005.
3. *Clarus Weather System Design – System Architectural Description*; Mixon/Hill, Inc.; August 2005.
4. *Clarus ESS Survey*; Cambridge Systematics, Inc.; December 2005.
5. *Clarus Weather System Design – Systems Engineering Analysis of Clarus-Related Systems*; Mixon/Hill, Inc.; September 2005.
6. *Location Referencing Message Specification (LRMS)*; SAE J2266; Version 1, November 2004.

1.5 *Overview*

This document provides an assessment of the relevance of existing software systems to the needs and design of the *Clarus* system. User needs for the system, upon which this architecture is based, are documented in the Concept of Operations (ConOps) and further developed in the High-Level System Requirements. The high-level design is documented in the System Architectural Description.

The remainder of this document consists of the following sections and content:

Section 2 – Operational Concept provides a description of the *Clarus* system.

Section 3 – Assessment Criteria describes the basis against which the various existing systems are to be assessed.

Section 4 – Assessment of Architectural Alternatives provides an assessment of architectural alternatives for key features of the *Clarus* system.

2 OPERATIONAL CONCEPT

The *Clarus* ConOps provides extensive discussions of the operational context, objectives, constraints, and system functions. These concepts are illustrated through discussion of an overall framework and operational scenarios for various user communities. Operational characteristics of the *Clarus* system itself are a subset of the overall framework and scenarios. The processes to be implemented in the *Clarus* system have been distilled from the framework in the ConOps and are shown in Figure 2 and Figure 3 below. This description focuses specifically on those functions to be fulfilled by the *Clarus* system and generalizes the interfaces based on the data types (rather than source types).

From the overall system perspective, the *Clarus* system will take in environmental data and metadata, and provide environmental metadata and qualified environmental data on request. The system will perform these operations based on data sharing agreements that define the terms of access and on quality control parameters used in assessing the incoming data. The system will need access to the environmental data networks and servers, and will need to provide network access for users requesting information.

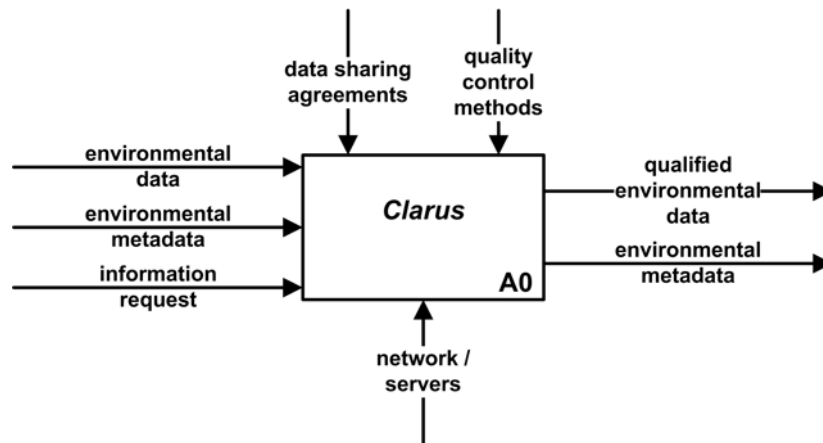


Figure 2 – *Clarus* System Process Context

Within the overall context, the *Clarus* system will collect, assess the quality of, and disseminate environmental data. The collection process locates, obtains, and stores the data in a common data structure, subject to access agreements. The quality control process applies one or more quality checks and associates quality flags with the data. The qualified data and the associated metadata are then available for dissemination, subject to any constraints specified in the data sharing agreements.

There will be multiple sources of data for the collection process, each potentially in its own format. Each source of data will also provide metadata describing the source and conditions surrounding the source. Terms under which data can be accessed from each source will be identified in data sharing agreements with the source organizations. Data are collected from the sources, interpreted from the source formats, and stored in a common data structure.

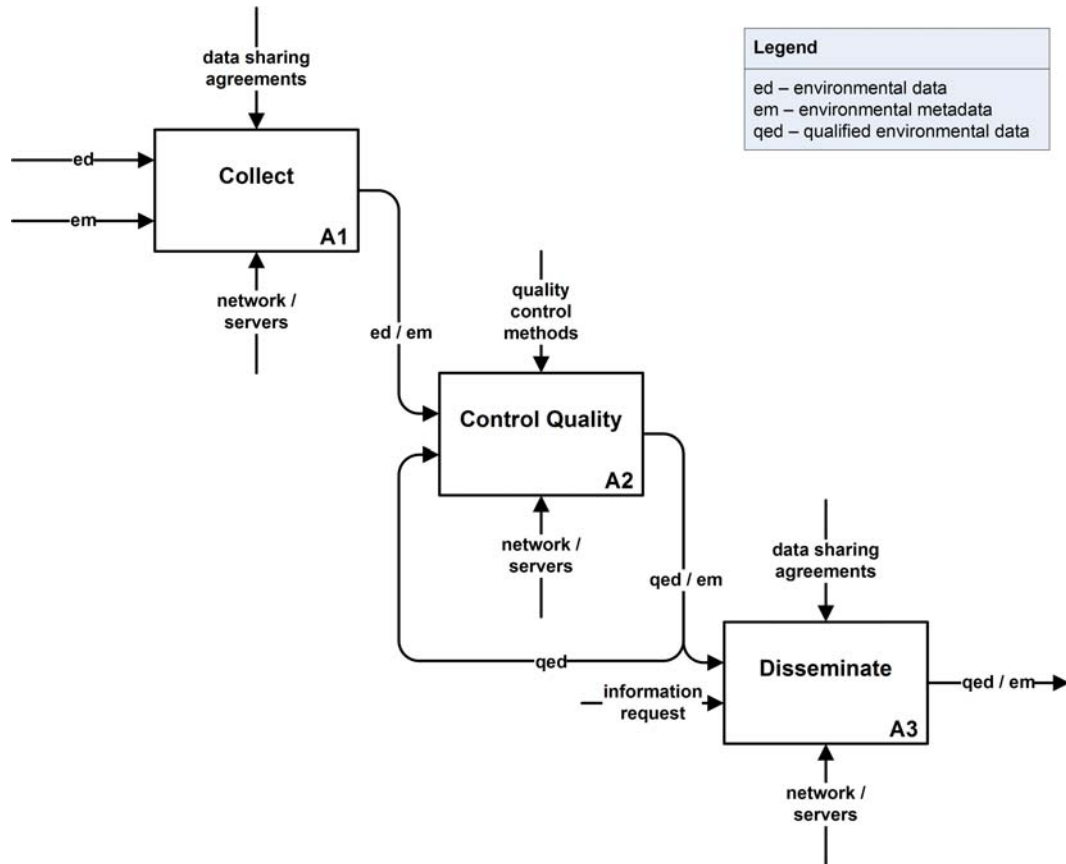


Figure 3 – High Level Clarus System Processes

The quality control process will implement one or more quality checks of the environmental data. Each quality check will be based on a set of rules for comparing the data to other models and data sets. Comparison data sets can include previously qualified data shown as a quality feedback loop in Figure 3. It may be necessary to derive or infer additional data from the original environmental data in order to complete some quality checks. Quality flags are assigned to the data according to the specific checks performed.

The data are disseminated in response to information requests directly from users with access to the data or automated processes based on data values and quality flags. In particular, data source organizations may be notified of data sets that do not pass particular quality checks. Data sharing agreements may constrain the data sets, formats, or distribution lists for data dissemination.

These essential Clarus functions provide the context for assessment of architectural alternatives to key features relative to Clarus needs and system design.

3 ASSESSMENT CRITERIA

The value of any analysis depends on the care used in identifying criteria on which to base the assessment. Bad decisions are just as frequently made from inappropriate (or incomplete) criteria as they are from erroneous information. The body of criteria should consider the full context and scope of the assessment. In addition, care should be taken to assure that the criteria are based on needs, and not on abstracted descriptions of a solution. Objective criteria describe what the system needs to do, but do not prescribe what solutions might do it.

In this case, assessment criteria must address the *Clarus* operational concept, as captured in the high-level system requirements. These requirements provide a broad view of the context and scope of *Clarus* in terms of what *Clarus* will do, under what controls or constraints it will do it, and generally with what resources it will do it.

Requirements on the *Clarus* system contained in the High-Level Requirements Specification (Ref. 2) fall into four broad categories: functional, data, interface, and performance. Existing systems will be qualitatively assessed in these areas to gauge relative applicability to *Clarus* objectives.

- Functionally, *Clarus* will:
 - gather observations from fixed and mobile environmental data “collectors” across North America;
 - provide continuous quality control (i.e., assessment) and flagging of the data through a variety of methods, logging the quality control process;
 - disseminate data on request or by subscription, according to pre-established data sharing agreements; and
 - administer the process by managing access, issuing change notices, keeping logs, and retaining the environmental data according to the data sharing agreements.
- *Clarus* data will:
 - be based on the NTCIP, TMDD, and CMML standards;
 - include atmospheric weather, surface weather, and hydrologic data;
 - include sensor metadata; and
 - include location, time and date stamp, and source for all observations.
- *Clarus* interfaces will:
 - be based on industry standards;
 - include provisions for direct manual data entry;

- disseminate data based on a variety of query techniques, subject to data sharing agreements;
- respond to dissemination requests on one-time and subscription bases; and
- allow for system administration.
- *Clarus* will be designed to:
 - minimize latency in data collection, quality assessment, and dissemination;
 - scale to 470 million *current* observations;
 - scale to 600 concurrent users; and
 - scale to 6000 registered users.

As discussed earlier in this section, these criteria describe what the system will do, not how it might be designed to do it. The *Clarus* System Architectural Description documents perspectives on how the *Clarus* system could be designed to meet the requirements, but it is not itself a formalized design. The architecture is therefore used in this document only as a basis for structuring the discussion of the alternatives.

4 ARCHITECTURAL ALTERNATIVES ASSESSMENT

This section discusses eight areas of concern with the *Clarus* system for which architectural alternatives are to be considered. The areas discussed include service topology, data interchange methods, contributor assignment, geo-referencing methods, station identification, data cache and repository, message formats, and quality checking methods. Each area is evaluated on its applicability and impact on the *Clarus* system with a description of its capabilities and implementation risks.

4.1 *Service Topology*

Network *topography* is a description of physical structure as it relates to communication components such as cables, switches, and routers. Star, tree, bus, ring, and combinations thereof are a few examples of network topographies. In contrast to network topography, network *topology* describes the logical structure of the network as it appears to connected systems.

With the advent of modern, logically switched communication networks, such as the Internet, network topography has become less important to application design. Connected systems need not be concerned with the underlying communication infrastructure. For the purposes of this discussion, the *service topology* describes the logical arrangement of *Clarus* services similar to the way network topology describes the logical arrangement of communication components.

Alternatives for the *Clarus* service topology are bracketed by two options: a centralized service, in which all *Clarus* functions are fulfilled in a central communications node, and a distributed service, in which the *Clarus* functions are logically distributed. For the purposes of this discussion, intermediate solutions that distribute some *Clarus* functions are included in the distributed alternative.

4.1.1 Centralized Service

The centralized service alternative for a service topology appears familiar because of its similarity to a star topography, even though the physical location of the *Clarus* equipment is irrelevant. In this alternative, the *Clarus* system would have a single communication point into which all raw environmental data and metadata would flow. This same point would qualify all the received information and respond to all incoming information requests.

A centralized system requires sufficient communication bandwidth to carry all of the incoming data and queries, as well as every outgoing data response. The data store would need enough capacity to maintain seven days' inventory of environmental data and associated metadata. The centralized system must have enough processor power to import and qualify the entire raw environmental data stream, and to respond to all incoming requests for information. The limiting factor in all these considerations is the volume of observations to be processed, which is directly related to the geographic distribution and resolution of those observations.

Environmental conditions and their impacts are area effects, and the observation data needed to characterize those conditions will be of commensurate scale and complexity. A single centralized system would therefore need to accommodate the qualification of data from hurricane-prone areas in the southeastern U.S. and winter storm data from the western mountain states. The centralized *Clarus* system would be responsible for configuring and calibrating the entire range of quality checking algorithm sets to qualify the data variety it receives.

4.1.2 Distributed Service

A distributed service topology is arranged along a bus or backbone where the individual systems can be spatially distributed but coupled together by a standard network infrastructure. This service configuration would contain multiple *Clarus* systems that work independently to acquire and quality control incoming environmental data, but makes the information available to all *Clarus* nodes. Although each such *Clarus* system (or node) could be similar in size and scope, the management of interactions between nodes becomes more complex as the number of nodes increases.

A distributed service topology allows for each *Clarus* system to be optimally tuned for processing environmental data characteristic of its specific region. For example, coastal areas could use quality checking algorithms that assess tidal effects.

The data processing load is likewise reduced at each individual *Clarus* installation. Only the environmental data from the established area for a particular *Clarus* installation needs to be retrieved. Even though the quantity of data requests would remain similar to the centralized approach, the requests would be reduced at each *Clarus* installation. For example, parties in the southeastern United States will infrequently request qualified environmental data for northwestern Canada.

More communication infrastructure is used in a distributed network. Each *Clarus* system would need its own network connection. This improves the *Clarus* network's overall resilience and availability. The expense for the additional communication infrastructure will be similar to obtaining the large amount of bandwidth needed with a centralized approach.

A distributed service topology does create one issue not present in a centralized system: how interested parties find specific information from among the distributed services. The Internet, which is itself an example of a distributed services topology, uses a network of domain name servers to resolve human-readable names to their logical addresses. The *Clarus* system could adopt a similar approach. The *Clarus* program could maintain a registry service—similar to a telephone book—that end users may query for *Clarus* system addresses and the geographic areas they cover. The *Clarus* program could also work with COTS products that can fulfill the same functionality, although the cost of these products is likely to be prohibitive.

The registry service is a system supporting administrative and management aspects of the *Clarus* program. As a system, the registry service also has the

options of adopting a centralized or distributed topology. In this case, distribution adds more complexity than necessary for a registry service to be successful. A single registry service contact point with the requisite communication, load balancing, failover, and backup infrastructure could easily handle all of the *Clarus* system lookup needs for the medium term. An initial centralized solution is easier to implement and maintain, and does not prevent the future distribution of the registry service should it become necessary.

4.2 *Data Interchange Methods*

The *Clarus* system collects environmental metadata and unqualified environmental data, performs quality checks, and disseminates environmental information to interested parties. Data acquisition methods are relevant to both the collection and dissemination processes. This section discusses alternative methods of implementing the exchange of data at the collection and dissemination interfaces.

Any interchange of information between two parties involves a minimum necessary set of conditions and messages. For example, there has to be some physical means of communication between the parties. Likewise, both parties have to be aware of and equipped to send and receive the messages. The alternatives discussed in this section assume the existence of those minimum conditions, and focus on the specific sequence of steps needed to complete an exchange.

4.2.1 **Polling Process**

For data input, a polling process would use an available list of environmental sources with their publishing format. The process would then request information from each source and convert the response based on its published format into an internal storage format for quality checking and dissemination. If no new information is available, the response can indicate there is nothing available or the *Clarus* system can discard the redundant information.

The data output polling process is similar to the data input polling process. Interested users know where to request information, their desired format of the information, and an estimate of the data update frequency. At the appropriate time, requests for information are sent to one or more *Clarus* systems indicating the desired output format. The *Clarus* system applies the requested filters to the available data, formats the data into the desired output, and sends the response to the requestor. The user can determine if information is fresh and discard the redundant data.

A polling process has some advantages over the two methods discussed below. It is easier to implement and is somewhat more secure. If the *Clarus* system were open to having sources publish directly to it, then it would be easier to flood the system with erroneous information. Polling for information helps with this situation as only responses to pending requests are expected. Everything else can be ignored. In the HTTP communication cases, the channel can use the HTTPS protocol and secure the information transfer with asymmetric key certificates.

4.2.2 Publish/Subscribe Process

In a publish/subscribe process, the entity that wants information builds a subscription that contains a destination, data formatting criteria, and filtering criteria. The subscription is then sent to the information publisher and is stored until it expires, is revoked, or is changed. When information at the publisher is updated and meets criteria in a subscription, the information is packaged as requested and sent to the destination. Subscription criteria can be time-based, event-based, or both.

The *Clarus* system data input process using a publish/subscribe method would use a configured list of environmental information sources to build subscriptions and send them to the sources with itself as the recipient. In this case, the *Clarus* system still knows the incoming data format and extracts the data itself, rather than requiring the originator to format the response first. The publishing environmental data contributors will then fulfill the subscriptions and transmit data to the destination *Clarus* system as needed.

Environmental data users could also retrieve information from the *Clarus* system using a subscription. The users would create a subscription request containing the query information for the environmental data or metadata and send the subscription to the appropriate *Clarus* system. When new information that meets their criteria is available, the *Clarus* system formats the information for the users and transmits it to them.

A publish/subscribe data acquisition method has the advantage over a pure polling method of reducing redundant data and unnecessary requests, freeing communication and processing capacity for other purposes.

4.2.3 Notify/Retrieve Process

A variation on the publish/subscribe method is notify/retrieve. Rather than building a subscription request that can be complex and difficult to define, a simple notification request is used instead. The notification request indicates an interest in information when the information is updated, on a schedule, or both. The notification request is similar to a subscription, but without the need for filtering attributes.

The system managing environmental data (*Clarus* or some other contributor) sends a small notification to each party with a registered notification request. The notified recipient then has the option to retrieve the data on their own independent schedule. This method is similar to a magazine or newspaper being delivered to a mail box or driveway. People who live in the house can see that their materials have arrived and can choose to retrieve it at their convenience.

4.2.4 Hybrid Push/Pull

There are two disadvantages to the publish/subscribe and notify/retrieve data interchange methods. The first is that if a subscription or notification response is lost, then the intended recipient has no way of knowing that data considered important to them was available. The second disadvantage of either interchange method is that a contributor may be forced to implement a different protocol than

is used in their existing system. This may prevent contributors from joining the *Clarus* program.

In contrast with these interchange methods, a pure polling process knows that it is receiving the latest possible information. This occurs at the expense of communication bandwidth and processing resources to determine if the received information is new. None of the proposed methods by themselves is the best solution.

A hybrid solution can be implemented to balance the advantages and disadvantages of each data interchange approach. The *Clarus* system should always implement a polling data interchange. This is the easiest method to implement and enables the *Clarus* system to communicate with contributors that are unwilling or unable to alter their systems to support a subscription or notification protocol. The *Clarus* system should also implement the notify/retrieve interchange method. Building upon the existing polling implementation, this method simplifies the subscription request. Contributors wishing to take advantage of the notification mechanism may do so at their convenience. The hybrid approach ensures that the *Clarus* system will have the latest environmental information. The polling process will compensate for lost notifications at intervals that do not impact bandwidth or processing excessively.

4.3 *Contributor Assignment*

Clarus' effectiveness in providing a clear picture of surface transportation weather conditions will be driven in large part by the quantity and quality of information it can provide. Since that information does not originate within *Clarus*, but is collected from contributors, *Clarus'* success depends largely on its ability to draw data from a large pool of contributors with high-quality data. The process for identifying and enlisting contributors is therefore essential to the *Clarus* program and has design consequences for the *Clarus* system. That process has two initiating alternatives: contributors can either be self-appointed or nominated by the *Clarus* program.

A self-appointed contributor approach is appealing because it opens the *Clarus* system to a diverse set of data collection opportunities, providing a wealth of data for quality assessments. At the same time, complete openness would allow anybody who knew how to contact a *Clarus* system to indiscriminately register sensor stations and to send questionable information. Even if questionable data is unintentional due to poor maintenance or configuration, deliberate attempts to undermine the *Clarus* system's usefulness are much more easily perpetrated in this alternative.

Other organizations have proposed a rating system for evaluating managed sensor networks. The ratings categorize the effectiveness of maintenance and administration of a particular sensor network. The ratings also serve as a baseline configuration reference for a contributor to implement for achieving a desired rating. NERON, for example, has proposed a three level rating system: bronze, silver, and gold.

The *Clarus* program could implement a similar rating system and integrate it with the data sharing agreement processes. By creating evaluation criteria and placing administrative oversight on the contributors, those genuinely interested in participating in the *Clarus* program will gain value from the qualified data while meteorological amateurs and miscreants will be deterred.

4.4 *Georeferencing Methods*

Georeferencing establishes relationships between geographic locations and data associated with those locations. The language and absolute references used in a particular georeferencing context are not necessarily tied to an independent standard reference, and can vary even for the same geographic locations. Within the transportation community, for example, different jurisdictions may use different georeferences for the same highway location. While the Location Referencing Message Specification (LRMS) (Ref. 6) provides several alternative schemes for georeferencing, it does not mandate the application of any particular scheme to any particular architectural context.

The *Clarus* system can execute its quality checking processes on sets of sensor data regardless of their actual physical location as long as the algorithms understand their relative positioning. However, for the qualified environmental data to be useful for others, a generally agreed upon location standard will need to be adopted.

Departments of Transportation and other organizations responsible for maintaining the transportation infrastructure frequently use roadway names and mile markers. Mile markers typically begin at one state border and progress sequentially to the opposite state border. County planners have their own similar set of mile marker references that begin and end on county boundaries. Cities can log and maintain their own location reference sets as well.

Route numbers and mile markers work well as local or personal references because the descriptions are based on a specific contextual reference. Unfortunately, descriptive names are not easily mathematically transformed or translated to other location referencing schemes. There are standards that attempt to constrain and apply a repeatable and expected pattern to naming mechanisms, but most systems rely on a library of named locations within a particular context to resolve locations with respect to a more standard reference.

The Global Positioning System (GPS), originally deployed in 1978 as a navigation aid for sea vessels assigns latitude, longitude, and elevation in a spherical coordinate arrangement on the earth's surface. Longitude values are measured east and west and are between -180 and 180 degrees with the zero point being at the prime meridian in Greenwich, England. Latitude values are measured north and south and are between -90 and 90 degrees with the zero point being the equator.

GPS has evolved over the last twenty-five years into an indispensable tool for many industries, but is not entirely perfect. For example, there are slight variations in the spheroid model used to calculate the coordinates. It is possible,

however, to automate the translation between the differing model coordinates if needed.

To be successful, *Clarus* system contributors must agree upon the location referencing scheme. GPS coordinates are the logical choice due to their wide availability, easy translation between systems, usefulness in quality checking algorithms, and ease of presentation. The *Clarus* program will need to define the minimum precision required to describe an observation location in the program standards.

4.5 *Station Identification*

Station identification is important to the *Clarus* system because the source of an observation is a required metadata element. Every source identifier must be unique in order to ensure the integrity of the metadata. These observation sources are essential to linking observations with the relevant environmental sensor station (ESS) metadata, particularly when observation data is identified as having a quality issue. The potential errors can be reported back to the owning contributors to assist them with maintaining and troubleshooting their equipment.

It is reasonable to expect that contributors will want to maintain their own naming conventions for their equipment. The *Clarus* system's primary goal is to be able to uniquely identify the source of an observation without two different sources attempting to use the same name.

One method is for the *Clarus* system to generate unique identifiers—uniformly unique identifiers (UUIDs)—and assign them to contributors to use in their communications with the system. While this ensures uniqueness and usability by the *Clarus* system, it puts the burden of configuration control and namespace mapping on contributors that may not have the resources or willingness to implement it.

Alternatively, the *Clarus* program could establish a standard naming practice for deriving unique names. For example, a five-character alphanumeric name using two characters for the state abbreviation, two for the county abbreviation, and one as a sequence could be proposed. The rules for generating unique names could become complex, however, and, would not necessarily guarantee uniqueness in a heterogeneous network with a variety of contributors using existing jurisdictional boundaries as references. As with *Clarus* program assigned identifiers, a burden is placed on the contributor to name their equipment properly which may not be conducive for them to maintain their internal processes.

Contributors are likely to have established their own naming conventions for management and identification purposes similar to those needed for the *Clarus* system. A better implementation alternative is to allow contributors to submit their set of source names to *Clarus*. The system will then assign its own unique identifiers for tracking purposes along with the associated contributor station identifiers.

This alternative establishes a lookup mechanism for communicating with a contributor about their sensors. The *Clarus* system can internally manipulate

compact identifiers that conserve computing resources while tracking the association to a contributor. If feedback is necessary, the associated identifier information can be used to determine the owning contributor and identify the potential sensor station issues using naming conventions that leverage the contributor's existing management processes.

4.6 *Data Cache and Repository*

Alternatives for storing the environmental metadata and data do not involve discussing whether there will be a database or some other storage method involved. With the quantity and variety of information flowing into and out of the *Clarus* system, a database is clearly needed. The alternatives discussed in this section consider using an open source database as opposed to a commercial relational database product.

Software interfaces to relational database systems have been standardized for many years and typically allow open database communication (ODBC) connections and structured query language (SQL) commands. Database vendors typically have their own proprietary communication protocols, but they also support ODBC and other similar standards. Stored procedures—precompiled queries—are generally not portable, but the languages and syntax can be converted between systems if needed. Not all database products support stored procedures and they aren't required for the *Clarus* implementation, but may be beneficial from a performance standpoint.

Microsoft Access is widely used and on the low end when dealing with the pricing of commercially available databases. An Access database's performance is not likely to meet the *Clarus* system needs, however, and its maximum two gigabyte (GB) database size limit is definitely too small to accommodate the expected volume of environmental information.

The next step up in the commercial database cost spectrum is Pervasive. Pervasive has very good performance, storage characteristics, and features. Its fee structure is based on how many simultaneous database connections are used in the application. Pricing begins at approximately \$150 per connection and drops to \$100 or less per connection with volume licensing. This does not necessarily drive up the cost of a *Clarus* system deployment, as the pool of available database connections is shared among the executing processes. This design reduces the total number of connections and, consequently, the overall installation cost.

Open source databases are appealing for their low cost. Open source licensing is often free, but carries fees when employed in an enterprise system such as *Clarus*. PostgreSQL, Ingres, and MySQL are open source databases with licensing (support) fees of approximately \$2000 per server per year. The performance of these products is adequate for the *Clarus* system purposes. Specific features and support levels will be investigated more thoroughly when the detailed *Clarus* system requirements become available.

Microsoft SQL Server is another lower-cost commercially available enterprise class database. This software's latest edition, SQL Server 2005, has several different licensing models. The per-processor model is designed to allow

connections to the database to be multiplexed for applications such as *Clarus* and is the most appropriate choice. SQL Server 2005 is approximately \$19,000 per processor. Sybase is a direct competitor to Microsoft SQL Server at this price point.

Oracle is a more expensive commercial database. Performance, storage limits, and features are not issues with this product. Oracle is known for its complex licensing structure and higher cost, which may reduce its desirability for the *Clarus* system.

The relational database management system's base cost is not the only factor in determining overall data storage costs. One architectural alternative is to operate multiple databases, each containing a replicated copy of the current data. This configuration maximizes system performance by having a database and processor dedicated to fulfillment of each process such as data acquisition, quality checking, or dissemination.

4.7 *Message Formats*

As mentioned in the Section 4.3 discussion of contributor assignments, the success of the *Clarus* program ultimately depends on its ability to collect data from a broad group of contributors. The program is thereby motivated to remove any obstacles to contributing data, especially where the data would come from well-maintained sensor networks. Architecturally, the *Clarus* system must have reliable network connections to those contributors and must be able to understand the messages sent by the data collectors.

Unfortunately, there are nearly as many data formats as there are potential contributors. As described in Reference 5, *Analysis of Clarus-Related Systems*, other environmental data collection networks have dealt with this issue in varying ways. The Oklahoma Mesonet, for example, uses a very homogeneous set of collectors with a correspondingly selective range of formats. The Canadian RWIN system has prescribed its own CMML message format to homogenize the messages to be interpreted at the RWIN servers. NOAA's MADIS program works with a very heterogeneous set of contributors and has been successful in supporting a large number of data formats. The *Clarus* program has an opportunity to make a deliberate decision that balances the need to be as liberal as possible in accepting data in a variety of formats with the responsible management of program resources required to support those formats.

The modular services-oriented architecture of the *Clarus* system is intended to make the system easy to maintain as the user needs change. The system can therefore support as many input and output format modules as needed to interact with contributors and users of *Clarus* data. This situation has the potential of burdening the *Clarus* program with the entire expense of assuring system interaction and could create a software configuration and distribution mess. This is particularly important in an institutionally and geographically diverse deployment.

At the other end of the spectrum, *Clarus* could constrain the input and output formats to a few modules. The choice of which formats to support would be

driven by the volume and quality of data made available in each particular format, subject to any specific *Clarus* requirements on format and content. This reduces the system development and maintenance costs, but may discourage contributors and users if they have limited funds to make changes to their systems.

A balanced solution could benefit both the *Clarus* program and the meteorological contributor and user community. A community survey could be performed to determine the top choices for input and output formats. Results of the survey would be combined with data from the ESS Survey and the *Clarus* requirements to guide the selection and design of the data input and output services. NetCDF and HDF can serve as both input and output formats, while SHEF is an input format, and CMML is an output format.

A balanced implementation could also develop input and output format processor modules that would accept and produce flexible data formatting. These flexible ASCII—FlexI, for short—text files would, for example, look like comma separated variable (CSV) formats with a column header description. Such modules could be configured to change the expected data order, units, labeling, and header presence for either input or output processing.

Potential contributors appear to generally support some form of an ASCII text format. This alternative allows them to leverage their existing systems while migrating to a more standard format if so desired. The configuration responsibility can be shared between a *Clarus* system and the contributor community.

The *Clarus* system could easily include the expected ASCII data format with the information used to retrieve environmental data from a contributor. The ASCII format would then be used by input or output formatting modules to reorganize the data appropriately. A contributor could also dynamically change their transmitted format by including a header that defines the subsequent environmental data. *Clarus* input modules can then adjust their data parsing accordingly.

4.8 Quality Checking Methods

Quality checking processes are essential to *Clarus* and constitute the bulk of data processing done by the system. A quality checking module encapsulates well-defined mathematical methods operating on environmental data sets to determine the degree to which those observations are reliable. These methods vary widely, from simple range checks to more sophisticated spatial and temporal distribution algorithms. Unfortunately, there are currently no unified standards for the quality control of weather data. In addition, quality checking algorithms are under constant reanalysis and development as new operational experience and meteorological methods become available. This creates an implementation problem for *Clarus*: the system has to accommodate essential methods that may vary over time from one parameter or one data source to another.

Solutions for managing this variability are inherently complex. System implementations occupy a continuum between a few complex components in simple relationships and many simple components in a complex arrangement. Modular architectures attempt to constrain this complexity by encapsulating sets

of methods and defining specific interfaces and interactions for each such encapsulation.

At one extreme of the range of alternatives, the solution would be a single quality checking module that could be given specific instructions for processing any particular data set. A configuration language would need to be developed to interact with the module. This module would be complicated to create and maintain because it would have to contain its own instruction parser and be capable of handling any possible desired meteorological algorithm. In addition, maintaining the library of instructions for the module is inherently difficult. Since it is impossible to accurately predict *Clarus*' future quality checking needs, the module's robustness and efficiency may decline over time.

At the other extreme, modules that perform basic mathematical functions could be assembled to perform more complex quality checking functions using a scripting engine. This alternative creates the need for a script parsing module to interpret the desired interactions. The quality checking modules remain simple but their interaction complexity is pushed onto the scripting engine. Each module will likely have high performance, but scripting engines (even compiled ones) may have poor enough performance to not be acceptable in the *Clarus* system.

A balanced alternative may help reduce both the complexity and improve performance. Quality checking modules, each implementing a complete quality checking algorithm, could be orchestrated in a particular sequence. While each quality checking module might have code in common with other modules, consistency is maintained through class inheritance and shared software libraries. Keeping the sequencing module simple reduces the scripting complexity and performance impacts, while the quality checking modules maximize their performance internally.

APPENDIX A - DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

The following table provides definitions of terms, acronyms, and abbreviations to assist interpretation of this document. *The IEEE Standard Dictionary of Electrical and Electronics Terms* [B2], IEEE Standard 610.12-1990, or IEEE/EIA Standard 12207.0-1996 may be referenced for terms not defined here.

Term	Definition
AASHTO	American Association of State Highway and Transportation Officials
Acquirer	An organization that procures a system, software product, or software service from a supplier. (The acquirer could be a buyer, customer, owner, user, or purchaser.)
AMS	American Meteorological Society
API	Application programming interface. A well-defined set of functions commonly used by software to interface with libraries of reusable algorithms.
Architect	The person, team, or organization responsible for systems architecture.
Architecting	The activities of defining, documenting, maintaining, improving, and certifying proper implementation of an architecture.
Architectural Description (AD)	A collection of products to document an architecture.
Architecture	The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.
ASN	Abstract Syntax Notation
ASOS	Automated Surface Observing System
ATIS	Advanced Traveler Information Systems. A description for a group of interoperable services that enable dynamic management of transportation infrastructure and related activities. The project scope of the ATIS committee is to develop a minimum set of medium-independent messages and data elements needed by potential information service providers to deploy ATIS services, and provide the basis for future interoperability of ATIS devices.
CAP	Common Alerting Protocol. An open, non-proprietary standard data interchange XML format that can be used to collect all types of hazard warnings and reports locally, regionally and nationally, for input into a wide range of information-management and warning dissemination systems.
CCTV	Closed Circuit Television
Clarus	The Clarus system. An environmental data sharing system that collects, evaluates, and disseminates environmental data gathered from a geographically diverse set of environmental sensors.
CMML	Canadian Meteorological Markup Language
Collector	An electronic device used to convert environmental sensor electrical signals into environmental condition measured values and store them for retrieval.
ConOps	Concept of Operations

Term	Definition
Contributor	A managing agency or organization that owns and/or operates a set of environmental sensor collectors.
CSI	Cambridge Systematics, Inc.
DATEX	Data Exchange. A European standard effort for center-to-center data exchange.
DMS	Dynamic Message Sign
DOT	Department of Transportation
DSS	Decision Support System
ED	Environmental data. This data has not been processed by any quality checking algorithms.
EDR	An environmental data request sent to retrieve available environmental sensor information.
EM	Environmental metadata. Information about an environmental sensor station.
EMR	Environmental metadata request. A data request sent to retrieve information about environmental sensor stations.
ESS	Environmental Sensor Station
FAA	Federal Aviation Administration
FHWA	Federal Highway Administration
GRIB2	GRIdded Binary data format, Edition 2. An information format used to transmit grid-based weather forecasts from contributing offices to the NDFD and also one of the primary forms used to transmit the NDFD grids to weather information customers and partners
HAR	Highway Advisory Radio
HDF	Hierarchical Data Format
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol. A communication standard for transmitting and receiving documents and other types of data over the Internet.
HTTPS	Secure Hyper Text Transfer Protocol.
ICC	(Clarus) Initiative Coordinating Committee
IMT	(Clarus) Initiative Management Team
in situ	From Latin, “in situ” means “in place.” As applied to meteorological data, it refers to data specific to a (fixed) point of observation.
ISP	Information Service Provider
ITE	Institute of Transportation Engineers
ITS	Intelligent Transportation System
ITS America	Intelligent Transportation Society of America
Life Cycle Model	A framework containing the processes, activities, and tasks involved in the development, operation, and maintenance of a software product, which spans the life of the system from the definition of its requirements to the termination of its use.

Term	Definition
MADIS	Meteorological Assimilation Data Ingest System
MDSS	Maintenance Decision Support System
MDT	Mobile Data Terminal
Metadata	In common information systems use, “metadata” is “data about data.” Within the meteorological community, this use has been extended to include data about objects related to weather observations. For example, location data for an ESS becomes metadata for the observation data.
MHI	Mixon/Hill, Inc.
MS/ETMCC	Message Set for External Traffic Management Center Communication.
NASA	National Aeronautics and Space Administration
NDFD	National Digital Forecast Database. A database supported by the National Weather Service that contains gridded forecasts of several ground-based weather elements such as temperature, humidity, and chance of precipitation.
NetCDF	Network Common Data Form
Network topography	A description of physical structure of a network as it relates to communication components such as cables, switches, and routers.
Network topology	A description of the logical structure of a network as it appears to connected systems.
NHI	National Highway Institute
NIST	National Institute of Standards and Technology
NOAA	National Oceanic and Atmospheric Administration. United States National Oceanic and Atmospheric Administration. A governmental administrative body responsible for managing programs and resources for weather and oceanographic science.
NTCIP	National Transportation Communications for ITS Protocol
NWOS	National Weather Observing System
NWS	National Weather Service
OCS	Oklahoma Climatological Service
OMB	Office of Management and Budget
PDA	Personal Digital Assistant
PMP	Project Management Plan
QC	Quality Checking.
QED	Qualified Environmental Data. Environmental data that has been evaluated by quality checking algorithms and contains a quality assessment flag.
QEDR	Qualified Environmental Data Request. A data request from environmental service providers and contributors to retrieve qualified environmental data from <i>Clarus</i> for value-added product delivery and quality feedback purposes.

Term	Definition
RWIS	Road Weather Information System. A unique system consisting of many meteorological stations strategically located alongside highways that allow the state Departments of Transportation to make more informed decisions during storms. Specialized equipment and computer programs monitor air and pavement temperature to make forecasts regarding how the weather impacts the operation and maintenance of the highways.
SAE	Society of Automotive Engineers. A group of engineers, business executives, educators, and students from more than 97 countries who share information and exchange ideas for advancing the engineering of mobility systems.
SAE	Society of Automotive Engineers. A group of engineers, business executives, educators, and students from more than 97 countries who share information and exchange ideas for advancing the engineering of mobility systems.
SEP	System(s) Engineering Process
SHEF	Standard Hydrologic Exchange Format
STWDSR	Surface Transportation Weather Decision Support Requirements
STWSP	Surface Transportation Weather Service Provider
System	A collection of components organized to accomplish a specific function or set of functions.
System Architectural Description (SysAD)	A collection of products to document a system's architecture.
TBD	To Be Determined
TCP/IP	Transmission Control Protocol/Internet Protocol.
TMC	Traffic Management Center
TMDD	Traffic Management Data Dictionary.
TOC	Traffic Operations Center
TRB	Transportation Research Board
UML	Unified Modeling Language
USDOT	U.S. Department of Transportation
UTC	Universal Time Code
View	A representation of a whole system from the perspective of a related set of concerns.
Viewpoint	A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.
VII	Vehicle Infrastructure Integration
VSL	Variable Speed Limit
WIST	Weather Information for Surface Transportation

Term	Definition
WRS	Weather Response System. A regionally-based service that electronically collects and processes weather forecast information into a coherent presentation for the purposes of traffic management and roadway maintenance.
XML	eXtensible Markup Language. A flexible text markup language used to create standard information formats that share both the format and the information to enable the interchange of structured data.