Copy No._______

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Proposed Architecture for Data Sharing in the Networked Underwater Warfare Project

Anthony W. Isenor

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2005-159
January 2006

Canada

This page intentionally left blank.

Proposed Architecture for Data Sharing in the Networked Underwater Warfare Project

Anthony W. Isenor

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2005-159
January 2006

Principal Author

Anthony W. Isenor
Defence Scientist

Approved by

David G. Hazen
Head, Technology Demonstration Section

Approved for release by

Kirk Foster
DRP Chair

Abstract

This work discusses topics related to data sharing and understanding in a network-enabled environment. A critical component of a successful implementation of network-enabled operations (NEOps) will be the client's ability to judge the importance of the received data as well as understand the content of the received data. In the NEOps environment, many clients will enter the network unaware of the available data assets. A discovery process is required for the client to first identify the available resources. Once identified, the client will require information on the structure used to deliver the data to the client. Then, the client will need information on the details of the data items present within the structure. This work proposes an architecture suitable for the discovery and understanding process. The architecture is based on a vocabulary, or dictionary of terms, and a definition of data structures. An example implementation is provided using extensible markup language. The Networked Underwater Warfare Technology Demonstration Project underway at DRDC Atlantic provides an implementation focus for the data sharing concepts presented in this work.

Résumé

La présente étude porte sur des questions liées au partage et à la compréhension des données dans un milieu réseaucentrique. Un élément d'importance cruciale pour le succès de la mise en oeuvre d'opérations réseaucentriques (OR) tient à la capacité du client de juger l'importance des données reçues et d'en comprendre le contenu. Dans le milieu des opérations réseaucentriques, de nombreux clients accéderont au réseau sans connaître les sources de données disponibles. Un processus de découverte est nécessaire pour permettre au client d'apprendre à identifier ces sources. Une fois les données identifiées, le client aura besoin d'information sur la structure suivant laquelle les données lui sont fournies. Puis, il lui faudra de l'information sur les détails des données présentes dans la structure. La présente étude propose une architecture qui permet au client de repérer et de comprendre les données du système. Cette architecture se fonde sur un vocabulaire, un dictionnaire de termes et une définition des structures de données. Un exemple de mise en oeuvre utilisant le langage de balisage extensible est présenté. Le projet de démonstration de technologies (PDT) sur la guerre sous-marine en réseau, en cours à RDDC Atlantique, offre un cadre de mise en oeuvre pour les concepts de partage de données présentés ici.

This page intentionally left blank.

Executive summary

Proposed Architecture for Data Sharing in the Networked Underwater Warfare Project

Anthony W. Isenor; DRDC Atlantic TM 2005-159; Defence R&D Canada – Atlantic; January 2006.

Background

In networked operations, a temptation will exist to provide data over the network by simply placing the data on a network accessible computer. However, data accessibility alone will not meet the needs of network-enabled operations. Data clients need more than simple access to data. Clients must have mechanisms available to judge the importance of the available assets. Part of this judgement process will involve an understanding of the data content.

A critical component of content understanding involves the metadata descriptions that support the content. These metadata will need to address both the structure of the data being provided via the data asset, but also the data items within the structure. Structure descriptions can be addressed using existing computer technologies like extensible markup language. However, the description of data items within the structure involves the definition of vocabularies that define and describe these data items.

Principal results

A viable architecture for identifying and understanding data that exits in a sharing environment is presented. The architecture is described conceptually, as well as demonstrated in a proof-of-concept style using an extensible markup language implementation.

Significance of results

In a network-enabled operation, a systems ability to understand the data asset is critical to the utilization of the asset. Systems that make up the information network, for example the Global Command and Control System (GCCS) or the Joint Consultation Command & Control Information Exchange Data Model (JC3IEDM), will likely share data by applying descriptive tags to the data. These tags will describe the data content, but will likely be based on the vocabulary of the particular system. An architecture that provides the ability to interpret and understand the vocabulary will assist the individual systems when judging the importance of the content.

Future work

The definition and description of data content provides for increased understanding. However, the actual transfer of data through the network will likely be limited by the available bandwidth of the non-physical network. The next effort for the Networked Underwater Warfare Technology Demonstration Project is to develop shared data structures that will take into account the limited bandwidth by minimizing data flow. Possible methods to reduce the flow include prioritization of data delivery and operational context at the time of data request.

Sommaire

Proposed Architecture for Data Sharing in the Networked Underwater Warfare Project

Anthony W. Isenor; DRDC Atlantic TM 2005-159; R & D pour la défense Canada – Atlantique; January 2006.

Situation générale

Lors des opérations en réseau, il sera tentant de simplement mettre les données dans un ordinateur raccordé au réseau. Toutefois, les opérations réseaucentriques exigent plus que le simple accès aux données. Les utilisateurs des données ont en outre besoin d’avoir accès à des outils leur permettant de juger l’importance des ressources offertes, jugement qui implique entre autres la compréhension du contenu des données.

Un élément d’importance cruciale pour comprendre le contenu des données a trait aux descriptions des métadonnées relatives au contenu. Ces métadonnées devront tenir compte non seulement de la structure de l’information fournie par la source de données, mais aussi des éléments de données présents dans la structure. Les descriptions de la structure pourront se faire à l’aide de technologies informatiques existantes, par exemple le langage de balisage extensible. Toutefois, la description des éléments de données présents dans la structure nécessite l’établissement de vocabulaires définissant et décrivant ces éléments de données.

Résultats

Une architecture viable pour l’identification et la compréhension de données présentes dans un milieu partagé est présentée. Cette architecture est décrite de manière conceptuelle et fait l’objet d’une démonstration de validation de principe utilisant une mise en oeuvre du langage de balisage extensible.

Portée

Lors d’opérations réseaucentriques, l’exploitation du système exige la compréhension de la source de données. Les systèmes qui constituent le réseau d’information, par exemple le GCCS (système mondial de commandement et de contrôle) ou le JC3IEDM (Joint Consultation Command & Control Information Exchange Data Model) partageront probablement des données au moyen d’étiquettes descriptives appliquées sur celles-ci. Ces étiquettes décriront le contenu des données, mais seront probablement établies à partir du vocabulaire de chaque système particulier. Une architecture qui permet l’interprétation et la compréhension du vocabulaire aidera les systèmes constituants du réseau à déterminer l’importance du contenu des données.

Recherches futures

La définition et la description des données facilitent leur compréhension. Toutefois, le transfert réel des données sur le réseau sera probablement limité par la largeur de bande disponible du réseau non physique. Les prochains travaux du projet de démonstration de technologies (PDT) sur la guerre sous-marine en réseau viseront le développement de structures de données partagées qui tiendront compte de la largeur de bande limitée en réduisant au minimum le flux de données. À cette fin, on exploitera entre autres la priorisation des transmissions de données et le contexte d'exploitation au moment de la demande de données.

Table of contents

List of figures

List of tables

1. Introduction

The military is moving towards a networked environment. This environment promises to have the right data available to the right people at the right time. This promise is driving an abundance of ideas related to improvements in common pictures, promulgation of command intent, resource and sensor sharing, etc. Details of the issues related to implementing such an environment are also beginning to be realized. As a method of examining where we are heading in terms of a fully networked military, it is helpful to consider where we are presently.

Military systems have traditionally focused on developments oriented toward a particular task. Often these tasks require some combination of computing resources and skilled operators. The computing resources are often numerical implementations of detailed algorithms that address a particular calculation, while the skilled operator typically guides the input and interprets the output. The process has traditionally been oriented towards what is perceived as a single task or function.

The military community starting point is in fact, very similar to other communities including the meteorological and oceanographic research communities. In these communities, the initial conditions may be represented by a collection of systems that were developed independently, to address a particular problem. These systems required some type of data input and produced a data output or data product. The systems typically required a skilled and knowledgeable operator, familiar with the system inputs and outputs. The data streams may have been real-time based, historic, or a combination of the two.

After the system processed the data stream, the output was passed to another application for the processing of some other aspect of the problem. Again, the *model*[1] was repeated with inputs, outputs and operator. In these cases the processing could be quite elaborate, based on complex algorithms and addressing multiple processing steps within the application. This processing required considerable computing resources.

Another important aspect of the processing was its orientation toward a single establishment, platform, or organization. Each organization, or often each group of people within an organization, had their own developed applications that addressed their particular needs. Many organisations attempted to maximize software reuse by promoting a particular software language or library. However, often the intense upgrade cycle of software packages meant there was a leap-frog effect between competing software environments, where one month product A had the environment with the most beneficial tools, while the next month product B had a new release with even more tools. This often resulted in disparities in development environments within a single organization.

During this development, the data had a reduced role in the process. Many concentrated on the development of the software, compatibility of software packages, or upgrades. The only data requirement was that the data be available and in a format that could be read. If any more information regarding the data was required, the scientist or operator could always be asked as the processing was conducted in the same organization as data collection.

[1] A glossary of terms is provided at the back. Italics are used to identify terms present in the glossary.

The move to the networked environment is changing the above scenario. In all the communities mentioned above, the most basic change will be due to the fact that the processing is not necessarily at the location of, or near, data collection. It is still likely that initial processing is conducted at or near the sensor, but further processing will be remote to the organisation responsible for collecting the data. This reduces one's ability to ask local experts about the data.

The implications of this type of processing model have already been experienced in the oceanographic community. International programs to collect, process, and distribute data have been ongoing since the mid-1980's. These programs were built on a *central archive data model*, where there is one location responsible for the assimilation of data collected for the programme. Requests for data are then made to the central archive and the data are distributed from that point to other users.

Many communities are now investigating the move to a *networked archive data model*. Although the time scales are different between the central and networked models, the critical data issues are very similar between the two models. Time scales are different because in the central archive model, the data are typically received after they are processed and intensely scrutinized by the collecting organisation. In some communities, this step can take many years to complete. In the networked archive model, the data are available more quickly but possibly at a reduced level of quality. Of course the central archive utilizes the network in data transfers. The issue here is not the hardware or infrastructure being used to move the data, but rather the philosophy or concepts behind the central archive data model versus the networked archive data model.

One critical issue common across both models is the recipients understanding of what they are receiving in the data transfer. In this case, the recipient could be either a human *user* or an *application*. The understanding of the data and the descriptors used for the data are commonly referred to as a *semantic* problem.

The semantic understanding of data has a long history, likely dating back to the first measured quantities. For example, in the oceanographic community we may consider temperature measurements and, in particular, the progression from initial measurements being made using a thermometer in a bucket of water, to electronic measurements based on frequency measurements of an oscillating crystal. Both measurements result in a temperature, but both are collected with different sensors, different procedures and different levels of accuracy.

Using these temperature data in a processing stream has obvious issues. Due to the varying characteristics of the data, one would not expect these two temperatures to be interchangeable in all calculations. However, when someone receives the data, how are they to know which temperature is from which source? This type of question is addressed through the use of semantics and metadata.

Metadata has been described as 'data about data'. A section in this document will explain more about metadata. At this point it suffices to say that the metadata will help the recipient distinguish between the two temperature measurements in the above example. In a networked environment, the important aspect of metadata use is whether or not the metadata is transferred with the data to the recipient. This is a key issue to be addressed in the conceptual networked archive data model.

There are many other issues that should be kept in mind during the transition from the central archive to network philosophy. For example, it seems only natural that the more data and information one has available via the network, the better one's decision will be - but is this really true? There are important issues related to the applicability of the available data (i.e., lots of data available, but none applies to your request) and the quality of the data (i.e., the data you require exists, but the quality is so poor you cannot use it in the decision making process).

Examples of both issues are available in our everyday experience. Consider for a moment the result from a standard Internet search. It is likely we have all experienced the plethora of hits returned from a web search engine and the resulting feeling of, 'where do I begin?' Similarly, consider the volume of information received daily in email. In both cases the data volume is large while the information volume may be small. As well, often in the case of web resources the quality of the information is questionable, as everyone has the ability to present material on the web. For the user, the problem becomes one of identifying the relevant information.

In the fully networked military, are words like 'net-centric', 'network enabled', 'interoperable', etc. In this environment, the implementation of concepts often leads the research that is intended to investigate the potential benefits of the concept. In the case of the networked military, the initial assumption appears to be that more data makes for a better decision. However, this type of assumption may not be based on research so much as it is based on anecdotal evidence.

Methods of identifying relevant information are an important part of the networked model. However, the actual identification of the relevant information can also be formulated in terms of a relevance model. A *data relevance model* attempts to address one key aspect of the networked paradigm - the issue of the right data[2]. For example, one relevance model may be based on semantic keyword searches. In this model, data would be searched for keywords that represent important items to the client. Other models may involve spatial-temporal searches, where data are identified based on proximity to an event important to the client.

For the Networked Underwater Warfare (NUW) Technology Demonstration Project (TDP), a relevance model has been indirectly proposed by Lefrancois [1]. Lefrancois has identified 12 (recently revised to 14) information types relevant to the multi-static problem addressed by NUW. Each of the information types were then examined based on a typical tasking encountered during an underwater warfare (UWW) operation. Thus, this relevance model is based on the identification of present tasking and the data types important to this tasking. As the tasking changes through an operation, so does the relevance of particular data.

1.1 Significance of NUW Data Sharing Architecture

This report is attempting to outline a *model* and *architecture* for sharing data within the NUW project. During this process, it is important that we recognize the significance of this architecture within the larger context of planning for the future Canadian military. The main objective of NUW is the demonstration of improvements to the UWW operation through the use of a common information management infrastructure [2]. This is supported in part by the development of a

[2] The full aspect is the right data, to the right people, at the right time.

networked data exchange system to generate the multi-platform Common Operating Picture (COP). The networked data exchange system is an important part of NUW. In fact, the exchange system is an enabler that provides the possibility of a Canadian Forces (CF) service-level integrated approach to UWW operations.

The NUW project will utilize an air force maritime patrol aircraft (MPA), a land-based or reach-back cell, surface research ship, Maritime Coastal Defence Vessel (MCDV), and possibly a submarine. These assets will be linked through the networked data exchange system. There is also the possibility of linking some small data subset to an instance of the Canadian implementation of the Land Command and Control Information Exchange Data Model (LC2IEDM). However, this type of integration does not necessarily have to stop at the service level. Conceptually, the data exchange could also support organizations outside the Canadian Department of National Defence (DND) such as other government departments (OGD) and non-governmental organisations (NGO).

To understand the significance of what NUW is attempting to construct, consider the integrated exchange between these platforms in relation to military planning for the future. Military planners have recognized the importance of such a collaborative information environment when developing the CF Target Integration Model (TIM) [3]. The TIM is a conceptual model that contains various components (e.g., data, fusion, decision support) and the relationships between these components (Figure 1). Within the TIM, the theory is that the information exchange and the resulting collaboration help enable a shared understanding. These things combine to ultimately lead to efficiencies in task and mission execution. The initial TIM target is 2008, and is thus described as TIM08. TIM08 provides the framework for discussing NUW's contribution to the future CF.

The TIM was developed within the framework of the C4ISR[3] Campaign Plan [4]. Using the guiding objectives within Strategy 2020 [5], the TIM concept evolved to help address three 2020 objectives - decisive leaders, globally deployable, and interoperability[4].

The TIM is illustrated in Figure 1 and is summarized as consisting of the 11 components identified in Table 1. These components were recently outlined at a C4ISR coordination workshop [6], whose focus was the coordination of Canadian C4ISR defence research projects. The components were taken directly from the TIM, with no formal definitions assigned to the components.

The coordination workshop recognized the NUW project as contributing to components five, seven, eight, nine and 11 (see Table 1). By understanding these contributions we will place NUW within the TIM context. This process will also help establish, the importance of the data sharing architecture to NUW and thus the TIM.

---

[3] C4ISR - Command, Control, Communications Computers, Intelligence, Surveillance and Reconnaissance

[4] As recognized by an Auditor General review [7] of the C4ISR Campaign Plan, there is no established Canadian definition for interoperability. Although such a definition could be generated to support the efforts of the NUW project, this is beyond the present scope of this report. Instead, we will use the word interoperability to loosely describe the process of utilizing the same data across different processing systems, for different applications.

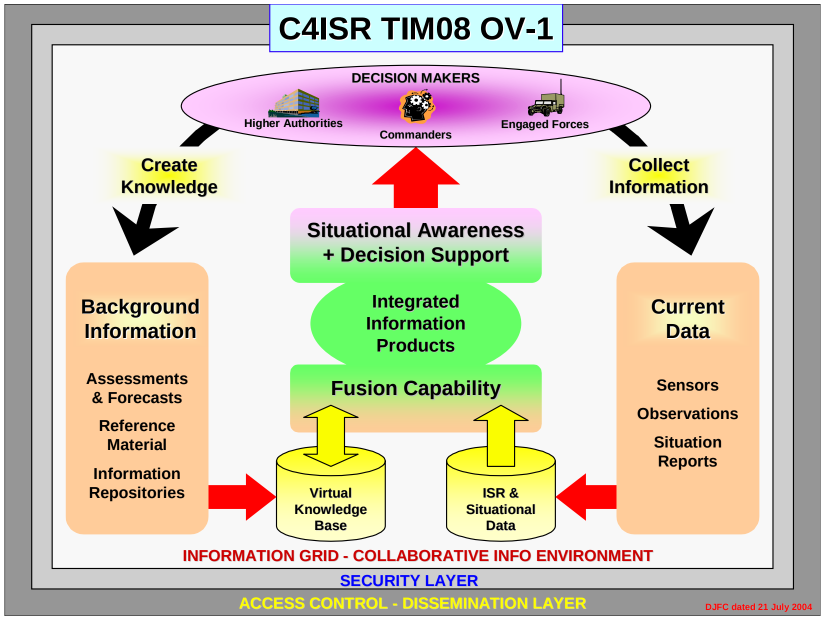

Figure 1: The TIM conceptual environment defined in the C4ISR campaign plan [4].

TIM component five is labelled "Global ISR & Operational Data (Current)". This is represented by the cylinder labelled "ISR & Situational Data" as shown in Figure 1. This cylinder represents real time data. Historic data (Table 1, component 4) is represented by the the "Virtual Knowledge Base" (Figure 1). Although the NUW project will utilize historical data from the Atlantic Meteorological and Oceanographic Centre (METOC), the project focus is on real-time data links and utilization. The real-time aspect is due to the NUW focus being on the collaborative execution of a UWW operation. In this sense, NUW will create and access a Collaborative Information Environment (component seven) for UWW operations.

The enhanced data and "Fusion Capability" (Figure 1) is also being investigated under NUW (component eight). In a networked environment, there are more available data resources as compared to the single platform case. New fusion algorithms and techniques need to be developed to take advantage of the diverse data resources from multiple sources, taking into account, for example, differences in data granularity and accuracy.

The NUW project will also be contributing to the development of an enhanced Common Operating Picture (component nine), enhanced because of the new data and information sources that will be combined and utilized. This COP will consist of data from sonobuoys, sonars, radars, etc. The remote sites (e.g., the METOC centre) will also be contributing historic and current data to assist the operator compilation and understanding of the local environment.

Table 1: TIM components as identified from Figure 1 and used in the DRDC C4ISR coordination workshop.

Component Number	C4ISR TIM Component
1	Access Control – Dissemination Layer
2	Security Layer
3	Information Grid
4	Virtual Knowledge Base
5	Global ISR & Operational Data (Current)
6	Sensors & Current Info Sources
7	Collaborative Information Environment
8	Fusion Capability
9	Common Operating Picture
10	Decision Support
11	Tasking & Feedback Mechanisms

NUW also intends to develop and implement decision support aids (component 10). At the C4ISR coordination workshop, "Decision Support" was not identified as a TIM component where NUW would contribute. This was because the brief project report supplied to the workshop did not mention the development of the decision aids. However, NUW plans to develop aids that will assist in the deployment of sensors, to maximize the spatial-temporal coverage of the sensor suite.

The "Tasking & Feedback Mechanisms" (component 11) refers to the influence the compilation of information has in determining the tasking of the assets. An enhanced understanding of the surroundings will result from an understanding of relevant data, which in turn leads to adjustments to the operational tasking. Decision aids play an obvious role here, as the aids help the decision makers evaluate and react to new situations.

In summary, the NUW project contributes to many of the TIM components that exist, starting after the sensor level (Table 1) and moving toward the decision and tasking level. The broad applicability of NUW to the TIM components highlights NUW as an example research implementation of the latter half of the TIM. As such, the architecture being built for NUW data sharing will likely provide useful input to developments that take the CF toward a more complete information sharing environment.

1.2 Outline

This report concentrates on some of the key issues related to data exchange in support of the NUW project. In particular, this report deals with the transfer of meaningful data. In that respect, this report represents a contribution to the overall NUW implementation plan for the data requirements as previously identified [1].

The report begins by first outlining the basic types of clients that may be considered in the networked environment. Section 3 then establishes data utilization steps, or alternately the steps a client follows to utilize the data. The utilization of data provides the foundation terminology to be used in the process of obtaining and understanding the received data. In this process, metadata plays a key role and Section 4 describes metadata and how it may be used as a support for the data. The metadata terminology is then introduced as a vocabulary. Section 5 describes types of vocabularies and subtle differences in these types. A description of dictionaries and why dictionaries are an important component for the understanding of data content is also included. Next in Section 6, a particular type of metadata is described. The United States (US) Department of Defence Discovery Metadata Specification (DDMS) describes the metadata required to describe a data asset. Finally, Section 7 combines all the previous discussion points to construct a proposed architecture for the NUW Project. Extensible markup language (XML) is used to provide a proof-of-concept implementation of the architecture.

2. Model of Client Categories

The ultimate end product of a system is often linked in some way to a human requirement. However, the system as a whole may be viewed as a group of processes. Each individual process doesn't directly address the requirements of the end product, but rather contributes to intermediate requirements of the system. These intermediate requirements may be considered application-based requirements, where the internal processing of the application requires these intermediate results. The human-based requirements are more often addressed by the application (i.e., the group of processes).

The requirements of the human and application are often different. However, the similarity between the two is that both are expecting something from the preceding part of the system. In this respect, both are clients of the preceding part of the system. Within this document, the term *client* will be used to include both the human (or user) and application. Note that clients have certain expectations on the outputs of the preceding part of the system. In this context, the term client is not being used to describe a client-server model. Rather, the client is identified as someone or something that is requesting and expecting something from a system component.

This concept of a client can easily be extended to individual system components, reaching the various functions that are utilized within a system. However, here we are not concerned with the internal system architecture that relates to internal functions. Rather, we will use the model to describe system-to-system or user-to-system situations.

The success of a process may be measured by its ability to address client needs. Therefore, it is important to understand client requirements on the process. To understand these requirements we need to understand the clients of the process.

Using the perspective of the client is a specific example of a more general requirements analysis approach that considers the viewpoints on the system [8]. Viewpoints are perspectives that can be based on the components of a system. Viewpoints can be established from *data assets*, components of the system, or receiver's of services from the system. The client categories being explained here are a piece of the larger group 'receiver's of services'. This analysis helps define the client types before analysing the services required to meet the client demands on the system.

To assist in this understanding, we must also keep in mind that we are attempting to create an information exchange system. This system will be delivering data and information to the client. From the viewpoint of the delivery system, the client is whatever happens to be requesting the data or information from the system. From the viewpoint of the client, the system is providing a function, asset, or service that is required. However, the exact client viewpoint depends greatly on the level of knowledge the client posses when approaching the system. Thus, we develop a *client categorization model* based on the amount of initial knowledge the client possesses regarding the data asset they are attempting to access via the delivery system. Here, the data asset includes both the available data and functions.

The process of categorization is always prone to criticism. Categorization tends to box items in one category or another, while not allowing the items to exist within more than one box.

Although this issue is recognized, we proceed with the categorization in part to develop the terminology for discussing client knowledge.

In terms of NUW, the data sources will be an assortment of sensors on or deployed from platforms such as ships, aircraft and submarines. The data from these systems will likely be processed from raw form at the collection site, with the data then entering some type of distribution system.

In the case of NUW, similar data systems will exist on the platforms. This will simplify the distribution function. In particular, the basis of the processing and distribution application will be the software environment known as the System Test Bed (STB) [9]. The STB is a configurable suite of software that utilizes display capabilities of the US Common Operating Environment (COE). The software suite provides a set of tools for the processing, analyses and display of sonar and related data. All the STB applications are built to utilize a CORBA-based (CORBA - Common Object Request Broker Architecture) data store, called the Data Server (DS) [10]. The STB DS forms the backbone of the data repository used for the NUW project.

As noted previously, the initial client knowledge forms the basis for the categorization. Of course, the initial level of knowledge may change if the client repeatedly accesses the data asset. However, this simply means the client categorization is a function of client familiarity with the asset. In the proposed model, we define the following three levels of client:

- Category Three - This is the highest level of client knowledge regarding the data asset. We consider this category to include those clients with extensive previous knowledge of the data asset, its structure, and data content. At category three, the attained client knowledge is so extensive that it likely originates directly from the original designers or creators of the asset. In this category, users or other designers have access to the original designers of the asset.
- Category Two - At a reduced knowledge level is the client that possesses knowledge of the existence of the data asset and the associated functions, without possessing the detailed structure knowledge of the data asset. In this case, the details of the data and data structures contained within the asset are not known. At level two, the client recognizes the existence of the asset but does not possess knowledge on the details of the internal structures.
- Category One - The lowest level of knowledge for a client is level one. At this level the client has no previous knowledge of the data asset. The client is not aware that the data asset exists, nor are they aware of the internal structure of the data asset. This level of knowledge is characterized by a client entering a network with no knowledge of the assets available within the network.

Implied in the above categorization is a level of client knowledge that allows the client to connect to the network where the data asset exists. Another important implication is that this model for categorization has nothing to do with network connections, protocols or actual process of transferring bits through some wire or wireless interface. This categorization is based only on client knowledge that pertains to the data and the structure used to store the data.

An analogue for this categorization may be made to a customer entering a store. A new customer may enter a store with no prior knowledge of the store. Perhaps they discovered the store by talking with a friend or by searching the local telephone directory (both of which may be considered data assets and both being part of the discovery process). This customer has no prior knowledge of the store layout, product line, or support staff knowledge. This customer is a category one client. Similarly, a customer may enter a store with some previous experience and perhaps knows the store layout and the level of staff support that can be expected. This is a category two client. Finally, a customer may enter a store knowing exactly what they want, the section where the item is stored, and where in the section the particular item is found. This is a level three client.

The most general client category is level one. In this category, the client requires the greatest amount of information to succeed in utilizing the data asset. The client at this level has two initial requirements, namely:

- to discover the existence of the asset, and
- to assess the content of the asset.

These requirements are important because they create the need for two separate, but related, information sets. In particular, these two information sets must be capable of providing the information that allows the discovery of the asset and also allows assessment of the content. Note that neither information set directly provides data to the client, at least not data in the context of the data asset. More correctly, the information sets are metadata that support the client requirements.

2.1 Client Levels and the STB Data Server

The STB presently exists as a research tool and represents a component of the shipboard processing associated with a research sonar processing system. The data server represents the data storage mechanism for the STB.

The data server (DS) is an object broker, capable of storing data, data descriptions and/or methods to requesting clients. The DS stores sequences of bits, with the content being whatever is applicable to the writing and reading applications. In this regard, the DS is capable of storing data item names, descriptions of the data, and the data values. In an object paradigm, the DS would store the data values with the methods used to manipulate these data values. These data objects would then be passed to applications, permitting the applications to manipulate the data values in ways only described by the methods.

Although the data description and object capabilities exist in the DS, neither have been utilized by present implementations of the STB. Without data description or method implementation, external clients are unable to exist as category one or two. This is because the client has no way of knowing the data exist within the data server, nor would the client know of the meaning of the data contained within the data server.

In previous implementations, only category three clients could communicate with the DS. In this state, all knowledge a client has of the structures within the data server is established by the sharing of information between system designers. This approach is acceptable at the current evolution of the STB. When designers are creating a system based on a collection of software components, it is natural for them to share the details of the data structures being used. However, when a system-of-systems is being constructed, the procedure starts to break down. The number of developers involved becomes too large to manage the developer-to-developer communication that is required to build the necessary interfaces.

3. Model of Data Utilization

Now that the client categories have been defined, we proceed to outline the procedure followed by a client to utilize a data asset. Essentially, this utilization involves the expansion of the two initial client requirements noted in the previous section. This model of data utilization is based heavily on previous work from the United Kingdom (UK) Natural Environmental Research Council (NERC) DataGrid[5] (NDG) Project.

The NDG effort began in 2002 [11] with the general aim of creating a virtual organisation to share environmental data. The NDG project is attempting to provide a discovery and usage capability for a data holding, by linking a wide range of heterogeneous data holdings under a single framework. There are seven key requirements of the NDG [12], namely to:

1. Provide discovery and access of data without prior knowledge of the holding,
2. Provide functionality beyond the original user community,
3. Provide discovery and access beyond original discipline for which it was collected,
4. Hide heterogeneity of data source,
5. Allow pre-presentation processing (e.g., sub-querying, transformations, consolidation),
6. Deliver data to the desired place in the desired format, and
7. Allow limited server-side processing.

NDG also established a model for data acquisition and utilization [13]. The process of utilizing the data asset may be viewed as a sequence of steps. The NERC team has suggested eight steps in this utilization process. Clarifying the discovery of the asset, here we suggest the modification of the NDG utilization steps to better address the net-centric paradigm. The revised steps are as follows:

- *Discovery* – The process of searching and finding the data asset.
- *Authentication* – The process of verifying that the client attempting to access the asset is indeed who they claim to be.
- *Authorization* – The process of determining if the authenticated client is permitted to access the asset being requested.
- *Data Identification* – The process of searching and finding the data that is required for the particular process or activity.
- *Extraction* – The process of retrieving the data from the repository on which it initially resides.
- *Subsampling* – The adjustment of the obtained data to the exact sampling frequency characteristics required for subsequent analyses.
- *Regridding* – The adjustment of the obtained data to the exact spatial-temporal characteristics required for subsequent analyses.

[5] The NDG team prefer upper and lower case combination 'DataGrid'.

- *Formatting* – The modification of the format or structure of the data file to meet the requirements of the local processing system.
- *Processing* – The actual calculations associated with the use or incorporation of the obtained data into analyses that meet the requirements of the research.

The subsampling and regridding components are intended to address item five in the NDG key requirements list. These steps may not be necessary in the NUW implementation.

The above model of utilization is a process and is not directly related to the architecture used to implement the process. Each step in the utilization model does not necessarily correspond to a single operation and may consist of many processes. As well, steps may be implemented in multiple layers within an application. For example, the discovery step may involve the discovery of a service that in turn accesses data. Alternately, the discovery step may be more direct where the discovery identifies a data asset such as a database. At a finer granularity, the discovery step may also involve the identification of a table or record level object within a database or data asset.

Each of the above discovery examples has a unique but related requirement for descriptive data. The required descriptive data must describe the asset, service, or data record to be discovered. The description must be in sufficient detail to allow independent assessment of the resource. This descriptive data is in fact the metadata that supports the discovery process.

Building the metadata repository that supports data discovery is not a sufficient condition for discovery; however, it is a necessary condition. The discovery process relies on the metadata to the point that the metadata must exist, it must be accessible, and must be *syntactically* and semantically understandable by the client. Here, accessible implies that the metadata exist in a common and known location, or be registered through common procedures. Syntactically understandable means the metadata must be readable by the client while semantically understandable implies that the metadata has the form, structure and content that the client can properly interpret.

In any system building process, considerable attention is often directed toward the physical construction of the system, with insufficient attention directed to the data content. However, in the networked environment, the content must be capable of providing the system with enough information to allow a judgement of the applicability of the asset. The process of judgement is implied in the 'data identification' step noted above. The systems ability to judge the usefulness and applicability of an asset will depend critically on the metadata content found as a result of the discovery process.

4. Metadata

Metadata is a complicated topic. As well, it is at a sufficient level of abstraction to be somewhat difficult to understand. Here, we attempt to define and describe metadata.

4.1 Definition

Many groups and organizations describe metadata as 'data about data'. However, this definition makes if difficult to quantify exactly what metadata is. Perhaps a better definition of metadata is that *metadata are the values of characteristics that qualitatively or quantitatively describe or support a resource*. In this case, any data asset is considered a resource. This definition provides several advantages over the more traditional definition.

The central point of the definition is the resource. A resource can be any data or service asset that is available to the local or networked environment. The resource is described using characteristics. These characteristics may be either qualitative or quantitative. The value of the descriptive characteristic is the metadata.

As an example, consider a *dictionary* of data terms. These terms can be considered part of the elements or items within a data structure. In turn, the data structure is filled with data to form data records. Suppose the dictionary contains a term 'latitude'. The dictionary would likely contain a descriptive characteristic called 'definition'. As an example, for the term 'latitude', the definition characteristic may contain 'the angular distance of a point from the equator of the earth'. The value of the descriptive characteristic 'definition' is the metadata that supports the term 'latitude'.

Metadata may also include quantitative descriptions of the resource. For example, a quantitative characteristic that supports latitude may be the range of acceptable values. If latitude were being used to describe the position of an object on the earth, then a quantitative limit on the range may be -90 degrees to +90 degrees or similarly, limits defined in terms of North and South.

This content or description is the metadata that describes the single term 'latitude'. Given this content, we see one role of metadata is to provide the semantic understanding of the terms used within a particular resource. In the case of the example, the metadata provides the semantics of the data item 'latitude'.

Metadata may also support a complete data set. In this case, the differences between describe and support are important. Describe implies the citing of details to provide a more realistic view of the data. For example, the latitude range defines values that directly describe the allowed content of the latitude data. Support implies that the metadata provides a level of assistance to the data, but does not directly define or limit the data. Support also includes the support of processes applied to the data asset. For example, a supporting characteristic may be the internet protocol (IP) address of the computer where the data asset may be obtained. This type of metadata supports the discovery of the data asset, but does not describe the data asset.

In terms of functional uses, metadata contributes to the process of distributing, advertising, using and combining data assets. Internationally, these functions are being explored in community-based efforts some of which are focused on marine data. Experts in the Marine Metadata Interoperability (MMI) Project [14] are helping to explain many of the metadata issues by providing definitions, guides and examples to clarify the use of metadata in these functions. The concepts being addresses in the MMI are directly applicable to the net-centric paradigm evolving in the military community.

4.2 Metadata Categories

The first requirement of a level one client is to discover the available assets on the network. In this scenario, the metadata that supports data discovery is the first level of metadata accessed by the client.

Other communities have examined the metadata requirements for supporting data discovery. The marine data community has attempted to define the levels of metadata required for automated systems to describe and support data assets on a network. The NDG effort has defined six levels of metadata. These levels are labelled archival (recently renamed to Climate Science Modelling Language, CSML), browse (recently renamed to Metadata Objects for Links in Environmental Science, MOLES), summary, discovery, collection and extra. The levels of metadata description were developed in a marine context, but are directly applicable to any shared network of data assets such as would exist in a military context. Isenor and Lowry [15] provide a terse summary of the metadata types as described by NGD.

Of particular interest for the NUW project is the model followed in the CSML and MOLES implementation. CSML is a structure used for the storage of the metadata required to support the use of the data asset. This type of metadata includes spatial-temporal coverage, definitions of coordinate systems, definition or pointers for parameter terms and data or pointers to data. The CSML concept (Figure 2) has a single CSML record describing an entity of a data set (e.g., one XBT profile in a series of profiles). The holder or owner of the resource would generate this CSML record. A user entering the system would use online software to create a CSML record that describes the data set they would then like to obtain. A software layer then utilizes the user created CSML description to query entity specific CSML descriptions, combining those data entities that match the user query. In this way, a new data set is constructed to meet the requirements of the user.

MOLES is a structure used for the storage of the metadata required to support discovery metadata generation and browse services. The *Discovery (D) metadata is defined as the metadata that populates the discovery portals*. D metadata is designed to be searched by clients looking for data sets. Discovery metadata comprises totally public domain information encoded in records conforming to established standards such as Dublin Core, Document Interchange Format (DIF), or ISO19115. Many such services already exist; for example, the Global Change Master Directory (GCMD). GCMD utilizes the DIF structure for metadata records. DIF can be generated automatically from MOLES. However, other structures such as ISO 19115 compliant metadata records may also be generated from MOLES. The basic idea is that a single MOLES repository can support the generation of multiple metadata structures that can then be used in existing search services (Figure 3).

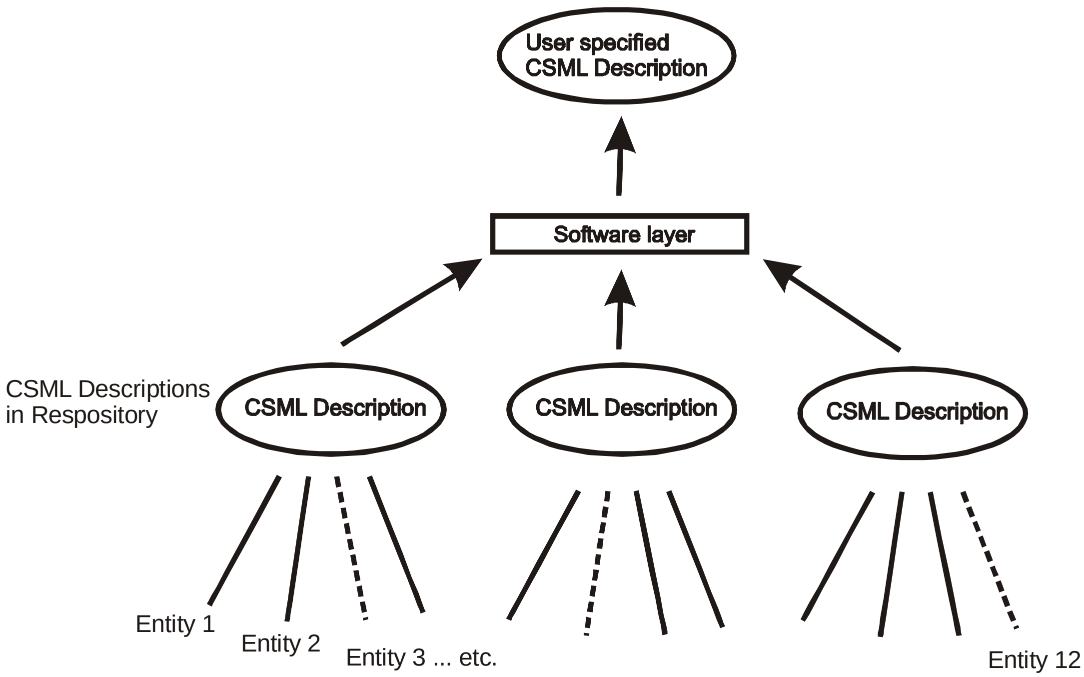

Figure 2: Conceptual model of how CSML metadata combine multiple data entities into a single data set as specified by the user. The dashed line indicates the entities identified as meeting user needs. These entities are from disparate data sources. The entity metadata records are combined to form a single, consistent CSML description for the user.

These models show two important points. First, the model for the CSML shows how metadata can be used to describe data sets as defined by the holder of the data. However, metadata descriptions can be used to redefine the data set based on requirements of the client. Second, the model for the MOLES shows how abstracting metadata structures to a higher level can be advantageous. The abstraction allows the creation of structures compliant with international metadata standards, thus addressing the need to provide consistent metadata in different international standards.

4.3 Metadata as a Resource Descriptor

Using metadata as a support for the data discovery function is one metadata usage that is easily understood. However, other views of metadata may be used to elucidate the meaning of the term metadata. For example, a unit of metadata may be considered to consist of a descriptive characteristic (e.g., termed a property), a value for this characteristic (e.g., termed a value), and the subject that the metadata refers to (e.g., termed a resource) (see [16] for further description).

This model is also the basis of the Resource Description Framework (RDF) [17]. RDF was developed by the World Wide Web Consortium (W3C) to represent metadata for web resources, where the term web resource can include anything identifiable on the web as well as things retrievable from the web. The RDF model uses the resource, property, value combination with a slightly different terminology, namely a subject, predicate and object, respectively.

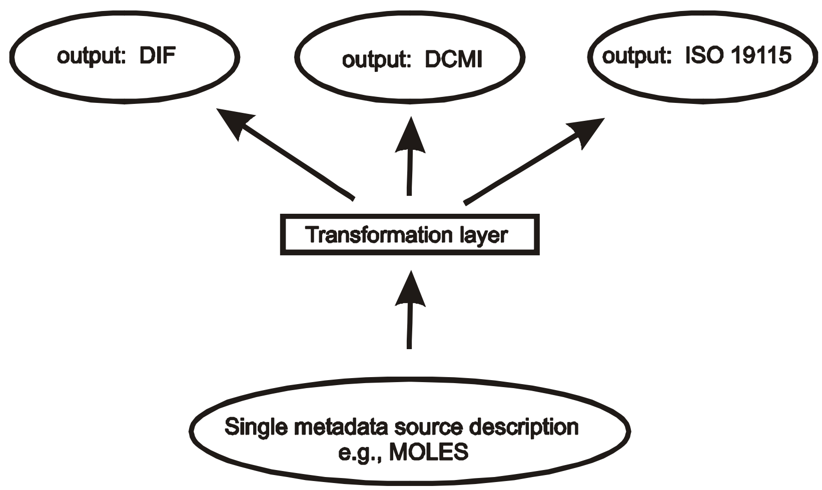

Figure 3: Conceptual model of how metadata abstracted to the level of MOLES can be used to populate discovery services using an assortment of metadata structures.

As an example of the resource, property, and value combination set, consider a data asset that is labelled sensor information. Sensor information may include the spatial-temporal position of the sensor, including latitude, longitude, altitude and the time the sensor was at that spatial position. However, in this example we consider a moving sensor and so the heading and speed of the sensor is also included in the information set.

In this example, the resource may be considered the sensor information set (the resource identifies the data or data set that is to be described). A property of the resource would be speed (the property is an identifier that represents an attribute or characteristic of the resource). The value of the property could be 10 knots (the value represents the content of the property). The property, resource, value set could be represented as a structure in a programming language, a record in a database or a comma separated variable length record. This metadata model is scalable from a single datum upward to collections of data assets.

5. Controlled Vocabulary

5.1 A Vocabulary

In many professional communities, the terminology used to communicate is often community specific. In the previous section we noted the use of the word ‘altitude’ as part of the spatial-temporal position of a sensor. Altitude is a term typically associated with aviation and specifically the height of something above a reference point (e.g., sea level). Altitude is not commonly used to describe the depth of something below sea level. In the community interested in underwater platforms, the term ‘depth’ is commonly used.

The terminology for a community collectively represents a specialized *vocabulary* for that community. When these vocabularies are formally managed, they become a *controlled vocabulary*. Controlling the vocabulary is useful because it helps the community avoid misspellings and avoid the use of arbitrary words that cause inconsistencies. Avoiding misspellings and inconsistencies helps avoid misunderstandings (i.e., promotes understanding). One form of metadata content is the terms that are used within a particular vocabulary.

Previously, we noted that the term ‘latitude’ could be defined within a dictionary. It was also noted that ‘latitude’ could be part of a larger collection of data, called ‘position’. With these two terms we begin to form a vocabulary for geospatial data. However, the two terms actually have two different purposes. The term ‘latitude’ is more closely associated with a datum value and as such is contained within a data vocabulary. This is because the term ‘latitude’ could be assigned a value representing a north-south position on the earth. The term ‘position’ is really a grouping of many terms, such as latitude, longitude, depth (or altitude) and perhaps time. However, the term ‘position’ could not be associated with a datum value. Position refers to the collection or group of terms. ‘Position’ is actually part of a discovery vocabulary.

A data vocabulary is a collection of terms that identify or name the individual data items in the subject community. For example, the term ‘latitude’ would be contained in the data vocabulary as this term applies to a data item. In the MMI project, this type of vocabulary is known as a *parameter usage vocabulary* (PUV). The parameter usage vocabulary would contain the formal parameter names, definitions, units, etc. and may be used within a data file or structure to label the data items.

A discovery vocabulary typically names a group of data labelled to assist in the discovery of data items that are in some way related. For parameters, the MMI refer to this type of vocabulary as a *parameter discovery vocabulary*. A PDV is a group of terms used in the discovery process. The PDV terms typically represent a collection of terms from one or more parameter usage vocabularies.

Discovery vocabularies are typically hierarchical, containing labels that often represent groups of other labels, ultimately relating to the PUV. This often results in high level terms being broad, such as ‘atmospheric’ to represent all atmospheric data at the asset. However, discovery vocabularies do not apply only to parameters. Discovery vocabularies could be related to platforms, sensors, geographic areas, etc. As examples, a platform vocabulary could include ‘ship’, with the subcategory ‘frigate’. This type of vocabulary would allow the distinction between particular platforms.

The discovery metadata vocabulary represents a sequence of successively higher conglomeration of terms. The particular grouping of terms within the discovery metadata vocabulary is related to the issue of data usage. The particular grouping that makes most sense to a client will be that grouping that most directly answers a particular question posed by the client. For example, if a user wants to know all the sensors on a ship, then grouping the sensors by platform is very sensible. However, if the user wants to know all the sensors of a particular type, then grouping by sensor type is most sensible. The individual questions posed by the user helps to establish the discovery metadata grouping of most interest to the user. The problem is that in a networked environment, there will exist a great diversity of users. For example, some users will be interested in grouping by platform while others will want the grouping by sensor.

Within NUW, we are particularly interested in the PUV and the PDV. For both parameter usage and parameter discovery vocabularies, the labels must be known and defined. For example, ‘velocity’ is a somewhat common term and one may consider it to be obvious. However, definitions are still required even between very similar communities (e.g., a westerly ocean current moves water towards the west while a westerly wind moves air towards the east). However, other data or discovery terms may be even less obvious. For example, ‘waveform type’ may be well known in one specialized subject area but unknown in another. Alternately stated, vocabularies are often community and sometimes domain specific.

5.2 A Dictionary

In most commercial database systems or spreadsheets, the common representation of a data unit is termed a table or sheet. In both cases, the data unit is presented in rows and columns. Typically the columns are named, where names indicate something about the data values in that column. In most cases the column name provides some hint about the data value in the column. Note that the name choice is at the user’s discretion – the applicability of the name to the data values is dependent on the user. Once named, the name is stored with the data internal to the data management system.

As noted in Section 2.1, the DS is capable of storing data names, descriptions, and even methods in its current configuration. However, present implementations have not utilized this aspect of the DS. Present implementations of data storage in the DS have not included the naming of the data in the structure housing the data. The sequence of bytes that represent the data values are not labelled within the DS so the DS has no internal information about what is stored in the byte sequence.

Of course this knowledge must exist in some location. In the case of present implementations, the knowledge of what is in the sequence of bytes is within the structure that writes the sequence. Of course the structure reading the sequence may also contain this knowledge. However, initially, only the writing structure knows the meaning of the data value sequence. As well, the writing structure is the authority on the content of the byte sequence.

It is important to realize that this exercise is attempting to liberate the information within the DS to outside clients. This liberation deals not only with the data structures but also the descriptors of those structures. Here, we use the word descriptor to represent the information that in the general sense would occur as column names associated with a database table or spreadsheet.

The structures and descriptors are initially set with the write statements that provide the content to the DS. Thus, the programmers who create the write statements are the ones defining the descriptors for the data within the DS. In this situation, two programmers could use the same descriptor name for data values that are defined, measured or observed differently. In a development environment where structure information is being passed programmer-to-programmer, this may not cause a problem as one programmer can explain in detail, the exact meaning of each data value. In this way, an individual programmer gains the detailed knowledge about the content of the structure and the associated data.

However, this procedure breaks down when there is no programmer-to-programmer communication. This is often the case in large developments or in unconnected developments where it is extremely difficult to identify and contact the programmer with the knowledge required to explain the data content. In this situation, the user is often left to examine the content and judge its use for the particular application.

Note that knowledge of the structure does not refer to the syntax of the structure. Syntax, meaning the details of data types (e.g., float, integer, string, specific ordering, etc.), is different from the knowledge required to judge whether or not some data item meets the need of a particular application. The judgement process is in many cases dependent on metadata associated with the data.

An example of this would be the beam-forming calculations performed using towed-array data. Depending on the beam-forming calculations, the beam angles may be in the range of -90 to +90 degrees, or in the range 0 to 180 degrees. As well, there may be ambiguity in which end of the array is being used as the reference location for the angular measurements. In this case, knowledge of the reference information is required to understand the beam angles. With a minor adjustment, the reference direction could accommodate most applications, but only if the receiving system knows the reference information.

Similarly, the data accuracy may also be part of the judgement process. Again using a beam example, the beam-form information from a sonobouy is considerably different as compared to a towed array. Both data may be reported as beam angle. In the case of the sonobouy, the accuracy may be 15 degrees while in the case of the towed array, it may be less than five degrees. The receiver of the data must have sufficient information available to judge if these data are useful for their particular application.

The definition of the data items or descriptors is a complex and intricate issue. Those defining the descriptors must examine and decide if similar descriptors (e.g., ship latitude, aircraft latitude, sonobouy latitude) have the same definition. In most cases, it is unlikely that the same definition applies as differences in accuracies, processing methods, etc. will result in different definitions. As well, careful consideration must be given to the content of the dictionary with regard to the judgements being made by the clients. With each descriptor, the defining party must think about things like the data accuracy, precision, ranges, and methodology used to obtain the data value.

A dictionary represents a formal structure for the storage of this information. The formalization provides a distinctive advantage. First, the formal dictionary structure provides those defining the content with a guide to the important information. Second, the formal structure allows the storage of the information in granular form. This means that the information regarding the dictionary descriptor is not stored as a single unit, but rather, contains structure in itself. In this form, parsing of the dictionary content is much easier.

Within the context of the DS, it is important to note that dictionary descriptor definitions will be a controversial process. At the moment, the programmer has complete freedom in defining and storing data in the data server. Formalizing the definition process will have consequences. The added burden of forming definitions will have cost implications because it will add overhead to the process of adding new data descriptors to the DS. As well, it will likely expose short-cuts taken by the programmers in initially defining the descriptors. In such a process one may expect to find issues such as the following:

- the same name or descriptor used for two different data units
- different names used for the same data unit
- data values that are stored using inappropriate names

In reality, the items named in the software structure that is used to write to the DS are not important. What is important is the content of the item and how this content is described. It is the description that will be used to judge the usage of the data for other applications, not the name used in the write statement that stored the data value.

5.3 Proposed Dictionary Structure

For clients to be successful in discovery and utilization of the data asset, the discovery and data vocabularies must be defined, accessible and understood by the clients. One method to accomplish this is to create a dictionary to support the terms used in the discovery and data vocabularies. Such a dictionary addresses the issue of a controlled vocabulary and may be modelled after a common language dictionary.

In 2002 the International Council for the Exploration of the Seas (ICES) and the Intergovernmental Oceanographic Commission (IOC) jointly created a study group to examine marine data exchange systems using XML. This study group, commonly referred to as the SGXML (Study Group on XML), developed a dictionary structure intended to aid in the discovery and mapping of oceanographic parameter terms. The intent was to allow access to the ocean parameter dictionary terms thereby allowing the community of interest the ability to query and identify existing dictionary terms. The SGXML hoped that by providing such a dictionary of terms, users would reuse existing terms rather than develop new terms.

This SGXML examined three very fundamental requirements of a data exchange system, namely, dictionaries, metadata and data structure. All three topics were examined from the perspective of ocean data. However, the results are directly applicable to making term definitions available to any system. Two of these topics, metadata requirements and the dictionary, are useful when considering the data structures within the data server.

The SGXML also recognized the need for the mapping of codes used within a single dictionary term. A single term may be defined and described in such a way that it is common across many systems. However, internal to the system, individual codes or abbreviations may be used to identify the specific term.

A simple example of this may be developed. Consider the bearing of a target from a platform. The bearing may be defined as either a relative or absolute angle. In the relative case, the bearing may be with respect to the ships heading. In the absolute case, the bearing is referenced to True North. Another acceptable variation would reference the bearing from magnetic north. The actual bearing angle may be defined in degrees or perhaps in terms of the System International derived unit of radians.

It is likely that such a common definition would be applicable to many systems. However, the systems may be storing or manipulating bearing data using an assortment of codes that identify the data. This is particularly true for legacy systems. For example, one system may refer to the bearing data as “brn” while another system refers to the same data as “bearing”. The SGXML structure allows the term bearing to be defined and also allows the term to be connected to two codes, in this example being represented by “brn” and “bearing”.

Now consider this example in terms of the STB data server. Within current DS implementations, only the client creating the data structure that contains bearing knows that bearing is present in the structure. At present, other clients only realize the presence of bearing through communication between designers. In effect, one system designer tells the other designer about the presence, structure and definition of the bearing (client category three situation). Moving the data server to client category 2 essentially means we are identifying the individual structure and data item to the system. This identification may be permitted either explicitly or implicitly.

Explicit identification would result in the data server actually storing the code and definition of the data structures and items. The data server is presently capable of storing this type of information. However, it is unlikely the present suite of applications would be able to utilize this information to automatically access the data structures within the data server. An implicit identification means that the item is defined externally to the data server. The implicitness comes from the fact that the data and definition are no longer directly connected within the data server.

The SGXML dictionary structure is well documented [15] with the structure shown here in Figure 4. The structure consists of identifying information such as the dictionary owner, a proper citation for the dictionary, a general description of the dictionary, and an example of the date structure used within the dictionary. The example date structure was included to allow legacy systems the ability to describe their dictionary terms without modification of dates.

```
dictionary
├dictionary_owner [1]
├dictionary_citation [1]
├dictionary_description [1]
├date_structure  [0,1]
└dictionary_entry [1,n]
  ├ dictionary_term [1] {instance [0,1]}
  ├ role [1]
  ├ definition [1,n]
  │ ├ definition_owner [1]
  │ ├ short_name [1]
  │ ├ creation_date [0,1]
  │ ├ change_date [0,1]
  │ ├ metholodgy [0,1]
  │ ├ unit_of_measure [0,1]
  │ ├ min_value [0,1]
  │ ├ max_value [0,1]
  │ ├ null_representation [0,1]
  │ ├ accuracy [0,1]
  │ ├ authority_citation [1]
  │ └ codeset [0,n]
  │    ├codeset_name [1]
  │    ├code [1]
  │    └codeset_owner [1]
  └synonym [0,n]
    ├synonym_instance [1]
    ├synonym_term [1]
    └synonym_owner [1]
```

Figure 4: Schematic of the SGXML dictionary structure being proposed for the NUW project.

The dictionary then has entries, with one entry for each dictionary term. An entry is equivalent to a single entry in a language dictionary. Each entry has one term and one role. The term is equivalent to a word in a language dictionary while the role is similar to the function of the word in a language dictionary. As well, the dictionary term may have synonyms. Synonyms allow multi-language capabilities and contextual slang words to be associated with the dictionary term.

Each entry for a particular term may possess multiple definitions. This is also similar to a common language dictionary where a single word may possess multiple definitions. Further details on the structure may be found in [15].

5.3.1 Modifications to SGXML Dictionary Structure

The dictionary structure proposed for this application (Figure 4) has been modified slightly from the original SGXML dictionary structure. The reasons for these modifications are documented below.

5.3.1.1 Instance Element

The first change deals with the <instance> element. Initially, <instance> was incorporated in the dictionary structure to address the numeric count of definitions. In a common language dictionary, the numeric count distinguishes the definitions. In an XML environment, a similar distinction may be made by the occurrence of the definition element. As such, it is unlikely <instance> needs to be mandatory within the structure. As well, the instance should be directly part of the definition as instance applies to the particular definition element. Thus, the first modification is to move the <instance> element to be a non-mandatory attribute of the <definition> element. The non-mandatory occurrence is because counting the definitions is an optional method of identifying the definition instance.

5.3.1.2 Accuracy Element

The second revision deals with the <accuracy> element. The initial <accuracy> element was defined as type float and as mandatory. However, the dictionary structure was intended for multiple types of definition. For example, a code set used to identify countries could be placed in the dictionary. In this case, a single definition would identify a country such as Canada in the <short_name>. Then, the code for Canada (e.g., CA [18]) would be included in the <code> under <codeset>. Other codes for Canada (e.g., CAN or 124 [18]) could be identified using the multiple occurrence of <codeset>.

However, the mandatory requirement for <accuracy> does not pertain to this particular type of definition. There is no accuracy associated with the definition for Canada. Thus, the second modification to the structure is changing the occurrence on the <accuracy> element to be optional.

5.3.1.3 Multiplier Element

The final modification is in the <multiplier> element. The <multiplier> was added to the SGXML work in an attempt to show unit manipulation between different codes [19]. However, this <multiplier> implementation for dealing with all possible unit conversions is inappropriate. This is because the implementation requires the repetition of conversion factors within a single XML document.

As an example, the conversion of kilometres to kiloyards within the dictionary would require the conversion factor to be explicitly included in all definitions that required the conversion. This leads to potential errors due solely to the repetition of the same conversion factor throughout the dictionary. As well, a dictionary should not be expected to hold the plethora of possible conversion factors.

To correct this problem, the <multiplier> element must be removed from the dictionary structure. If the requirement exists for such a multiplication factor, then a separate XML conversion document should be created to contain the necessary conversions between units. The conversion document could also contain more complex conversions.

5.4 Defining an Ontology of Terms

The proposed SGXML dictionary structure is by no means the only potential implementation mechanism for vocabulary management. In other communities, vocabulary management is being conducted using the Ontology Web Language (OWL) [20] [21].

OWL is a language constructed to describe the meaning and relationships behind resources. OWL is an XML based language that also depends on constructs from the RDF and Resource Description Framework Schema (RDFS). OWL can be used to define a hierarchy of object classes with associated properties and data types. Relationship between classes can also be introduced.

OWL could be used to construct class-subclass relationships that provide a hierarchy for data term definition. OWL would also allow the formation of relationships between terms in multiple vocabularies. This would potentially be useful in a network-enabled coalition, where the coalition members each have defined vocabularies. The marine community [21] is attempting to use OWL for this very purpose. Another useful feature of an OWL implementation is the relative ease of searching up and down the hierarchy for related, but perhaps unknown, terminology. This also provides a discovery mechanism for terms in the vocabulary.

In the case of the NUW Project, OWL represents a slightly more complicated implementation path. However, OWL does offer a known standard, which would be useful when interfacing with coalition members in a network-enabled operation. From this perspective, OWL is a more scalable solution. However, as a demonstration project the NUW TDP is not intended to produce a final, fully-scalable product. Thus, for NUW the simplicity of the SGXML structure is more appropriate for implementation.

5.5 Unit Descriptions

Units play an important part in the measurement of a quantity. Understanding units is important in a subject area that is dependent on data produced by measurements. The unit provides a standard base, which is critical for the comparison of values. Of course in a networked environment, the potential for value comparison is increased because more clients have access to the data.

The units associated with the data values are also critical for data use. Networked data are potentially useful for many purposes, with each purpose often linked to a legacy, or existing, application. These applications were typically constructed in a manner that assumes incoming data have particular units. In this case, the conversion of quantities from one unit to another becomes important.

Units are commonplace, but surprisingly difficult to deal with properly. Units have been the cause of very public errors at well-known organisations in space programs [22] and commercial airlines [23]. Dealing with units is a non-trivial process and errors resulting from incorrect unit conversion are often serious.

The System International (SI) is the internationally recognized set of unit descriptors. In a truly net-centric solution, it is likely that the SI system would be adopted for unit nomenclature. However, there are many occasions when communities that deal with particular subject matter do not use SI units. This is the case with naval tactical systems, which often use units such as kiloyards, nautical miles or degrees, all of which are not official SI units (although nautical miles is a recognized unit). Thus, in the specialized development for the NUW TDP, we recognize the need for specialized unit nomenclature and relax the requirement for SI units.

6. Discovery Metadata Structure

As noted previously, a discovery vocabulary typically labels a group of data terms that is relevant to a particular subject area. The discovery vocabulary may be used or presented within a discovery metadata structure. In this respect, the discovery metadata structure represents a component of the resource, property and value model. In this model, the resource is the data asset. The properties of the asset are defined by the metadata structure. The values are actually the discovery vocabulary that is the content of the metadata structure.

6.1 US Department of Defense Net-Centric Data Strategy

The US released the Department of Defense (DOD) Net-Centric Data Strategy [24] in May 2003. The strategy outlines the DOD vision of how the communities-of-interest, metadata, and the Global Information Grid (GIG) will be combined to form the net-centric environment. The vision has two primary objectives:

- increasing the data that is available to communities or the Enterprise
- ensuring that data is usable by both anticipated and unanticipated users and applications

Given these objectives, the Strategy outlines seven approaches or goals that when met, will achieve the stated objectives. These goals are [24]:

- to make data visible
- to make data accessible
- to institutionalize data management
- to enable data to be understood
- to enable data to be trusted
- to support data interoperability, and
- to be responsive to user needs.

Metadata plans a central role in the goals of data visibility, accessibility, understanding the data content, data trust, interoperability and response to user needs. In recognizing the importance of metadata, the US DOD has also released the Department of Defence Discovery Metadata Specification (DDMS) [25] in support of the discovery process.

6.2 US Department of Defense Discovery Metadata Specification

The DDMS is a metadata specification that identifies and describes characteristics that are important for the description of a data asset. This type of description describes the asset as a single unit. For example, the asset may have an associated publisher; it may have a title; a creation date; etc. These attributes pertain to the asset as a whole and do not describe the content of the asset. This level of description supports the discovery of the asset and initial assessment of the asset's applicability of use.

The US DOD has identified this type of metadata as a requirement for the Network Information Grid. The DOD has documented the metadata requirements in the Department of Defense Discovery Metadata (DDMS) Specification. This specification has been evolving since April 2003, with the latest release in July 2005 (Version 1.3) [25].

The DDMS clearly states the intent is to provide metadata for the discovery of data assets at the marco or summary level. In an example involving a database, the DDMS provides metadata at the database level, with a description, owner, etc. This type of metadata essentially advertises the existence of the database, with broad descriptions of the data content. The detail of content, such as individual parameters, is not typically included at this level of metadata description.

The DDMS is very well aligned with the Dublin Core Metadata Initiative (DCMI) [26] specification, with extensions beyond the DCMI to address the particular business needs of the US DOD. As an example of the extensions, the DCMI element "Coverage" is defined as specifying the extent or scope of the resource. The DCMI also defines element refinements for Coverage that includes spatial and temporal coverage. The spatial and temporal refinements are also elements in the DCMI, but are used to narrow the scope of the coverage element.

As examples, the DDMS extends the coverage by introducing refinements that include geospatial coverage and virtual coverage. Geospatial coverage provides information on the reference frame of the coordinates used in the resource. Virtual coverage identifies the one or more addresses on a computer network where the asset is located. Note that this definition does not specify information about the content of the asset, but rather the virtual location of the asset. Other elements, such as security, have also been added in the DDMS.

Other specific components within the DDMS assist in meeting the goals of the Net-centric Data Strategy. For example, the DDMS "Security" element contains 18 security information items such as the classification of the data asset, who classified the asset, the data producer, release restrictions, dates of classification, and exemptions. All of this information supports the accessibility goal of the Net-Centric Data Strategy. The accessibility is realized only when access is controlled via appropriate security metadata.

As noted above, the DDMS is well aligned with the DCMI. Librarians and computer scientists developed DCMI to address issues associated with online libraries. Thus, the metadata structure of DCMI is similar to the metadata used to catalogue books in a library. In a similar way, DCMI use in the discovery process is similar to the use of a card catalogue in the process of finding a book in a library.

Other groups have examined the DCMI for use in metadata descriptions in other areas of research. In the marine community, the DCMI applicability to discovery of marine data sets was questioned because of its granularity. The marine community was exploring metadata requirements for the automated discovery of data assets. The DCMI provides a high level descriptive ability as is evident by the descriptive components such as “title”, “subject” and “description”. Within the DCMI, these elements are defined as user defined free-text. The free text form means there may not be a standard content or structure for the descriptions. This makes use by automated systems very difficult.

The lack of structure that results from free-text descriptions is acceptable in some cases. For example, user driven search engines can utilize free-text by examining the content to identify key words and to some extent, the context in which the words are used. However, for automated systems that wish to use and manipulate the content, the descriptions need to be parsed to obtain those data required. Since the descriptions are free-text, the problem is the lack of consistency among the source descriptions. Since the consistency is lacking among the data provides, the parsing of free-text into consistently meaningful data is very difficult. As well, if a system needs to parse data out of free-text descriptions, it is likely that the metadata structure does not properly support the query system.

These difficulties result in the content of the DCMI and the DDMS elements being either inaccessible or at the least, very difficult to access from automated applications. Note that the use of such descriptive free-form text is useful for discovery using user-driven search engines and crawlers. These mechanisms can search documents for pre-existing elements, obtain the content of the elements and present the search results to the user. It is then the users task to understand the search results. From this perspective, the DCMI metadata elements are very useful. However, for automated systems that are extracting and using the metadata, the use of the DCMI descriptions is more questionable.

Although there are problems with using the DCMI for automated systems, the DDMS extensions help to alleviate these problems. The DDMS have taken the free-text definitions and extended DCMI components to reduce the free-text. For example, the <subject> element within DCMI was extended in DDMS to include components for <category> and <keyword>. These components allow subject specification based on a controlled vocabulary (i.e., <category>) and natural language (i.e., <keyword>). The <category> element is suited to automatic systems searching for resources because the controlled vocabulary can be known and utilized by the automated system.

The DDMS is the US DOD standard for the GIG. As such, it is instructive for the NUW TDP to attempt to use the standard in an actual application. It is important to note that the DDMS deals specifically with the metadata associated with the resource. The implementation or method of storage of the metadata is not part of the DDMS. In this way, the DDMS represents a *content model*. As per the examples in the DDMS documentation, the implementation can involve free text, XML, HTML, etc. For the current project, the XML tools that support the DDMS should be utilized.

7. The Proposed Architecture

The goal of this investigation is to identify and document a potential data server implementation that would allow discovery of data server content in a network utilizing disparate systems. Before describing the proposed architecture, it is important to reiterate an important point raised in Section 2.1. The data server is quite capable of storing both data values and the associated names and descriptions for these data. Applications could be built to interface between the data server and those disparate systems, allowing these systems to query the data server to determine if particular named data items exist, locate those items and utilize the associated data. However, applications already built for the inhouse STB system would need to be modified to accommodate the data structures in the data server; because these structures would now be linked to data item names and descriptions. As well, the additional linking and searching would slow the applications accessing the data. Access speed was an important design consideration during the initial DS development. As an alternative, the data server could also support the storage of data objects – data and methods used to access those data. Using this approach, the inhouse applications could access the data objects and manipulate the data values based only on the methods within those objects. The application serving external disparate systems could provide the complete data object or data values from those objects.

Both of these approaches are viable and are in fact more general solutions to the problem of data discovery and access from the data server. However, both solutions involve considerable inhouse application modification. To avoid these modifications, as alternative approach is proposed here.

The proposed NUW architecture utilizes the concepts noted in previous sections. As an initial summary, the proposed architecture:

i. will address client category one access,

ii. will provide full definition of terms to clients,

iii. will provide metadata for the discovery of data assets, and

iv. will address the need for both internal and external data structures.

Figure 5 provides a schematic of the proposed architecture.

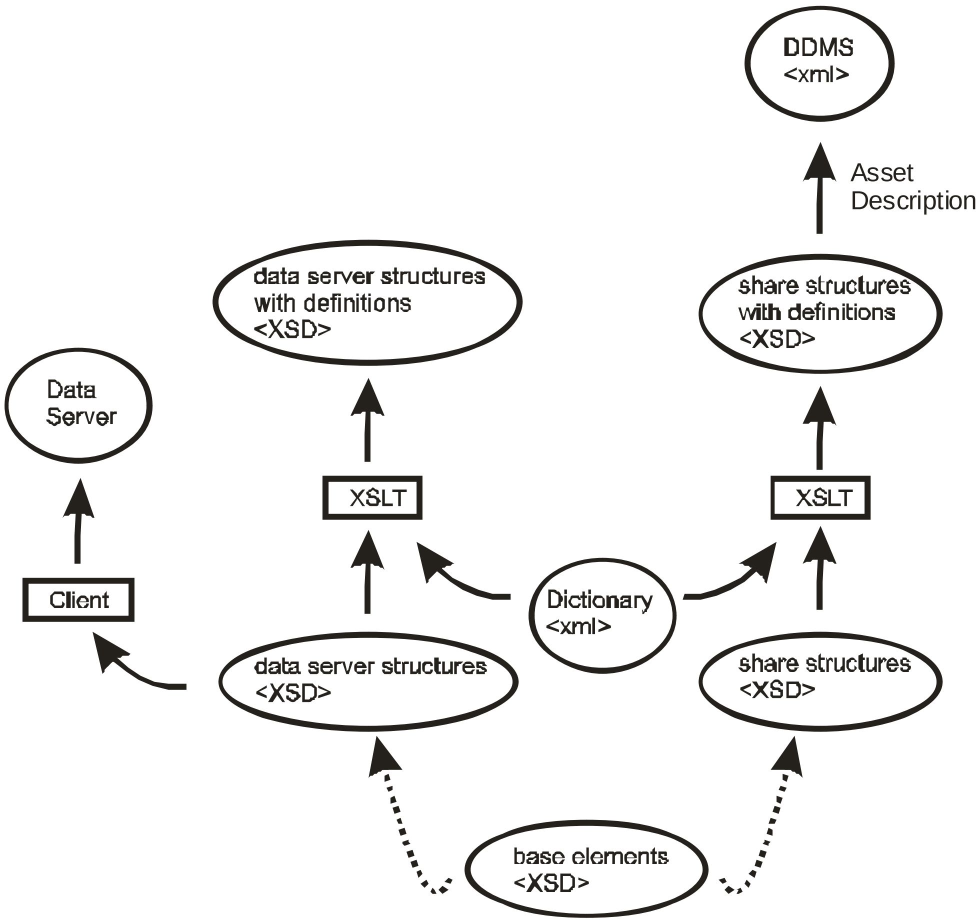

Figure 5: An architecture that takes advantage of the XML dictionary of terms and programmer created schemas. Programmers would create a set of base elements, including those elements in both the structures for private (data server structures) and public access (share structures). XSLT would be used to document the share structures using a consistent set of descriptions in the dictionary.

7.1 Dictionary and Structure Definition

The proposed architecture shown in Figure 5 has the base elements, the data server structures, shared structures, and the dictionary of data items and structures. The base elements, data server structures and shared structures are all described using XML schemas which are created by the program developers. The base element schema contains full definitions of all elements. The two structure schemas combine the elements into data structures to be used by either the data server or shared among platforms. Only elements defined in the base element schema may be used in the structure schemas.

An example of a section from a base element schema is shown in Figure 6. This shows a typical definition of an XML element, which can utilize the full schema functionality. A simple base element schema in a well-formed and valid XML document is provided in Annex A.

```
<?xml version="1.0"?>
<schema xmlns:drdc="http://www.drdc-rddc.gc.ca"
xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="frequency">
    <simpleType>
      <restriction base="float">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
```

Figure 6: A section of the base element schema that defines an element known as frequency.

The data server or shared structure schemas utilize the elements defined in the base element schema. This utilization is via an inclusion of the base element definitions within the data server or shared structure schemas. This is shown using a small section of the shared structures schema (Figure 7). The <xsd:include> statement allows access to all elements defined within the base elements schema.

This type of referencing provides considerable control over the data items used in the data server or shared structure schemas. For example, once a particular element is defined in base elements schema it would be consistently used across the data server and shared structures schemas. This is because only one definition of the data item exists – that being in base elements. Figure 7 shows how the frequency element is referenced in the shared structures schema, with the reference being to the defined frequency element in base elements. A well-formed and valid XML document for the shared structure schema is shown in Annex B.

The base elements schema provided an XML definition of a element. Next, consider the description of the frequency element. As with the definition, there is only one description of the data item. This description is contained in the dictionary document. Following the SGXML dictionary format, the description of the frequency data item is shown in Figure 8.

The dictionary entry in Figure 8 describes frequency. There are four pieces of information in the dictionary entry that should be highlighted. First, the <role> element is being used to identify an individual data item or a data structure. In Figure 8, the <role> has a value of ‘item’ indicating that the term ‘frequency’ is a data item. Second, the optional <min_value> element describes the minimum inclusion values for the data item. These values place limits on the expected content of the data item being defined. Third, the <code> defines the actual name of the data item. Finally, the <drdc:type> defines the computer-based typing of the data item. A well-formed and valid XML document for the dictionary is shown in Annex C.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:include schemaLocation="base_elements.xsd"/>
      <xsd:complexType name="waveform_type_x">
            <xsd:sequence>
                  <xsd:element ref="frequency"/>
                  <xsd:element ref="duration"/>
                  <xsd:element ref="cpa_time"/>
            </xsd:sequence>
      </xsd:complexType>
```

Figure 7: A section of the shared structure schema that utilizes the frequency element defined in the base elements schema.

```
<dictionary_entry>
      <dictionary_term>frequency</dictionary_term>
      <role>item</role>
      <definition instance="1">
            <definition_owner>DRDC</definition_owner>
            <short_name>band centre frequency</short_name>
            <creation_date>2005-03-09</creation_date>
            <change_date>2005-03-09</change_date>
            <methodology>The centre frequency of a band is computed from
the geometric mean of the lower and upper cutoff frequencies of the
band.</methodology>
            <unit_of_measurement>s^-1</unit_of_measurement>
            <min_value>0</min_value>
            <codeset>
                  <codeset_name>STB Codes</codeset_name>
                  <code>frequency</code>
                  <codeset_owner>STB</codeset_owner>
                  <drdc:type>float</drdc:type>
            </codeset>
      </definition>
</dictionary_entry>
```

Figure 8: A section of the dictionary that describes the frequency data item.

The final piece is the XSLT code (provided in Annex D). This code combines the term description from the dictionary with the schema definition from the data server or shared structure schemas. The result is a documented schema available for both the private and shared structures.

The documentation method utilizes the <annotation> and <documentation> elements within the schema definition. The <documentation> element is flexible in its allowed content, allowing both free text and structured content. In this case, the structured descriptions will be obtained from the dictionary. In this case, we examine the dictionary content for a particular term. The content within the dictionary <definition> element is included in the documented schema within the <documentation> element. Together this could appear in the schema document as shown in Figure 9. A well-formed and valid XML document for the documented shared structure schema is shown in Annex E.

This provides the advantage of prompting those creating the data items with the content expected within the documentation. It also provides content that can be easily manipulated by the XML parsers.

```
<xsd:element ref="frequency">
  <xsd:annotation>
    <xsd:documentation>
      <definition instance="1">
        <definition_owner>DRDC</definition_owner>
        <short_name>band centre frequency</short_name>
        <creation_date>2005-03-09</creation_date>
        <change_date>2005-03-09</change_date>
        <methodology>The centre frequency of a band is computed from the
geometric mean of the lower and upper cutoff frequencies of the
band.</methodology>
        <unit_of_measurement>s^-1</unit_of_measurement>
        <min_value>0</min_value>
        <codeset>
          <codeset_name>STB Codes</codeset_name>
          <code>frequency</code>
          <codeset_owner>STB</codeset_owner>
          <drdc:type>float</drdc:type>
        </codeset>
      </definition>
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

Figure 9: A section of the documented shared structure schema.

Once created, the documented schemas are validated. Validation is the process where all data items present in the schema are checked for consistency of definition and form. Online or application tools are available for validating an XML document against an XSD. In this cause, the validation uses the XSD file and validates against the allowed rules for creating a schema.

Maintaining the primary source of definitions within the dictionary has advantages to maintaining this within the schema. First, multiple uses of the data items will only need to be documented once in the dictionary. The XSLT processing can automatically verify the existence of the term in the dictionary. Also, the dictionary provides separation of the definition from the schema and thereby places importance on dictionary maintenance as a function in itself.

The data server schema can then be used to describe the content of an operational instance of a data server. This type of server structure description may also assist system designers trying to access the content of the data server. At present, the structure design for data server structures is contained in the write statements of the programs writing data to the DS. Using an XML-based definition of the data server structures, complete with data item definitions, would assist in the communication of structure information between designers.

7.2 DDMS Usage

Two resources are considered for description using the DDMS: the share structures and the dictionary of terms. Services may also be described using the DDMS, but the requirements for specific services have not been dealt with in this study.

As noted previously, the DDMS specification provides metadata for the data assets available through the network. A single DDMS XML document may be used to specify the metadata associated with a single data asset. The single asset nature of the description is a result of the XSD definitions that are provided as an example implementation of the DDMS.

The exact implementation of the DDMS in a system with multiple resources is not actually specified in the DDMS or example XSD. The DDMS only describes the metadata characteristics that are identified as important descriptors for a resource. The XML implementation of the DDMS is represented by the XSD document that defines the components of the DDMS in terms of XML elements. However, the schema definition by default specifies the placement and relationships between elements in XML documents that comply with the XSD representation of the DDMS. This complicates the issue slightly, as the XML implementation has resulted in additional constraints on the metadata definition simply as a result of the XML implementation and not a result of the specification.

To describe multiple assets, we could use the DDMS metadata schema to build a NUW specific schema for multiple resources. For example, we could package the DDMS <Resource> elements into a user-created metadata structure. The XML document that validates against the new schema would not be a DDMS document, but may be considered a DDMS catalogue. The problem here is that no formal specification exists for the XML DDMS catalogue.

A second option is to describe the metadata assets as themselves, collectively, being an asset. In this approach, each data asset would have an associated metadata description contained in an XML document. This XML document would validate against the schema provided with the DDMS. In an environment of n data assets, we would have n XML documents containing the metadata descriptions. Then, a final XML document is added (now we have n+1 documents) which collectively describes all the XML metadata documents as a single asset. The collective XML document identifies the individual XML documents. The grouping of the collective is arbitrary and thus may be based on location, topic, etc. In this way, a DDMS XML document can be used to list the assets available via other DDMS compliant documents.

The XML implementation of the collective DDMS may also be used to describe the services that are available to external applications. These services would then be described further in specific DDMS XML documents. For example, one service may provide the shared structure schema. Another service may provide definitions of the terms in the shared structures, with the definitions originating from the dictionary.

However, this collective metadata implementation does stretch the interpretation of the specification. The DDMS identifier category would need to include multiple identifications for the multiple resources. However, the identifier category was intended for multiple identifications of a single resource, not single identifications of multiple resources.

Another option also exists for describing multiple resources. The OWL could be used to define the group of resources that represent the collective resource. This would allow the DDMS defined metadata documents to validate against the DOD provided XSD, while maintaining the view of multiple resources collectively being considered as a single resource.

The complexity of introducing another language such as OWL into the NUW data sharing architecture seems to be a slight over complication. The requirement to populate multiple resources into a single, searchable, catalogue is adequately met by including a root level XML tag that encapsulates all the DDMS compliant <Resource> elements. This appears to be a viable approach to the multiple resource problem.

8. Concluding Remarks

This work has attempted to outline data sharing issues that will be important in the net-centric or network enabled paradigm of operations. The Networked Underwater Warfare Technology Demonstration Project provides a focal point and implementation mechanism for the development of ideas surrounding these data sharing issues.

The introduction of the NUW project in relation to the CF Target Integration Model provides the high level context for what NUW is attempting to accomplish. The NUW Project is delving into many of the TIM components and in doing so, is exploring new concepts related to the connectivity of military platforms such as MPAs and submarines. Some of the data sharing complexity between these platforms is dealt with in this work.

In a networked environment, many clients entering the network will likely be unaware of the data assets that are available in the network. A discovery process will be required for the client to first identify the available resources. Once identified, the client will require information on the structure used to deliver the data to the client. Note that this is not necessarily the structure used to house the data in the source system.

Once the structure is identified, the client will likely need information on the details of the data items present within the structure. In this work, we propose the inclusion of this type of information in a dictionary of terms, which will define the various data items present in the shared structures. Services can be used as a client interface to the dictionary. However, the important point is not the service, but the availability of definitions to aid client understanding, to allow client judgements on the asset, and to build client trust in the data source.

The community working on network-enabled applications must realise the importance of metadata content and structure. The system described in this work is in reality, a system of metadata. The XML documents that describe the dictionary, the data structures, the schemas and the asset descriptions, are forms of metadata. The actual data to be shared in the NUW Project has not been described here.

The flexibility of creation and use of XML documents for controlling the metadata descriptions makes XML a viable implementation mechanism. However, XML does introduce the overhead of tagging the content. Often, the tags can occupy a significant amount of space in the resulting data file. Data volume issues will be a consideration in NUW as data will be moving through non-wired networks. However, if the intent of the NUW project is to research the network-based sharing of data assets, then data volume issues should not be used as a reason for dismissing an XML implementation. Other potential solutions may exist to alleviate the volume problem, while maintaining the use of XML. For example, smart transfers (e.g., sending only data updates), tag compression or file compression may be used to reduce data volumes.

The fundamental problem addressed in this work is the sharing of information related to the meaning of data items and data structures. This is required to provide a level of understanding for the clients accessing the data. However, the proposed architecture involving the STB data server introduces one key challenge. Past implementations of the data server has utilized considerable developer-to-developer communication. This has resulted in the data structures that are not fully described within the data server. This type of implementation enhances the speed of data access but does not promote a developer-independent sharing of the meaning of the data within the structures.

In present implementations, the definition of the data content isn't contained within the data server. Presently, content definition originates with either the data server configuration file or program write statements. However, this information is not passed to the data server. If we are going to liberate the content of the data server to allow discovery of this content, then we need to have in place a mechanism that first liberates the structure of the content. The structure and associated data definitions will allow clients an understanding, and thus a judgement, as to the usefulness of the data.

We must also be careful that we avoid the creation of an architecture that requires two or more independent sources to create the data server structures and client accessible structure information. Ideally, those creating the structure will only have to update or modify one description of the structure. This one modification will then flow through the system, updating the data server as well as any structures documented for external clients.

We also have to be cognisant of the data server acting as the receiver of information. In a fully networked operation, it is conceivable the data server could take the role of data receiver. In this case, the foreign source would be defining both the structure and data definitions. This introduces an assortment of problems. In this case, a software layer would likely be introduced to transform the incoming data stream into a form suitable for storage in the data server. Other applications would need to then recognise the presence of these new data, and utilize these data in the processing.

The incorporation of the STB data server into an operation based on the network-enabled paradigm presents many challenges. Metadata will play a key support role in this process of understanding the data and structure descriptions. These concepts will be critical to moving the paradigm toward a fully interoperable suite of processes capable of utilizing data assets from heterogeneous sources.

References

[1] Lefrancois, Marcel. (2004). Information Exchange Requirements for Networked Underwater Warfare. (DRDC Atlantic TM 2004-168). Defence R&D Canada – Atlantic. October 2004.

[2] Pederson, Richard. (2005). NUW Project Status Meeting Presentation. (unpublished presentation). June 11 2005.

[3] C4ISR Campaign Plan Interim Report Target Integration Model 2008. (2005). Annex G. June 19 2005.

[4] Canadian Forces C4ISR Campaign Plan. (2003). C2I2 JCAT Secretariat OPI: Director Joint Force Capabilities (DJFC). June 27 2003.

[5] Shaping the Future of the Canadian Forces: A Strategy for 2020. (1999). June 1999.

[6] DRDC C4ISR Coordination Workshop. (2005). Ottawa, Canada. June 14, 2005.

[7] Report of the Auditor General of Canada to the House of Commons. (2005). Chapter 4 National Defence - C4ISR Initiative in Support of Command and Control. April 2005.

[8] Sommerville, Ian. (1996). Software Engineering. Addison Wesley Longman Limited. p.742.

[9] Sonar Test Bed Reference Guide. (2002). DRAFT Version. 08 October 2002.

[10] Lawrence, S. (2003). Description of the Sonar Test Bed Data Architecture. DRDC Atlantic CR 2003-042. Defence R&D Canada – Atlantic.

[11] Lawrence B.N., R. Cramer, M. Gutierrez, K. Kleese van Dam, S. Kondapalli, S. Latham, R. Lowry, K. O'Neill and A.Woolf. (2003). Proceedings of the U.K. e-science All Hands Meeting, 2003. S.J.Cox(Ed) ISBN 1-904425-11-9, see http://ndg.nerc.ac.uk/publications.htm . (Access Date: September 13, 2005)

[12] O'Neill K., R. Cramer, M. Gutierrez, K. Kleese van Dam, S. Kondapalli, S. Latham, B.N. Lawrence, R. Lowry and A.Woolf. (2003). Proceedings of the U.K. e-science All Hands Meeting, 2003. S.J.Cox(Ed) ISBN 1-904425-11-9, see http://ndg.nerc.ac.uk/publications.htm. (Access Date: September 13, 2005)

[13] Lawrence, Bryan, David Boyd, Kerstin Kleese van Dam, Roy Lowry, Dean Williams, Mike Fiorino, and Bob Drach. (2002). The NERC DataGrid. (Project Proposal submitted to NERC on the 31st of January 2002).

[14] Graybeal, John and Luis Bermudez. (2005). Marine Metadata Interoperability - Adding Science Value. See http://marinemetadata.org/. (Access Date: September 13, 2005).

[15] Isenor, Anthony W. and Roy K. Lowry. (2004). Final Report of the ICES/IOC Study Group on the Development of Marine Data Exchange Systems using XML. DRDC Atlantic ECR 2005-005. Defence R&D Canada – Atlantic.

[16] Bermudez , Luis E. (2004). ONTOMET: Ontology Metadata Framework. Phd Thesis. Drexel University. December 2004.

[17] Manola, Frank and Eric Miller (Editors). (2004). RDF Primer. W3C Recommendation. 10 February 2004.

[18] ISO 3166 Code Lists. (2005). See http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/index.html. (Access date: June 27 2005).

[19] Isenor, Anthony W. (2003). XML Based Manipulation of Codes Exchanged Between Data Systems. DRDC Atlantic TM 2003-132. Defence R&D Canada – Atlantic. August 2003.

[20] Smith, Michael K., Chris Welty and Deborah L. McGuinness. (2004). OWL Web Ontology Language. http://www.w3.org/2004/OWL/ (Access date: July 2005).

[21] Bermudez, Luis, John Graybeal, Anthony W. Isenor, Roy Lowry, and Dawn Wright. (2005). Construction Of Marine Vocabularies In The Marine Metadata Interoperability Project. MTS/IEEE Oceans 2005. Washington DC. September 18-23 2005.

[22] Isbell, Douglas, Mary Hardin, Joan Underwood. Mars Climate Orbiter Team Finds Likely Cause of Loss. Press Release at http://mars.jpl.nasa.gov/msp98/news/mco990930.html (Access date: July 2005).

[23] Nelson, Wade H. The Gimli Glider. (Copyright WHN 1997). Soaring Magazine. http://www.wadenelson.com/gimli.html (Access date: July 2005)

[24] Department of Defense Net-Centric Data Strategy. (2003). US Department of Defense Chief Information Officer. May 9 2003.

[25] Department of Defense Discovery Metadata Specification (DDMS). (2005). Deputy Assistant Secretary of Defense. 29 July 2005.

[26] Dublin Core Metadata Initiative. (2005). See http://dublincore.org/. (Access date: June 27 2005).

This page intentionally left blank.

Annex A Example Base Element Schema

```
<?xml version="1.0"?>
<schema xmlns:drdc="http://www.drdc-rddc.gc.ca"
xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="frequency">
    <simpleType>
      <restriction base="float">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
  <element name="modulation_frequency">
    <simpleType>
      <restriction base="int">
        <minInclusive value="0"/>
        <maxInclusive value="200"/>
      </restriction>
    </simpleType>
  </element>
  <element name="duration" type="date"/>
  <element name="bandwidth">
    <simpleType>
      <restriction base="float">
        <maxInclusive value="300"/>
      </restriction>
    </simpleType>
  </element>
  <element name="weight">
    <simpleType>
      <restriction base="float">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
  <element name="amplitude">
    <simpleType>
      <restriction base="float">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
  <element name="check_sum" type="int"/>
  <element name="sequence" type="int"/>
  <element name="bytes" type="int"/>
  <element name="status" type="int"/>
  <element name="status_reserved" type="int"/>
  <element name="modulation_index" type="float"/>
  <element name="name_bytes" type="int"/>
  <element name="type_name_bytes" type="int"/>
  <element name="shading_name_bytes" type="int"/>
  <element name="name_reserved" type="int"/>
  <element name="name" type="string"/>
  <element name="type_name" type="string"/>
  <element name="shading_name" type="string"/>
  <element name="delay">
    <simpleType>
```

```
      <restriction base="float">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
  <element name="check_word">
    <simpleType>
      <restriction base="int">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
  <element name="time">
    <simpleType>
      <restriction base="double">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
  <element name="sensor_time">
    <simpleType>
      <restriction base="double">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
  <element name="cpa_time">
    <simpleType>
      <restriction base="double">
        <minInclusive value="0"/>
      </restriction>
    </simpleType>
  </element>
</schema>
```

Annex B Shared Structure XML Schema

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="base_elements.xsd"/>
  <xsd:element name="share_structures">
     <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="waveform_y" type="waveform_type_y"/>
          <xsd:element name="tactical" type="waveform_type_y"/>
        </xsd:sequence>
     </xsd:complexType>
   </xsd:element>


   <xsd:complexType name="waveform_type_y">
      <xsd:sequence>
          <xsd:element ref="frequency"/>
          <xsd:element ref="duration"/>
           <xsd:element ref="cpa_time"/>
      </xsd:sequence>
   </xsd:complexType>


   <xsd:complexType name="waveform_type_x">
      <xsd:sequence>
         <xsd:element ref="frequency"/>
         <xsd:element ref="amplitude"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

Annex C Dictionary of Terms


```
<?xml version="1.0"?>
<dictionary
xsi:noNamespaceSchemaLocation="file://c:\Anthony\Projects\NUW\Data_server_discov
ery\XML\parameter_dictionary_v2.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:drdc="http://www.drdc-rddc.gc.ca">
	<dictionary_owner>DRDC Atlantic</dictionary_owner>
	<dictionary_citation/>
	<dictionary_description/>
	<date_structure/>
	<dictionary_entry>
		<dictionary_term>waveform_type_y</dictionary_term>
		<role>structure</role>
		<definition instance="1">
			<definition_owner>STB</definition_owner>
			<short_name>waveform type</short_name>
			<creation_date>2005-04-07</creation_date>
			<change_date>2005-04-07</change_date>
			<methodology>A particular waveform used to investigate
characteristics of underwater sonar return signals.</methodology>
			<codeset>
				<codeset_name/>
				<code>waveform_type</code>
				<codeset_owner/>
			</codeset>
		</definition>
	</dictionary_entry>
	<dictionary_entry>
		<dictionary_term>wavetrain_data_type</dictionary_term>
		<role>structure</role>
		<definition instance="1">
			<definition_owner>STB</definition_owner>
			<short_name>wavetrain</short_name>
			<creation_date>2005-04-07</creation_date>
			<change_date>2005-04-07</change_date>
			<methodology/>
			<codeset>
				<codeset_name/>
				<code>wavetrain_data_type</code>
				<codeset_owner/>
			</codeset>
		</definition>
	</dictionary_entry>
	<dictionary_entry>
		<dictionary_term>frequency</dictionary_term>
		<role>item</role>
		<definition instance="1">
			<definition_owner>DRDC</definition_owner>
			<short_name>band centre frequency</short_name>
			<creation_date>2005-03-09</creation_date>
			<change_date>2005-03-09</change_date>
			<methodology>The centre frequency of a band is computed from
the geometric mean of the lower and upper cutoff frequencies of the
band.</methodology>
			<unit_of_measurement>s^-1</unit_of_measurement>
```

```
                        <min_value>0</min_value>
                        <codeset>
                                <codeset_name>STB Codes</codeset_name>
                                <code>frequency</code>
                                <codeset_owner>STB</codeset_owner>
                                <drdc:type>float</drdc:type>
                        </codeset>
                </definition>
                <definition instance="2">
                        <definition_owner>STB</definition_owner>
                        <short_name>modulation frequency</short_name>
                        <creation_date>2005-03-09</creation_date>
                        <change_date>2005-03-09</change_date>
                        <methodology/>
                        <unit_of_measurement>s^-1</unit_of_measurement>
                        <min_value>0</min_value>
                        <max_value>200</max_value>
                        <codeset>
                                <codeset_name/>
                                <code>modulation_frequency</code>
                                <codeset_owner/>
                                <drdc:type>int</drdc:type>
                        </codeset>
                </definition>
        </dictionary_entry>
        <dictionary_entry>
                <dictionary_term>duration</dictionary_term>
                <role>item</role>
                <definition instance="1">
                        <definition_owner>STB</definition_owner>
                        <short_name>duration in time of ping</short_name>
                        <creation_date>2005-03-09</creation_date>
                        <change_date>2005-03-09</change_date>
                        <methodology>The temporal duration of a sound source.  Used
to indicate the length of time an acoustic ping is released into the
water.</methodology>
                        <unit_of_measurement>s</unit_of_measurement>
                        <codeset>
                                <codeset_name/>
                                <code>duration</code>
                                <codeset_owner/>
                                <drdc:type>date</drdc:type>
                        </codeset>
                </definition>
        </dictionary_entry>
        <dictionary_entry>
                <dictionary_term>amplitude</dictionary_term>
                <role>item</role>
                <definition instance="1">
                        <definition_owner>STB</definition_owner>
                        <short_name>amplitude of waveform</short_name>
                        <creation_date>2005-04-07</creation_date>
                        <change_date>2005-04-07</change_date>
                        <methodology>The amplitude of a waveform.  Defined as 1/2
the peak-to-peak oscillation.</methodology>
                        <unit_of_measurement></unit_of_measurement>
                        <min_value>0</min_value>
                        <codeset>
                                <codeset_name/>
                                <code>amplitude</code>
```

```
                        <codeset_owner/>
                        <drdc:type>float</drdc:type>
                </codeset>
        </definition>
</dictionary_entry>
<dictionary_entry>
        <dictionary_term>check_sum</dictionary_term>
        <role>item</role>
        <definition instance="1">
                <definition_owner>STB</definition_owner>
                <short_name>check_sum</short_name>
                <creation_date>2005-04-14</creation_date>
                <change_date>2005-04-14</change_date>
                <methodology/>
                <unit_of_measurement></unit_of_measurement>
                <codeset>
                        <codeset_name/>
                        <code>check_sum</code>
                        <codeset_owner/>
                        <drdc:type>int</drdc:type>
                </codeset>
        </definition>
</dictionary_entry>
<dictionary_entry>
        <dictionary_term>time</dictionary_term>
        <role>item</role>
        <definition instance="1">
                <definition_owner>STB</definition_owner>
                <short_name>time from ship 1</short_name>
                <creation_date>2005-03-09</creation_date>
                <change_date>2005-03-09</change_date>
                <methodology/>
                <unit_of_measurement>s</unit_of_measurement>
                <min_value>0</min_value>
                <codeset>
                        <codeset_name/>
                        <code>time</code>
                        <codeset_owner/>
                        <drdc:type>double</drdc:type>
                </codeset>
        </definition>
        <definition instance="2">
                <definition_owner>STB</definition_owner>
                <short_name>time from ship 2</short_name>
                <creation_date>2005-03-09</creation_date>
                <change_date>2005-03-09</change_date>
                <methodology/>
                <unit_of_measurement>s</unit_of_measurement>
                <min_value>0</min_value>
                <codeset>
                        <codeset_name/>
                        <code>sensor_time</code>
                        <codeset_owner/>
                        <drdc:type>double</drdc:type>
                </codeset>
        </definition>
        <definition instance="3">
                <definition_owner>STB</definition_owner>
                <short_name>time from aircraft</short_name>
                <creation_date>2005-03-09</creation_date>
```

```
                    <change_date>2005-03-09</change_date>
                    <methodology>Time as indicated from the Canadian Patrol
Aircraft.</methodology>
                    <unit_of_measurement>s</unit_of_measurement>
                    <min_value>0</min_value>
                    <codeset>
                           <codeset_name/>
                           <code>cpa_time</code>
                           <codeset_owner/>
                           <drdc:type>double</drdc:type>
                    </codeset>
             </definition>
      </dictionary_entry>
</dictionary>
```

Annex D XSLT Code to Generate Documented Structure Schema

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:drdc="http://www.drdc-
rddc.gc.ca" version="1.0">

<xsl:output method="xml" indent="no"/>

<xsl:template match="*|@*|text()">
	<xsl:copy>
		<xsl:apply-templates select="*|@*|text()"/>
	</xsl:copy>
</xsl:template>

<xsl:template match="xsd:complexType">
	<xsl:variable name="nameatt" select="@name"/>
	<xsl:copy>
		<xsl:apply-templates select="@*"/>
		<xsl:element name="xsd:annotation">
			<xsl:element name="xsd:documentation">
				<xsl:copy-of
select="document('dictionary.xml')/dictionary/dictionary_entry[dictionary_term=$
nameatt]"/>
			</xsl:element>
		</xsl:element>
		<xsl:apply-templates select="xsd:sequence"/>
	</xsl:copy>
</xsl:template>

<xsl:template match="xsd:sequence">
	<xsl:copy>
		<xsl:apply-templates select="xsd:element"/>
	</xsl:copy>
</xsl:template>

<xsl:template match="xsd:element">
	<xsl:variable name="nameref" select="@ref"/>
	<xsl:copy>
		<xsl:apply-templates select="@*"/>
		<xsl:element name="xsd:annotation">
			<xsl:element name="xsd:documentation">
				<xsl:copy-of
select="document('dictionary.xml')/dictionary/dictionary_entry/definition[codese
t/code=$nameref]"/>
			</xsl:element>
		</xsl:element>
	</xsl:copy>
	</xsl:template>
</xsl:stylesheet>
```

Annex E Fully Documented Structure Schema

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:drdc="http://www.drdc-rddc.gc.ca">
  <xsd:include schemaLocation="base_elements.xsd"/>
  <xsd:element name="share_structures">
    <xsd:annotation>
      <xsd:documentation/>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="waveform_type_y">
    <xsd:annotation>
      <xsd:documentation>
        <dictionary_entry>
          <dictionary_term>waveform_type_y</dictionary_term>
          <role>structure</role>
          <definition instance="1">
            <definition_owner>STB</definition_owner>
            <short_name>waveform type</short_name>
            <creation_date>2005-04-07</creation_date>
            <change_date>2005-04-07</change_date>
            <methodology>A particular waveform used to investigate
characteristics of underwater sonar return signals.</methodology>
            <codeset>
              <codeset_name/>
              <code>waveform_type</code>
              <codeset_owner/>
            </codeset>
          </definition>
        </dictionary_entry>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element ref="frequency">
        <xsd:annotation>
          <xsd:documentation>
            <definition instance="1">
              <definition_owner>DRDC</definition_owner>
              <short_name>band centre frequency</short_name>
              <creation_date>2005-03-09</creation_date>
              <change_date>2005-03-09</change_date>
              <methodology>The centre frequency of a band is computed from the
geometric mean of the lower and upper cutoff frequencies of the
band.</methodology>
              <unit_of_measurement>s^-1</unit_of_measurement>
              <min_value>0</min_value>
              <codeset>
                <codeset_name>STB Codes</codeset_name>
                <code>frequency</code>
                <codeset_owner>STB</codeset_owner>
                <drdc:type>float</drdc:type>
              </codeset>
            </definition>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
```

```
      <xsd:element ref="duration">
        <xsd:annotation>
          <xsd:documentation>
            <definition instance="1">
              <definition_owner>STB</definition_owner>
              <short_name>duration in time of ping</short_name>
              <creation_date>2005-03-09</creation_date>
              <change_date>2005-03-09</change_date>
              <methodology>The temporal duration of a sound source.  Used to
indicate the length of time an acoustic ping is released into the
water.</methodology>
              <unit_of_measurement>s</unit_of_measurement>
              <codeset>
                <codeset_name/>
                <code>duration</code>
                <codeset_owner/>
                <drdc:type>date</drdc:type>
              </codeset>
            </definition>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element ref="cpa_time">
        <xsd:annotation>
          <xsd:documentation>
            <definition instance="3">
              <definition_owner>STB</definition_owner>
              <short_name>time from aircraft</short_name>
              <creation_date>2005-03-09</creation_date>
              <change_date>2005-03-09</change_date>
              <methodology>Time as indicated from the Canadian Patrol
Aircraft.</methodology>
              <unit_of_measurement>s</unit_of_measurement>
              <min_value>0</min_value>
              <codeset>
                <codeset_name/>
                <code>cpa_time</code>
                <codeset_owner/>
                <drdc:type>double</drdc:type>
              </codeset>
            </definition>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="waveform_type_x">
    <xsd:annotation>
      <xsd:documentation/>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element ref="frequency">
        <xsd:annotation>
          <xsd:documentation>
            <definition instance="1">
              <definition_owner>DRDC</definition_owner>
              <short_name>band centre frequency</short_name>
              <creation_date>2005-03-09</creation_date>
              <change_date>2005-03-09</change_date>
              <methodology>The centre frequency of a band is computed from the
```

```
geometric mean of the lower and upper cutoff frequencies of the
band.</methodology>
              <unit_of_measurement>s^-1</unit_of_measurement>
              <min_value>0</min_value>
              <codeset>
                <codeset_name>STB Codes</codeset_name>
                <code>frequency</code>
                <codeset_owner>STB</codeset_owner>
                <drdc:type>float</drdc:type>
              </codeset>
            </definition>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element ref="amplitude">
        <xsd:annotation>
          <xsd:documentation>
            <definition instance="1">
              <definition_owner>STB</definition_owner>
              <short_name>amplitude of waveform</short_name>
              <creation_date>2005-04-07</creation_date>
              <change_date>2005-04-07</change_date>
              <methodology>The amplitude of a waveform.  Defined as 1/2 the
peak-to-peak oscillation.</methodology>
              <unit_of_measurement/>
              <min_value>0</min_value>
              <codeset>
                <codeset_name/>
                <code>amplitude</code>
                <codeset_owner/>
                <drdc:type>float</drdc:type>
              </codeset>
            </definition>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

List of symbols/abbreviations/acronyms/initialisms

C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CF	Canadian Forces
COE	Common Operating Environment
COP	Common Operating Picture
CORBA	Common Object Request Broker Architecture
CSML	Climate Science Modelling Language
D	Discovery (type of metadata defined by NDG)
DCMI	Dublin Core Metadata Initiative
DDMS	Defence Discovery Metadata Specification
DIF	Document Interchange Format
DND	Department of National Defence (Canada)
DOD	Department of Defense (US)
DRDC	Defence Research and Development Canada
DRP	Document Review Panel
DS	Data Server
GCCS	Global Command and Control System
GCMD	Global Change Master Directory
GIG	Global Information Grid
HTML	Hypertext Markup Language
ICES	International Council for the Exploration of the Seas
IOC	Intergovernmental Oceanographic Commission
IP	Internet Protocol
ISO	International Organisation for Standardization
JC3IEDM	Joint Consultation Command & Control Information Exchange Data Model
LC2IEDM	Land Command and Control Information Exchange Data Model
METOC	Meteorology and Oceanography Centre
MOLES	Metadata Objects for Links in Environmental Science
MPA	Maritime Patrol Aircraft
MMI	Marine Metadata Interoperability
NDG	NERC DataGrid
NERC	Natural Environmental Research Council

NEOps	Network Enabled Operations
NGO	Non-Governmental Organisation
NUW	Networked Underwater Warfare (Technology Demonstration Project underway at DRDC Atlantic)
OGD	Other Government Department
OWL	Ontology Web Language
PDV	Parameter Discovery Vocabulary
PUV	Parameter Usage Vocabulary
R&D	Research & Development
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SGXML	Study Group on XML (official name: ICES/IOC Study Group on the Development of Marine Data Exchange Systems Using XML)
STB	System Test Bed (previously known as the Sonar Test Bed)
TDP	Technology Demonstration Project
TIM	Target Integration Model
TM	Technical Memorandum
UK	United Kingdom
US	United States
UWW	Underwater Warfare
W3C	World Wide Web Consortium
XBT	eXpendable BathyThermograph
XML	eXtensible Markup Language
XSD	XML Schema Definition
XSLT	eXtensible Stylesheet Language Transformation

Glossary

application

a piece of processing software that executes on a computer.

architecture

the method and design of a structure, process or application, or some collection of these. An architecture outlines a plan for the construction of the structure, process or application.

authentication

the process of verifying that the requesting client is indeed who they claim to be.

authorization

the process of determining if the authenticated client is permitted to access the data being requested.

central archive data model

a data storage model where one location is responsible for the assimilation of data collected for a programme.

client

collectively, refers to a user or application that places particular demands on a system.

client categorization model

a model used to group similar clients. In this work, the grouping is based on the level of knowledge the client possesses when accessing the system.

content model

a description of the data and information that is applicable to a specific topic. In terms of a *data space*, the content model describes all subject matter, such as individual data items, that make up the data space.

controlled vocabulary

a managed vocabulary.

data asset

a resource that includes both the data and the functions available to support the resource.

data identification

The process of searching and finding the data that is required for the particular process or activity.

data relevance model

a conceptual view and description of a process that is used to identify data that are important to a particular client, based on a client requirement.

data space

an abstract space that contains all data that pertains to the subject matter. All data that are relevant to the subject matter are part of the data space associated with that subject.

dictionary

a list of terms or names important to a particular subject or activity along with discussion of their meanings and applications

discovery

the process of searching and finding the data that is required for the particular process or activity.

extraction

the process of retrieving the data from the repository on which it initially resides.

formatting

the modification of the format or structure of the data file to meet the requirements of the local processing system.

metadata

the values of characteristics that qualitatively or quantitatively describe or support a resource.

model

a conceptual view and description of something that may not be directly observable.

networked archive data model

a data storage model where no one location is responsible for the data assimilation (i.e., no central repository). Various locations contribute historic and quasi-real-time data to the entire community.

parameter discovery vocabulary

a group of terms, with each term representing a collection of parameters from one or more parameter usage vocabularies.

parameter usage vocabulary

a controlled vocabulary containing the formal names, definitions, units, etc. for parameters.

processing

the actual calculations associated with the use or incorporation of the obtained data into analyses that meet the requirements of the research.

regridding

the adjustment of the obtained data to the exact spatial-temporal characteristics required for subsequent analyses.

subsampling

the adjustment of the obtained data to the exact sampling frequency characteristics required for subsequent analyses.

semantic

the meaning of a term. Semantics are often related to a specific subject area. For example, the same term used in two different subject areas may have two different meanings.

syntactic

the formal rules for constructing data structures or data elements.

user

a human that interacts with an application. May also be considered an operator.

vocabulary

a set of specialized terminology used to communicate in a specific community.

Distribution list

Document No.: DRDC Atlantic TM 2005-159

LIST PART 1: Internal Distribution by Centre:

2 DRDC ATLANTIC LIBRARY FILE COPIES
3 DRDC ATLANTIC LIBRARY (SPARES)
1 D. HAZEN
1 M. E. LEFRANÇOIS
1 R. STUART
1 W. A. ROGER
1 G. R. MELLEMA
3 B. TOPP
1 R. PEDERSON
1 G. HEMPHILL
1 J. THORLEIFSON (DRDC Corporate)
1 R. SCHWARTZ
2 A. W. ISENOR (1 CD COPY, 1 HARD COPY)

19 TOTAL LIST PART 1

LIST PART 2: External Distribution by DRDKIM

1 NDHQ/DRDKIM 2-2-5
1 NORBERT HACHE, DMSS 8-2-4
National Defence Head Quarters
101 Colonel By Drive
Ottawa
1 JOHN SONES, DMSS 7-3
National Defence Head Quarters
101 Colonel By Drive
Ottawa
1 GREG WALKER, DSTM 3
DRDC Corporate
305 Rideau Street
Ottawa
1 R. BARRY WALKER, DMIMR
National Defence Head Quarters
101 Colonel By Drive
Ottawa

1 DONNA WOOD, DST C4ISR 4
DRDC Corporate
305 Rideau Street
Ottawa

6 ____________________________
TOTAL LIST PART 2

25 TOTAL COPIES REQUIRED

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) Defence R&D Canada – Atlantic PO Box 1012 Dartmouth, Nova Scotia B2Y 3Z7	2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C, R or U) in parentheses after the title.)

Proposed Architecture for Data Sharing in the Networked Underwater Warfare Project

4. AUTHORS (First name, middle initial and last name. If military, show rank, e.g. Maj. John E. Doe.)

Anthony W. Isenor

5. DATE OF PUBLICATION (Month and year of publication of document.) January 2006	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.) 58	6b. NO. OF REFS (Total cited in document.) 26

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.) 11cg01	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) DRDC Atlantic TM 2005-159	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

This work discusses topics related to data sharing and understanding in a network-enabled environment. A critical component of a successful implementation of network-enabled operations (NEOps) will be the client's ability to judge the importance of the received data as well as understand the content of the received data. In the NEOps environment, many clients will enter the network unaware of the available data assets. A discovery process is required for the client to first identify the available resources. Once identified, the client will require information on the structure used to deliver the data to the client. Then, the client will need information on the details of the data items present within the structure. This work proposes an architecture suitable for the discovery and understanding process. The architecture is based on a vocabulary, or dictionary of terms, and a definition of data structures. An example implementation is provided using extensible markup language. The Networked Underwater Warfare Technology Demonstration Project underway at DRDC Atlantic provides an implementation focus for the data sharing concepts presented in this work.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Networked underwater warfare; NUW; metadata; standards; XML; extensible markup language; data discovery; network enabled operations; NEOps; interoperability; vocabulary; data dictionary; system of systems;

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca