2006 CCRTS
The State of the Art and the State of the Practice

"With All Your Knowing, Get Understanding: Conceptual Interoperability and the Net-Centric Value Chain"

Topics: Net-Centric Data Strategy;
Conceptual Interoperability;
Formal Ontology

Charles D. Turnitsa
Dr. Andreas Tolk

Virginia Modeling, Analysis and Simulation Center
7000 College Drive
Suffolk, VA 23435
cturnits@odu.edu

Abstract: *The Net-Centric Value Chain is a picture of how various methods and technologies can be applied to migrate Data to Information, from Information to Knowledge, and from Knowledge to Awareness. We have, at VMASC, derived and refined the Levels of Conceptual Interoperability Model. This paper will show how the varying levels of the Net-Centric Value Chain are achieved by attaining successively higher levels of conceptual interoperability; where our current state of the art is in progressing through these two paradigms; and how the next stage (moving from Information to Knowledge) can be achieved by applying various conceptual interoperability enabling methods, including data linguistic methods such as a data grammar, or a formal ontology. This enables a topology merging the Net-Centric Value Chain, with the Levels of Conceptual Interoperability Model, along with a suggested roadmap of applications to achieve the desired level of each.*

1. Introduction

The concept of Effects Based Operations mandates the formulation and enablement of Agile Organizations, which are capable of net-centric activities. These sorts of Organizations, by their nature, have needs for Agile Information Sources. System to system interoperability (including service to service interoperability, which is of potential greater interest to net-centric activities, especially within the Global Information Grid) is not only the current goal of many research programs, but successful adoption of this interoperability is a requirement for success. At VMASC we have an ongoing research program looking into the areas of system (and service) composability and interoperability, which has led (among other things) to the Levels of Conceptual Interoperability Model. This model gives not only a codification of the various quanta of interoperability that can be attained, but it also suggests a number of means and methods for achieving such quanta.

The data produced and consumed by net-centric entities, such as an Agile Organization, are required to attain different levels of value, to serve different goals. These levels of data are shown within the Net-Centric Value Chain, which proceeds from data to information, from information to knowledge, and from knowledge to awareness. These different levels of value, especially where an Agile Organization (which is required to compose with other organizations) is concerned, require interoperability. We show within this paper how the various methods captured within the Levels of Conceptual Interoperability can support not only the various levels of the Net-Centric Value Chain, but also to enable the transformation of data between the levels.

2. Agile Interoperability

A network-centric view of the world shows us that for an organization to be successful in satisfying its own operational goals, such an organization must be able to adapt to different operational environments. As each new operational goal presents itself, the organization (if it is successfully aware and perceptive of the true goal) senses that a certain capability, or portion of its total resources and assets, are required to achieve that operational goal. Rarely is this the entire organization. Equally rarely, in a modern world of specialization and purposeful reasons for existence, will all of the required resources and assets be found within one organization. In that case, certain resources and assets of an organization will have to be combined with the resources and assets of one or more other organizations, forming a new, ad-hoc organization that has just the right combination of resources and assets required to

achieve the operational goal. This sort of ability to re-align, and re-combine, at will, is what makes an organization an agile organization.

2.1. Agile Organizations have Agile Information System Needs

Traditional organizations, in the age of information warfare, require a great number of different systems to perform operations. These systems provide the organization with the ability to communicate, sense, plan, and a host of other tasks. The traditional view of an organization supports that the organization will have access to all of the systems it normally needs to exist, and all of the systems that it requires to have in order to perform successful operations with its own organic assets and resources. However, what in the case of an agile organization?

In the case of an agile organization, as portions of an organization must rely on portions of another organization – in effect creating a new, temporary organization for the purposes of achieving success within an operation – those two organizations will have reliance on a distinct set of systems. The ad-hoc agile organization formed from two or more existing organizations will therefore have a reliance on a number of systems that, in each case, at least part of the ad-hoc organization was not aware of earlier. This has several important implications.

2.2. Implications of Agile Interoperability Requirements

The foremost of these implications is that an organization that must rely upon a system that it is not aware of earlier must be trained on such a system. Certainly, the counter argument exists that organizations can receive training on systems that are part of the larger pool or organizations that they must combine within (for instance, US Army organizations might train on a number of different systems from the other branches of US DOD that the Army organization might have to potentially use), however this is not a realistic solution, as the number of possible systems far outnumbers the amount of available training time and resources that the members of an organization have access to. In addition, when we consider that warfare in the future is likely to involve organizations composed of coalition partners, the number of possible systems multiplies that might be involved is multiplied to a very high total.

A secondary implication for organizations, related to but distinct from the training issue, is one of non-familiarity with the output of previously unknown supporting systems. It is quite possible that the output of a system (an unfamiliar system) might be needed for another system (a familiar system) that an organization may be relying on. In that case, the familiar system has needs for data and information within a certain format, which the unfamiliar system may not be able to provide. These differences may be in protocol or format, or more importantly, they may be semantically disjoint. An example might be where the location reporting capability of a Coalition partners GIS system may report locations using a particular type of coordinate system, and then a targeting system which is relying on that report may be expecting location information using a different coordinate system.

In both of these implicational cases, we can see the need not only for the existence of agile organizations, but also an agility aspect extended to the systems that support such organizations. Such systems, and complex systems of systems, must be assembled in an as-required fashion, in order to satisfy the goals of effects-based operations in a network centric

world. The successful interaction of such systems is referred to as interoperability, and where it is applied to the systems of agile organizations, we refer to it as agile interoperability.

3. Levels of Conceptual Interoperability

In the domain of simulation to simulation interaction, as well as simulation to C2 (and C2 to C2) interaction, VMASC has been involved in a great deal of studies, activities and experimentation to solve some of the problems hinted at in the preceding section. In doing this, we have found it very useful to derive a system whereby we may quantify and identify the various problems of interoperability. The results of this effort have been presented in the form of the Levels of Conceptual Interoperability Model, or LCIM. The original LCIM was presented [1] several years ago, and has been recently referred to in a number of research projects, both within and without VMASC. As the model continues to be worked with and understood, it is under refinement. The first major revision of the model has appeared in a number of publications [2], and the model continues to grow, and be applied.

3.1. The LCIM as Originally Envisioned

The LCIM draws heavily on several models that went before it. Notably, these include the "Levels of Information Systems' Interoperability" model (LISI) [3], and the NATO C3 Technical Architecture Reference Model for Interoperability (NMI) [4].

The LCIM, as originally presented [1] consisted of a description of five layers of interoperability, with increasing levels of conceptual understanding between the two systems.

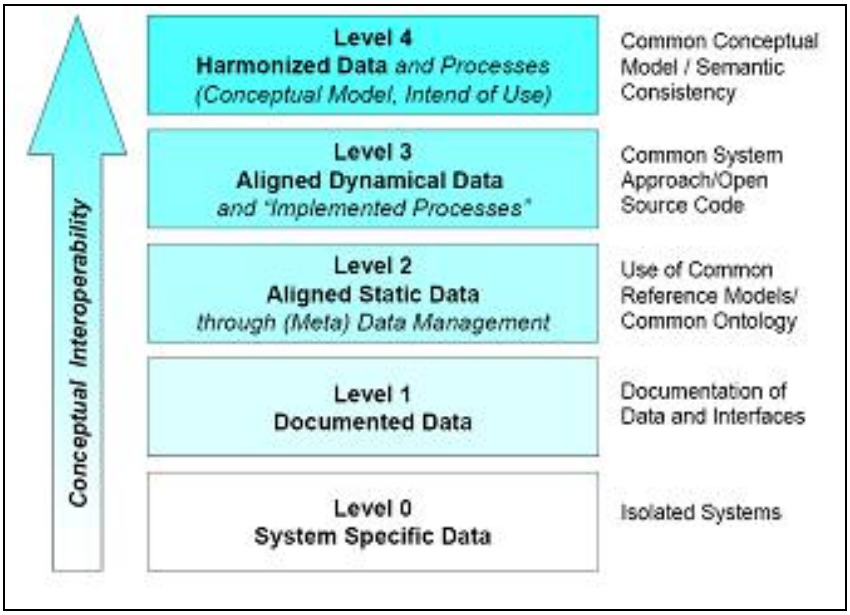

Figure 1 - LCIM

As can be seen from diagram 1, these levels run from isolated systems, which have no need (or ability) to interoperate with other systems, on up through systems with a common conceptual model, allowing for semantic consistency in their communication with each other. Following are a few paragraphs describing this model and the top four layers.

3.1.1. Documented Data

The idea represented by this level of interoperability, is where the data to be exchanged is documented, providing for a standardized method of referencing the data. An excellent example of this would be through the Object Model Template (OMT) that is part of the High Level Architecture (HLA) [5]. Interoperating systems at this level remain closed "black boxes" to each other. The only interaction is through documented data guides.

3.1.2. Aligned Static Data

The systems that are interoperating are aligning their data to a static, but standardized, referential data model. An excellent example of this from the simulation world, is how all of the federated systems within an HLA federation would interoperate through the same shared Federation Object Model (FOM). The systems still remain "black boxes" to each other, but now have a standardized means of interacting with the same data elements.

3.1.3. Aligned Dynamic Data

The systems interoperating with this level of connectivity not only share a similar referential model with each other, as above, but they also have a shared view of how the data is employed by each other. This does not mean that the systems employ the data in the same way, but they do understand its employment. One method for describing its use would be through the descriptive view of a system afforded by a tool such as the Unified Modeling Language, or UML [6].

3.1.4. Harmonized Data

The systems interoperating have a clear understanding of each other's conceptual model. There is a clear model and a clear understanding not only of the data to be exchanged, and how each system employs that data, but also of the assumptions behind the employment of the data, and the concepts that are represented by the data. At this level, systems are said to be conceptually interoperating with each other.

3.2. The LCIM as Revised and Extended

As can be seen from the levels described in the original LCIM, the descriptions and definitions are based heavily on the data being interchanged, and the understanding of that data. Rather than concentrate on the data being passed between systems, it is equally important to concentrate on the use (with understanding) that a system makes of that data. This emphasis explains the main difference between the LCIM today, and the original model.

Getting to the point of where we are today has taken a few steps of research between the original specification and now. Starting in 2004, with the publication of [7] there was a view of the LCIM accommodating system consumption in addition to just the data being transferred. This introduced a re-orientation of the model, and renaming the levels to accommodate the new paradigm. This interim model consisted of four levels beyond level 0. This is very similar to the original model, with the new levels being Technical, Syntactic, Semantic, and finally, Conceptual. Increased investigation at the University of the Federal Armed Forces in Germany [8] suggested the inclusion of a Pragmatic level above the Semantic level. The original reason for this inclusion is to accommodate a level that has ontological understanding of the data, but not quite a full conceptual understanding – fitting in perfectly between a semantic level (where we have a taxonomy view), and the conceptual level (where we have a complete view). That view presented a total of six layers, very

similar to the modern view, but with the dynamic and pragmatic layers (see figure 2) combined.

The current LCIM has a number of additional levels over what the original LCIM afforded, and also redefines all of the levels to describe the systems involved, and their understanding, rather than the data they can ingest and produce.

Just as with the original LCIM, the basis for all interoperability builds on level 0, which is no interoperability at all. Systems at this level either cannot, or need not communicate with each other. The remaining levels, 1 through 6, describe interoperability-enabling communications of increasing amounts of understanding. This understanding is not only of the target system understanding data passed to it from a source system, but also the understanding made by a source system of what is **needed** by the target system.

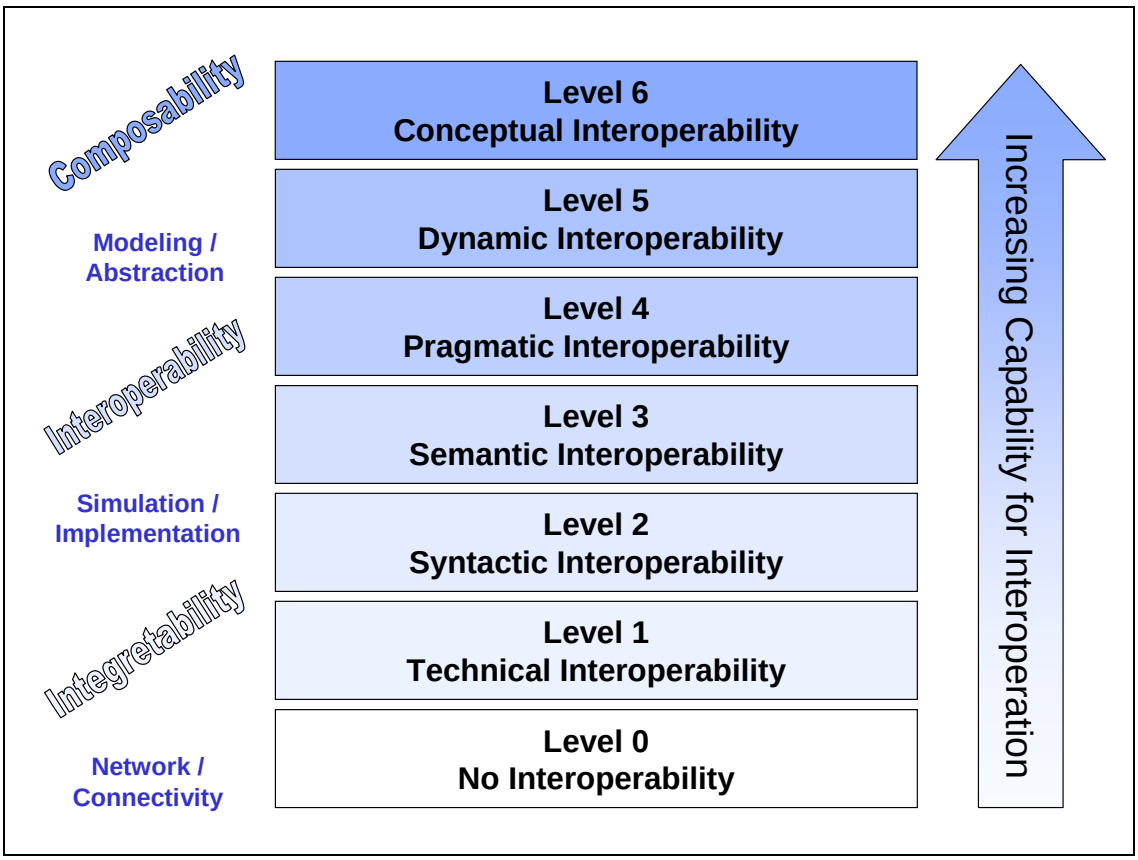

Figure 2 - Current view of the LCIM

3.2.1. Technical Interoperability

Technical Interoperability refers to a state where the interoperating systems do have a technical connection to each other (perhaps over a network), and perhaps a protocol exists for exchanging data between the various systems, but there is no alignment between the data elements being sent and received. The names of the elements, the order of sending and

receiving, and even the very nature of the data is not assured with any level of understanding between the systems.

An example of this sort of interoperability would be where a target system were to receive all of the data messages and requests to come out of a source system, and merely record it, without trying to make any sort of sense of that data. In this simple scenario, no need exists for the target system to make sense of the data received from the source system. In this case, the data is merely being recorded.

Technologies and methods accommodating this sort of interoperability are seen in systems connected employing TCP/IP (or other network protocols), and perhaps HTTP (although that is almost syntactic in nature).

3.2.2. Syntactic Interoperability

Syntactic interoperability is the next level up over the simples of technical connections. At this layer, there is a basic syntactic understanding of the data involved. By syntactic understanding, the intention is to refer to a systems ability to exchange the *right forms* of data, and in the *right order*. If we were talking about a strictly linguistic model rather than an information systems model, we would describe this sort of understanding as the knowledge of how a sentence is structure – is it subject-predicate-object? The knowledge of sentence structure ensures that the speaker can formulate sentences in the correct form, but not necessarily with any higher order of meaning.

An example of a system engaged in this sort of interoperability might be a roadway simulation system that is expecting a data item to be conveyed to it – the data item sent must needs be an object, perhaps the name of a type of vehicle, but no further understanding is requested or expected. Is it named in the correct manner? Does the name carry with it any sort of meaning? Does the name, as conveyed, provide for all the data to understand what is represented? The answers to these questions are not found at this level of interoperability.

Technologies and methods accommodate syntactic interoperability include the High Level Architecture, CORBA, SOAP, XML tagging, and various other service connecting technologies that would mandate a particular sequence and ordering of connectivity, without mandating any level of understanding within the data being passed in and out.

3.2.3. Semantic Interoperability

Communications at this level of interoperability begin to take on some sort of understanding. At the previous level, all that was required to satisfy level 2 was that the data be in the correct syntactic form (and of course, level 1 requirements are subsumed). At this level, not only are syntactic rules enforced decreeing the “form” of data being exchanged, but within that form, the correct type of data must be employed. Previously, we used the example from linguistics that not only must there be an adherence to the portion of the grammar (subject, object, predicate, etc) being conveyed, but here the data must also satisfy the contextual (or semantic) needs in its typology.

Extending the former example of a roadway simulation system requesting or requiring the name of an object, perhaps a vehicle, at this level we go a little deeper. For instance, the system in question might be requesting (from the source) not only a data element describing a vehicle type, but a vehicle name from a particular manufacturer, or country of origin, within certain parameters. A clearer example would be a road network simulation system

that is looking for a data element representing a vehicle involved in driving a particular route. To satisfy that system at the syntactic level, only a vehicle name (any vehicle name) was required. To satisfy the system now (at the semantic level), the correct type of vehicle name must be provided.

A method of enabling the interoperability of systems with this level of conceptual interoperability is via a central mediation service. If such a service is well designed, around a data model that is sufficiently complete to be able to accommodate all the data types that need to be exchanged between source and target, than it can go a long way towards ensuring semantic interoperability.

Such a service would have to be accommodating to all of the systems that would need to communicate through it, having a protocol of interaction that the systems in question could adopt, and also allowing for data representation that the systems in question could have their own data mapped to. An example of such a service has been recently discussed in [9].

As an enabler for semantic interoperability, a central referential data model must have a sufficiently complete taxonomy of data elements such that it can accommodate all of the various resolutions of data that the systems connecting to it might require.

Note that while we are beginning to convey data with some sorts of understanding at this level, there is still wide room for misunderstanding, as the following levels will illustrate.

Available technologies to enable semantic interoperability, and perhaps a referential data model, could include registered XML tag sets (so it is not only in a common ingestible format, but also any employing systems are aware of the taxonomy of the data), or a common federation object model (as envisioned in the High Level Architecture specifications).

3.2.4. Pragmatic Interoperability

The understanding required to have systems intercommunicate at the level of pragmatic interoperability is not only of form and type, but also of ontological meaning. A target system, in order to make full use of data sent to it from a source system, must not only receive the data in the correct semantic form, but it must also have an understanding of the source system's ontology, and the place the data received fills within that ontology.

The ontology of a system is the epistemology, or truth view, of the system's data model. What is meant by "truth view" is the sum collection of relationships between the data within the model, and all the defining parameters and assumptions that go into those data and their relationships [10].

One such method of describing the ontology of a data model is through a tool such as OWL [11], which is a method for describing and defining the ontology of a data model, showing not only the ontology of each data element in the sense of a biological ontology (meaning, how more specific concepts emerge from more general concepts), but it also shows a basic view of how these data entities relate to each other. An example would be if we had an ontology describing basic food items. A *grapefruit* is a specific concept descended from the more general concept of *fruit*. A *grapefruit* is also related to a *banana* in that both are commonly considered to be *breakfast foods* (within the ontology of our universe of discourse – basic food items).

Systems communicating with a level of pragmatic interoperability might use this understanding to give not only the correct type and form of data (as with the previous levels, 2 and 3), but it also can now be expected to serve the goals of the consuming system. For instance, taking our earlier example of a roadway simulation system forward even further, let us assume that a target system is awaiting a data transfer from a source system. That target system is awaiting a data element describing an object – a vehicle (level 2). The target system requests that the data element represent a particular manufacture of vehicle, perhaps one with the parameters of performing a particular task, such as driving along a route (level 3). With pragmatic interoperability, it can also be expected that the target system might be expecting a data element describing a particular type of vehicle, executing a driving task within a certain range of space and time parameters, and finally that the vehicle has the capability of performing the specific task within an understood set of limits. The source system not only has to understand what the target system is expecting, but also has to have some sort of ontological understanding of what the task in question is, and which of the particular vehicles described by its own data model are capable of performing that task.

Pragmatic interoperability requires that the systems involved have an understanding of the ontology of the data being exchanged. This can build on the idea of a common referential data model, as exhibited in the enabling example for level 3 interoperability. The difference between level 4 and level 3 is that the referential data model must also have a sufficiently complete ontology to be able to accommodate the ontological descriptions of the data for the source and target data models.

A view of where an accommodating core ontological view was able to accommodate data translation between two disjoint systems is given in [12].

Technologies and methods that could be developed to support pragmatic interoperability include ontology specification mechanisms (such as OWL), UML, and perhaps the Model Driven Architecture [13].

3.2.5. Dynamic Interoperability

The previous level of interoperability assumed that knowledge of the ontology of a source data model would be sufficient to describe all the understanding that a target system would need to know, in order to make sufficient understanding of the data. However this makes a very simplistic assumption, and that is that the pragmatic, or ontological, understanding of the source system's data model will remain universal. To get to this, the next level of interoperability and understanding between systems, there must be a more agile form of understanding than with a specific understanding. This level of communication is called (within the expanded LCIM) dynamic interoperability.

Pragmatic interoperability (level 4) assumes that a source system will be able to provide the data required by a target system, and that the parameters and characteristics of that data are well understood. But what if the necessary view or description of those parameters and characteristics change? Rather than having a single ontology to convey understanding of all data elements within a data model, there exists the need to have a system, or method, for conveying the particular instance of an agile ontology to an interoperating system.

The understanding of this sort of information – a situational ontology – must have a method for presenting that ontological data, perhaps through a dynamic service or requesting system.

This precludes it being captured within a simple referential data model (as with level 3). It also is beyond what would be needed by a referential data model with a necessarily complete ontology to support only one ontological view of a system. It could, perhaps, be based on a referential data model that is flexible enough to support dynamic definitions of several ontologies (and be sufficiently complete to accommodate all those ontological definitions). Working with such a meta-ontology would require a method of defining the ontological needs for a particular transaction, and then using that understanding to accommodate the transaction. An example of some of the work being done in this area is with such things as the DARPA Agent Modeling Language [14], or now with the work being down by the OWL-S (OWL ontology description for services). There doesn't appear to be a standard in this arena yet, but defining progress is made through several projects.

There is an ongoing project known as the Java Agent Development Framework (JADE) [15], which is seeking to enable heterogeneous agents to be able to interoperate. It is based on standards published by the Foundation for Intelligent Physical Agents [16], who have a series of specifications defining how heterogeneous agents, acting as brokers for systems, can interoperate and exchange data and services.

It should be pointed out that a complete UML representation is an example of how dynamic models could be represented. Not only are the data and system elements modeled, but there is also the idea of dynamism as seen through the changes to the relationships between system and data elements over time.

A simple example of dynamic interoperability could be seen if we extend (slightly) our ongoing series of examples surrounding a roadway modeling system. Let us assume that the target system is requesting from the source system a series of data elements (level 2) describing a particular manufacture of vehicle (level 3) that can successfully drive at a certain speed through a particular curvy patch of highway, with a specific brand of tires (level 4). This is fine, but what if an underlying parameter affecting data consumption or data entities within the target system changes. Let us suppose that we now have conditions within the synthetic environment that are part of the target system where the vehicle models have to operate in foul weather. That changes the ontological definition of "drive at a certain speed through a particular curvy patch of highway". We would need to be able to define that change to the ontological definition of our request.

OWL-S [17] supports this sort of change by defining the ontology of a data service. Such a service could be sensitive to particular state changes of the system it is interacting with. OWL-S provides for modeling such a service through four defining parameters, which are Input, Output, Preconditions, and Effects. This is very early research, as stated earlier, and needs to be extended, but it is the sort of data that will have to be conveyed to define the ontological conditioning of a data interchange. The clear benefit that the OWL-S work promises is the intention that it be used by executable systems.

Methods and technologies that might assist with enabling systems to interoperate at the Dynamic level include, in addition to OWL-S, others such as UML, and perhaps some of the newer work being done with such dynamic service frameworks as the Web Services Workflow model.

3.2.6. Conceptual Interoperability

We now reach the final level described within the Extended Levels of Conceptual Interoperability. This level defines communications between systems that is truly conceptual. True conceptual interoperability, or communication, is only available when complete understanding of the concepts inherent within the target and source data models is shared, or shareable, between the data models. This understanding implies not only the data, and their meaning, but also the associated relationships, defining parameters, and composing assumptions behind that data.

Attaining conceptual interoperability may be beyond the ability to use a referential data model, until such time as it is possible to have well-formed definitions of the various data parameters, assumptions, and relationships that define a concept.

Currently there are a couple of approaches that begin to define, in a rigorous fashion, a system and its underlying assumptions and concepts. One such method is in the relatively new extension of UML, the Systems Modeling Language (SysML). SysML is a method that is intended to allow enterprise architects to have the ability to define all of the interoperating systems within their architecture [18].

The DEVS system is another method for describing a simulation system and its underlying assumptions, for the purpose of enabling interoperability with other simulation systems. DEVS is the Discrete Event System Specification. It is a formalism developed to define the various systems within a simulation system of systems. It has grown beyond its original intent as just a description tool for discrete event systems, and now exists as a framework for defining and describing how a family of heterogeneous systems can interoperate and how their state changes and model interactions contribute to the meta system, or system of systems. Current developments in this community are seen in [19].

An example of true conceptual interoperability can be seen if we follow on from our earlier example of a roadway simulation. Only now, to better highlight the concepts of true interoperability, we will expand our example to include two other systems. The benefits of this example will be first to show what can be accomplished between systems, and second to introduce a new concept to our model – emergent behaviors. The first benefit is illustrated when we consider the systems and what they are now capable of.

Lets assume that the three systems are our original roadway system, which is a simulation of vehicles driving on different roadways, and exhibiting behaviors from a variety of different basic data and assumptions. Interoperating with the roadway system, in our example, are two other simulation systems. The first of these is a civic model that models the different aspects of a city – the demographics of the population, the infrastructure, and other basic data concerning the existence of a city. The second additional model in our system of systems is an economic model that simulates the trends in purchasing of a population based on the large commodities (such as automobiles) that will be purchases.

Our three fictional systems have attained true conceptual interoperability with each other. The method, for the purposes of our example, is not important. Only that they have a way to interoperate by exchanging data in such a manner that the systems understand the conceptual functions and activities, and the underlying parameters and assumptions driving those activities, of the other models they are interoperating with. The roadway model understands the data and functionality (at a conceptual level) of both the city model and the economic model. And they each in turn understand the other two. This conceptual understanding is

dynamic in nature, so they understand the changes that they themselves, and their partner systems, will undergo over time.

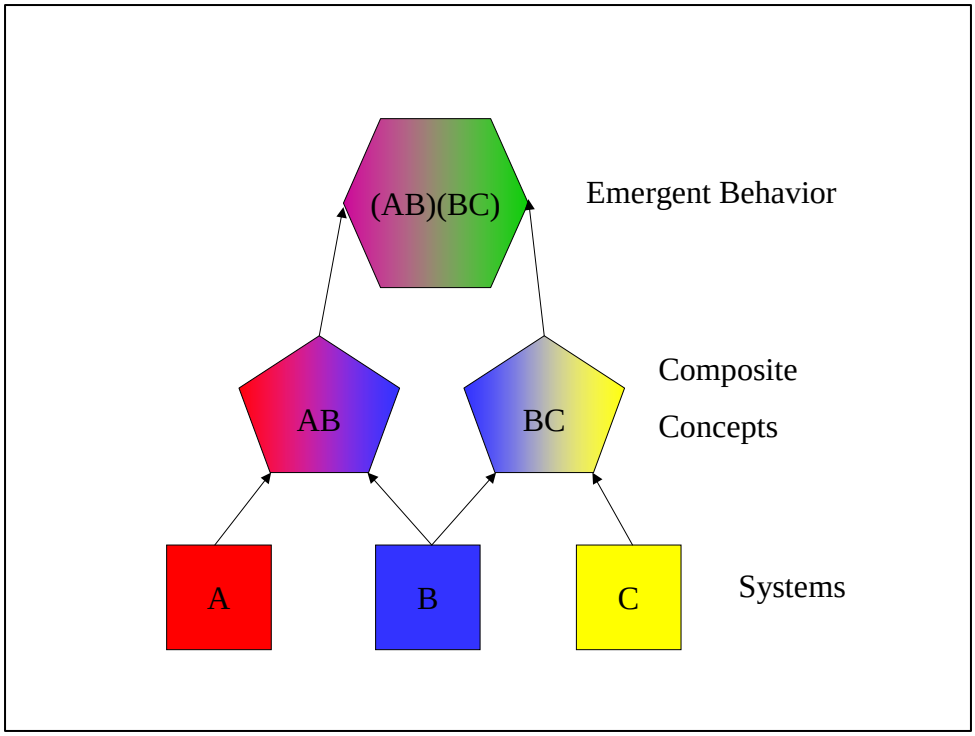

Figure 3 - Emergent Behaviors

The benefits of this conceptual interoperability should begin to be clear. The roadway model can now make use of information about roads that the city model has in its defining data. The purchases made by entities in the economics model will affect the taxes collected within the city model, and so on. The ideas, and the data defining them, that come out of two or more systems can be called composite conceptions. These composite conceptions are the immediate benefits of conceptual interoperability.

At the beginning of this example, however, there was mention of a new concept – emergent behaviors. These are behaviors within the system of systems that are dependent on the interaction of other systems, and the data that such behaviors are derived from are not available from the individual models. They only become apparent, and available, when the models coexist, and conceptually interoperate, over time. They are the results of what happens when our various composite conceptions begin to interact with each other and create other child concepts. Those second order child concepts are the components of emergent behavior.

In our notional system of systems, some examples of this emergent behavior might be this. As more segments of a particular part of the city purchase more and more of a particular type of vehicle, then the city model is capable of having its concepts of the state of maintenance of those roads that the demographics purchasing such vehicles might have on the roadway. Lets say that a certain class of citizen enjoys purchasing and driving sports utility vehicles (SUVs), and that they have a detrimental effect on the surface of roads in a certain part of

town. Then the use of SUVs, by the citizens purchasing them, will degrade the roads in that part of town, potentially driving down the value, and the economy, of that part of town, so the demographics change over time. None of these effects can be seen without the models interacting over time.

For supporting true conceptual interoperability there are several technologies and methods that are being developed. These include not only DEVS, but also the MDA, and even the Department of Defense Architectural Framework [20]. The key feature here is that it needs to have not only an ontological view of the data, as with level 4, but also the concepts of time related state changes, as with level 5. In addition there needs to exist an understanding of an overall system ontology, existing within time at different states, and a clear understanding of all the assumptions, rules, and relations of the various parts of the meta-system.

4. Net-Centric Value Chain

Alberts and Hayes [21] give a good overview on net centric operations and warfare. Within this body we see a number of attributes that an organization within the information age should exhibit. An information age organization is one that is capable of net centric operations and warfare, or to use our term here, an agile organization. One of the main ideas introduced is the value chain of net centric warfare starting with data and going over information and knowledge to awareness. This value chain was used in numerous articles and papers to cope with the benefits of recommended improvements for Command and Control (C2), among these the *NATO Code of Best Practice for Command and Control Assessment* [22] and many papers of the Command and Control Research and Technology Symposium (CCRTS), in the US as well as international. The four quality categories are defined as follows:

- The value chain starts with *Data Quality* describing the information within the underlying C2 information systems.
- *Information Quality* tracks the completeness, correctness, currency, consistency, and precision of the data items and information statements available.
- *Knowledge Quality* deals with procedural knowledge and information embedded in the command and control system such as templates for adversary forces, assumptions about entities such as ranges and weapons, and doctrinal assumptions, often coded as rules. In future systems, this agile component could be presented by M&S systems.
- Finally, *Awareness Quality* measures the degree of using the information and knowledge embedded within the C2 system. Awareness is explicitly placed in the cognitive domain.

In summary, the data describing the conducted and supported operations are of critical importance. When placed into context, data leads to information. Information applied in the form of agile models, offered as OR and M&S services within a SOA, will even enable the support of knowledge within C2 information systems. C2 quality is improved by an order of magnitude when a new level of quality is reached in this value chain. OR and M&S services support this critical development. The following figure introduced by Tolk (2004) depicts this.

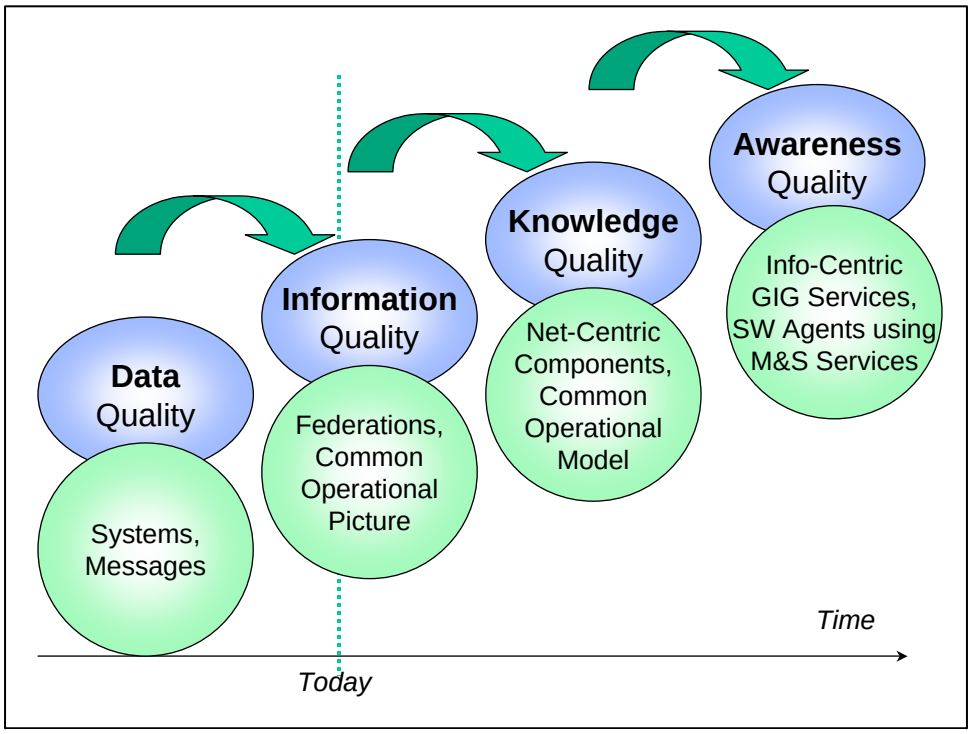

Figure 4 - C2 Improvements

The reasons for improved C2 quality improved are the following: Data quality is characterized by stand-alone developed systems exchanging data via text messages, such as the U.S. Message Text Format. By the introduction of a common operational picture (COP), data are put into context, which leads to information instead of data. The federated systems using the COP result in an order of magnitude improvement of the C2 quality, as decision makers shared this common information. The next step, which is enabled by service-oriented web-based infrastructures (but not yet operationally used), is the use of OR and M&S services for decision support. In particular simulation systems are the prototype for procedural knowledge, which is the basis for knowledge quality. Finally, using intelligent software agents to continually observe the battle sphere, apply models and simulations to analyze what is going on, to monitor the execution of a plan, and to do all the tasks necessary to make the decision maker aware of what is going on, C2 systems can even support situational awareness, the level in the value chain traditionally limited to pure cognitive methods. Although these topics are still issues of ongoing research, first promising results are documented in Tolk (2004).

In order to be able to make use of these potentials, C2 processes must be integrated and aligned. The data of C2 must be integrated seamlessly into this environment. National efforts are the basis to migrate C2 into the information age by digitizing tasks, missions, operation plans, and the commander's intent. Digitized Commander's intent in this sense goes beyond the paragraph defined in the tradition five-paragraph operational order of armies. It describes everything that is needed to describe an executing system to understand what a commander wants, which includes traditional orders. It is not envisaged that the paragraph that relates to Commander's intent in a traditional order would be replaced and would therefore remain as "free text."

5. Satisfying Data Quality

Each of the four strata of the net centric value chain will now be investigated. We will start with Data Quality, and proceed up the chain. At each level we will show what this level of quality means to an agile organization.

In doing this we will examine how the level of quality can enable the various goals of a net centric organization. These goals are listed here, but they are discussed to a much greater degree within [23].

- Sense-making capability
- The ability to work in a coalition environment (meaning a JIM – joint, interagency, multinational – environment)
- Knowledge of what the appropriate means (level of and type of force) are, for the mission
- The ability to orchestrate the appropriate means and to respond in a timely manner

While [23] states that the third goal is not in the purview of the C2 world, I believe that it is a goal that an Agile Organization must pursue, and that the value chain can shed some light as to how well it can be attained.

As mentioned above, the value chain begins with Data Quality. Data, from an organization's perspective, is found within the systems that are inherent to that organization.

5.1. Data Quality and the Agile Organization

An organization existing within the age of information warfare must have data. It must be aware of its surroundings, the activities of opposing forces, and the intentions and wishes of the hierarchical (or distributed) supra-organization that it is a part of. There is, however some risk involved when an organization is only aware of the data that it has available via its own organic sensing and reporting systems. In this case the organization is said to have achieved <u>only</u> a Data Quality of network-centric value. We will examine what this means, and show how Data Quality can be expanded to the next stage along the value chain, Information Quality.

Being possessed of only Data, especially data that is only derived from organic means, it is difficult to see how an organization can satisfy any of the four goals of a net centric organization, as laid out above. Without C2 data from sensors and units outside of itself, it cannot make any more than the must rudimentary tactical sense of a situation; without being able to exchange data with JIM partners, coalition operations appear impossible; without the

distributed knowledge from an extended ISR effort, it is not clear that an organization can know what the appropriate means should be; and finally without knowing what the appropriate means are, it is difficult to be able to orchestrate those means. Self-synchronization becomes difficult or impossible to achieve.

5.2. Data Quality and the LCIM

The Levels of Conceptual Interoperability Model capture the view of a system that does not have to interoperate with other systems, and term this Level 0 interoperability. Organizations that exhibit only a data quality of net centric value have systems that, at least from the perspective of organization-to-organization interoperability, have achieved level 0 alone of conceptual interoperability.

5.3. How do we progress from Data Quality to Information Quality?

In reality, very few modern (information age) organizations are only capable of data quality in their net centric valuation. The process of achieving information quality has already been achieved by most systems, and this is accomplished by having a basic interoperable capability, at the technical and syntactic levels (levels 1 and 2 of the LCIM), between systems and organizations. Once this is achieved, organizations can at least exchange basic information and be assured that it will be of the proper protocol and format to be understood by related organizations. At this point, we have moved past an organization operating alone, and relying solely on organic data sources. This early stage of additional quality, however, is only the beginning and should in now way be confused with true interoperability, as we will see below.

6. Satisfying Information Quality

Information Quality is achieved by an organization when it can take the data generated by its own systems, as well as the data imported from related organizations, and when the organization can apply some contextual sense to that data. The data has meaning, and becomes information. Rarely can this be done with relying solely on the data available organically to a unit, but it must be combined and augmented, via interoperability, with data from outside the unit.

An example of the difference between Data Quality, and Information Quality could be seen from the simplest of C2 systems – a report on the operational picture of a theater of operations. For an organization that has data quality, it might be aware of the fact that opposition forces occupy such a position in the battle space. Yet, with the improvement to Information Quality, that position in the battle space takes on additional meaning and context, when the organization is aware of how many allied organizations are within force projection range of the position, and also how many supporting oppositional forces might be around to strengthen the foe's position. This is context, and turns simple data into something imbued with semantic meaning.

6.1. Information Quality and the Agile Organization

Information Quality, is of course, a great leap above Data Quality in the net centric value chain, as each advance along the chain brings great benefits. We can see this as we consider how Information Quality can affect our four stated goals of an Agile Organization.

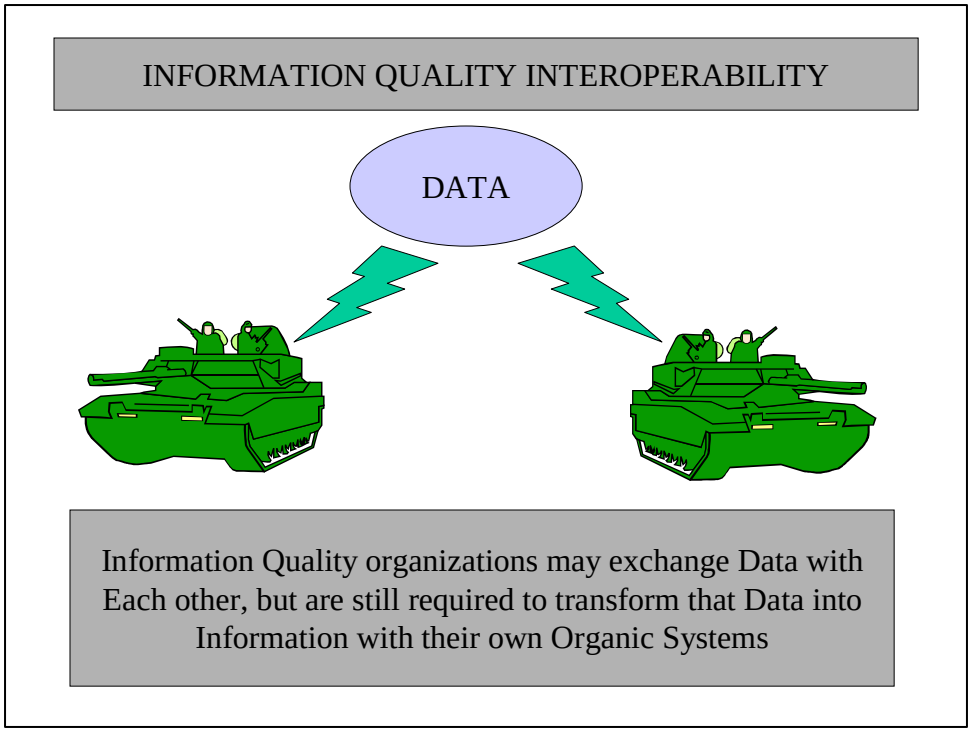

Figure 5 - Information Quality Interoperability

Once Information Quality is achieved, limited sense making is possible. The data that the organization has access to begins to have some context. However, without progressing to Knowledge Quality, that contextual data is limited, still, to the data that the organization itself can generate or receive – the ability for allied organizations to communicate Knowledge is not yet attained, so the organization is limited to what Knowledge it can generate out of received data.

The ability of an organization to work in a JIM environment is available at the most rudimentary level, but not to the full extent expected within a completely realized network centric environment. The reason for this is still because all Information that the allied organizations are capable of is Information that they can generate themselves from exchanged data, without being able to interoperate at higher levels, such JIM operations are limited.

Knowledge of, and of course the ability to orchestrate, appropriate means for a mission is limited to missions where the organization has available all facets of information pertaining to the mission. As this is highly unlikely in any single organization, this again is a limited aspect.

6.2. Information Quality and the LCIM

As organizations and their systems interoperate, and exchange data, we can show how Information Quality can be achieved by arriving at a new level within the LCIM. We have seen that Information Quality is the placement of data within context – giving it semantic meaning. Within an organization that means beginning to achieve some of the techniques described in the LCIM under Semantic Interoperability (level 3). Between organizations, however, the only interchange done by an organization at the Information Quality level is the exchange of data – the semantic endowment takes place within the organization. To achieve

that level of interchange, we have to (as we have already stated) support levels 1 and 2 (technical interoperability, and syntactic interoperability) between systems of different organizations. These levels are achieved by having established protocol and procedural methods for exchanging data between systems, such as an XML interface. To ease such exchange between large numbers of different systems, some standards for the protocol are extremely useful in accomplishing the goal in an efficient manner. This could be achieved, for instance, by having a repository of community wide accepted XML schemata.

6.3. How do we progress from Information Quality to Knowledge Quality?

In order to attain the higher level of Knowledge Quality, organizations must be able to exchange something more valuable than data (i.e. – Information). In order to do that, semantic interoperability must be achieved. But that is not enough. As we will see below, at the Knowledge Quality level of the value chain, not only is organic sense making required to be fully realized, but also the sharing of sense making information. In order to move fully from Information Quality to Knowledge Quality, the level of pragmatic interoperability (level 4 of the LCIM) must be entered. This allows the exchange of not only semantic information, but also with an understanding of the underlying concepts of the semantic values. This is how we begin to realize the transferal of doctrinal information.

7. Satisfying Knowledge Quality

Knowledge Quality is as much above Information Quality in the value chain, as Information Quality exceeds the capability of Data Quality. With Knowledge Quality, it now becomes possible to not only have understanding of data within context, but also the underlying implications of that context become apparent. Doctrinal understanding of the semantics involved, as well as knowledge of the assumptions and inherent concepts of the systems involved are well known and part of the data.

Interoperability for an organization at the Knowledge Quality level of the value chain implies not only the ability to exchange semantic information, but also to have pragmatic knowledge of the semantic information being exchanged. Not only is the meaning of the data clear, but also the implications of that data within context.

7.1. Knowledge Quality and the Agile Organization

Once an organization has attained Knowledge Quality in the value chain, it can truly begin to act as an Agile Organization. All four goals of a net centric organization are addressed at some level.

First, sense making is greatly improved, as now not only can an organization attempt a level of organic sense-making, but with the ability to interchange information with other organizations (and have some understanding of that information, within a semantic and contextual framework) allows for a much wider picture of sense-making to be achieved. The ability to also export sense-making information is finally achieved at this level.

The ability to operate in a JIM environment is finally becoming a reality. Now, with some contextual understanding of information, and its semantics, the doctrinal imbued information being exchanged makes sense, making it possible, for instance, to interoperate and exchange data within the battle management languages of the various organizations involved, and that communication can be understood with some chance of success.

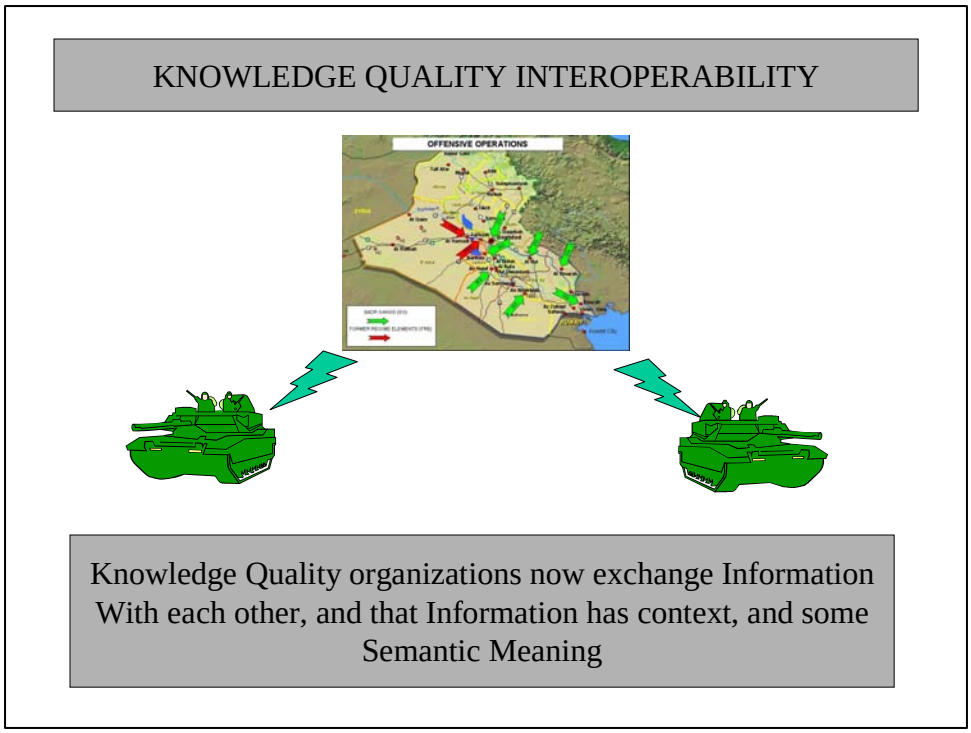

Figure 6 - Knowledge Quality Interoperability

As the semantics and context of battle space situations and entities become understood, the ability to judge the proper means and methods required for a mission are finally able to be comprehended. This may include not only knowledge from an organization's organic systems, but also from allied systems that export and interchange information, adding to the understanding of the situation.

Finally, the ability to carry out missions, with the proper means and methods, is beginning to be realized. The limiting factor at the Knowledge Quality level is that while there is a pragmatic understanding of the contextual placement of semantic information that pragmatic understanding has not yet taken on a dynamic quality. What are the forward-looking effects of the actions that an organization can take at this level? Awareness Quality is required to answer that question fully.

7.2. Knowledge Quality and the LCIM

Interoperability between systems within and between organizations operating at the Knowledge Quality level of the value chain requires systems capable of both level 4 and level 5 of the LCIM. These levels are, respectively, the semantic and pragmatic levels.

Although that is easily seen from the description of these levels, the underlying implication here is that a new field of research, barely touched so far within the C2 community, must come to fruition. That field of research is in realm of Ontology. Ontology, or rather a formal ontology, has been defined as "a specification of a conceptualization" [23, 24]. A definition of ontology (or, more specifically, a formal ontology), in clearer terms, is a formal model of the terms of a domain, with a clear model of what those terms mean, and how they can interact with each other (via taxonomical relationships, grammatical rules, and unambiguous definition). Once ontological analysis is completed, the resulting tools can help with the initial understanding of terms for disjoint systems, and can help operationally with the definition and translation of dynamic data interchanged between systems. Without these tools, we don't see how disparate systems can have any understanding of the data elements, and the concepts and implications behind those elements. They can know, in a semantic sense, the data being exchanged, but they cannot understand.

7.3. How do we progress from Knowledge Quality to Awareness Quality?

Achieving Awareness Quality for an organization is a very big leap over Knowledge Quality, which is difficult to achieve. With Awareness Quality, we are entering the cognitive domain. Knowledge about the situation and environment that an organization is operating in must be achieved, and also cognitive knowledge of how possible actions will change that situation and how future actions will be affected by changes in courses of action. All of this is complicated by the fact that no organization is likely to be operating alone, but rather in concert with other coalition partners. Knowledge of the implications of the actions of those allied organizations, and understanding of how they will most likely affect the situation, must also be understood. Making this change from Knowledge Quality to Awareness Quality requires the adoption of Dynamic Interoperability between systems, and even the beginning of Conceptual Interoperability between the organizations themselves.

8. Satisfying Awareness Quality

As mentioned, when an organization attains the Awareness Quality level on the value chain, that organization begins operating within the cognitive domain. Not only is sense-making possible in the current situation but also of future possible situations in a predictive manner. Knowledge can be interchanged between organizations, with not only semantic understanding, but also pragmatic and dynamic understanding of the context, and possible contexts, of that knowledge. Full understanding of the battle space, its entities and situation, and also of the missions and directed activities of an organization is achieved. With that full understanding, true agility is possible, as the correct awareness of a mission, and the correct assessment of how to apply the precise amount and type of effect is known.

8.1. Awareness Quality and the Agile Organization

An Agile Organization was identified, early in this paper, as an organization able to re-organize itself, and also combine with other organizations (or portions of organizations) in order to achieve a very specific effect. That effect must satisfy a particular mission goal. These two abilities, the ability to self-synchronize and reorganize, and also the ability to achieve a particular effect in order to satisfy a mission goal, are the two trademark qualities of net centric organizations and also effects based operations. The four goals of a net centric organization identified earlier are satisfied, fully, at the Awareness Quality level of the value

chain, so now we have an organization that is fully capable of net centric operations. This organization is a true Agile Organization.

8.2. Awareness Quality and the LCIM

To achieve Awareness Quality, we have stated that not only Dynamic but also Conceptual interoperability (levels 5 and 6 of the LCIM) must be attained. The reason for attaining Dynamic Interoperability within an organization, and also between organizations, is because it takes all of the understanding possible at the Pragmatic Level, and then applies that to dynamic situations. Not only is complete understanding of context and implication present, but also the contexts and implications of related situations. This implies, in addition to the formal ontology of a domain mentioned at the Knowledge Quality level, but now the formal ontology of a dynamic domain space – one where the goals and situations are shifting, therefore the conceptual meaning of entities and relationships are also shifting.

Conceptual interoperability is required at the Awareness level, because a clear untainted understanding of the situation is required, without taint or distortion emplaced by the exchanging systems. Conceptual interoperability is required because the underlying concepts of Knowledge being exchanged are distinct from the dialect of data being used to represent that knowledge. This can be seen from the example of a Dog. Consider the concept you have of a dog – four legs, a tail, etc. Now think of all the ways that different languages can represent the word *dog* (dog, hund, chien, hond, perro, cane, etc). Each of these languages introduces a certain amount of cultural addition to the basic concept of a dog. You may be conveying something about the underlying concept that you do not intend. This is the reason that ontological information in the biological and medical realms is based on Latin, a supposedly dead language – so no new inferences or assumptions can creep into the terminology used. With conceptual interoperability, we are able to exchange concepts free from bias and inference, removing all of the last potential for hidden meaning and ambiguity.

9. Conclusion

As we have seen, our state of affairs today in the C2 world is largely centered on the beginnings of the Information Quality level of the value chain. More is needed, especially if the vision of net centric warfare, and effects based operations, is to be realized. The methods described herein that show the way to progress from Information Quality to Knowledge Quality, and then to Awareness Quality are not going to come easily, but it will come. Continued research will open up these methods, as well as (no doubt) showing us new methods as we progress. After all, we have yet to achieve our own level of Awareness Quality in this field.

10. The Way Ahead

The achievement of full understanding at the Awareness Quality level in the value chain requires not only a formal and specific understanding of all of data representing all aspects of the battle space, it also requires the ability for systems to combine and merge together their abilities in a rapid, and ad-hoc manner. Such a situation is envisioned within a fully realized Global Information Grid environment, where service providers are available on demand to post and publish Knowledge, and every data perceiver is also a data publisher [25]. We see this high operational tempo of knowledge assimilation and aggregation as requiring a number of new technological enablers and techniques, including reliance on software agents (to help

produce and consume knowledge in near real time), as well as extensive embedded simulation systems (to help make predictive sense out of assimilated knowledge).

These capabilities add up to C2 systems having the ability to do their own, on demand, operations research into potential future situations. The implications here are vast – complete understanding of the entities and environment are required by such systems, as well as understanding of the pragmatics of doctrine, and the ability to exchange such high level knowledge with other systems.

This is a very steep prescription of what is required of our systems, yet we believe it is achievable. From where we are today, research must progress in the area of ontological representation of data, the enabling of software agents to be able to understand situations and retrieve and understand data about those situations, and the embedding of modeling and simulation capabilities within our C2 systems is a start.

11. References

[1] Tolk, A. and Muguira, J. A.; "The Levels of Conceptual Interoperability Model (LCIM)," Fall Simulation Interoperability Workshop, Orlando, FL, September 2003

[2] Turnitsa, C.D.; "Extending the Levels of Conceptual Interoperability Model," Proceedings IEEE Summer Computer Simulation Conference, Philadelphia, PA, 2005 IEEE CS Press

[3] Levels of Information Systems Interoperability, C4ISR Architectures Working Group, 30 March 1998, available at: DefenseLink Website: http://www.defenselink.mil/nii/org/cio/i3/AWG_Digital_Library/pdfdocs/lisi.pdf, latest version visited January 2006

[4] NATO Allied Data Publication 34 (ADatP-34): "NATO C3 Technical Architecture (NC3TA), Version 4.0," March 2003; obtainable via the NATO standard website: http://www.nato.int/docu/standard.htm

[5] Defense Modeling and Simulation Organization: HLA Technical Specification, IEEE Standard 1516, https://www.dmso.mil/public/transition/hla/techspecs/

[6] Object Management Group: Unified Modeling Language Resource Web Page, http://www.uml.org/

[7] Tolk, A.; "Composable Mission Spaces and M&S Repositories – Applicability of Open Standards," Spring Simulation Interoperability Workshop, Washington, DC, April 2004

[8] Hofmann, M. A.; "Essential preconditions for coupling model-based information systems," NATO M&S Group (NMSG) Conference on C3I and M&S Interoperability, RTO-MP-123, Antalya, Turkey, October 2003

[9] Tolk, A., Diallo, S., Dupigny, K., Sun, B., Turnitsa, C.; "Web Services based on the C2IEDM – Data Mediation and Data Storage," Spring Simulation Interoperability Workshop, San Diego, CA, April 2005

[10] Gruber, T.; "Toward Principles for the design of Ontologies used for knowledge sharing," Technical Report KSL 93-04, *International Workshop on Formal Ontology*, Padova, Italy, 1993

[11] World Wide Web Consortium; "OWL Web Ontology Language Overview," http://www.w3.org/TR/owl-features/, February 2004

[12] Doerr, M., Hunter, J., Lagoze, C.; "Towards a Core Ontology for Information Integration," Article 169, 2003-04-09, *Journal of Digital Information*, Volume 4 Issue 1

[13] Miller, J., Mukerj,i J., Editors; "MDA Guide Version 1.0.1," Object Management Group, June 2003

[14] Defense Advance Research Projects Agency, http://www.daml.org/, Darpa Agent Markup Language website

[15] Java Agent Development Framework, http://jade.tilab.com/, Telecom Italia Lab, latest version released January 2006

[16] Foundation for Intelligent Physical Agents, http://www.fipa.org/, latest version released January 2006

[17] Martin, D., et al.: "Bringing Semantics to Web Services: The OWL-S Approach," Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, San Diego, CA, July 2004.

[18] SysML Partners; "Systems Modeling Language Specification," Specification version 0.9, Object Management Group, January 2005

[19] Zeigler, B. P.; "DEVS Today: Recent Advances in Discrete Event-Based Information Technology," 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, Orlando FL, October 2003

[20] Department of Defense Architecture Working Group; "DOD Architecture Framework, V1.0," Department of Defense Networks and Information Integration Document Archive, February 2004

[21] Alberts, D.S. and Hayes, R.E.; "Power to the Edge, Command and Control in the Information Age. Information Age Transformation Series," 2003, CCRP press: http://www.dodccrp.org/publications/pdf/Alberts_Power.pdf

[22] NATO Research & Technology Organization, Studies Analysis and Simulation (SAS) Panel. 2002. NATO Code of Best Practice for Command and Control Assessment, Revised Edition. CCRP Press, www.dodccrp.org/publications/pdf/NATO_COBP.pdf

[23] Gruber, T.; "Toward Principles for the design of ontologies used for knowledge sharing," Technical Report KSL 93-04, *International Workshop on Formal Ontology*, Padova, Italy, 1993

[24] Tolk, A. and Blais, C.L.; "Taxonomies, Ontologies, and Battle Management Language – Recommendations for the Coalition BML Study Group," *Proceedings of the Spring Simulation Interoperability Workshop*, 2005, IEEE CS Press

[25] Department of Defense; "Data Sharing in a Net-Centric Department of Defense," *Department of Defense Directive 8320.2*, December 2, 2004

12. Biographical Information

CHARLES TURNITSA is the Lab Manager for the Battle Lab at Virginia Modeling Analysis and Simulation Center (VMASC) of Old Dominion University (ODU) of Norfolk, Virginia. In addition to his work as Lab Manager, he also performs as a system engineer and researcher on several VMASC research projects. He has served in many roles as an IT professional for over a decade, performing analysis, design, and development tasks on many research projects for NASA and the US Military. Currently he is pursuing a PhD in Modeling and Simulation from Old Dominion University.

ANDREAS TOLK is Senior Research Scientist at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU) of Norfolk, Virginia. He received his Ph.D. in 1995 and his Master in Computer Science in 1988 from the University of the Federal Armed Forces in Munich, Germany. He has over 14 years of international experience in the field of Applied Military Operations Research and Modeling and Simulation of and for Command and Control Systems. In addition to his research work, he gives lectures in the Modeling and Simulation program of ODU. His domain of expertise is the integration of M&S functionality into real world applications based on open standards.