Review of NATO CALS Data Model v. 4.0

Dr. John W. Sheppard
ARINC
2551 Riva Road
Annapolis, MD 21401
jsheppar@arinc.com

INTRODUCTION

The Diagnostic and Maintenance Control subcommittee of SCC20 is currently charged with producing standards supporting diagnostic and maintenance processes. The primary standard under development within the DMC is P1232, Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE). In addition, the DMC is actively defining standard metrics and characteristics for testability and diagnosability under P1522. Two related activities in the process of commencing relate to defining a standard information model, capturing test requirements (TeRM) and defining a standard information model for test and maintenance history information. It is anticipated that TeRM will provide information during product design and development that will shape development of test and diagnostic capabilities. AI-ESTATE will define information used to diagnose problems within the product once fielded. The standard metrics will be used to assess how well the system can be tested/diagnosed based on concepts defined within AI-ESTATE. The test and maintenance history information will be used to assess effectiveness of supporting the product in the field. The relationship between these models and standards is depicted in Figure 1.

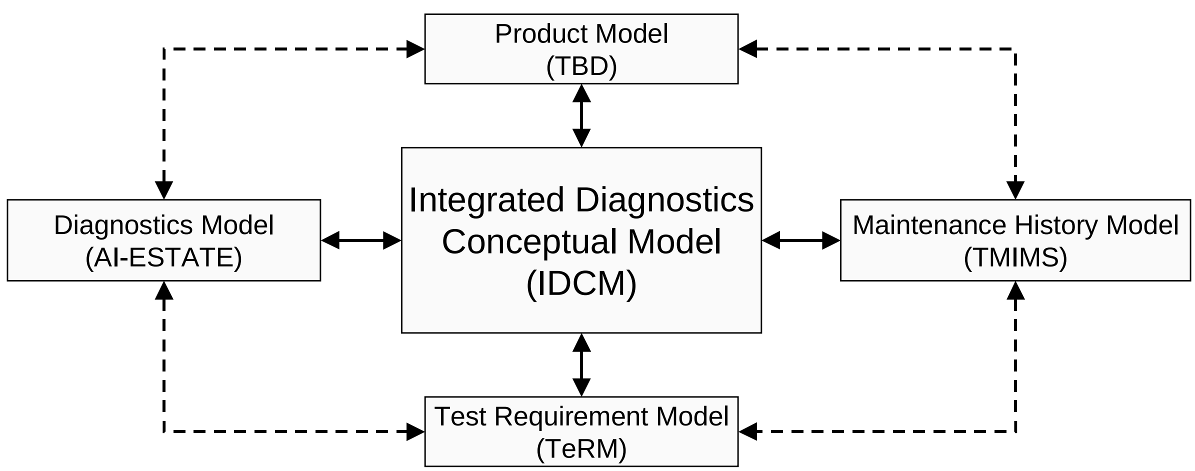

Figure 1. Integrating Models

What is missing in Figure 1 is the role of P1522 (Standard Testability and Diagnosability Metrics and Characteristics). It is expected that P1522 will draw from all of the models identified above, with specific emphasis placed on AI-ESTATE and TMIMS.

In response to an action item from the 00-D meeting of the DMC, a review of the NATO CALS Data Model (NCDM), version 4.0 was performed relative to requirements emerging relative to P1522 and TMIMS. Since the action item was assigned, it was discovered that the NCDM is now referred to as the NATO Product Data Model (NPDM); however, in this document, we will continue to refer to the model as the NCDM.

NATO CALS DATA MODEL

According to the NATO CALS Office (NCO) the NCDM is "a formal description of the data required to support the logistics process for the acquisition and support of major systems. (NCDM §1.1)." The NCDM was constructed with the recognition that information modeling occurs at three levels, in accordance with the three-layer architecture set forth by the ANSI/X3/SPARC Study Group on Database Management Systems. Specifically, the three layers identified include:

1. **The conceptual layer—**This layer serves as an integration layer. The layer does not specify implementation details, nor does it specify details from the perspective of any given application.
2. **The internal layer—**This layer contains the physical data model. The focus of the layer is the definition of storage formats; however, a requirement on the layer is that it be able to import data from other sources that conform to the semantics of the conceptual model.
3. **The external layer—**This layer provides a view of the conceptual layer from a particular perspective or application. It too is a conceptual model; however, it is specialized to include details from the target domain.

Within the three-layer architecture defined above, the NCDM has the role of the conceptual layer in the logistics process. The NCDM is intended to support exchange of logistics data between different operations and applications.

In terms of the evolution of the NCDM, it is interesting to note that the NCO is in the process of shutting down support for the model. It is believed that the model has reached a level of maturity such that no further support is warranted by NATO; however, the model has been supplied to the Product Life Cycle Support (PLCS) initiative to serve as a starting point for their activities. The hope is that PLCS will be able to take advantage of the work done on the NCDM rather than "reinventing the wheel." Note it is also anticipated that PLCS will be focusing more on defining models at either the internal layer (i.e., specifying formats) or the external layer (i.e., specifying views of the information). At this point, it is unclear what role PLCS might play relative to the goals of the DMC.

The NCDM is defined with a Core Model and several subsidiary models.

1. Product Configuration
2. Failure Analysis

3. Task Description
4. Technical Documentation
5. Logistics Support Anlaysis
6. Supporting Models

It is interesting to note that the Core Model "reaches down" into the subsidiary models for the purpose of obtaining the needed detail. Consequently, we can envision the Core Model as the integrating model for the NCDM.

THE CORE MODEL

The purpose of the Core Model as defined in NCDM §4 is to provide a high-level definition of various views of a product. Specifically, the model is constructed to facilitate viewing the model from three perspectives:

1. The product as required
2. The product as designed
3. The product as built

The as-required view of the product is supplied via a model of a product "concept." Product concept is defined as part of the Product Configuration Model schema and is referenced within the Core Model by the product design definition, the product instance definition, and an alias identification. The as-designed view is addressed more directly by the Core Model through product definition, product instance definition, and element definition. The as-built view (also as-used) view is also covered via the product instance definition as well as much of the Product Configuration Model.

Ultimately, the product entity represents the core concept of a product design. Various "breakdowns" of a product can be modeled via the Core Model that can parallel logistic breakdowns such as physical breakdown, functional breakdown, etc.

In examining the EXPRESS-G of the product design structure, it was interesting to note how parent-child relationships were represented. Where AI-ESTATE represents these relationships with direct, recursive attributes, NCDM does not. Instead, the NCDM creates a separate entity for a given relationship and identifies the "relating" and "related" parties in the relationship. The advantage to the NCDM approach is that it permits attributes to be associated with the relationship itself. This idea is analogous to resolving many-to-many relationships in relational data models via a relationship table.

Another interesting stylistic item within the NCDM became evident from examining the Core Model. The Core Model permits specification of attributes with "standard" values and "non-standard" values. This is similar to AI-ESTATE in the way it defines standard and user-defined outcomes for tests and diagnoses, except that the NCDM approach is rather clever. The approach is illustrated in Figure 2. Specifically, the value of a given attribute is tied to a SELECT type that selects between a standard value (defined in an enumerated type) and a non-standard value (defined as an attribute-value pair).

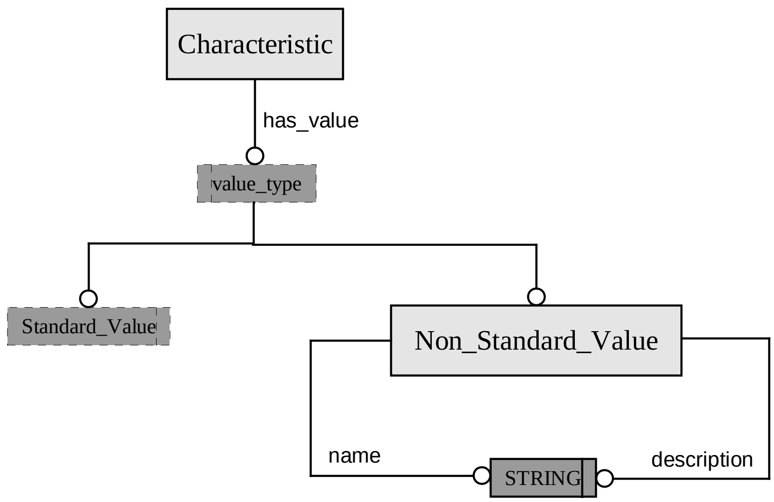

Figure 2. Construct for Standard/Non-Standard Values

The product instance definition relates directly to the repair_item as defined by AI-ESTATE. Specifically, the product instance definition identifies itself as a "realization" of a product concept. Further the product instance definition is an ABSTRACT supertype of a serial-numbered instance, a lot-numbered instance, and some kind of material. Of note about the material is the fact that the product can be identified with something that is not individually tracked.

Another interesting characteristic identified in the Core Model (which makes use of the Product Configuration Model) is the ability to track configuration changes of a product. Specifically, the product instance substitution identifies those instances that are substituted in a higher-level product as a result of upgrade, repair, or other reason. Since the substitution identifies the predecessor and successor product instances, it provides a means to track the changes within the model.

Another interesting observation about the NCDM is the fact that it treats a logistics activity as any maintenance activity. Specifically, logistics activity is defined to be an ABSTRACT supertype of one of corrective maintenance, preventive maintenance, adaptive maintenance, and perfective maintenance. Unfortunately, the Core Model does not take advantage of the definition of tasks in the Task Description Model to further refine the definition of the activities; otherwise, it is apparent this could be a rich construct for modeling maintenance activities.

Another nice feature of the Core Model is the ability to track usage history. This will be significant for both TMIMS and P1522 in terms of relating the usage to the maintenance history and tracking field maintenance statistics.

PRODUCT CONFIGURATION

The Product Configuration Model draws heavily from concepts defined by ISO 10303–Part 41 and ISO 10303–Part 44 (both of which were reference by IEEE Std 1232-1995). The Product Configuration Model provides the capability "to define and manage the configuration of complex items, over their full life cycle, to the serial number level (NCDM §5.2)." Essentially, the model proceeds by defining configuration "baselines" and then tracking modifications to the baselines.

The product concept is the starting point for a product configuration since it defines the requirements for the product. The product concept has several usage scenarios associated with it to serve as a means of specifying requirements from a user's perspective. In addition, the product concept includes several specifications corresponding to user requirements and combined using Boolean operators.

A particularly interesting notion introduced in the Product Configuration Model is configuration effectivity. The configuration effectivity identifies whether a particular configuration item is appropriate for use within a particular usage scenario, thus it can be used to identify preferred configurations of a product for satisfying particular user requirements.

The Product Configuration Model also defines an activity, which is defined to be identification of an activity that has been performed. Thus an activity is historical in nature and would be of direct use by TMIMS. Further, the activity entity includes a description of the task used to carry out that activity. Task descriptions are defined in detail in the Task Description Model.

FAILURE ANALYSIS

Given the scope of effort of the DMC, the Failure Analysis Model is of particular interest. The first thing noted in this model is the assertion that an "anomaly" is defined to be "a reason for doing something. There is something about the product that is not how it should be and so work must be done (NCDM §6.1)." The entity product anomaly is used to indicate that the product is not as it should be.

The NCDM description indicates that an anomaly can be either a failure or damage. However, the actual model shows an anomaly to be the root of a class hierarchy as shown in Figure 3. This model does not correspond to the description in that failure mode is a subtype of anomaly and damage is a subtype of failure mode.

An additional construct of interest within the Failure Analysis Model is a construct to enable cause-effect modeling. Upon an initial read of the section, it appears that the construct is intended to address issues found in a Failure Mode Effects and Criticality Analysis (FMECA); however, it is apparent that the effects addressed in this construct are not the same as effects in a FMECA. Specifically, the cause-effect information is captured with the product anomaly relationship entity that relates two or more anomalies together as "consequential failures"

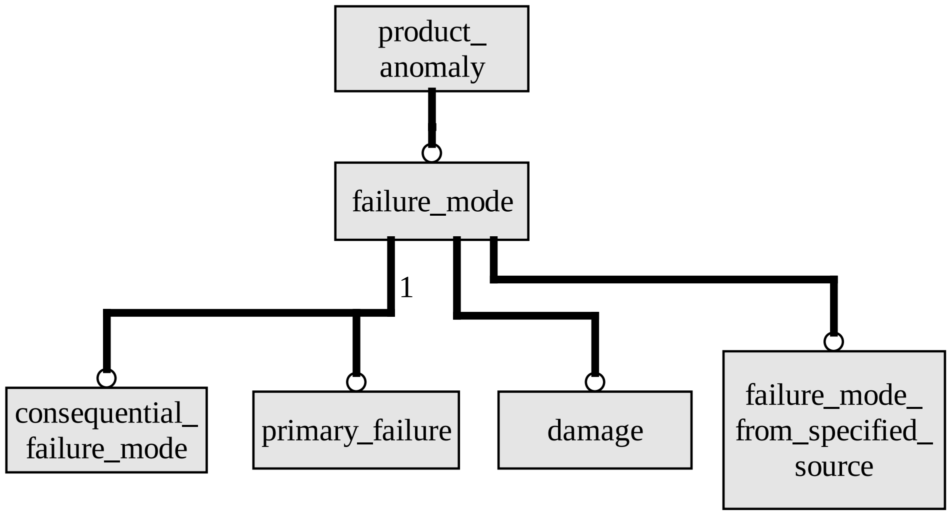

Figure 3. Anomaly Hierarchy

through the consequential failure relationship (NCDM §6.2.2). This is shown graphically in Figure 4. Some construct similar to this would be useful in certain kinds of diagnostics, especially when treating multiple faults; however, the implementation shown here is confusing and difficult to implement.

There is an additional confusing construct related to consequential failure relationships. Specifically, the product anomaly relationship identifies related and relating product anomalies. In addition, the consequential failure relationship, which is a subtype of the product anomaly relationship, identifies a relating anomaly. The type of the relating anomaly is consequential failure mode while the type of the relating product anomaly is product anomaly. Recall that a failure mode is a subtype of product anomaly, and consequential failure mode is a subtype of failure mode. The point of this discussion is that it is unclear how the relating product anomaly compares to the relating anomaly. The model provides no definition of relating anomaly that distinguishes it from the relating product anomaly. This indicates potential confusion and ambiguity in this part of the model.

The Failure Analysis Model also includes definition of detection methods for anomalies within the model. This is a very useful construct and can be related back to the test within AI-ESTATE; however, the detection method is categorized as one of a set of enumerated types: automatic test equipment, external equipment, human detection, manual test equipment, visual inspection, and other. First, it is unclear what the difference is between human detection and visual inspection (except insofar as human detection is a superset of visual inspection). Second, it was disheartening to note that the numerated list did not include any kind of on board or built in test equipment (except insofar as "other" includes it). Note that the detection method did not use the same construct for handling non-standard values as elsewhere in the model.

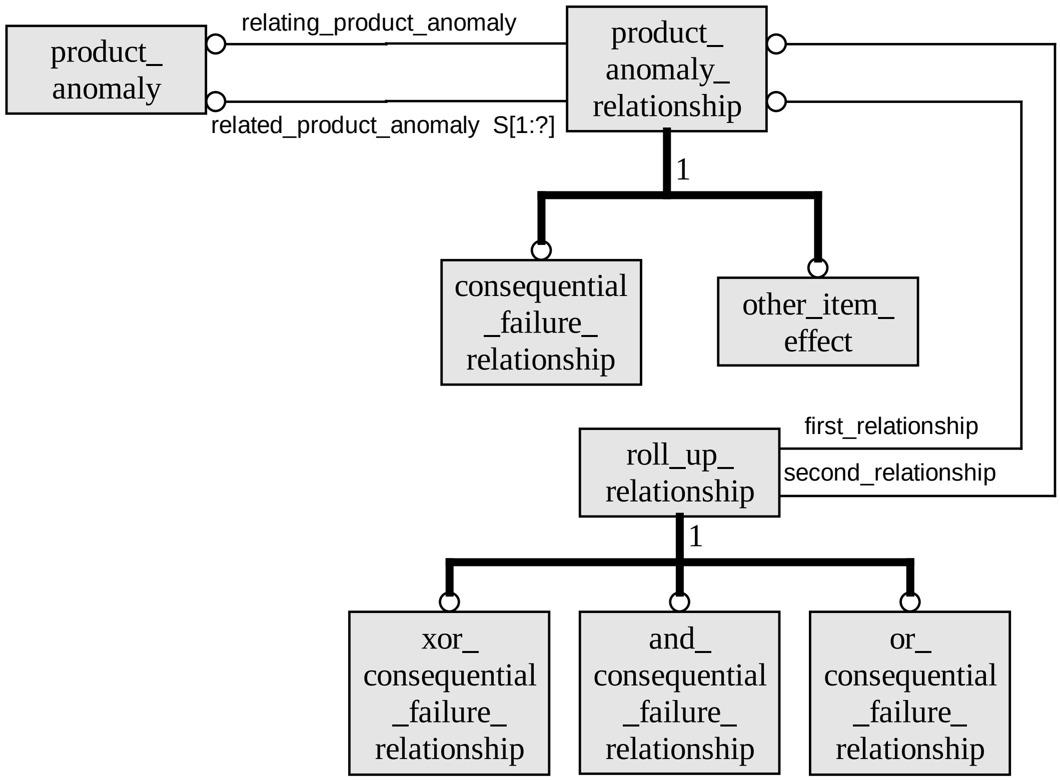

Figure 4. Cause-Effect Model

Within the AI-ESTATE model (Figure 5) is defined information relative to the severity and degree of a diagnosis (i.e., failure). Similarly, the NCDM specifies severity (within the Failure Analysis Model) and criticality (within the Task Description Model). Within AI-ESTATE, we seem to confuse these two concepts since we provide an attribute of diagnosis labeled severity with type criticality. Within the NCDM (§6.4.4), severity relates to a safety hazard and is enumerated as catastrophic, critical, marginal, minor, negligible, and none. Of course, this can be generalized beyond safety, but it is clearly intended to rate the severity of the failure mode relative to the ability to satisfy some requirement. On the other hand, criticality is intended to indicate "if failure to accomplish [a task] in accordance with system requirements would result in adverse effects on system reliability, efficiency, effectiveness, safety, or cost (NCDM §7.4.17)." Specifically, severity relates to the failure mode while criticality relates to a task. Otherwise, the two concepts appear to be virtually identical.

The AI-ESTATE model has also started to include information related to context within which a repair item is used and tested. The Failure Analysis Model includes information related to missions and mission phases that further extend the type of information captured relative to

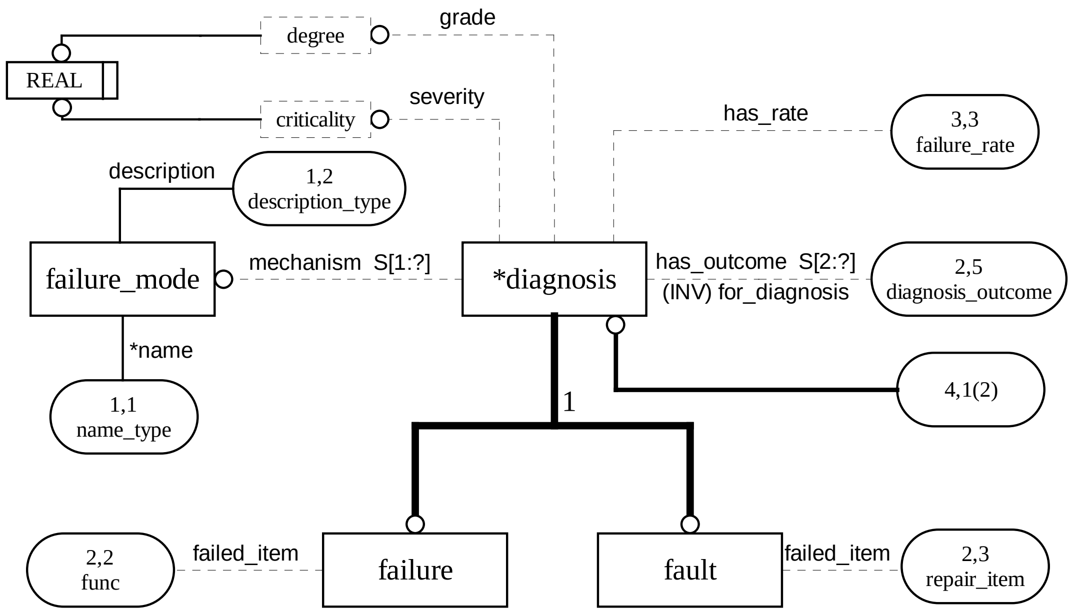

Figure 5. AI-ESTATE Diagnosis Submodel.

context. The construct used, while still not sufficiently detailed, is valuable and should be added to the context information included in AI-ESTATE.

TASK DESCRIPTION

In the previous section, we mentioned the Task Description Model. This portion of the NCDM provides an interesting and useful construct for modeling tasks (NCDM §7). Of particular note is the observation that a task need not be constrained to a simple sequence of actions. Note that AI-ESTATE limits both test and repair to a sequence of actions; however, it is not necessarily the case that this is an appropriate representation of these tasks.

In defining a task, The NCDM divides a task into a specification of “what is to be done” and “how to do it.” Further, in specifying how to perform a task, information on required resources (similar to AI-ESTATE) and decision logic is included. In fact, the NCDM states that the mechanism for specifying a task “is effectively equivalent to a simple programming language, rich enough to enable IETM style functionality to be provided directly from the data base that corresponds to the model (NCDM §7.1).”

A specific deficiency in the way the NCDM handles non-sequential tasks is in how it defines test conditionals. A test conditional is simply a way of testing some condition to determine which from among a set of alternative subtasks should be performed. The conditional is defined as a predicate; however, it is implemented simply as a text description. The order operator within AI-

ESTATE provides a formal definition of conditionals and could be used to improve on this construct by eliminating open-ended specification.

A particularly nice construct modeled within the Task Description Model is the structured task method. This entity is an ABSTRACT supertype of task method sequence (a sequence of steps), concurrent methods (steps performed concurrently), decision points (conditionals), terminating conditions, and looping methods. However, this construct could be simplified some by specifying the tasks functionally.

AI-ESTATE has struggled with how to expand the resource portion of its model. To date, the assumption has been that AI-ESTATE will rely on the Test Resource Information Model (TRIM) to fill in details on resources. However, the NCDM includes high-level constructs for resources that are more general than the TRIM. It is likely that the NCDM resources would serve as a closer match to handling resources (at least for now) than the TRIM.

It is interesting to note that the NCDM considers personnel to be a resource. This is reasonable; however, the NCDM goes a step too far by modeling personnel has synonymous with “personnel skill.” Skill level is generally considered to be a cost to be optimized. The skill portion of the NCDM is appropriate for inclusion in the cost model for AI-ESTATE, not as a part of the resource model; however, it is reasonable to include personnel as a type of resource that has an associated skill level.

TECHNICAL DOCUMENTATION

The NCDM recognizes that product data extends well beyond the type of information that can be specified formally in an information model. As a result, the NCDM includes a Technical Documentation Model that serves as a “catch all” for technical data (NCDM §8).

When the TMIMS project was first getting under way, it was expected that TMIMS would include ancillary technical data (or at least it would provide links to ancillary technical data) that would be helpful to support diagnostic and maintenance maturation. It is believed that such a structure is still needed. The NCDM manages the diverse information sources through specification of an information object. This information object is an ABSTRACT supertype of either a base information object (simple document) or a structured information object. The structured information object provides either a link to an external source (i.e., a document not in the database) or to additional information objects that are related in some way.

LOGISTICS SUPPORT ANALYSIS

The final major model within the NCDM is the Logistics Support Analysis Model. Recall that the NCDM treats all logistics activities as maintenance activities. Even so, additional information is required to support the logistics process besides just maintenance information. The Task Description Model began to expand into the realm of context by including mission information. The Logistics Support Analysis Model further extends into context by including a scenario and role model. The intent is to provide a means for identifying relevant portions of a large database based on usage scenarios rather than completely replicating the database and extending the database for each potential user (NCDM §9.2.1).

As a style issue, the NCDM Logistics Support Analysis Model includes an entity called "characteristic." The characteristic is analogous to the "generalized attribute" of a previous incarnation of AI-ESTATE, and it is intended to capture properties and characteristics in a fully general way. These characteristics can be used in one of five roles through the characteristic assignment type—allocated, calculated, measured, planned, and required. The last three are of particular interest. Note they correspond to an attribute value being measured (i.e., observed), planned (i.e., an expected value), or required (i.e., specified as a performance requirement). These can be applied directly to P1522 in characterizing metrics and characteristics as measured in the field, expected in the field, or required by a customer. Clearly, this distinction has benefit to TMIMS as well to assist in the maturation process.

An interesting subtype of characteristic is defined within the Logistic Support Analysis Model, supportability characteristic. This is actually an abstract supertype of one of the following—reliability characteristic, annual usage characteristic, testability characteristic, availability characteristic, and maintainability characteristic. Under reliability characteristic (as subtypes) we also find failure characteristic, inherent maintenance factor, and time between maintenance tasks. Recall that the LSA model is oriented towards maintenance tasks. It is interesting to note that the testability characteristics are defined with respect to detection, isolation (both consistent with P1522), and RTOKs and CNDs due to BIT. This is the only mention within the NCDM of BIT. It is encouraging that the NCDM avoids the issue of false alarms and focuses on the measurable RTOKs and CNDs.

SUPPORTING MODELS

Finally, the NCDM includes the definition of several supporting information models to capture details such as approvals, information about personnel and organizations, date and time information, and supporting resources. The NCDM also claims to be consistent with STEP Integrated Resources; however, it is unclear from the document how or where such IRs are utilized by the model or are consistent with the model.

ISSUES AND RECOMMENDATIONS

Based on the review of the NCDM, the following recommendations are offered:

1. Modify the AI-ESTATE construct for handling user-defined values to parallel the structure shown in Figure 2.
2. Adapt maintenance architectures within AI-ESTATE, TMIMS, and P1522 to be consistent with the logistics activity (Core Model) and maintenance concept (Logistics Support Analysis Model) entities.

3. Relate the AI-ESTATE repair_item to the usage history entities (Core Model) and configuration entities (Product Configuration Model) to better track configuration changes in the repair item.
4. Track activities within TMIMS analogous to the activity entity within the Product Configuration Model.
5. Include constructs to address cause-effect relationships between failure modes; however, do not follow the construct provided within the NCDM.
6. Parallel NCDM in definition of criticality and severity.
7. Enhance the AI-ESTATE model to include information on mission and mission phase within the context portion of both the Common Element Model and the Dynamic Context Model.
8. Extend the test and repair sequences to permit more than simple sequences of actions. Something similar to the NCDM could be provided; however, the programming language included in the NCDM may be going to far. Perhaps permitting specification of tasks within AI-ESTATE in a "functional" manner (i.e., limited to conditionals and recursion can do this). In addition, test conditions should be tied to conditionals as defined within the order operator of AI-ESTATE.
9. Examine the NCDM resource item in the Task Description Model as a possible expansion of the resource entity in AI-ESTATE.
10. Expand the AI-ESTATE resource model to include personnel. Expand the AI-ESTATE cost model to include skill level. Use the personnel skill entity within the NCDM Task Description Model as representative of what to include.
11. Utilize the information object construct from the Technical Documentation Model to support ancillary product information with TMIMS.
12. Adapt and apply the scenario and role portions of the Logistics Support Analysis into the AI-ESTATE model. This can be used to support P1522. As mentioned previously, also include constructs related to maintenance concept for the same reason.
13. Within P1522, specify a measure as one of the characteristic assignment types in the Logistic Support Analysis Model.
14. Utilize supportability characteristics as defined within the Logistic Support Analysis Model to the extent that these characteristics support the P1522 standard. Extend the characteristics as necessary; however, avoid extensions that would be contentious and difficult to define formally.

One significant issue arose while reviewing the NCDM. It is apparent that a great deal of work went into the definition of the NCDM, and the DMC should exploit this work to the maximum extent possible. Unfortunately, the NATO CALS effort appears to be on the decline. Based on discussions with Michael Danielsen, version 4 of the NCDM will be the last version. However, it is also unclear how PLCS will play in this.

As a standards committee we need to decide among three approaches:

1. Interface our models with NCDM v. 4, recognizing there will be no further development.
2. Extract and modify relevant portions of the NCDM for our use and maintain this subset within our committee.
3. Hope that PLCS will adopt and continue to develop the NCDM with input from us. Then interface our models with the evolving PLCS models.

The first alternative is problematic due to the fact DMC has no way to correct the deficiencies identified in the model relative to our work. However, the advantage to using the current version of NCDM is that the model is locked.

The second alternative is problematic due to the added complexities associated with creating and maintaining an additional model. However, the advantage to using and maintaining an AI-ESTATE-specific version of the NCDM is that we have control and can tailor the model to meet our needs.

The third alternative is problematic for two reasons. First, it is unclear at this stage what role the NCDM will play with NCDM. It is also unclear how PLCS will play in the standards arena. Mike Danielsen indicated a desire by US DoD and Industry representatives in PLCS to focus on international standards rather than NATO-only solutions; however, the how PLCS plans to do this is unknown at this time. Second, the models maintained by PLCS would most likely be a moving target for the foreseeable future (at least until PLCS and its efforts become well established). Interfacing with such a moving target is always a challenge at best. Of course, the advantage to working with PLCS is the ability to draw on them as a resource for purposes of completing work we need and increasing our visibility and credibility in the industry.