Paper for the 8th International Command & Control Research & Technology Symposium

Title: *Time-Based Tree Graphs for Stabilized Force Structure Representations*

Submitted by: Sam Chamberlain
U.S. Army Research Laboratory
ATTN: AMSRL-CI-CT
APG, MD 21005-5067
410-278-8948
Fax: 2934 / 9223
wildman@arl.army.mil
http://www.arl.army.mil/~wildman

and

Chris Leeds
Chief, Combat Service Support MTOE Division
U.S. Army Force Management Support Agency
ATTN: MOFI-FMA-SA
9900 Belvoir Road, Suite 120
Fort Belvoir, VA 22060-5578
703-805-4189

Time-Based Tree Graphs for Stabilized Force Structure Representations

Dr. Sam Chamberlain

U.S. Army Research Laboratory
ATTN: AMSRL-CI-CT
Aberdeen Proving Ground, MD 21005-5067
410-278-8948
wildman@arl.army.mil
http://www.arl.army.mil/~wildman

Mr. Chris Leeds

Chief, Combat Service Support MTOE Division
U.S. Army Force Management Support Agency
ATTN: MOFI-FMA-SA
9900 Belvoir Road, Suite 120
Fort Belvoir, VA 22060-5578
703-805-4189

Abstract

A basic, and often key, component of any battlefield representation is the forces involved. Obtaining high-resolution force structure data has always been a major task. This is true whether the representation is for simulated or actual operations. However, the problem extends far beyond simply obtaining a single force structure snapshot. The real challenge is maintaining the data, especially when numerous other programs are creating and linking their data to the force structure. This paper describes an approach for maintaining consistency in a high-resolution database of Army units that is undergoing continual change due to force modification. The use of time-based tree graphs is proposed as a technique for providing stability and maximizing the retention of existing entities to minimize the effect to systems that use the data. In a network-centric context, an easily accessible repository called the Army Organization Server (AOS) is under development that will contain the evolving, default force structure of the Army.

1. Introduction

In June 1998, pursuant to a discussion with the Director of Research and Strategic Planning, OASD(C3I), a study was conducted to answer the question: "With all this great technology, why can't our system interoperate?"[1] The resulting report[2] cited three voids that perpetuate the inability to integrate systems; they are:

- The lack of a common naming convention,
- The absence of a central theme for data integration, and
- No designated authoritative sources of information.

The proposed solution has evolved into three components.

---

[1] OASD(C3I): Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence. Conversation with Dr. Dave Alberts, Director, Research & Strategic Planning, June 1998.

[2] Chamberlain, Sam; *Default Operational Representations of Military Organizations*, Army Research Laboratory Technical Report: ARL-TR-2172; February 2000; see: http://www.arl.army.mil/~wildman/PAPERS/tr2172.html

The first is the adoption of a common naming convention, called *enterprise identifiers*, or EIDs, to uniquely tag data across the enterprise.[3]

The second is the assertion that the central theme through which all battle command processes converge is the fundamental concept of *force structure*. Consequently, a formal representation of force structure is required as a foundation to integrate other battle command concepts.

Finally, it is the actual force structure data, not its theory or a model, which is required by the battle command system users. Historically, in the absence of data, those in need create it themselves. Force structure data is no different. As a result, there are several different sources of force structure data available throughout the Army, in various forms and different levels of detail. Clearly, they are not synchronized and require extensive effort to keep current. A recommended solution is to have the official domain experts (those who design and develop force structure as a profession) provide and maintain this data for the user. However, in doing so, the requirements of the user must be included in the design of a repository and the processes that are used to maintain it.

2. Background – Tree Graphs

A convenient, mathematical tool for describing force structure is graphs, or more specifically, graph theory. A common manifestation of a graph is an organization chart. Because of the basic principles of military command, military organizations are conveniently represented via hierarchical organization charts that describe the aggregation and composition of clusters of people and equipment. A ***graph*** is composed of a set of ***nodes*** connected by a set of ***links.*** In mathematical vernacular, the nodes are called *vertices* and the links are called *edges*. An example of a graph is illustrated in Figure 1 where a graph called "G" is composed of a set of vertices, V, where V = {A,B,C,D,E}, connected by a set of edges, E, where E = { (A,B),(A,C),(A,D),(C,E),(C,F) }. There are many ways to connect the nodes listed in V, and the structure provided by E is just one.

A ***tree*** is a special type of graph that is *fully connected* (i.e., every node is linked to at least one other node) and there are *no cycles* (i.e., when links are traversed, only one path exists between any two nodes). When these two criteria are met, the graph must be a tree. Org charts are trees. Normally, one node is selected as the "beginning," or top, of the tree and is named the root node. Figure 1 summarizes several tree graph terms and illustrates how they are easily exploited to denote hierarchical organization charts.

---

[3] For explanations of this solution, see: https://ess.arl.army.mil. Also, the following papers are available:
Chamberlain, Sam, *Implementation of an Enterprise Identifier Seed Server for Joint and Coalition System,* Proceedings of the 7th International Command and Control Research and Technology Symposium; Quebec City, Quebec, Canada; 16-20 September 2002; http://www.dodccrp.org/Activities/Symposia/7thICCRTS/Tracks/pdf/109.PDF
and
Chamberlain, Sam; *An Enterprise Identifier Strategy for Global Naming Across Arbitrary C4I Systems*, Proceedings of the 6th International Command and Control Research and Technology Symposium; US Naval Academy, Annapolis, MD; 19-21 June 2001; Presented 19 June 2001. http://www.dodccrp.org/6thICCRTS/Cd/Tracks/Papers/Track2/059_tr2.pdf.

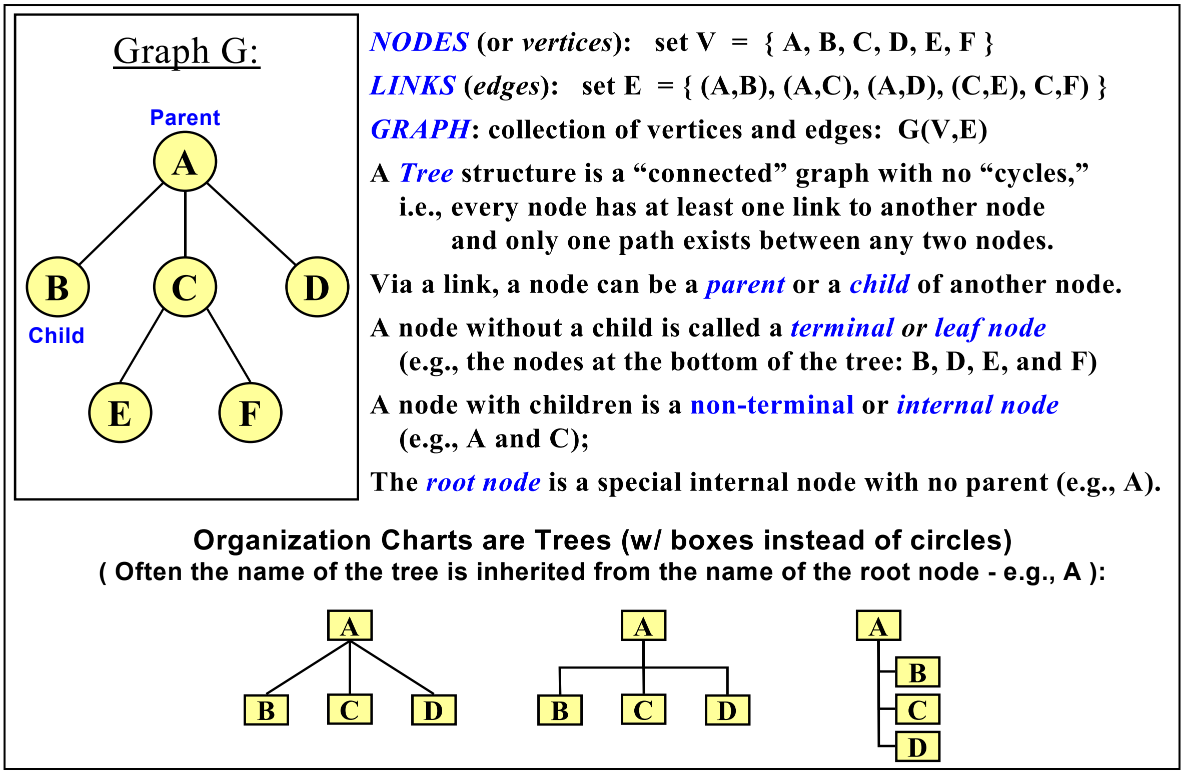

Figure 1: Tree Graph Definitions and Terms

Terms like "unit" or "structure" are ambiguous in isolation; therefore, a more formal definition is required. Using the tree graph formalism, a node of a tree can be named an "organization" (e.g., node A in Figure 1 is called Organization A.). The term "association" can be used to refer to a link of a tree (e.g., the line connecting nodes A and C, denoted by (A,C), in Figure 1 can be called association "AC", or more specifically, organization-association "AC"). An organization chart is a tree graph composed of a set of nodes and a set of links, or in this new vernacular, a set of organizations and a set of organization-associations. For convenience, the graph can be called a "unit." Thus, a unit (a graph) is composed of organizations and organization-associations.

The action of moving from node to node along the links of a graph is called "traversing" the graph and there are numerous, well-known algorithms for doing this. The links and nodes of a graph may include additional attributes to allow them to be filtered (i.e., selected or deselected) during the traversal process. This allows different paths to be followed by applying parameter constraints during the traversal process, as is illustrated in Figure 2. The left-most graph, marked "Base," shows all the nodes and links with the addition of a label *a*, *b*, or *c*. To traverse this tree, one provides a set of permissible labels to be used during the traversal process. The middle graph illustrates the case in which only nodes and links with a label of *a* or *c* are included. The right tree illustrates the case in which only nodes and links with a label of *b* or *c* are included. One can include as many different labels as necessary to describe the different path combinations.

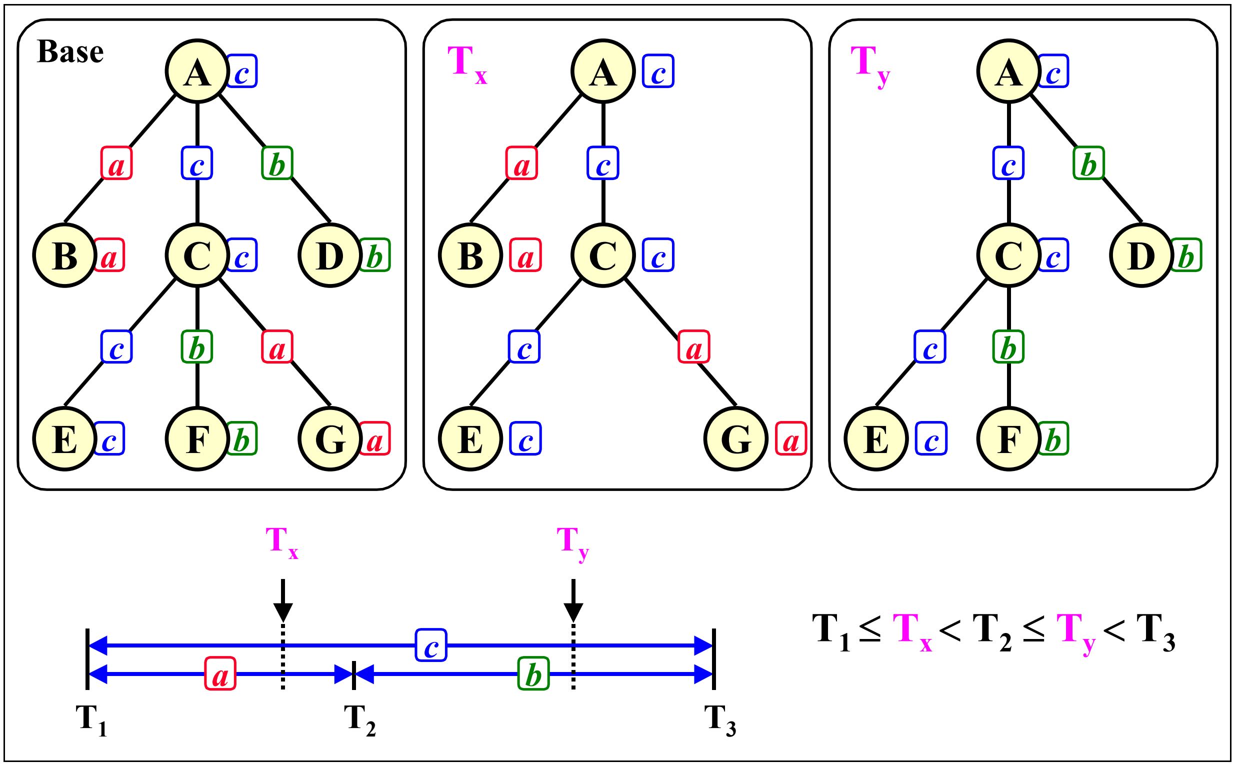

Figure 2: Time-Tagged Nodes and Links

To simplify the annotation and selection process, a sequence of always increasing (or decreasing) numbers may be used.[4] A convenient set of numbers that meets this criterion is *time*. Consider a graph where every node and link has a time interval included in its definition (i.e., a start time and an end time). The lower part of Figure 2 includes a timeline that denotes three time intervals using the three times T_1, T_2, and T_3. The time period from T_1 to T_2 is given the label *a*, the period from T_2 to T_3 is labeled *b*, and the period from T_1 to T_3 (the concatenation of periods *a* and *b*) is labeled *c*. One can now apply the time-based meaning of these labels to the same labels in the graphs of Figure 2 and use any time on the time line as a value to selectively traverse or filter the nodes and links of the graph. Any node or link whose time interval includes the provided time is included in the traversal process.

The middle graph shows the result of selecting time T_x (from the timeline in Figure 2), which is included by both time intervals *a* and *c*. The right graph shows the result of selecting time T_y, which is included by both time intervals *b* and *c*. This technique provides a simple mechanism for building selectable graphs using a single parameter (i.e., time) even though there may be many different intervals (i.e., labels) associated with the nodes and links of the graph. Notice that this technique may be used with *any* sequence of always increasing (or decreasing) numbers. Time just happens to be a very familiar, and natural, choice because many processes are based upon it.

---

[4] In mathematics this is called a monotonic function. In this case, it is a monotonic increasing function. See: http://newton.dep.anl.gov/newton/askasci/1995/math/MATH136.HTM.

3. The Current Process and Data Modeling

In the current U.S. Army force structure documentation process, documents called Modification Tables of Organization and Equipment, or MTOEs, describe the authorized structure of military forces. There are about 4900 MTOEs that cover the active, reserve and National Guard components of the Army. These MTOEs may change several times per year. Force structure data is an integral part of automated battlefield information systems. Managing the force structure data within these systems is a challenging task that is currently accomplished via manual means. A major objective is to move towards an automated system to manage this increasingly voluminous data that is becoming higher in resolution due to the impending requirement for soldier level data. For this reason, the primary impetus for the design and implementation of a data representation is the ease and simplicity of automated maintenance. To be a viable system, force structure updates must occur in a manner that is automated and transparent to the battle command system user.

The task of maintaining this data can be significantly simplified by selecting a representation that reduces the effects of modification by isolating and minimizing changes. To achieve this objective, a new approach for representing force structure is proposed that uses timed tree graphs, like those of Figure 2. This allows one to execute "tree traversal" algorithms to arbitrarily move up and down the tree selecting and deselecting nodes and links based upon time. However, a major task is the conversion of the current, discrete, document based system into one that is continuous, timed-based, and uses formally specified tree graphs. To understand how the current system can be transformed to meet these new requirements, one must first understand how the current documentation system is configured.

An interesting feature of the MTOE is its dual personality; it is used to describe both generic force structure, like its counterpart the TOE, and to describe real units. The TOE is a requirements document that describes the model case and is used as the starting state for building an MTOE. Associated with every resource in a TOE is a value that indicates the number of assets required. To develop the MTOE, the model (TOE) is analyzed and adjustments may be made to the requirements, to include the addition of qualifying information, or in some cases, adding or deleting requirements for, or the amounts of, personnel and/or equipment. The final authorizations are based upon many other variables and constraints, such as budget, unit priority, and resource availability. In most cases, the authorized amounts are equal to the required amounts, but sometimes authorizations are reduced below the requirement. So the name Modification TOE is quite appropriate.

The TOE and MTOE have nearly identical structures, most notably, the use of multipliers in the document. There will be multipliers for both the number of resources required and the number authorized. There may be 18 Automatic Riflemen (AR) required and 18 authorized. This approach is adequate for describing generic structures; but in real units, where each of the 18 ARs must be tracked individually, there must be 18 separate individual ARs. Clearly, there is a difference between the definition of the position named AR and the 18 separate entities that are the actual instances of the definition. This is a common trait of data models for which a thorough description is beyond the scope of this paper. Suffice it to say that a real unit is an instance of a definition; and in this case, the instances are called *organizations* (nicknamed org) and the descriptions are called *organization-types* (nicknamed org-type). Therefore, there will be two different structures of trees: *org-type trees* that contain multipliers and *org trees* that do not. The org tree contains instances of the org-type tree nodes that are expanded based on the multipliers in the links of the org-type tree. These structures are illustrated in Figure 3 using the example of the platoons authorized within a company.[5]

From the org tree and org-type tree structures, one can understand why org-trees are necessary to describe real units. It is not possible to use multipliers to represent individual units. On the battlefield, one cannot track "3 X Rifle Platoon" that is represented as a single link and node in the org-type tree. Instead, this must be expanded into three separate rifle platoons named (in this example) 1st Platoon, 2nd Platoon, and 3rd Platoon. This capability has been a primary impetus for the development of an Army Organization Server (AOS) that will provide automated battle management systems with a source of force structure data that meets their operational requirements.

From Figure 3, one can see that an org is an instance of an org-type. This is explicitly denoted by the horizontal, dashed line, called an "IS-A" link, that associates each org with the corresponding org-type from which it was established; for example, "1st Platoon IS-A Rifle Platoon," as are 2nd and 3rd Platoons. There may be many (e.g., hundreds) of instances that refer to a common org-

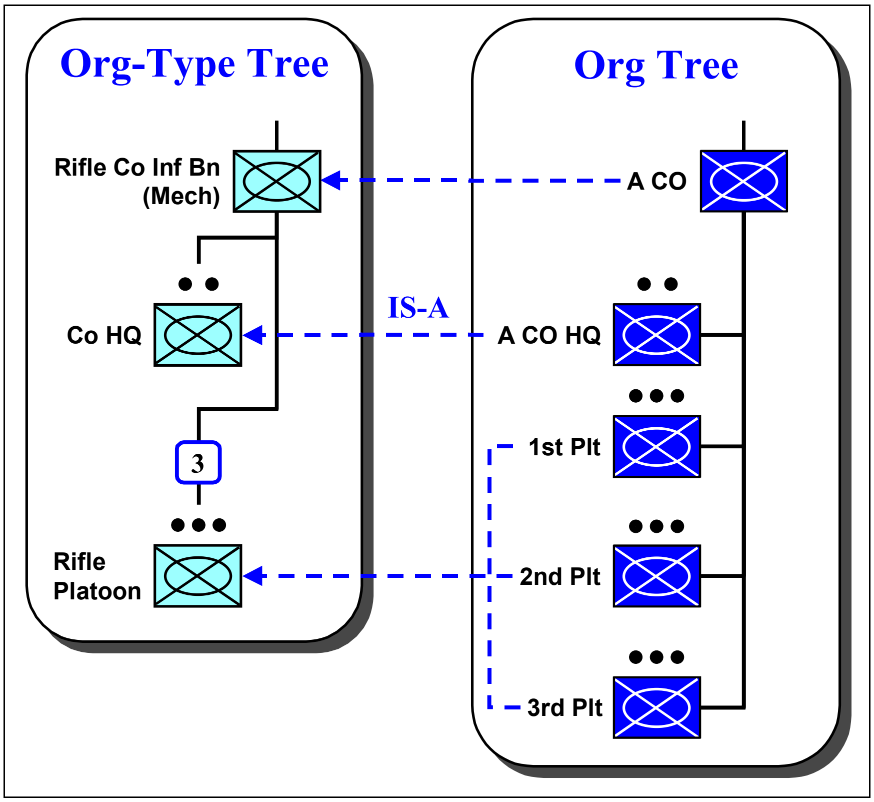

Figure 3: Org and Org-Type Trees

5 A popular data model using this terminology and semantics is the Land Command and Control Information Exchange Data Model, or LC2IEDM, that is a proposed NATO Stanag.
See: https://akea-cio.army.mil/admg/html/datamodels.asp; Army Knowledge Online account required).

type, and this allows information common to all the instances to be stored in a single place via the org-type tree.

Semantically, the org tree represents real units while the org-type tree contains the descriptions of those units. This causes confusion when comparisons are made with the current MTOE implementation because of the MTOE's dual nature. An MTOE is used to describe both generic force structure and real units. In other words, it contains facets of both orgs and org-types: *an MTOE is used to represents a real unit, but it does so using multipliers*. This is the quandary; semantically an MTOE maps to an org tree, but structurally it maps to an org-type tree (with multipliers). This circular contradiction appears to cause a clash in mapping between the current process and future model-based schemes. However, this problem is easily fixed.

The dual nature of MTOEs is exemplified by the fact that there are four primary identifiers associates with an MTOE: a DOCNO (document number), a CCNUM (command and control number, which is analogous to a version number), a UIC (unit identification number), and an EDATE (effective date – the date the unit's status will be compared to that MTOE). A minimum of two of these identifiers is required to identify an MTOE. From the perspective that an MTOE is simply a modified TOE, the DOCNO/CCNUM combination is the natural identification scheme. A DOCNO is derived from the TOE identification number (called a standard requirements code, or SRC) and identifies a type of unit (e.g., an "INF BN MECH (FXXI)," a Force XXI structured Mechanized Infantry Battalion). The CCNUM identifies the version of the document and has four digits: a two-digit version number followed by a fiscal year. So a CCNUM of 0103 would be version 01 in FY03.

From the perspective that an MTOE represents a real unit, the UIC/EDATE combination is the natural representation. A UIC identifies a military unit (typically a battalion for an MTOE) and the EDATE indicates the date for which the unit must meet the specification of the MTOE. So if one queries an MTOE database with either an UIC/EDATE or a DOCNO/CCNUM combination, a single result will be returned.

Figure 4 illustrates this dual nature of an MTOE. It is a complex diagram that shows the relationship between DOCNO, CCNUM, UIC, and EDATE. There are nine boxes representing nine different MTOE documents that contain authorization details for one or more unit. At the top of each box is the CCNUM for that MTOE; and the boxes are positioned, left to right, in increasing CCNUM order. Four of the boxes represent the older "L-edition" structure (initially designed in the late 1980s) and are distinguished by a CCNUM with underlined (green) letters. The other five boxes represent the newer "F-edition" (Force 21) structure and are distinguished by a (purple) CCNUM without an underline. The nine boxes reflect the perspective that a DOCNO and CCNUM define an MTOE that can be represented by an org-type tree. In this case, one would state that there are nine MTOE documents present, identified with different CCNUMs.

To the left of the boxes is a list of the four UICs of the real units that are established by this series of MTOE documents.[6] Dots are placed in the boxes for the MTOEs that are used by the units –

[6] WAGL, WAGN, WEZE, and WEZK are the UICs of the four mechanized infantry battalions in the 1st Cavalry Division: 1st-5th, 2nd-5th, 2nd-7th, and the 1-9th Cavalry Battalions, respectively.

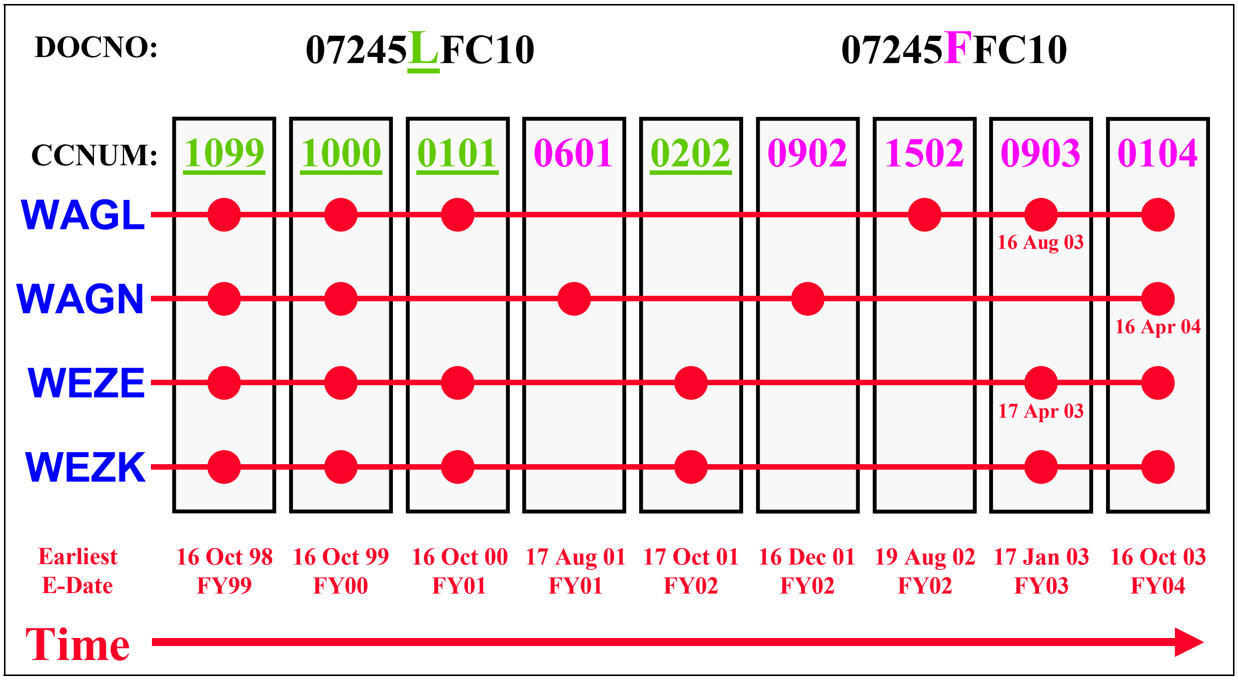

Figure 4: Time-Ordered List of MTOE Documents

clearly, every unit does not use every MTOE. Below each box is the earliest EDATE of the units that use that MTOE. Other EDATEs are listed inside the box next to the dot of the units that has that EDATE. When no other EDATEs are listed, all the dots in the box share that same EDATE. From this perspective, one would state that Figure 4 denotes 23 MTOE documents. This identification technique is implemented by placing a list of the UICs and their EDATE in the header of a document specified with a DOCNO/CCNUM. In this example, a set of dots located in the same box represents the units (each with a UIC) that share a common structure and authorization specification, although each unit may have a different date (EDATE) to meet that specification. The defining question is: *are there nine or 23 MTOE documents displayed in Figure 4*? The problem is that the answer is yes.

The solution is straightforward: an MTOE, as it is currently defined, cannot be represented by a single tree. The "boxes" are represented by org-type trees and the "dots" are represented with org trees. The result is that an MTOE and a TOE have nearly identical structures that are represented using org-type trees with multipliers. Once the generic structure is decided, an org tree is created for each real unit; and this process includes expanding the multipliers into individual, trackable entities. Finally, every node of an org tree is linked to the corresponding node in the org-type tree from which it is established. Authorization information (e.g., personnel and equipment) is associated with the nodes of the org-type tree, so that this information is maintained in a central location that is shared by all the real units represented by org trees. Once this design feature is in agreement, the issues associated with implementing timed trees can be addressed.

4. Time in Org and Org-Type Trees

Figure 3 illustrates three structures: the org-type tree, the org tree, and the association between them (i.e., the "IS-A" links). Now consider the case in which every node and link has a time interval associated with it. The interval is denoted using a start value (called *s_date*) and a termination value (called *t_date*). Although all the nodes and links use the same format to define this time period, the use of this interval is different depending on the structure.

In an org tree, which represents a real unit, the time is real time. In this structure the term effective date is appropriate as the time periods associated with each node or link do indeed represent real time. If one wants to see the default structure of a unit on 15 Oct 2002, then that date is used during the traversal process to select only those nodes and links with time periods that encompass that date. Any node or link whose time period does not include the given date is excluded. Figure 5 illustrates a situation in which a unit changes structures on a given date. In this example, the org tree has six nodes and five links. All the nodes and links, except the right most, have the same associated time period that extends back and forward in time. However, the right most node and link have a different associated time period. In this case, the right most node and link are no longer viable as of 17 Aug 2001. Therefore, if one does a tree traversal using a date on or before 16 Aug 2001, the right most node (D Company) will be included; if a date on or after 17 Aug 2001 is used as the criteria, the right most node will not be included. In other words, this is how one represents the fact that on 17 Aug 2001 this unit changes structure and the current use of the term EDATE continues to be appropriate. In the current vernacular, it is correct to state that 17 Aug 2001 is the EDATE for the transition of this unit from one structure to another; other units may have the same or different EDATE.

Now consider the org-type tree. Recall that this tree contains the generic template (with multipliers) that defines the structure and associated description about personnel and equipment that are used by the org trees. Several org trees may refer back to a single org-type tree. It is important to understand how the different links that are associated with the org-type nodes of the tree differ in their use.

First, within the org-type tree, there are the "vertical" links that denote the parent-child relationships, with multipliers, among the nodes. These are illustrated in Figure 6 with some of the associated time periods annotated. These links are used by the force development community to traverse up and down the tree as the sequence of modification is developed and specified.

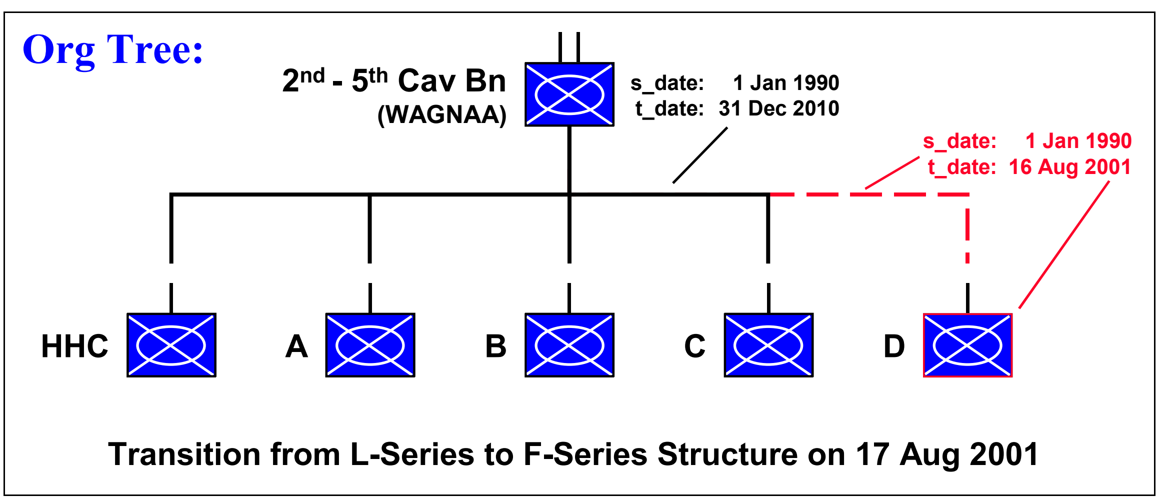

Figure 5: Example of a Time Based Org Tree

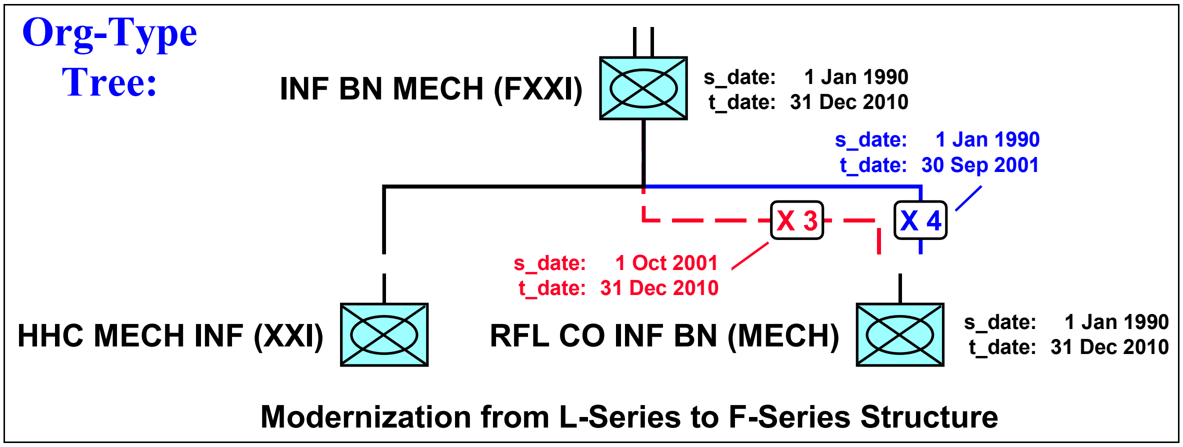

Figure 6: Example of a Time Based Org-Type Tree

Figure 6 portrays the generic case for the situation in Figure 5. There are two links between the top and right-hand nodes with mutually exclusive time periods. The link with the multiplier of 4 is valid until a date of 30 Sep 2001, at which time the link with the multiplier of 3 becomes valid. So if the org-type tree is traversed using a date on or before 30 Sep 2001, the battalion is authorized four rifle companies. If the tree is traversed using a date on or after 1 Oct 2001, then three rifle companies will be authorized. Ultimately, it is anticipated that org trees can be built and maintained automatically via the exploitation of these links.

For org-type trees, the actual value of time used to define the time periods is irrelevant. Recall that the org-type tree represents the nine boxes of Figure 4. These boxes indicate a relative state of evolution, and so it is with the org-type tree. Because there may be many org trees referring to a single org-type tree for their authorization data, a single, real date, like an EDATE, is not meaningful. All that is required of a date is that it reflect a state of evolution, and that a later date indicates a later state of evolution (i.e., a point farther in the future). In other words, the time periods associated with the nodes and links of the org-type tree refer to an independent timeline representing *evolution* time. This is in contrast to the time periods of the org tree that reflect real time. Instead of an EDATE, an org-type tree reflects a modification date, or MDATE, that is simply an indicator of relative time. Therefore, the MDATE and EDATE do not have to correspond in any relationship other than a relative one. When one traverses an org-type tree, an MDATE must be provided to execute the node and link selection process. This is analogous to stating: "show me the generic template for a typical Mechanized Infantry Battalion with a modification state of 1 Oct 2001." This is the fundamental reasoning behind the process of evolution for the force developer.

Second, there are the "horizontal" links between the nodes of the org tree and their corresponding node in the org-type tree; these are the IS-A links of Figure 3. To obtain the information about what personnel and equipment are authorized for the particular real organization, the IS-A link is traversed from the org tree node to the corresponding node in the org-type tree. End users and battle command systems will rarely, if ever, traverse up and down the vertical links of the org-type tree. They are interested in the real, not generic, unit structure, so they traverse up and down the org tree and only refer to the nodes of the org-type tree (likely unknowingly) to obtain the associated authorization data. From the perspective of users outside the force development

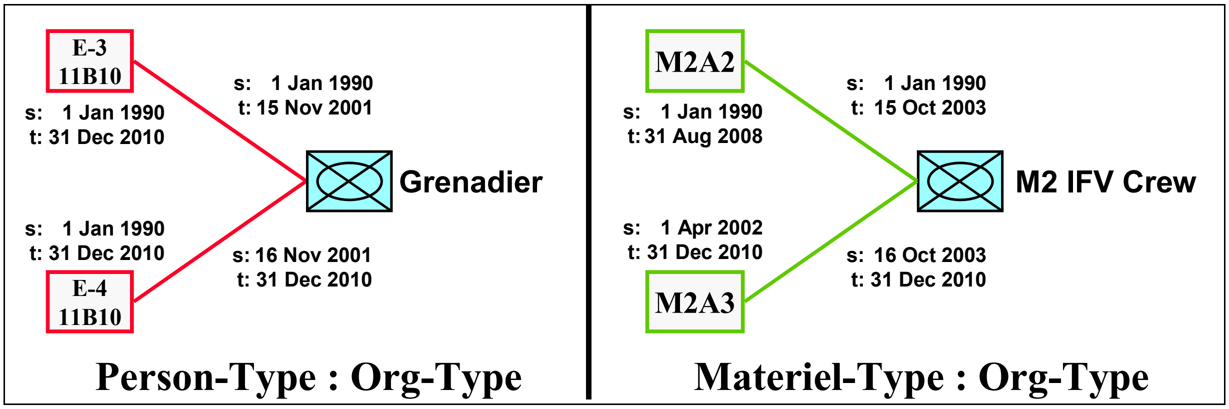

Figure 7: Authorization Information in the Org-Type Tree

community, the vertical links of the org-type tree are of minimal use and the purpose of org-type nodes is to store common authorization data.

Analogous to the vertical links between the nodes of the org-type tree, there are links, with multipliers, between the org-type nodes and personnel and materiel type entities as are illustrated in Figure 7. Although the details of these entities will not be discussed, the links to them follow the same pattern as those between org-types. (In Figure 7, the multipliers on the links are all equal to 1 and are not shown to simplify the diagrams.) Like every node and link in the force structure graph, those used to associate personnel and materiel data with an org-type node have an embedded time period that indicates a relative modification state. Therefore, as the force developer traverses the org-type tree, personnel and materiel information is filtered based upon the MDATE provided to the traversal process.

The left side of Figure 7 illustrates the link with personnel data, and on the right is a materiel example. Both examples include two mutually exclusive links. The personnel example denotes a transition point (beginning on 16 Nov 2001) at which time the pay grade requirement for a grenadier position changes from an E-3 to an E-4. Similarly, the materiel example denotes a transition point (beginning on 16 Oct 2003) at which time the authorized vehicle for the M2 infantry fighting vehicle crew is switched from an M2A2 to an M2A3 variant. Just as before, the MDATE chosen for the tree traversal process will determine which of these links are selected for traversal. Both of these examples show mutually exclusive cases: one cannot be both an E-3 and an E-4, nor is one authorized both an M2A2 and an M2A3. In both cases either one or the other is selected. However, this does not have to be the case. A typical example is when a new piece of equipment is added without replacing another.[7]

As previously explained, there are two ways in which one traverses to an org-type node: one is from another org-type node while moving up and down the org-type tree echelons, as in Figure 6; and the other is from an org node via the IS-A link, as in Figure 3. In both cases, an MDATE must be provided so that the correct personnel and equipment authorization information is obtained. In the first case, one already has an MDATE, as it is required to traverse the org-type tree. However, an MDATE must be specifically provided when accessing the org-type node via

[7] The challenge of synchronizing the many MDATES is an interesting one, but beyond the scope of this paper.

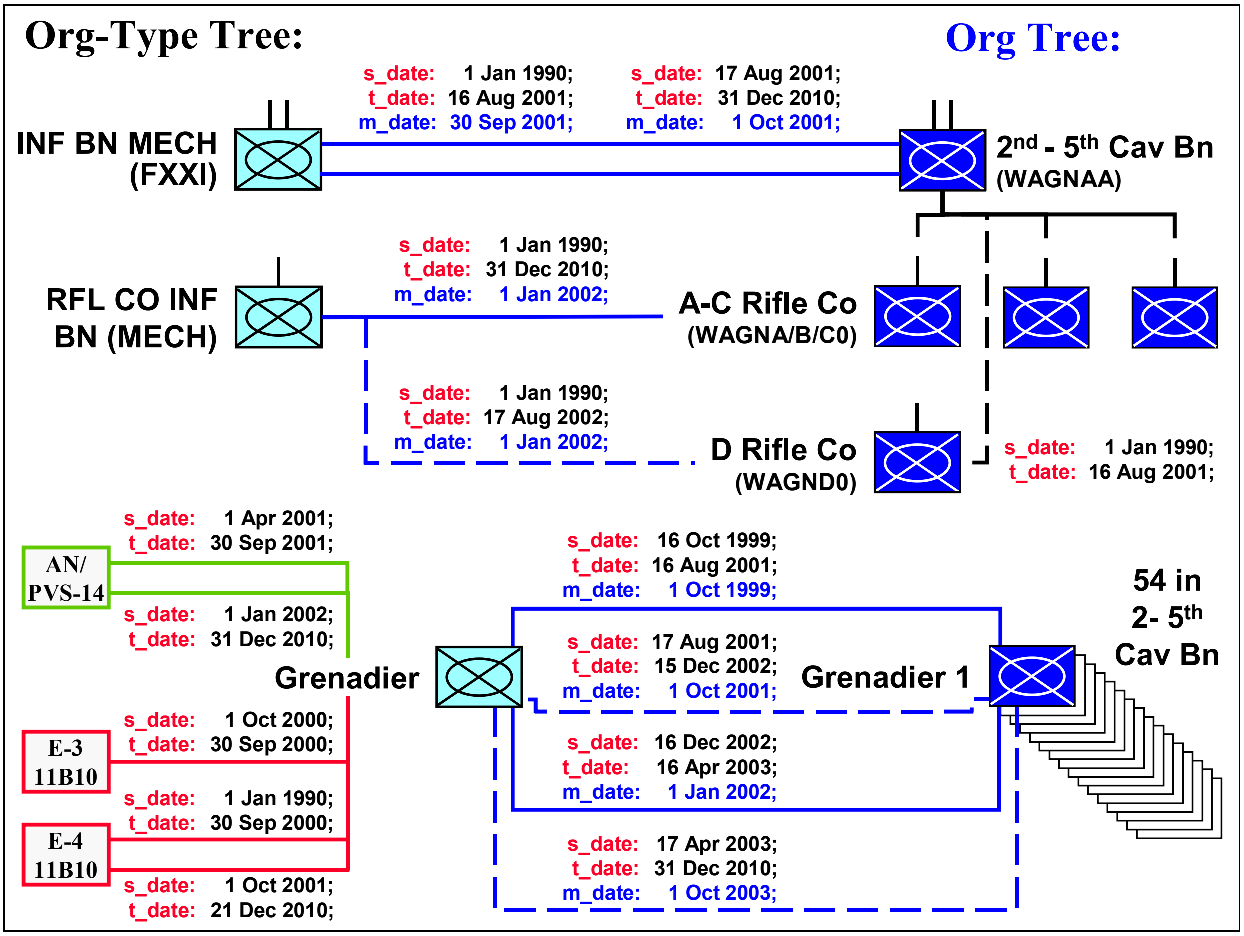

Figure 8: Combining Org and Org-Type Trees

an IS-A link. This requires that the IS-A links have an associated MDATE in addition to the time period indicators (s_date and t_date). This attribute is called the m_date.

The situation becomes interesting when the features of Figure 5 and Figure 6 are combined to produce Figure 8. This is a subtly complex diagram that contains several interesting results. Clearly, these are not complete org or org-type trees, but only selected slices to illustrate specific features. Also, although every node and link has an associated time period embedded in its structure, only a few are shown to simplify the diagram. The parent-child links of the org-type tree are also not shown for the same reason. Note that the m_date attribute is included only with the IS-A links that associate the nodes of the org tree that represent real units, with their corresponding node in the org-type tree that represents the generic case and contain the authorization data.

Starting at the bottom of the figure and working up, one can see that many grenadier billets (in the org tree) refer to the single grenadier position in the org-type tree.[8] When an EDATE is chosen to

[8] A distinction is made between a billet and a position. A billet is a real organization (an org) to which a person can be assigned. In this example, there are 54 grenadier billets in the battalion. A position is the description of a billet; it is an org-type and contains the authorization information about the billet. So technically, an MTOE Line Number denotes a position, and not a billet.

traverse an org tree, different IS-A links are selected based upon that date and the time periods specified on the links. The four links between billet Grenadier 1 and position Grenadier denote that the billet Grenadier1 has four distinct modification states during four different time periods:

```
Modification level (MDATE) 1 Oct 1998 FROM 16 Oct 1999 TO 16 Aug 2001 (EDATE);
Modification level (MDATE) 1 Oct 2001 FROM 17 Aug 2001 TO 15 Dec 2002 (EDATE);
Modification level (MDATE) 1 Jan 2002 FROM 16 Dec 2002 TO 16 Apr 2003 (EDATE); and
Modification level (MDATE) 1 OCT 2003 FROM 17 Apr 2003 TO 31 Dec 2010 (EDATE).
```

The continuous and mutually exclusive characteristics of these links are clear. The MDATE specified in each IS-A link is used by the traversal process to selects the appropriate personnel and equipment authorization links that are associated with the position (i.e., an org-type node), based upon the embedded time period of those links. This is the identical approach that is used to traverse any tree. The result is to relate the EDATE of a real unit with the MDATE of the generic unit. In this example, the result is that the soldier occupying the billet Grenadier1 is authorized to be of pay grade E-4 continuously through all its EDATES; and beginning on 16 Dec 2002, he is authorized an AN/PVS-14 night vision goggle. These criteria can be applied to all 54 grenadier billets or the links can be tailored individually for different results.

The four links created for Grenadier1 correspond to the right most four dots of unit WAGN in Figure 4. However, in this example, two of the four links have no affect on the result; these two links are denoted with dashed lines. Because of this, they can be removed and the same results will be produced. This is a common occurrence when mapping one-for-one between MTOEs and timed tree graphs. It is caused by an MTOE property known as the “parent unit” in the current documentation process. The ramifications of this will be discussed shortly, but for the moment it should be realized that unnecessary IS-A links can be produced and should be avoided.

Continuing up the diagram, one should recognize the pattern of the top set of nodes from Figure 5 and Figure 6. This is the case in which the battalion (Bn) loses a company (Co) as of 17 Aug 2001. This is specified by the t_date embedded in the link to (and the node) “D Co” from the battalion node and causes both the node and link to expire for dates after 16 Aug 2001. The same is true for the t_date embedded in the IS-A link to its associated org-type node, which is no longer useful after 17 Aug 2001. However, notice that the IS-A link from the other companies (e.g., Co A) is singular. This is because nothing has changed at the org-type node during the course of the specified time period. The “RFL CO INF BN (MECH)” org-type node has no associated personnel or materiel links, only links to children org-types, as is illustrated in the left tree of Figure 3. Even though there have been many changes elsewhere in the org-type tree of the battalion, the links to the children org-types (i.e., 1 X Co HQ and 3 X Rifle Platoon) have remained constant. Therefore, there is no need for multiple links.

There are two IS-A links between the top org and org-type nodes. However, this is actually optional. As previously explained, the purpose of the IS-A link is to provide common authorization information, maintained via the org-type nodes, to the org nodes. Like the company nodes below it, the “INF BN MECH (FXII)” org-type node has no personnel or materiel information directly associated with it; it has links only to its children org-type nodes. Two IS-A links may be used to reflect the change from four to three rifle companies, as is illustrated in Figure 6 for the generic battalion. However, it is highly unlikely that an end-user would require knowledge about the generic case when this information is already reflected in the org tree for the real unit. Therefore, to be technically complete, two IS-A links are shown. But this can be reduced to a single link, as was done at the company echelon directly below it, if the end users decide that it is not required to distinguish between the structural changes in the generic case (i.e., in the org-type tree). Note that the full org-type tree still remains, but the links between the org-type nodes are not selectable when accessed via the IS-A link from the org tree. If only a single IS-A link is used (with a single m_date attribute), then any results will be based on the selection criteria of the single m_date attribute.

5. The Retention Problem

The primary impetus for this study was the development of a scheme to maximize the stability of force structure data as it progresses through the evolution process. In other words, the objective was to eliminate, prevent, or minimize unnecessary modifications that might cause aggravation to systems already using the data. With the current MTOE system this is not the case, based in part on the concept of a parent unit.[9] In tree graph terms, a parent unit corresponds to the designation of a particular org-type node as an ad hoc to form an org-type sub-tree. Typically, the parent unit for an MTOE document is a battalion. By definition, a change to any of the hundreds of element of the battalion is considered a change to the MTOE and requires that a new MTOE document be produced. An extreme example of this is between the MTOE documents labeled 1502 and 903 in Figure 4. In this case, the two MTOEs are identical except for a six-word sentence in the narrative of the header. Yet, a new document was published based upon this change even though the org-type trees produced by these two documents are identical.

The primary notion behind the approach presented in this paper is that as time progresses, every entity is assumed to be unchanged unless stated otherwise. Because every node and link has an embedded time period, at any given time the force structure graph will have an earliest and latest time within its components that defines the current *epoch* of the graph. As time progresses, the epoch should slide towards the future as obsolete entities are pruned from the past and new ones are added to the future. The *time horizon* of the force structure graph is defined by the date furthest in the future that is embedded in any node or link. The force developer controls the time horizon as the evolution process progresses into the future.

The reader may have realized by now that the key feature that enables this time-based approach to function so elegantly is that the lifetime of an entity can be extended by simply increasing the value of the termination date. If this is done uniformly to all the nodes and links, then they all continue to be included in the graph. This is the default case. Every node and link is retained, with its node or link identifier, as the time horizon is increased and the epoch slides forward. Thus, stability is the norm. Only when changes are required are the start, termination, and modification dates adjusted. This is what maximizes entity retention. If only one item changes

---

[9] Parent Unit: from *AR 71-32, Force Development and Documentation - Consolidated Policies*, 3 Mar 97.
a. A parent unit is an MTOE numbered unit of battalion or equivalent level, or a numbered company, battery, troop, platoon, detachment or team, that is not an organic element of a battalion. The 5th and 6th positions of a UIC that end in "AA" identify an organization as a parent unit.
b. TDA units organized under a unique TDA number assigned by HQDA.
See: http://books.usapa.belvoir.army.mil/cgi-bin/bookmgr/BOOKS/R71_32/GLOSSARY

from "one MTOE to the next," then only those changes within the large tree structure are executed. Everything else remains the same, because the only modification to the vast majority of entities is the extension of their termination dates to form a new time horizon.

Figure 9 is a modified version of Figure 4. The basic difference is that the document boxes have been reordered and replaced with vertical lines.[10] The CCNUMs have been retained at the top of the diagram for reference and a modification timeline has been applied to the bottom with MDATES assigned, in relative order, to the documents.[11] In the bottom half of the diagram, three examples of authorization changes are provided. First is the change of the battalion's organizational structure from four to three rifle companies. Second is a personnel change for a grenadier billet between pay grade E-4 and E-3. Third is the addition of a new item of materiel, the AN/PVS-14 night vision goggle. Next to each of these examples is a line that represents the period of time that these features are in effect, based upon their presence in the MTOE documents.

The dots for the four units whose UIC are listed on the left side of the diagram retain the same

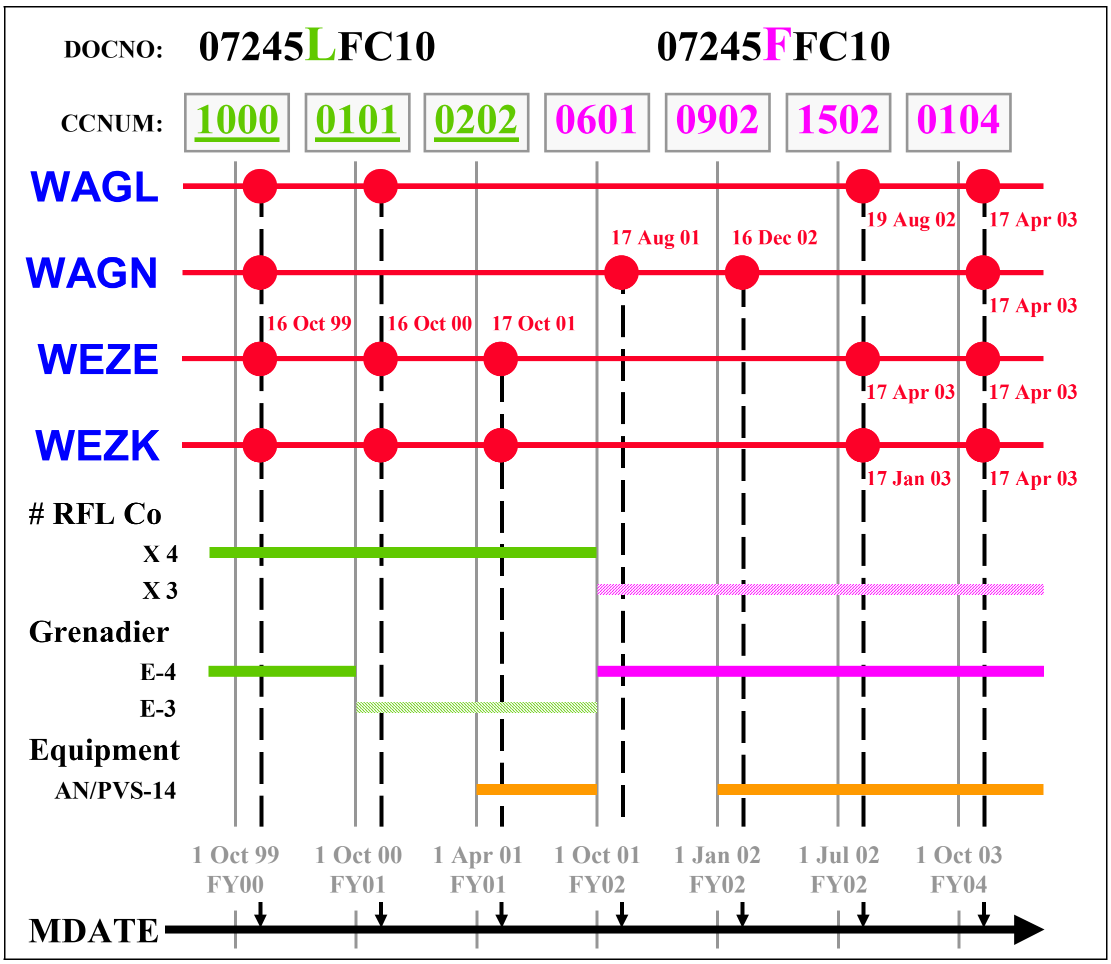

Figure 9: Time-Based MTOE System

---

[10] The 1099 document is not shown for simplicity. The 0903 document has been removed because it is identical to the 1502 document, except for a sentence in the header data. .

[11] Modification time is a monotonically increasing function using relative, not actual, time.

meaning as before, although in a slightly different context. Extending from the sets of dots are vertical dashed lines that indicate a point on the modification timeline (an MDATE) for which the unit has a synchronization point. Next to the dot is the EDATE to indicate the real date at which time that unit is expected to comply with the characteristics associated with the MDATE. If only a single EDATE is provided, then all the units aligned at that point share the same EDATE. If the vertical dashed line crosses a horizontal time period of the three examples, it means that at that MDATE those features are in effect. For example, the left most set of dots indicates that all four units must be modernized to the state indicated by MDATE 1 Oct 99 by 16 Oct 99, and at that MDATE, they have four rifle companies and the grenadiers are E-4s. As explained before, Figure 9 can be represented using a single org-type tree and four org trees (one for each real unit). With the modification timeline present, the CCNUMs can be removed; and all that is required is an identifier for the single org-type tree (to replace the DOCNOs), because the MDATE alone is sufficient to define the modification characteristics.

There is an issue as to just what modification order means; that is, why were the MTOE documents reordered to the sequence presented in Figure 9? A *precedence of modification* is proposed that places modifications in sequences based first on changes to the organization tree structures, then on changes to personnel requirements, and finally on changes to materiel. The documents in Figure 9 are ordered based on this precedence.

There is no mathematically correct modification order, but there are reasons that this sequence was selected. The organizational structure domain is placed first because of the tenet that force structure (represented by the org and org-type trees) forms the foundation by which all other battlefield objects are associated. Changes to the structure of the org-type tree can affect all the changes that occur elsewhere, especially the numbers of entities authorized. Therefore, changes to the structure of the tree are taken into account before any changes to the objects that are attached to the tree (e.g., personnel and materiel authorizations). As an example, there are significant structural changes caused by the transition from an L-series to an F-series organization (e.g., as is exemplified by the change from four rifle companies to three, changes in fire team structure, and the movement of combat support personnel out of the HHC). Because the highest precedence is given to organizational structure, one could argue that any F-series organization is more modern than any L-series. In Figure 9, this results in reversing the order of the sequence of CCNUMs 0202 and 0601, because 0202 was the last L-series document and 0601 was the first F-series document. The personnel domain was placed second because it closely follows changes to the organizational structure. When representing force structure down to the billet level, all personnel changes are associated with org-type nodes that are leaves of the tree (i.e., located at the bottom of the tree). Finally, materiel changes are considered.

The information in Figure 9 is ordered, from left to right, in modification sequence as defined by the proposed precedence of modification. Fictitious MDATEs have been applied to the timeline to provide synchronization points; these dates are merely relative, and any set of increasing dates can be used. In the examples, the time periods for organizational structural (four versus three companies) are clean and continuous because these features were considered first when ordering the document information. In the personnel domain, the grade of the grenadier has a short toggle between E-4 and E-3; but this only affects two of the four units. If the information were left in the original MTOE document order, there would be two toggles. For materiel, there is now a gap in the time period when there was none before. However, this is easily handled and a synchronization point with an MDATE has been created to allow the real units to specify this set of features.

If the three examples presented were the *only* features included in the trees (and in reality, they are only three of hundreds), then two of the synchronization points are unnecessary. In this diagram, only the first five MDATES define unique combinations of the example features. The last two MDATES (1 Jul 02 and 1 Oct 03) have the same combination as the fifth MDATE (1 Jan 02), and therefore, can be coalesced (i.e., all four dots would line up with a single MDATE).

This returns one to the topic of parent unit. Hundreds of characteristics (organizational, personnel, and equipment) can change from MTOE document to document, but in reality only a small percentage actually changed. For the end user, whose focus is the org tree of their own unit, and not the org-type tree, the concept of parent unit is unimportant. At the technical level, all the end user cares about is that for a given EDATE (used to traverse the org tree), an MDATE is provided in each IS-A link that selects the correct personnel and materiel information from the corresponding node in the org-type tree. For the org tree users perspective, the MDATE can be different in every IS-A link; this is of no concern.

However, for the sanity of the force developers, whose task requires them to traverse up and down the org-type tree, MDATES must be synchronized to some higher level. To date, this is done through consolidation at the "parent unit" (typically battalion) echelon. In terms of the time-based trees, this means that an MDATE represents the same modification point (i.e., state) for all the nodes and links below the node designated as the parent unit. Formally, one can say that the parent unit is the highest org-type node in a tree for which all the nodes and links below it are synchronized to the same MDATE. Pragmatically, this means that the concept of parent unit can remain even when the force structure process transitions from one that is document based to one that is time-based. The beauty is that this is simply another option for the force developer. As the situation dictates, and the org-type tree morphs into new forms, the echelon at which MDATES are synchronized can change to accommodate the needs of the development process.

6. Summary

This paper has presented a force structure documentation scheme that transforms the current force development process from one that is document-based to one that is time-based. Force structure can be represented as tree graphs where every node and link of the force structure tree is assigned a time interval that indicates when that node or link is viable. The primary notion behind this approach is the assumption that, unless explicitly stated otherwise, every entity remains unchanged as time progresses. The key feature that enables this time-based approach to function so elegantly is the uniform extension of the time intervals associated with every element of the force structure tree. Thus, stability is the norm.

By using a time-based approach, the maintenance of numerous documents that depict snapshots of a specific organizational entity at a given time disappears. Instead, there is an evolving force structure tree with many different forms and options that are differentiated merely through the application, or filtering, of time. One can imagine a knob that, through turning, allows one to advance or retard time so that one can observe the evolution of an organization's structure, equipment, and personnel. In essence, the force structure representation changes from a discrete form to one that is continuous; and although it may superficially appear to be more dynamic, it is actually more stable. This is because evolution is achieved through minimal modifications of a tree graph that is annotated with time intervals on each node and link.

This approach has been proposed for inclusion into the development of the Army Organization Server (AOS) by the force development community within the U.S. Army. The AOS will provide information to battle command systems and other end users about four primary domains: (1) real organizations, (2) generic organization templates, (3) materiel items, and (4) personnel qualifications. These last two entities describe the type of equipment and personnel authorized for the units. Ultimately, the AOS will possess a top (or root) organization node called "Department of the Army" and extend down to the individual, billet level. Between these extreme levels are all the intermediate, operational entities required to function on the battlefield. This includes teams, squads, elements, sections, platoons, or any other entity required to conduct the primary operations of a military unit, whether it be active duty, reserve, National Guard, or administrative. It is the AOS that the fielded units, and in particular, the digitized units, will use as the digitized source of authoritative force structure reference information.

However, inserting and maintaining the reference data into battle command systems, now named the initialization capability, is only the beginning. Locally, the digitized units will add a significant amount of data to their battle command system databases. This includes information about soldiers, the property-book, standard task force configurations, communications, plans, and a wide variety of other information. Because of its fundamental nature, almost all of this information is ultimately associated with the default force structure. Consequently, any changes to the force structure, let alone unnecessary ones, are not easily tolerated and must be kept to a minimum and carefully executed, so that the integrity of the database is strictly maintained. It is for this environment that this approach was developed. The objective is to provide a source of force structure data that, although it is constantly being updated, is consistently and rigorously maintained in a form that supports automated tools to expeditiously apply changes with a minimum of human intervention and aggravation, and with no or minimal adverse affects to the applications that use that data. It is expected that documenting force structure using time based tree graphs will prove to be a major facilitator towards accomplishing this goal.