

The Case for Non-Humanly Intelligible Record Identification

September 2010

The Opportunity

VA continues to amass vast quantities of data and information about their facilities from existing systems. New systems being implemented such as BIM and SAM/MAXIMO will increase the quantity and quality of data available. To leverage this wealth of data to intelligently manage the facility life cycles, all systems and users must be able to seamlessly share and integrate the available data.

Three key data elements are at the center of any effort to link or share data: **Locations**, **Assets**, and **Projects**. Having a reliable and efficient way to link these fields in the variety of systems and tables in use is absolutely essential. The basis for such reliable links rests in the key elements being rock solid in their unchanging uniqueness and accessibility.

Background

There was a time when data storage was expensive, processing was slow, and little flexibility was available in combining and displaying information for users. Modern databases have dramatically changed the landscape. They now have the capability to store data inexpensively, retrieve it efficiently, and present it flexibly.

Good database design is crucial to long term success and maintainability. On the one hand, user involvement is absolutely needed in defining operational requirements and business processes. On the other hand, users tend to think in terms of “flat files” ... like tabular reports and Excel spreadsheets, with a single record per line. The conflict arises when users with a flat file comfort level and legacy system experience define or *dictate* data handling and storage policies. There is a tendency to confuse operational requirements with database structure design, especially when it comes to record identifiers. Typically, users want information imbedded in the identifiers. For a number of reasons, that is generally not a good idea.

This is not to say that users don't need recognizable information and identifiers which reinforces their confidence in the system that is storing or presenting the data. However, the presentation of this recognizable information/identifier and its use for indexing, linking, and integrating systems are different issues. Referenced data where the identifier is subject to change, from either management decisions or process evolution, presents challenging data integrity issues. When persons or systems mishandle the identifier changes, data will become “orphaned” and the database itself and any derived data will quickly have credibility issues. Fully integrated central relational databases for *all* information could largely resolve the issue, but they are not a practical real world solution when integrating with other systems.

Unique Identifiers

The difference in the use of identifiers is crucial. The *user* only needs **uniqueness in context**. The *system* needs **absolute uniqueness** across all users and uses. In the context of a specific building at a known site, the user is perfectly comfortable with an ID of “AHU-1” for an air handling unit or “2103” for a room number. No further identification is needed in that context. Since that simple identifier does not work when multiple buildings or multiple sites are involved, the issue becomes how to achieve absolute uniqueness. The easy way has been to put the burden on the user: create and use a long, structured, *humanly intelligible* identifier. A typical example would combine **Station + Campus + Building + Wing + Floor + Room**, leading to something like 695-MIW-70-A-2-2103 for the location of the AHU which itself would be identified as 695-MIW-70-AHU-1 or more complex as 695-MIW-70-A-2-2103-AHU-1. A still more complex situation exists when the building has only one floor and no wings, so the user will need to provide a value, such as a “1” for the floor and a null filler for the floor at their respective locations for the ID string to work correctly. This type of identifier increases the possibility for human induced errors.

Another drawback of the humanly intelligible identification strings occurs when the sub-elements contain numeric information that is not padded with leading zeros to make all number strings the same length (i.e. if the largest building number is 1000, then building 70 would need to be padded to 0070.) Sorting using the long identifiers is not user friendly and when combined with unpadded numbers often presents annoying results ... i.e., building 70 (695-MIW-70) sorts after buildings 100 or 5000, but before building 8. To get the desired results on the AHU described above, the identifier would need to become 695-MIW-0070-A-02-2103-AHU-01.

Social Security Number

A more familiar example is a persons name and social security number (SSN). No individual is identified solely by their name because of the lack of uniqueness. Uniqueness can be achieved in the context of a home address (State, City, Address, Unit) plus the name. However, the location information is clearly not desirable as part of the permanent ID because of the issues associated with changing addresses. Notwithstanding that the SSN has some information embedded in it (based on the location of issue state and group), it is essentially a non-humanly intelligible identifier that is not subject to change, although the name, location, marriage status, etc. of the identified person may freely change.

Examples

There are some notable recent examples of the impact when the need for permanent uniqueness was disregarded:

Example 1: Federal Real Property Profile (FRPP)

The Department of State incorporated a number of humanly intelligible data elements in the Unique ID for their FRPP data submission. When a core identifier for two posts was changed or swapped, it created a situation of irreconcilable issues with duplicate identifiers and/or identifiers with imbedded information that was obviously inconsistent. Humans reviewing the data were constantly confused and data maintenance became burdensome as each new person encountering the

anomaly had to be assured that it was not erroneous. Actual errors were more difficult to detect. Any fix to the FRPP data for the two posts became a band-aid approach, since the “permanent” identifier still contained humanly intelligible data elements that are subject to change in the future.

Example 2: Project Numbers

The Department of State incorporated two and *sometimes* three humanly intelligible data elements in their facility project identification numbers. Within their XX-YY-1234 numbering schema, the XX categorized the project, the YY designated the post, and the 1234 was normally a sequence/serial number within a post’s projects but in some cases, individually designated numbers were used for specific project scopes (8032 for physical security upgrades, etc.) Problems arose when post identification codes were changed, as with their FRPP data. Simply changing the old project numbers was not a viable option since a variety of hard copy construction, procurement, and accounting documents as well as drawings containing the original project number could not practically be changed. And furthermore, situations arose where the same scope identifier was needed for a new project several years after a previous one had been completed. None of the workarounds were good. In the end, the fixes caused more confusion than the benefits that they were perceived to provide.

In each of these examples, the data managers sought to manually generate the unique identifier and did not recognize the possibility or probability of future changes or conflicts.

Identification Schemas

There are numerous possible identification schemas. Flexibility and usefulness is counterbalanced by complexity and implementation. Simple schemas can be implemented quickly but have limited flexibility and real usefulness. As system designers recognize these trends, there have been alternatives generated to humanly intelligible identifiers.

In the figures below, the blue fields are the key/cross reference data for the computer. Yellow highlights user recognizable data/information.

Figure 1 below portrays a single humanly intelligible identification schema for a location: Station=695, Campus=MIW, Building=70, Wing=A, Floor=2, Space=2103 . It is very simple in concept, but places the major burden on the user for mental translation and greatly limits the flexibility of integrating data across systems. It suffers from the sorting and filtering issues described above.

Humanly Intelligible ID	Classification	Name	Description
695-MIW-70-A-2-2103	Room	2103	Main Conference Room

Figure 1. Humanly Intelligible Single Identifier

Figure 2 is a slightly better method in that it formally parses the identifier into its constituent elements that overcomes the need for padding numbers and makes it easier for computers and users to sort and filter. However, linking or integrating it with other systems and data is challenging because its multi-part key must be matched to a correspondingly complex key. On the surface its strength would appear to be the consistency of the structure for all records. In fact, this is a major weakness. It forces rigid adherence to a structure that is not consistent with all existing data and is subject to change with requisite impacts on data maintenance and system modifications.

Station	Campus	Building	Wing	Floor	Space	Classification	Name	Description
695	MIW	70	A	2	2103	Room	2103	Main Conference Room

Figure 2. Humanly Intelligible Segmented Identifier

Figure 3, a segmented identifier with links to multiple source tables, shows a still better data solution, but is still awkward to implement with its multi-part key and dependency on numerous tables when linking or interfacing with other systems or data. Sorting, filtering, and reporting options are enhanced and simplified with the availability of more sophisticated queries. However, like the segmented identifier schema in Figure 2, its rigid structure is a serious weakness both in terms of accommodating existing data and responding to futures changes to the schema structure. (For display simplicity the identifiers shown are meant to be representations of GUIDs.)¹

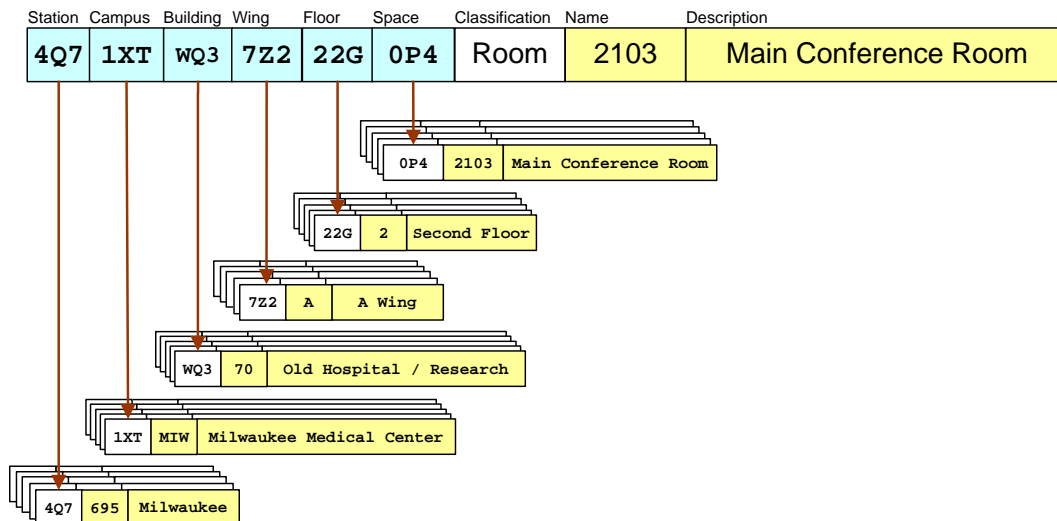


Figure 3. Non-Humanly Intelligible Segmented Identifier

¹ A **Globally Unique Identifier** or **GUID** is a special type of identifier used in software applications to provide a unique reference number. The value is represented as a 32 character hexadecimal character string, such as {21EC2020-3AEA-1069-A2DD-08002B30309D} and usually stored as a 128 bit integer [Wikipedia]

Figure 4 depicts a single identifier with no humanly intelligible identifiers or elements for the database. The system creates a truly unique identifier for elements such as location, assets, and project (i.e. a GUID), and have all links and references utilize it in a true parent-child relationship. Unlike previous depicted schemas, it is not limited by a predefined or rigid data structure. Having a single table with all internal references simplifies integration with other systems. The figure portrays a parent-child identifier that would have the flexibility to adjust to changes or differences in identification schemes without losing the ability to virtually reconcile those differences through the record classification. There would be no need for fillers or placeholders to confuse or burden the user. The human users would be able to read an associated identifier, placed within its context. To maintain absolute data integrity, the GUID would need to be used for linking to other records and integrating with other systems. The red lines and arrows represent the parent-child relationships, so the system can work up or down the family tree. The user would see: Room 2103-Main Conference Room, Second Floor, A Wing, Old Hospital/Research Bldg, Milwaukee Medical Center, Milwaukee. Or, based local preference: Milwaukee, Milwaukee Medical Center, Old Hospital/Research Bldg, A Wing, Second Floor Main Conference Room, Room 2103. In the local context, as much or as little of the identification information as desired could be displayed, all accessible in the family tree from the database through the single GUID for the room.

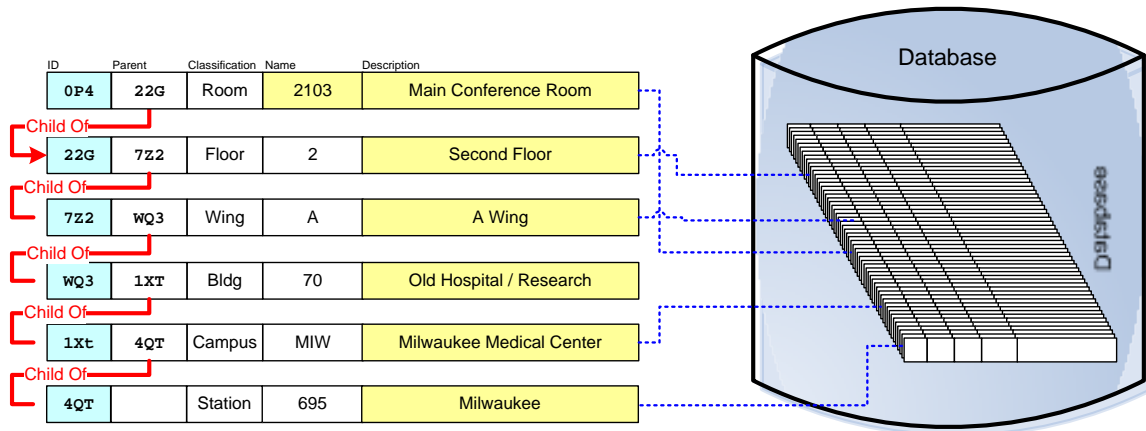


Figure 4. Non-Humanly Intelligible Parent-Child Relationship

Avoiding humanly intelligible identifiers may seem counterintuitive and be somewhat more difficult to initially implement, but the long term benefits in terms of data integration and adaptability to change far outweigh those concerns and costs.

VA MAXIMO Pilot

The VA is embarking on an important replacement of its legacy VistA system with MAXIMO under the FLITE/SAM program. MAXIMO versions 6 and prior required the Location ID to be unique within the system. This meant that to handle something as simple as AHU-1 in multiple buildings, the implementation had to artificially ensure uniqueness. Typically this was accomplished by prefixing or suffixing the facility number to the location identifier. Depending on the implementation, this makes it easier to sort and select by either facility or by location within facilities, but not both. When the

implementation includes multiple sites, then the artificial uniqueness must be expanded to include an additional element which further complicates flexibility in sorting and selection.

MAXIMO version 7, which is being used by the VA in the Milwaukee SAM Pilot, removes the *requirement* for the uniqueness in the Location ID field. It uses a system generated unique number (long integer) for all internal system references. When utilized, this has the advantage of simplifying the location and facilitates more flexibility in sorting and selection if the system utilizes the parent-child information available to it. Unfortunately, the team implementing the pilot is leading an implementation with the user maintained artificial uniqueness in the Location ID field, as required by prior versions.

Location classification should lead to flexible Location ID instead of full pre-defined structured strings. Legacy data will not conform to a rigid standardization schema. Some assert that the goal within the VA is for a level of standardization that has been lacking with the current VistA AMES/MERS implementation. Standardization can be achieved without forcing conformity for all elements. With the mountain of existing data in AMES/MERS, electronic drawings, and hard copy drawings, forced conformity in the Location ID would create an additional, largely manual, translation layer that adds complexity, especially at the facility worker level. The very real risk is that data maintenance and access will become cumbersome and overbearing, creating a state of reduced credibility that necessitates even more inefficient manual “data calls” for facility oriented information for VA management.

Asset identifiers (e.g., English naming) should similarly not be standardized for the sake of distant management and absolute uniformity across the VA. Typing and classification need to be standardized. Naming should make the best use of available documentation and past good practices. Then the MAXIMO system can present the data in a standardized manner for both VA program management and individual facility management.

A US Army Corps of Engineers (USACOE) has been addressing these identification issues and one of their Engineering Research and Development Center, Installation Technology Transfer Program reports noted that:²

“LOCATIONS and ASSETS require a unique identifier. These are used in the interface for identifying assets and locations and internally for creating the linkage between them. In conventional usage, MAXIMO has been populated ‘by hand’ and these identifiers have been hand-crafted to satisfy this dual use. Data generated from [*data transfer definitions like*] IFC (Industry Foundation Class) or COBIE (Construction Operations Building Information Exchange) is not subject to this manual review and **the tension between these two purposes** [*human*”

² USACOE, COBIE Data Import/Export Interoperability With the MAXIMO Computerized Maintenance Management System, by Nicholas Nisbet, November 2008 (ERDC/CERL CR-08-1 [DRAFT] Page 10). Bracketed explanatory information, IFC and COBIE definitions, and bolding added for emphasis.

identification and computer linking] and the balance of priorities was ultimately irresolvable without a specific project.

“...The recommendation remains that the best choice where data reuse is required is to use the unique 22 character Global Identifier provided by IFC. If the data transfer is a one-off transfer, then other identifiers may be used, such as the shorter locally unique identifier provided by XSLT transformation engines.”

Recommendation:

In planning for FLITE/SAM it is important to remember that the system will likely be used for decades. No one has a crystal ball into the need for future changes, however, just as there have been significant changes in the past decade there will likely be similar ones in the coming decades.

While the current implementation for MAXIMO will not result in any loss of data, it will set a precedent for the Beta and final roll-out. Forward thinking and planning for the future flexibility will allow simpler and less burdensome implementation with increased capabilities. At a minimum, the MAXIMO implementation should:

- Use parent-child relationships for locations and assets.
- Include a GUID identification field in addition to MAXIMO’S long integer, capable of being either generated or imported from other source data.
- Turn off the unique requirement for the visible ID.
- Cause MAXIMO to display the traditional identification string based on the parent-child relationships, not a manually maintained identifier.
- Allow and facilitate simple searches that can be segmented and refined rather than less intuitive long string searches.
- Use the GUID in all cross system linkages, references, and integrations.
- Readily present the GUID with all exposed and/or exported data.

Prepared by: Malcolm Junkin, Design + Construction Strategies (DCStrategies)
Under contract to: National Institute of Building Sciences (NIBS)
For: VA Office of Construction and Facilities Management (CFM)

Design + Construction Strategies, LLC
11 Dupont Circle, Suite 600
Washington, DC 20036
202.222.0610
www.dcstrategies.net