VCDE Workspace – Missing Values Reasons Report: Recommendations and Open Issues

August 12, 2005
Harold Solbrig, David Aronow, Rebecca Crowley, Jim Kadin, Kim Klinger

Introduction

This is a report of the "missing value reasons small group" looking into how caBIG should handle Missing Values (MV) and Missing Value Reasons (MVR) in data elements.

We first summarize our recommendations so far, and then summarize the open issues we have identified. We hope to get help from the broader VCDE workspace on how to resolve the open issues.

Summary of Recommendations

The small group tasked with looking at Missing Values and Missing Value Reasons has the following recommendations to the VCDE workspace:

1. **There should be a standard terminology** of Missing Value Reasons for use across the caBIG infrastructure.
2. **The use of a MVRs is very context specific** – a term might be an MVR in one use context but a "meaningful" value in another. Specificity of names of terms and clarity of value meaning is essential.
3. **The use of a term as a MVR should be unambiguously clear that it is a MVR**. We think this is best accomplished by having:
    a. A specific branch of the EVS terminology hierarchy for MVR concepts. The MVR small group will provide a seed set of terms and definitions to EVS, and the caDSR curators should extend the initial roster of concepts as needed.
    b. There should be a specific conceptual domain in the caDSR for MVR value domains. (so the use of a term as a MVR is clear from its value meaning?)
4. caBIG needs to support two ways of implementing MVRs:
    a. **MVRs stored in a data field as permissible values** along with other "meaningful" values. Most existing systems and data fields work this way.
    b. **Having a separate linked data field to hold MVRs.** For example, a data field for zip code might have a separate "zip code MVR" data field for specifying why the zip code is missing.

    Note that this second method is required or desirable for some data types like Booleans, numerics, date/times, and free text fields where it is impossible or undesirable to code MVRs in with the "meaningful" values.

    This second method appears to require the ability to link a "MVR Data Element" with a Data Element that holds the "meaningful" permissible values somehow.
5. The following aspects of the use of a data field (in a form, application, database, message, etc.) are all context dependent:
    a. Is the field required or optional (i.e., might it be missing)?
    b. If the field is missing, will an MVR be supplied?

c. Which specific MVRs are supported for this field?

For example, in one application the zip code must be filled in (is required). In another application, zip codes may be missing, but there are no MVRs. In a third application, zip codes may be missing, but missing zip codes require one of several specific MVRs.

To enable semantic interoperability, for each data field exchanged, aspects (a)-(c) must be specified somewhere.

How and where this information (this metadata) is best specified is unclear to us. See Open Issue A. below.

We are open to how the caDSR experts think this is best accomplished.

We are of the belief that MVRs are a characteristic of a Data Element, not a Data Element Concept, so there can be different Data Elements with identical Data Element Concepts, but whose Value Domains include different MVRs.

6. **MVRs should be implemented as a relatively simple, stable, high level set of MVRs**. We think there is a law a diminishing returns: Certainly, the more specific a MVR is, the more clear it is and the easier it is to differentiate it from "meaningful" values. However, we don't feel it is productive to specify a detailed set of MVRs unless there are compelling reasons (use cases) from the caBIG community .

Open Issues and Concerns

The small group has been unable to come to consensus or decisions in the following areas and would like input form George Komatsoulis, Frank Hartel and/or Avinash Shanbhag. The issues below suggest where these issues lie.

A. **Optionality: George Komatsoulis, Frank Hartel and Avinash Shanbhag.** It is unclear how best to specify the optionality and MVR distinctions (see recommendation #5 above). Does this information belong in the caDSR or in application level documentation or some combination thereof? Are all these distinctions aspects of a Data Element, the use of a Data Element in an application, or something else?

   Optionality is typically not specfied for Data Elements in the caDSR. MVRs are related to permissible values and therefore seem to be an aspect of Data Elements in the caDSR. However optionality of data fields is tightly related to MVRs: if a data field supports MVRs, then it is optional. So this seems contradictory.

   Then there is cardinality of a field (say in a message) which is a generalization of optionality. Where does that fit in?

B. **Too many CDEs: George Komatsoulis**. Will these recommendations lead to a plethora of CDEs in the caDSR and either become too complicated to implement or more trouble than they are worth?

   These recommendations appear to imply distinct CDEs be created for each distinct combination of MVRs needed - plus different CDEs for the two ways to

represent MVRs listed above, plus (potentially) additional CDEs for required data fields.

Is it reasonable to have an application use a CDE that supports MVRs but then not use the MVRs when a data field is missing? This might allow for fewer CDEs needing to be defined but seems to miss the point of the metadata.

C. **caGrid implementation: Avinash Shanbhag and George Komatsoulis** How do applications on the caGRID 'know' about optionality of fields and MVRs? (please refer to recommendation #5, above). The VCDE WS small group wishes to implement the the functionality stated in Recommendation 5 See above) but would like input and descriptions form the Architecture WS as how these would be implemented on the caGrid. It seems that for a field in a message on the grid, the following pieces of information need to be easily available:

   a. Whether the field is missing or not.
   b. Whether or not it supports MVRs (and which MVRs)
   c. If the field is missing, what is its MVR (if any). At the very least, we need to differentiate MVRs from other "meaningful" values.
   d. Whether it supports MVRs as part of its permissible values or as a separate field.

      Some small group members think that at the messaging level, only the more general "separate, linked MVR data field" should be supported to simplify the message structure. However there is concern about the added work of translating data fields that code MVRs as permissible values when creating or handling a message.

   Perhaps some of this information needs to be available via the message schema rather than the message itself. In general, it would be nice to know which fields are required and which are optional, otherwise any application processing the message must assume that each field is optional which would seem to complicate any application handling the message.

   We feel that these are primarily architectural issues, but they are core aspects of how optionality and MVRs are handled. Is it the purview of the small group to make recommendations along these lines to the architecture workspace?

D. **Heirarchy or flat structure for MVR terms: George Komatsoulis and Frank Hartel**. The small group does not have agreement on whether to structure the MVR concepts in EVS hierarchically as done in HL7. We submit this issue to arbitration to decide for us.

   A hierarchically organized terminology might help to clarify the MVR meanings. It is unclear at this point what other uses might be made of relationships between the MVR concepts. There is concern that maintaining such a hierarchy will make adding future MVR concepts more difficult as they will have to fit into the existing hierarchy or will lead to a restructuring of the hierarchy.

E. **MVR Best practices: George Komatsoulis**. We do not have agreement on what constitutes best practice guidelines or functional requirements for MVRs. We submit this issue to arbitration to decide for us.

   Some small group members feel that the "separate, linked MVR data field" is the practice that should be encouraged, although both methods need to be supported,

because it is more general (i.e., it applies to fields that are not enumerated value fields) and because it separates the MVRs from the “meaningful” values better.

Other members feel that this is more complicated and creates a heavier burden on systems and system implementers.

F. **MVR contained in Terminology or in Metadata. George Komatsoulis and Frank Hartel.** The introduction of missing value reasons into the EVS may be of concern because it would appear to cross the boundry between concepts and data, which typically separates the terminology and data standards repository. However, in this case an important ancillary consideration is that the decision to store missing value reasons within the caDSR (unconnected to a value meaning code) would considerably complicate the work-process of annotating UML models. The small group anticipates that creation of value domains will eventually be supported by the semantic connector and UML loader. If this process cannot be used for attributes which are missing value reasons, then developers will need to separately create these within the caDSR. Because of the importance of missing value information - parallel secondary work processes should be avoided if at all possible. The small group recognizes the need to carefully weigh the needs and requirements of terminology development against the practical aspects of tooling and work processes in this case, and seeks additional guidance on this issue.

   Therefore, we are asking EVS and/or any other significant parties to recommend in the future how MVRs would be added as pure terminology into EVS and where they might be placed. To date, missing value type terminology is located in the path of “"NCI Administrative Concept" ->"Business Rules" -> "Support Grant Application". As additional terminology requests are made in order to meet the need of additional Objects, Properties, and Permissible Values within the caDSR, what will be the process for their creation within EVS?