

Adaptation - "the pluses and minuses"?

Clearly one of the most critical issues contributing to user problems with implementation of the SDSFIE lies in the ability of the user to "tailor" the content to meet local information needs. Compliance was continually an issue as new Releases of the Standard often just moved the SDSFIE further from the desires of the individual most impacted. The SDSFIE over the years has offered some "tailoring", principally in the form of "Filters", which permitted the user to select which Features would be included in the final dataset. So, instead of getting all 1200 plus features, some custom feature set could be predefined. Although this helped, it did little to address all of the issues.

Keeping in mind that the SDSFIE is a Feature - Attribute - Value (FAV) Standard, there was never any attempt to compromise on attributes and values in Filters. All the `fr_stuff`, `to_stuff`, and `coord_stuff` was still in the tables, even though the use of these fields was almost never exploited. Domain lists were continually expanded, while certain users wanted to use only one or two values. At the same time, local user needs could only be considered compliant if the user could figure out a way to get his data requirements into the SDSFIE. The more clever the user, the closer the structure would look to their own. This entire situation is an invitation to a Standard that partially satisfies many, and completely satisfies few, something like what exists in the SDSFIE. Some form of flexibility in the SDSFIE is essential.

The trick is how to make the SDSFIE flexible, while not giving up on the premise of data sharing, universal understanding, discipline, and rigor. The secret lies in what the SDSFIE team is calling **Adaptation**. Adaptation is simply constructing a standard, and the tools, that permits the "adds and deletes" in a structured way so as not to compromise the objectives. While XML is a Standard, it has been successfully adapted to lots of creative uses, without compromising the overall objective of what XML was built to do. Something similar can be done with the SDSFIE.

In an attempt to clarify terminology, the pluses and minuses to the SDSFIE are being called **Extensions** (the add-ons necessary to meet the local need) and **Profiles** (the elements of the Standard; Features, Attributes, and Values not required in the local dataset). While this requires somewhat greater understanding of the local information need, it will result in a dataset that is much closer to the user's ideal schema. And all of this can be accomplished at a variety of levels.

Let's say, for example, that GeoBase wants to rigorously define 50 SDSFIE Features as being required of the Air Force. They define that Profile (the reduction of Features, Attributes, and Values) they require. If they wish to add any elements for the Air Force, they will do that in an organized way through a Web Site that enforces the rules and records the results. This USAF SDSFIE (with Profile and Extension) becomes the starting point for Major Commands, who can repeat the process, all the way down to users. At the local level, the specified Profiles and Extensions result in a recorded, documented dataset that has been thoroughly tailored.

What this means practically, is that some of what is in the SDSFIE today (in Release 2.600) might go away. It might be included in a USACE Extension, or even replaced by a link to a Business System that interfaces with the SDSFIE. The good news in all of this is that the Standard will be simplified, and it will more easily meet local user requirements without the need to "cheat". Additions and Modifications to the Standard will be derived from recorded extensions, and web sites and services will be created to facilitate the documentation of this "customization".

It's a large undertaking, but worth the effort. It won't be available tomorrow, but through adaptation, the overall usefulness and acceptance of the standard will increase. It's a large step in the right direction.