Chapter 2. Taking to Heart the Essential... > XBRL Is Powerful but Not a Complete Solution or System

2.4. XBRL Is Powerful but Not a Complete Solution or System

Although XBRL provides a substantial foundation of functionality to build on, XBRL itself isn't a complete running system any more than XML is a complete solution. XBRL offers a basis for delivering a diverse and rich set of functionality to achieve a system's goals, but it isn't a complete business solution in and of itself. If you had to describe what XBRL provides in a word, that word would be **leverage**. You can build a broad range of specific systems by using XBRL in a fraction of the time that it would take if you were to start from scratch.

On the other hand, given that XBRL is general in nature, it meets the needs of a broad set of information exchange use cases, which means two things:

- **XBRL isn't perfect for any specific application of XBRL** because XBRL isn't trying to be perfect for any one thing.
- **XBRL is perhaps more than you need for your specific application of XBRL** within your system.

The last point isn't a problem: Just use the pieces you do need and ignore the rest. Keep these important points in the back of your mind:

- **XBRL isn't a complete solution or system**. XBRL is a standard that provides a tremendous amount of leverage, but it is not in and of itself a complete solution or system. XBRL is used within solutions or systems.
- **XBRL can come in many dialects**. XBRL is a general-purpose language. Business systems have many different goals, which you can achieve in many different ways. XBRL users do so in different ways, basically creating what amounts to different dialects of XBRL that don't necessarily all interoperate well with each other unless they're made to do so.
- **Software tools may not be interoperable**. A key to effective information exchange is software interoperability, which XBRL International works hard to enable. However, if two different software vendors support XBRL within their applications, those applications may not necessarily be 100 percent interoperable. For example, if one software tool supports a module of XBRL (such as XBRL Dimensions) that another software tool doesn't support, you may have interoperability problems. Software tool interoperability must be created; it's hard work but achievable.
- **Business systems may not be interoperable**. If two business systems implement XBRL within the systems, you have no guarantee that the two systems will interoperate, unless they're specifically **designed** to be 100 percent interoperable. For example, one system may support a module of XBRL (such as XBRL Dimensions) that the other system doesn't support. Or, if one system implements proprietary features (such as the addition of a non-XBRL attribute to a concept, deviating slightly from the standard) and the other one doesn't, the two systems may not interoperate correctly. As with software-tool interoperability, business-system interoperability is achievable, but it does take work.
- **XBRL taxonomies may not necessarily be interoperable**. If two groups create two XBRL taxonomies that use different architectures, those XBRL taxonomies may not be interoperable. Creating different taxonomy architectures isn't a characteristic of XBRL; it's a characteristic of how much (or how little) the two different systems work to make their systems interoperable. Systems sometimes process completely different sets of information, and interoperability isn't necessary. Or, systems can process the same type of information but implement XBRL taxonomy characteristics in different ways, so the systems aren't interoperable. For example, the US GAAP XBRL taxonomy, the IFRS XBRL taxonomy, and the EDINET XBRL taxonomy are all used for financial reporting, but these XBRL taxonomies are significantly different. (You can see a comparison of these three XBRL taxonomies at [www.xbrl.org/TCF-PWD-2009-03-31.html](http://www.xbrl.org/TCF-PWD-2009-03-31.html).)
Extensibility makes interoperability even harder. If you create an XBRL taxonomy to meet the need of a static reporting problem (for example, an unmodifiable form), interoperability is easier to achieve than if you use XBRL's extensibility. If you're not using extensibility features, you don't have to deal with all the ways users may extend the XBRL taxonomy. If you use XBRL's extensibility, you have to ensure that those extending your XBRL taxonomy do so in consistent ways and in only the ways you intend them to extend the XBRL taxonomy. The XBRL specification doesn't cover how to extend an XBRL taxonomy and not break the information model, so you must build your system to handle these sorts of issues.

Application profiles reduce risk and make interoperability easier. To make implementing an XBRL system easier, you can create an application profile, which articulates an XBRL architecture (meaning it's 100-percent compliant with some specific constrained subset of XBRL), and all the automated validation rules you need to effectively constrain XBRL taxonomies and XBRL instances within a system against that XBRL architecture. You can even use an existing application profile that works correctly.

A downside of using someone else's application profile is that you must live within its constraints. If you can, implementing XBRL is substantially easier because you have less work to do, and the risk of the system not working as expected is reduced. (Chapter 12 discusses application profiles.)

Extending XBRL with XML extensibility is a possibility. In many areas, XBRL does allow XML-type extensibility, and you can find many good reasons to use it. For example, you can add attributes to an XBRL concept, if you need to. If you're working in a closed system (where you have complete control over all of that system's aspects), XML-type extensibility can be a good strategy. However, this approach can have serious drawbacks. For example, if you add an attribute to an XBRL concept, an XBRL processor may be able to read that attribute, but the application won't know what to do with it. XBRL created extensibility the way it did to avoid just this situation. You need to be conscious of these types of things when you consider how to use XBRL. Be aware of both positive and negative impacts.