

GJXDM Technical Overview

(An Astonishingly Brief) History

The GJXDM is the result of a long history of data dictionary efforts. These efforts came and went. Eventually, a major data dictionary reconciliation effort occurred. Combining results from many of these past efforts, it created the Reconciled Data Dictionary (RDD). This data dictionary formed the basis of the GJXDM.

Governance (Also Astonishingly Brief)

The Global XML Structure Task Force (XSTF) sets the direction of the GJXDM. The XSTF meets regularly, both in person and by phone, to guide, review, and resolve issues pertaining to the model. The XSTF consists of a wide variety of individuals from the justice community. Representatives on the XSTF come from the courts, state and local law enforcement, probation and parole, Federal government, the vendor community, academia, and elsewhere. The Global Advisory Committee directs the XSTF. Global is shorthand for the Global Justice Information Sharing Initiative. The Department of Justice, Office of Justice Programs, funds GJXDM development.

Tech Concepts

The genesis of the model lies in the reconciliation effort that gathered together data dictionaries from a multitude of sources and, per the name, consolidated them into one master data dictionary.

This result, while impressive, was still just a flat data dictionary, simply a list of elements and descriptions. The key to the model is the application of structure to that dictionary. This addition of structure turned what was a GJXDD (Global Justice XML Data Dictionary) into the current Global Justice XML Data Model (GJXDM).

Adding this structure revolved around creating objects and pulling out commonalities between objects which were then encapsulated into higher level objects. Confusing? Here's a quick example. Suppose you have an arrest subject and a police officer. These are two different objects and there are things about them that differ. An officer has a badge number; something most subjects lack. In turn, the subject has things an officer lacks, such as the location where the subject is detained. On the other hand, both of these objects share many types of information. Both have a name, for example.

In a normal data dictionary, there would be separate elements for the name of each of these objects. There would be something called "ArrestingOfficerName" in addition to an

"ArrestSubjectName". In the data model, common information such as this is brought up to a higher level and encapsulated into an object called "Person". This Person object contains all the types of information common to all people, regardless of their exact role in the justice system. More specific types of people, such as an arresting officer, are derived from this Person object and inherit all these person-ish types of information. But they also get additional types of information that are more specific to the exact sort of person they are. So, there would be an element called "ArrestingOfficer" that is derived from Person and inherits things like a name, in addition to having officer-specific information such as a badge number. (In the model itself, the term is "ArrestOfficial" instead of "ArrestingOfficer". We'll use the model's terminology from this point on.)

This process of finding commonalities and encapsulating them into higher-level objects was done for each and every element resulting from the RDD plus many more. Starting with 16,000 elements, the end result was the 2800-some elements and types of the GJXDM. (By "end result" don't be fooled into thinking that the model is done. It's not.)

Elements and Types

This object-oriented structure is achieved through the use of types. The model contains over 2300 elements, which define the semantics of the information in the model. The model also contains over 500 data types. These types create the structure and define how elements relate to each other.

One way to view this structure is as a set of two hierarchies. Each element in the model has a defined spot within both of these hierarchies. One hierarchy details "has-a" relationships. This hierarchy defines what sorts of information an object has. An arrest official "has-a" badge number. It's often helpful to think of this as a container relationship. An arrest official object "contains" a badge number. Reflexively, a badge number is contained within an arrest official. Some objects, like the high-level Person object, are not contained within any other object. They exist on their own at a top level. Other such examples are generic objects like "Activity" and "Organization".

The other hierarchy details "is-a" relationships. This hierarchy defines what sort of thing an object is. An arrest official "is-a" person. These can also be thought of as being derived type relationships. An arrest official is "derived" from the more general Person object.

The model handles these derivations via the types. Technically, an ArrestOfficial object isn't derived from the Person object. Rather, there is a type, called PersonType, from which an ArrestOfficialType would be derived. Then there is a Person element of type PersonType and an ArrestOfficial of type ArrestOfficialType.

Unlike the "has-a" hierarchy, all derived types ultimately derive from a single top-level type called "SuperType."

The combination of these two hierarchies defines the structure of the model and each element's place within it.

References and Relationships

Many relationships between objects are represented by the "has-a" and "is-a" relationships inherent in the GJXDM structure. But the array of possible relationships between objects is larger than any model could directly represent. Inter-personal relationships alone would be unmanageable.

To represent these sorts of relationships, the model provides for reference elements to define relationships between objects in a more flexible manner. There are many ways to use these references and the community is still working out best practices.

Code Tables

Code tables are an inevitable part of any justice system. Many different justice agencies create and maintain many different national-level code tables. The model itself does not contain these code tables. Instead, the tables are maintained in an XML format as external schemas. The model then imports these tables. This allows the tables themselves to be maintained by the agencies that normally would maintain them.

Schemas

The data model is a conceptual set of elements, types, and relationships. To actually use the model, we instantiate it using XML technologies. In particular, XML Schema is used to validate the XML-based exchanges. XML Schema is a means of defining how an XML document must be formatted. It controls which elements can be used, how many times they can appear, and how they can be organized within the XML document.

The model as a whole is represented as one large schema. As the model is rather large, the resulting schema is also rather large. This can be unwieldy. The data model introduces the idea of a subset schema. A subset schema is a pared-down version of the model. Following certain rules, you can omit sections of the model that are not needed in your particular application. While subset schemas can be created by hand, the Georgia Tech Research Institute (GTRI) provides an online tool to generate subset schemas for you. Called the Schema Subset Generation Tool (SSGT), this tool allows you to choose the elements and types appropriate for your exchange. The SSGT then creates the subset schema, following the appropriate rules.

The model also introduces the concept of a document schema that acts a wrapper around the rest of the elements. (Recall that, in terms of container relationships, some elements

exist at a top level instead of being contained within a higher element. The document schema provides a top-level element under which these other elements can be contained.)

Tools

The GJXDM is a very large model. Understanding it can be a difficult task. Fortunately, there are some tools available to help you navigate and explore the model. (Unfortunately, there are a mere handful of tools. But any tool is better than none.)

The Georgia Tech Research Institute (GTRI) provides two tools to help you explore the model. One is simply a spreadsheet that lists out element and type information in a variety of ways. The spreadsheet contains internal hyperlinks that allow you to easily jump from element to element and type to type. The spreadsheet also organizes elements into broad categories and provides an overall view of the type-derivation hierarchy.

A second tool from GTRI is the Model Viewer. Once a standalone product, the Model Viewer is now incorporated within GTRI's Schema Subset Generation Tool. The Model Viewer functionality provides detailed information about individual elements and types. The Model Viewer functionality also allows you to search through names and descriptions to find the element or type that you need.

Both these GTRI-provided tools are freely available at no cost to the justice community.

While GTRI provided the earliest tools for exploring the model, other tools exist. The National Center for State Courts developed an exploration tool for the data model called "Wayfarer". Wayfarer provides detailed information about elements and types and the relationships between them. It also provides detailed searching capabilities. While similar to the Model Viewer in the kind of information presented, Wayfarer uses a database representation of the model to derive all sorts of additional information, including:

- ✓ All elements that a specified element can contain, including all inherited elements.
- ✓ The reverse, every element that can inherit the specified element.
- ✓ All elements of a particular type, or of any type derived from that type. In other words, when you look at PersonType, you can see every element that is personish in nature.
- ✓ Searching of elements, types, and code tables.

Wayfarer is available both as an online resource and as a localized set of files suitable for installing on a laptop. Wayfarer is freely available at no cost to the justice community.

For those with an interest in learning more about Wayfarer, there will be a SIG session devoted to the tool. The session will feature an in-depth demonstration, plus time for questions and discussion.