Common Activities in Data Interchange Format (DIF) Development

Peggy D. Gravitz
COLSA Corporation
Huntsville, AL  35806

Jack Sheehan
Defense Modeling and Simulation Office
Alexandria, Virginia

Thom McLean
Georgia Institute of Technology
Atlanta, GA  30332

Keywords:
Data Interchange Format, metadata, standards, process

ABSTRACT: Data Interchange Formats (DIFs) are key components of a number of toolset architectures that are being developed to support M&S activities. Prominent examples of DIFs are used for sharing of High Level Architecture (HLA) object models, Conceptual Models of the Mission Space, and environmental data. The purpose of this paper is to document some of the common processes and products that are typically included in the development of DIFs. Although the individual DIF specifications are developed to address different applications within the M&S development and execution process, the development steps for many DIFs are similar. Therefore, we present a process for the sequence and description of these steps. The intent here is to be descriptive not prescriptive, illustrative not exhaustive.

1. Introduction

Data Interchange Formats (DIFs) are key components of a number of toolset architectures that are being developed to support M&S activities. The purpose of this paper is to document some of the common activities, processes, and products that are typically included in the development of DIFs. Although the individual DIF specifications are developed to address specific applications within the M&S development and execution process, the development steps for many DIFs are similar. Therefore, we present a process for the sequence and description of these steps. The intent here is to be descriptive not prescriptive, illustrative not exhaustive. However, we hope that potential DIF developers can benefit from the ideas presented here to assist in their DIF development activities.

1.1 What are DIFs?

For the purpose of this paper, we limit the scope to include DIFs we create in support of M&S activities. We also do not address runtime simulation interfaces, such as an RTI, but limit ourselves to data exchange that occurs outside of the simulation execution. A DIF defines how data will be exchanged between applications. We define a DIF as a formal specification of the structure and format of data to be interchanged between data producers and consumers. A DIF is specified sufficiently so that tools and applications can be developed which may support the specific subject area to exchange information which conforms to the DIF both for data input and output. A DIF has the following characteristics:

• Independent of the communications method used to exchange information,
• Encourages the development of related tools and applications that can be easily integrated with existing tools and data, and
• Becomes the blueprint for mapping the data elements from one database to another.

1.2 Why are DIFs important?

Modern simulation architectures are increasingly heterogeneous systems. The definition of the synthetic environment, the design and parameterization of the federate simulations, and the analysis of the simulation
results environment, may be conducted by several independent applications. These applications frequently need to access a common set of data for the simulation. Examples of this type of data may be configuration data, initial conditions, or scripts for federate behavior. Moreover, several types of federations may rely on existing data sets, or data set standards. Military simulations, for example, rely on large sets of existing intelligence data. Engineering simulations may rely on large sets of scientific performance data. Much of the data that simulations require may already exist. However, the data may not exist in the format the application requires or with an access method that is available to a particular user community.

The “real-world” information required by new information systems may already exist in a database created with a different purpose in mind. However, the substantial investment in these legacy systems must be retained without significant revision [1]. The underlying cost/capability trade-off does not support a “one size fits all” solution. Therefore, data interchange formats are being created for “Large Grained Re-use”:

When there is no method of exchanging the data between systems, users develop interfaces between each pair of systems or formats. However, if a standard format is used, simple exchange mechanisms can be created that facilitate data interchange. The development of an interchange standard with defined mappings to existing systems provides the best means for efficiently exchanging information between databases [2]. Figure 1 illustrates the reduced number of interchange methods required to share data. Each database can map through a DIF, rather than to each of the other databases with which information must be shared.

1.3 Examples of DIFs

The following prominent examples of DIFs are supported by the Defense Modeling and Simulation Office:

- The HLA Object Model Template (OMT) DIF is used for exchanging complete object models among development tools, libraries and data dictionaries.
- The Synthetic Environment Data Representation Interchange Specification (SEDRIS) is used for natural environment data provided under the direction of the M&S Executive Agents. Having a common representation of the physical environment is critical to M&S systems. SEDRIS provides a uniform and effective method to describe, reuse, and interchange environmental data.
- The Unit Order of Battle DIF (UOB DIF) is used for exchanging military unit Table of Organization and Equipment (TO&E) data. The UOB DIF and the accompanying data access tool automates the M&S startup data load process for developers and users by providing Unit Force Structures and Task Organizations information.
- The Conceptual Models of the Mission Space (CMMS) DIF is used for exchanging functional descriptions of military operations and tasks.
- The Integrated Data Dictionary DIF enables information interoperability among HLA FOMs and SOMs and CMMS. This is accomplished by capturing and maintaining data dictionary requirements of the HLA Object Model, CMMS, and the Knowledge Acquisition Tool (KAT).

1.4 DIF Definition Styles

A DIF may be described using a formal technique or an informal agreement. Informal techniques can be as simple as a data model and a table to describe a common delimited format. There are several formal possibilities such as Backus Naur Form (BNF), Tagged Grammar Format (TGF), Extensible Markup Language (XML), or Interface Description Language (IDL).

Typically, DIFs have been described using an extended notation of BNF. While a BNF definition allows very complete control over the format of the DIF, such fine
control may complicate parsing and using the DIF formatted data. Recognizing this, the DMSO has developed a BNF extension, Tagged Grammar Format (TGF) which allows automatic parser and interface generation.

Backus-Naur Form is attributed to John Backus and Peter Naur [3]. BNF was invented to describe the syntax of Algol 60 in an unambiguous manner. Since then it has become widely accepted and used by most authors of books on new programming languages to specify the syntax rules of the language. BNF allows the specification of a format down to the individual character level. While BNF does allow control, it does not provide a guarantee of correctness or parseability.

Use of a BNF specification does not naturally lead to a simple specification. Several existing DMSO DIFs have been specified using BNF. In developing these DIFs, a common set of conventions is being used to facilitate development.

An extension of BNF, which allows automated parser and interface generation, has been sponsored by DMSO [4]. By associating tags with certain non-terminals and lexicons, a unique canonical interface can be generated from a TGF grammar. A TGF DIF is a complete language specification for a data file. Accompanying software can alleviate much of the development requirement for reading and writing DIFs.

Extensible Markup Language is a subset of Standard Generalized Markup Language (SGML) promulgated by the World Wide Web Consortium (W3C) [5]. XML describes a class of data objects called XML documents. It makes it possible for developers to create interoperable dialects of markup languages. To define a XML dialect, a Document Type Definition (DTD) is created. A DTD suitable for defining data interchange formats. XML allows for in-line, or externally referenced DTDs.

Extensible Markup Language is an emerging W3C standard that may simplify specification and use of many types of DIFs. Finally, complicated systems such as SEDRIS may use Interface Description Language to specify the interface to SEDRIS conformant data.

As the XML standards evolve and stabilize, there will doubtless be a number of commercially available tools to perform parsing and interface procedures. XML DIF files will resemble HTML files in their form. A DTD expresses structure only. The format of the file is defined by XML. In this way, an XML based DIF will lack some of the representation power of a BNF based DIF. However, in many cases, XML will provide sufficient expressive power for data interchange.

Several criteria may influence the selection of a DIF definition language. They include: 1) the ability for parser generation, 2) interface generation, 3) existing tools, and 4) human readability. Data analysts and developers will need to read and understand the data format and structure being represented during implementation.

2. Common DIF Development Activities

Some of the common activities, process and products that are typically included in the development of DIFs are provided in the following sections organized into seven sequential steps. Figure 2 illustrates seven steps that have been used successfully by DMSO to develop DIFs. However, in any methodology that you use to develop the DIF, the most important consideration is an organized approach that can be validated against each participant’s requirements once completed, then tailored for the specific user(s).

Step 1: Identify The Need for a DIF and a Strategy for Managing the DIF

First, the need to promote the sharing of common information is identified by a group of users. The particular subject area should have interest and applicability to the group of users. The subject area will be analyzed and documented to provide a standard view of the information with a consistent and standardized data format [1]. Data requirements are collected, analyzed, and prioritized from the individual subject area participants who require data exchange for accomplishing their task(s).

At this time the strategy for managing the DIF should be addressed. An agent will need to be assigned the responsibility of coordinating the development of the DIF, as well as maintaining the DIF once completed.

Step 2: Develop a Logical Data Model (Schema) and Specify Use Cases

In order to arrive at this common view of the data the goal for the group of users is to build a logical data model. This model is also known as a Subject Area
Information (SAI) Model [2] and serves as a center point when developing translations of the data among systems. The SAI Model is a repository for collecting and analyzing data requirements from participants’ existing database implementations.

In order to build the SAI Model, the group of users must communicate the requirements of their individual database systems. Existing physical data structures, models and documentation must be analyzed to determine their current implementation requirements. Each participant’s system could have common information but dissimilar structures, usage or data dictionary. However, because there is sufficient need to share this information, developing a DIF will allow the individual data to be shared among participants.

Once individual data descriptions have been collected, an understanding of the data structure is acquired by building individual physical data models of their current databases. Once the individual physical aspects are documented and understood, then the requirements from the models must be prioritized and merged into an integrated SAI or logical model.

After the logical model is complete, the users in the group provide use cases to describe their database application’s point of view and interface requirements for the data. Definitions and usage of the data complete the understanding of the physical attributes.

**Step 3: Determine and Build the Physical Representation of the DIF**

The attributes of the data in the logical model are expanded with physical attributes into a physical data model that serves as the foundation for defining the DIF. The attributes of the data elements in this physical model facilitate the description of the elements in the DIF. This model also provides a map for common access to the data.

**Step 4: Determine DIF Definition Style and Build Physical DIF**

Based on the use case requirements defined by the users of the group and the metastandards available, the best method or DIF definition style is selected in preparation for publishing the DIF package. There are
several methods that can be used to describe the required format for the data as previously discussed in paragraph 1.4.

Once the preferred DIF method is chosen, the physical description of the DIF is created. There are two basic constructs for describing data elements with a DIF [4]. The first method uses delimiters to mark the boundaries of a data item. Parentheses can be used to denote the beginning and end of a structure or item. Another form of delimited format is comma-separated values. This is common in table definitions where the comma indicates the end of the column and an end-of-line character delimits the row. Sometimes these two methods are used together.

**Step 5: Packaging the DIF**

The DIF package can then be published and may take the form of a specification. This documentation should include background information on the subject area being represented, the purpose and scope of the DIF, the format definition (BNF, XML) description, the actual subject area DIF definition description, consistency rules with constraints, a DIF glossary with data element definitions, and a example DIF file. The example should include a real world sample of data within the current subject area and a use case from the group of users.

**Step 6: Post and Review DIF**

Once the DIF has been packaged, the draft DIF can be posted and distributed to the participants for review. Application developers will be able to evaluate and determine its usefulness in interchanging data between their tools and other applications. Participants should check the DIF for completeness to ensure that the DIF has captured the data elements required for their systems to interchange data.

**Step 7: Publish and Maintain DIF**

Once the DIF has been reviewed, it can be published so that both original and potential participants can access and implement the DIF to exchange information. Application implementers should be encouraged to download and implement the DIF and provide feedback. After a period of time, the data requirements for the DIF subject area will change and new databases will be constructed. As these changes occur, a periodic review of the DIF should occur and appropriate revisions made.

**3. Summary**

We have discussed the definition, purpose and application of DIFs. Examples have been described that demonstrate the utility and effectiveness attained by applying a standard data interchange methodology. We have summarized some of the common DIF development activities that have been used successfully in support of many programs. These steps are not part of a rigid prescribed process but are being successfully applied by several developers of DIF packages.

The use of DIFs offer the potential to share valuable and reusable data among numerous existing and future systems even though these systems may differ by varying technical, organizational or physical perspectives. DIFs also offer a practical way to share information and to facilitate the establishment of partnerships among programs.

**4. References**


**Author Biographies**

PEGGY GRAVITZ is a senior systems consultant with COLSA Corporation. She has served as the project leader at COLSA for several DMSO-sponsored projects including the M&S Directories, the MSRR, the Authoritative Data Sources (ADS), the CMMS-Registration Template (CMMS-RT) DIF and has been...
supporting DoD data standards development and initiatives for over 6 years. She has a MS in Systems Management and a Masters in Business Administration. She has over 15 years experience as a data engineer and systems analyst in information management systems and technology.

THOM MCLEAN is a Research Scientist II at Georgia Tech Research Institute and a graduate student in the College of Computing. He received a B.S. in Aerospace Engineering from the U.S. Naval Academy in 1984, and a M.S. in Computer Science from Texas A&M University in 1992. His thesis topic is the Analysis of Real-Time Distributed Simulation. His other research interests include scientific computing, meta-database, and message sequence charts.

JACK SHEEHAN is the Director, Data Engineering Division, Defense Modeling and Simulation Office (DMSO) and the Technical Director for the Conceptual Models of the Mission Space (CMMS) Program. As the DMSO Data Engineering Director, he is responsible for the Modeling and Simulations (M&S) Functional Data Administration Program and data standards development for several on-going activities. These activities include the High Level Architecture, CMMS, and representations of the environment, systems, and human behavior, discovery and cataloging of DoD Authoritative Data Sources for M&S. He has over 12 years experience as a Senior Systems Engineer conducting basic research, architecture definition, engineering design, prototype implementation, and at-sea feasibility demonstrations for U.S. Navy combat systems. His M&S experience includes development of a CORBA-based real-time architecture which extends Distributed Interactive Simulation standard messages to distributed objects including surface ships and submarines, fixed and rotary wing aircraft, and heavy armor ground forces.