Data Modeling as a Tool to Understanding Simulator Data Requirements

Roy O. Scrudder
Computer Sciences Corporation
Fort Huachuca, Arizona

ABSTRACT

A common problem in the training simulator community is the effective sharing and reuse of data. Each new simulator effort tends to set up its own infrastructure to acquire the data necessary to run the simulator. This was one of the specific problems addressed by the Universal Threat System for Simulators (UTSS) program. The goal was to define common data requirements for a family of training devices and to map these requirements from source databases to the individual data elements required by the training simulators.

The Integration Definition for Information Modeling (IDEF1X) data modeling language was employed with a reverse engineering methodology to accomplish this task. Using this methodology, the UTSS team constructed IDEF1X data models to document the data required for each of the simulators to be served by UTSS. The format of the data varied widely from simulator to simulator, with some simulators employing relational database management systems while others used formatted ASCII files. The resulting data models were then integrated into a common data model which described the data requirements for the family of simulators. Source databases were also modeled using the reverse engineering methodology and the resulting data elements were mapped to the common data requirements model.

In addition to identifying the common data requirements and the sources of data necessary to meet these requirements, several side benefits were realized. Many of the simulator administrators gained additional insight into the rules governing their data. This discovery and documentation of the rules governing the simulator data provide the opportunity to improve the quality of data, thereby reducing down-time from data errors encountered at run-time. Another major benefit of this effort is the identification of a common set of data requirements, data sources, and a data architecture that can be used for the development of new simulators.

ABOUT THE AUTHOR

Roy Scrudder is a Computer Scientist with the Systems Engineering Division of Computer Sciences Corporation. For the last 7 years he has worked in support of the Electronic Proving Ground, Fort Huachuca, Arizona. He is currently responsible for data requirements modeling and integration for the UTSS program. He has also conducted data modeling training and participated in the development of a reverse engineering and schema integration methodology for data requirements in support of the Defense Modeling and Simulation Office. Mr. Scrudder holds a B.S. degree in Applied Mathematics from the University of Tennessee at Chattanooga.

Data Modeling as a Tool to Understanding Simulator Data Requirements

THE NEED FOR EFFECTIVE DATA SHARING AND REUSE

Current Practices

Presently, there is a void in effective sharing and reuse of reference data in the training and simulator community. These reference data range from weapons and platform performance and characteristics data to data describing behavior and tactics of objects in a simulation environment. Because of the lack of sharing and reuse of reference data, there is a massive duplication of effort and expense in gathering and maintaining the data necessary to support the training and simulator community. In most cases, stovepipe systems are set up to extract data from data sources and structure the data into a format compatible with a single simulator (see Figure 1). However, many of these systems access the same data from the same source databases and documents. These data sources include databases and publications developed by the intelligence community as well as unclassified sources such as vendor-supplied information and the Janes series of publications.

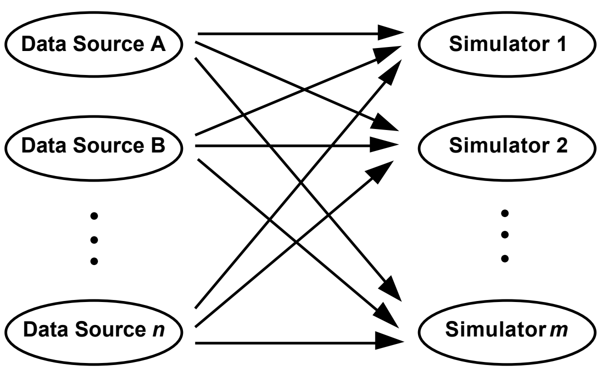

Figure 1. Duplication of Interfaces

Management of extracted reference data varies widely from simulator to simulator. Some simulators use relational database management systems (RDBMS) to manage the reference data, but the majority of simulators maintain the reference data in text files. Furthermore, there is no common way of documenting the data required to support a simulator. Before data can be effectively shared and reused in the training and simulator community, there must be a common way of describing the data required. This technique must not only capture what data are required to support a simulator but must also describe the relationships between elements of the data.

Requirements for the UTSS Program

The Universal Threat System for Simulators (UTSS) program was initiated with data sharing and reuse as two of its primary goals. UTSS is a Joint Service Program led by the Naval Air Systems Command (NAVAIR PMA205). UTSS is chartered by the Joint Technical Coordinating Group - Training Systems and Devices (JTCG-TSD), with funding support from the Defense Modeling and Simulation Office (DMSO). [UTSS-SS]

UTSS goals include building a common repository of reusable data for a family of simulators used in the training community, herein after referred to as the target simulators. UTSS implements a centrally located data repository to house the reference data (see Figure 2).

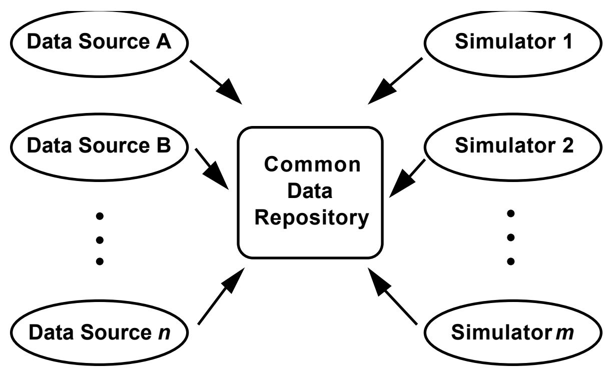

Figure 2. Use of Common Data Repository

Eventually the common data repository will be replaced by a distributed system, where reference data will be maintained at the site responsible for producing the data. In this distributed system, there will be no central warehousing of data. Data will be pulled from the sources and provided to the target simulators in the format they require.

The initial operating capability for UTSS is based on the data requirements from four simulators:

- F-14D Mission Flight Trainer (MFT)
- F-16 Unit Training Device (UTD)
- Modular Semi-Automated Forces (ModSAF) Simulator
- UH-1N Threat Environment Simulator (TES)

UTSS needed a technique to support the statement of data requirements for the target simulators as well as the source databases. This technique also needed to support the integration of data requirements into a single conceptual view which would serve as the design for the UTSS's central data repository.

Data Modeling Solution

The UTSS program adopted the methodology proposed by the Joint Data Base Elements for Modeling and Simulation (JDBE) project for modeling and integrating data requirements. The JDBE methodology was developed in support of DMSO. The JDBE methodology stresses the development of data standards from the bottom up [JDBE]. This fits well with the UTSS goal of establishing a common set of data requirements based on existing simulators.

The JDBE methodology employs the Integration Definition for Information Modeling (IDEF1X) language for stating data requirements. Using this methodology, the UTSS team developed IDEF1X data models to describe the data requirements for each of the target simulators. Continuing with the JDBE methodology, the UTSS team integrated the data requirements into a single data model which combines the requirements from individual target simulator data models.

For UTSS, another key part of the JDBE methodology included the development of mappings. Mappings describe the algorithms and conversions necessary to transfer data between systems using different data representations. The UTSS team developed mappings to describe the required transformations between the central data repository and the target simulators. Additional near-term efforts include the development of mappings to describe the transformations between source databases and the central data repository. These mappings are integral to the JDBE methodology and eliminate the need to make expensive changes to existing systems to conform to a single data standard. The mappings also allow the UTSS program to support simulators which deal with different levels of abstraction for the same types of data.

The IDEF1X Language

The IDEF1X language was developed in the commercial sector. The first use of IDEF1X in the Department of Defense (DoD) was as part of the US Air Force's Integrated Computer-Aided Manufacturing (ICAM) project. It has since been established as a Federal Information Processing Standard [FIPS 184] and has been designated as the standard data modeling language for use in DoD's data standardization program [8320.1-M].

The IDEF1X language is a form of entity relationship modeling. Like other entity relationship modeling languages, IDEF1X describes entities (those things or concepts about which information is kept), attributes (the properties of entities), and relationships between entities [Bruce]. IDEF1X does this through graphical models with supporting metadata. Metadata consists of data that describe data. An IDEF1X data model is not a database. It contains no actual data. Rather, it is a specification of data requirements.

The relational model is the basis for the IDEF1X language. In simple terms, the relational model views data as tables containing individual instances of the entity described in the table. For example an F-16 would be an instance in the table describing aircraft. The columns are the properties of the entity. For the aircraft table, examples of columns would be wingspan, maximum fuel weight, and maximum speed.

Another feature of the relational model is the way that relationships are represented. The relational model assumes that there are one or more columns that uniquely identify the instances of the entity. These are known as keys. For example, an aircraft name would uniquely identify one type of aircraft, and there would be no two aircraft with the same aircraft name. To tie two tables together, the key columns are repeated in each table and hold the same values.

As previously mentioned, simulators vary widely in the way they represent and store reference data. Of the four simulators addressed initially for the UTSS program, only one (F-16 UTD) uses a RDBMS to manage reference data. However, the UTSS team was able to use the IDEF1X language to describe the data requirements and relationships for all of the target simulators

DATA MODELING FOR UTSS

Physical Data Modeling

The first step in modeling the data requirements for each of the target simulators was the development of physical data models. The physical data model strives to capture the structure of the data exactly as it is represented in the simulator's legacy database. In this context, the term "database" denotes a collection of data, whether in an RDBMS or in text files. Figure 3 shows a small portion of the physical data model for the F-14D TES, expressed using the IDEF1X language.

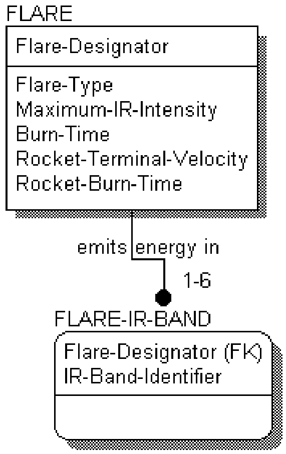

Figure 3. Example from a Physical Data Model

In the IDEF1X language, the tables are known as entities and are represented as boxes. This example shows two entities, FLARE and FLARE-IR-BAND. The table columns are known as attributes and are shown as text within the entity box. Those attributes that form the key are shown above a separating line within the entity. The line between the boxes shows the relationships between FLAREs and FLARE-IR-BANDs. The label "1-6" denotes how many FLARE-IR-BANDs must be specified for any "FLARE".

This small example shows the power of the IDEF1X modeling technique. This very concise graphic states many important rules that must be maintained for a valid set of F-14D TES data:

- Each FLARE is described by a Flare-Designator, a Flare-Type, a Maximum-IR-Intensity, a Burn-Time, a Rocket-Terminal-Velocity, and a Rocket-Burn-Time.
- Each FLARE has a unique Flare-Designator.
- Each FLARE-IR-BAND is described by a Flare-Designator and an IR-Band-Identifier.
- Each FLARE-IR-BAND is uniquely identified by the combination of a Flare-Designator and an IR-Band-Identifier.
- Each FLARE emits energy in one to six FLARE-IR-BANDs.

In addition to the information conveyed in the graphical model, there are additional metadata recorded about the entities and attributes. A definition is recorded for each entity and attribute. Attribute definitions include the units of measure, if applicable. For each attribute, a data type and precision are stated (e.g., floating point to two decimal places). Finally, a domain rule is stated for applicable attributes. Domain rules specify the range of allowable values, either by a range constraint (e.g., 0.0 to 100.0) or by a list of allowable values (e.g., aircraft, ground vehicle, ship).

A complete physical data model is very thorough in specifying a set of data requirements.

The initial implementation of UTSS only dealt with performance and characteristics data. It did not address tactics or behavior data. With these constraints, the physical data models for the target simulators varied in size from 42 to 97 entities and from 301 to 437 attributes.

Logical Data Modeling

Next, the data requirements were further refined. The technique of normalization was applied to the physical data models. Normalization removes ambiguities from a data model, providing a clear definition of the data requirements. The resulting logical data model usually describes additional rules not described in the physical data model. Figure 4 shows a refinement of the FLARE entity from the previous example.

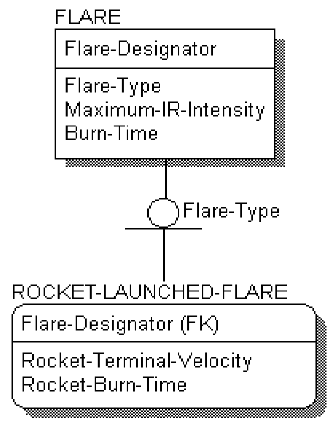

Figure 4. Example from a Logical Data Model

This example uses an IDEF1X construct known as a generalization hierarchy (the circle with a bar underneath) to show that there are subtypes of FLAREs. Specifically, this data model states that a FLARE may be a ROCKET-LAUNCHED-FLARE. In addition, this model is less ambiguous because it states that Rocket-Terminal-Velocity and Rocket-Burn-Time only apply to ROCKET-LAUNCHED-FLAREs. Further it states that the FLARE's Flare-Type is the attribute that determines whether a FLARE is a ROCKET-LAUNCHED-FLARE.

Data Model Integration

The next step in the JDBE methodology required the integration of the logical data models for each of the target simulators into a single conceptual data model. Duplicate entities were combined into single entities and duplicate attributes were eliminated. For example, if two or more of the target simulators' logical data models contained entities describing aircraft, the resulting integrated data model had one aircraft entity.

This entity contains the superset of attributes from the aircraft entities of the target simulators' data models.

Often, two or more target simulators describe the same type of data at differing levels of resolution. For example, one simulator might characterize the thrust of a missile during each stage of flight, while another simulator might require only an average thrust. In cases like these, the higher resolution representation (thrust per missile stage) is always included in the integrated data model. The lower resolution data (average) thrust can always be derived from the higher resolution data.

Mappings

After an integrated data model is created, it is important to record the mappings of attributes from the integrated data model to the target simulator data models. This is the key to transforming data between a database derived from the integrated data model and the target simulators. The same is true for mappings between source databases and the integrated data model.

Many of the mappings are simple equivalencies, but some are more complex. A common type of mapping is a simple algebraic equation used to convert units of measure. For example, the integrated data model may represent maximum missile speed in feet per second. However, one simulator may require this data element in miles per hour. In this case, the mapping is the simple conversion from feet per second to miles per hour. Another example of mapping is the situation stated earlier with missile thrust. In this case, the appropriate mapping from individual state thrust to one simulator's requirement for average thrust may be to perform a weighted average of thrust per stage using the burn time per stage as a weighting factor.

One critical set of mappings describes the equivalence of keys, those attributes that uniquely identify instances of entities. A pervasive problem in the training and simulation community is the lack of standardization of the keys used to identify those things that are simulated. For example, one simulator may reference an M1A1 Abrams tank as an "M1A1", where another simulator denotes the same tank as a "Abrams Tank" and a third simulator denotes it as an "M1-A1." It is critical that one set of nomenclatures be adopted for use in the integrated data model, and that the translations to the individual target simulators' data models are understood. This involves the development of look-up tables.

RESULTS

The approach described here was found to be very effective for modeling the data requirements for UTSS. The data requirements for each of the target simulators were accurately captured. The data models produced by this process were presented to the simulator subject matter experts at each target simulator site. At each site, the UTSS team provided a 30 minute training session on reading and understanding IDEF1X data models. With this minimal training, the subject matter experts were able to understand the IDEF1X data models for their simulators and verify that they accurately portrayed the simulator's data requirements.

The subject matter experts at each target simulator site found the data models to be very useful and informative tools. Because of the structured approach, a complete picture of the simulator data requirements was produced in a concise manner. In all cases, data models documented data requirements at a level of detail not previously achieved at the simulator sites. The subject matter experts gained new insight into the data required for their simulators. The rules resulting from the data models allow the simulator site personnel to more thoroughly check their data and eliminate many costly errors.

Another benefit of this process was the development of a standard set of data requirements. New simulator development projects can use the standardized set of data requirements as a baseline, saving costly analysis time. The data models and associated metadata are being submitted through the DoD data standardization program as candidate standards.

CONCLUSIONS

The process of using data models to describe data requirements has proven beneficial. The methodology described here provides an effective way of integrating multiple sets of data requirements. Even when applied to single simulators, not as part of an integration project, the process provides valuable documentation of data requirements. The process of data modeling can also be used in a top down fashion to define new and unique simulator data requirements.

Future efforts for UTSS include the addition of more target simulators and source databases. As additional data requirements are defined, the UTSS integrated data model and the central data repository based on the data model are expanding to provide greater support to existing and future simulators. Real benefits are achieved as new simulator developments use the UTSS representations and data. Simulators save money and receive more reliable data by using the UTSS repository.

REFERENCES

[Bruce] Bruce, Tom, *Designing Quality Databases with IDEF1X Information Models*, 1992, Dorset House Publishing.

[FIPS 184] *Integration Definition for Information Modeling (IDEF1X),* 21 December 1994, National Institute of Standards and Technology, Gaithersburg, MD.

[JDBE] *Joint Database Elements for Modeling and Simulation (JDBE), Methodology Manual*, February 1995, Electronic Proving Ground, Fort Huachuca, AZ.

[UTSS-SS] *System Specification for the Universal Threat System for Simulators (UTSS)*, 18 November 1994, Naval Air Systems Command, Arlington, VA.

[8320.1-M] *DoD 8320.1-M: Data Administration Procedures*, March 1994, Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Arlington, VA.