Toward a Data Logging Data Interchange Format: Use Cases and Requirements

Björn Möller and Fredrik Antelius, Pitch Technologies, Sweden
Ryan Brunton, JHU/APL, United States
Tom van den Berg and Remco Witberg, TNO, The Netherlands

bjorn.moller@pitch.se
fredrik.antelius@pitch.se
ryan.brunton@jhuapl.edu
tom.vandenberg@tno.nl
remco.witberg@tno.nl

Keywords:
Simulation, Training, After Action Review, Interoperability, Data logging, HLA, DIS

ABSTRACT: *Many use cases for distributed simulation depend on the effective analysis of simulation data after the simulation has completed, sometimes even years later. While many proprietary data loggers exist, logs are stored in proprietary formats often tailored for the specific use case for which each tool was designed and the specific data model used in a given simulation.*

This paper suggests that it is both possible and desirable to exchange, archive, and reuse simulation event log data using a standardized format for data interchange. The authors propose that such a format should be developed. There are several differences in the requirements between runtime formats and interchange log formats, with long-term reusability trumping runtime requirements for performance and space efficiency.

Finally some solutions are suggested for several of the identified technical challenges. These include support for arbitrary data models, simple yet expressive metadata, log size, and complexity. Some use cases for which the suggested format would be most useful are also given.

1. Introduction

Well-documented data can be highly useful over time, for purposes more or less related to the original purpose. This is most obvious in the scientific world where more than 500-year-old astronomical observations are useful in the study of comets. Another example is outdoor temperature measurements where many western countries have continuous data series of more than 150 years that can now be used to study trends and fluctuations in the climate.

This paper covers the use and reuse of simulation data. It describes a number of cases where the reuse of simulation data is highly valuable, both on a short term and long term basis. However the reuse of data is challenging today due to the lack of a standardized file format for archived data. Such a data format is suggested and some preliminary requirements are given. A number of possible solution approaches are also given. A SISO study group is suggested in order to further explore this topic.

1.1 Data logging in simulation

Simulation data is logged today for a variety of purposes, depending on the type of simulations [1]:

Training simulations, where the main purpose is a training effect, commonly log data for after action review purposes (AAR). In this case data is often played back in the original training systems together with additional "God's Eye" views. In some cases the trained staff is offered a lightweight application that can play back the data, sometimes called a "take-home package".

Analysis simulations are often used mainly to produce data, not a training effect. The logged data is an intermediate product that is used to produce the final data, which may be statistical summaries and graphs.

Test and evaluation simulations, where the behavior of a system under test is evaluated early in the development cycle and certainly before the system arrives in the target environment. The logged data is used for analysis and comparison with expected or intended system behavior.

A slightly different use case is **federation development** where well-known data can be used to feed a federation under development for everything from basic testing to verification of correct behavior. During integration different teams can exchange data for preliminary testing of

interoperability, before the different federates are connected for real.

1.2 Reuse of logged data

Logged data can be used for the original purpose as described above or reused for other purposes. The closer the connection is between the original producing simulation systems and the systems where it is reused, the more tailored and optimized the data exchange format usually is. Unfortunately this may lead to that the data becomes less reusable. Data format and semantics may be well known to the developers and may be adapted to peculiarities of particular simulators. A number of federation agreement topics may be implicit and not well documented. A number of particular conditions of the original simulations and scenario data may be lost. These are some challenges that must be dealt with when reusing simulation data.

1.3 Exchange of data between data loggers

Today many larger companies and organizations have several different internally developed data loggers in use in different projects. There are also several COTS and GOTS data loggers available. A lot of applications have data loggers built into the application. The choice of data logger in each project is based on functional requirements, cost, timing, availability and even personal preferences. It is unlikely that everyone will standardize on one common data logger.

Still teams using different data loggers, for example in international simulation or in integrated project teams would benefit from exchanging data in a standardized format.

As an example, two of the largest after action review (AAR) systems in use within the U.S. military are the Joint Digital Collection, Analysis, and Review System (JDCARS) and OneSAF AAR [2]. Both of them support data collection and AAR activities, yet both utilize proprietary and incompatible relational database tables for data storage. This means that even when data in one system is logged for a simulation event using a data model supported by the other (e.g., RPR FOM, MATREX) the logged information cannot be utilized in the other tool.

Similarly commercial offerings, such as the Pitch Recorder [3] and MAK Data Logger [4] utilize highly optimized storage solutions to accommodate runtime data capture that are proprietary to the capturing tool. This incompatibility can restrict the ability of simulation designers to pick the best-suited tool for their exercise needs by constraining their choice to compatibility with previously logged data and/or previous software investments by other participants or other departments within their own organization.

1.4 Reuse of data over time

Time goes by. This introduces several challenges with respect to the reuse of data:

Computer architectures and basic data representations evolve. Popular 36 bit architectures for scientific calculations are no longer commonly available. In the 90's the PC technology evolved from 16 to 32 bits and today the 64-bit architecture are becoming popular on the desktop. While ASCII and EBCDIC character representations are still in use, the trend in many languages is to move towards Unicode.

Standards in general and the domain specific **data exchange models** in particular evolve and are extended and replaced over time. Older standards are forgotten, in particular if they are poorly standardized or if the de-facto standard deviates from the written standard.

Software applications evolve or disappear, making it difficult to open older file or data base formats.

Limitations of the simulators and scenarios that were used to produce the data and even the purpose of the original federations may be forgotten.

The **purpose** of using simulation and even approaches and mind-sets of the people that build simulators evolve. Early simulation builders probably did not think of Simulation Based Acquisition or Simulation Based Design. Still their logged data may be useful in these newer contexts.

And finally **expertise** with simulators and historical data used in or derived from earlier simulation exercises disappears, because people change job or retire and with them the knowledge.

All of the above challenges need to be addressed for any data that is stored for reuse over time.

1.5 The potential of data reuse

It is difficult to predict the full range of uses to which logged data may eventually be reused should it be extensively archived, much in the same way that new uses are being routinely found for archived scientific data (e.g., particle accelerator results, climate data, etc.).

Traditional data reuse includes:

- Reuse for training or analysis as scenarios for additional simulation exercises.
- Additional analysis based on existing data.
- Test and integration.

Several additional uses may include:

- Cross-tool AAR, where tools other than the capturing logger are used for domain-specific AAR and where data from several logs may be aggregated, combined and stored in a new log, or an existing log extended.
- VV&A of a simulation environment to (help) determine if the simulation environment is fit for purpose. The logged data may serve as evidence.
- Long term archival for analysis potentially long after the generating simulation is no longer available.
- Reverse-engineering simulation components based on recorded behavior; potentially useful should the original simulation behavior be required but the original components are no longer available (e.g., code lost, original company no longer in business, hardware obsolete, etc.)

2. Tentative Requirements for a Standardized Data Format

The need for a standardized log file format was identified as early as 1995, when STRICOM developed a standardized format for DIS logs in parallel to development of the DIS standard [5], while an initial approach to requirements for an interchange format viable for archival purposes was documented in the Standardizing Army After Action Review Systems (STAARS) report in 1996 [6]. Despite these early moves, no such standardized data interchange format for simulation event logs exists today.

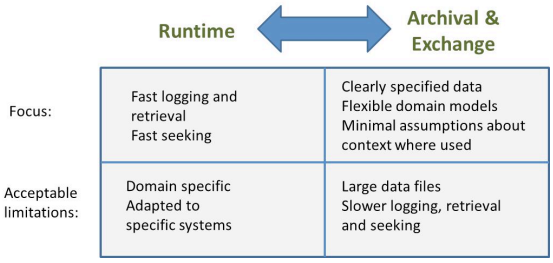

Figure 1: Conflicting Requirements

This section lists some requirements for a data interchange format (DIF). Note that a data format for long-term archival that supports any information model will have certain requirements that conflict with a high-performance runtime data format that applies only to one class of simulators.

Runtime data formats need to focus on the performance in retrieving and seeking data. It is acceptable to adapt it to a specific domain information model and to adapt it to peculiarities of certain existing systems. There is little need to capture all the details of the data formats, interpretation and context inside of the data since individuals that are knowledgeable about the systems are available.

Archival and exchange data formats need to be flexible to adapt to different domain information models. The data formats and, to a reasonable degree, the assumed context needs to be clearly specified. No assumption can be made about which systems that will consume this data.

2.1 Basic requirements

Flexible Information Model: No particular domain specific representation can be assumed, like in DIS. Any simulation DIF needs to be adaptable to arbitrary data models, both to support architectures like HLA which natively support such models today and because of the need to adapt as existing fixed-model architectures evolve (e.g., DIS) and new architectures are developed.

Multi-architecture Support: Support for logging data generated via multiple simulation architectures (e.g., DIS [7], HLA [8], TENA [9]) is a critical requirement for format adoption in the U.S. where no one architecture has, or is trending toward, a monopoly position. Often multiple architectures are utilized in a single simulation event through the use of gateways and bridges between architectures, requiring that a complete log of the simulation event support all used architectures.

From an international perspective it may be just as important to support other types of protocols, in particular voice communication in training scenarios. Link 16 is another candidate that is widely used in the air defense domain.

One possible approach would be to make the format neutral to the standard or protocol used rather than to include special adaptations to a particular protocol.

Another possible approach is to bridge all data into a single architecture that supports flexible domain models, like HLA, where the data is then logged.

Reproducibility: Data must be recorded sufficient to reproduce or reconstruct the captured exercise data as originally produced.

Entity Lifecycle: The simulated entity lifecycle must be recordable (see also [1]). This can be seen as an extension of the previous requirement, but has further utility for example for VV&A.

Supporting Data: There is a "need to archive planning, communications, and administrative data along with other data types" [6]. Especially in live simulations, out of band data is often critical to put the simulation events within context. This context can be essential for analysis and training evaluation,

and includes information like measures of performance (MOP), planning data, administration information, communications data (e.g., Link 16), etc.

Large Data Volume Support: Management of large volumes of data, possibly in excess of file size limitations, needs to be supported. Large and/or long running exercises can generate incredibly large volumes of data. This data can easily exceed the file size limit imposed by the operating system, and may require the use of more than one physical volume for storage.

2.2 General Meta Data

Many of the preceding requirements imply a requirement for metadata separate from the DIF itself. This metadata provides critical information like:

- File size limits, information on ordering of the resulting data log files.
- The architecture(s) and data exchange model(s) (DEM) in use.
- The supporting data required to reproduce or interpret simulation results.
- Recording identification information, including simulation name, points of contact, targeted domain, use cases, etc.
- Recording simulation logical and clock time: start, end, and per-data log file temporal bounds.

2.3 Meta Data about the data formats

In addition to the general meta data referenced above, the information common to the utilized simulation Data Exchange Models (DEM(s)) needs to be included with the log as meta data in order to interpret the recorded data. This meta data includes:

- Definitions of data types, records, etc.
- Time representation.
- Object class and interaction definitions.
- Attribute and parameter type information.

2.4 Runtime data

Runtime data capture should ensure that data is "archived in a manner that allows reproduction of the data stream exactly as received and supports analysis across exercises." [6] At minimum this requires the storage of:

- Entity lifecycle events (e.g., object creation and deletion).
- Event timestamps (both clock time and simulation time if present).
- Data sufficient to associate logged events with the DEM entities being logged.
- Architecture-specific events necessary to support analysis in use cases where such data is necessary (e.g., ownership management, federate join events, execution control events, etc.).

3. Potential Solution Approaches

The requirements stated above have led the authors to the following potential solution approaches to a simulation data logging DIF:

Full logs should be structured as an archival format with metadata separate from event log information. This has precedent both within the world of traditional backup and archiving and software development (e.g., JAR [10] files and OSGI [11] bundles in Java). Supporting data would be included in the archive; utilizing domain-specific standard interchange formats with the archive meta data tying supporting data to the event data. An example of this approach in the 3D domain is the Keyhole Markup Language Zipped (KMZ) format used by Google Sketchup which consists of a ZIP archive containing a Collada file with 3D data, a Keyhole Markup Language (KML) file with GIS mapping data, and various texture files, with archive metadata tying the various included files together [12]. By structuring the log in this way:

- The data is kept separate from the metadata
- Architecture-specific DEMs can be included in the archive and referenced within the data log without impacting the data log format itself.
- Supporting data can be maintained in parallel and associated with simulation data using archive metadata.
- Data logs can be split according to arbitrary file size, temporal duration, etc., while maintaining ordering even when split across physical volumes.

Both metadata and data should (if possible) be encoded as XML. XML has the advantage of being both human and machine readable, and has become the de facto standard for encoding arbitrary data interchange formats.

This latter point would allow the reuse, either through direct inclusion or through hyperlinks within the meta data URL, of existing XML-based format standards already within use within the simulation community. These include, but are not limited to:

- The Object Model Template (OMT) DIF [13] utilized to create DEMs for HLA simulations.

- Identification information from the OMT/Base Object Model (BOM) standard [14].
- Point of contact, history, and security information from the M&S COI (MSC) Discovery Metadata Specification (MSC-DMS): standard [15].

Any DIF should be "optimized for generality." [1]. While separating the meta data from the data and linking data elements directly to their native DEM definitions goes a long way toward meeting this goal, the DIF format itself should have simple, expressive syntax and semantics in order to make it easy to parse by both a human reader and computer software.

The following recommendations may serve as a foundation from which this goal can be met:

- All events should be annotated with both clock time stamps and (optional) logical time stamps. This supports the separation of event logs into size or time constrained files, the reconstruction of file ordering should the meta data be lost or corrupted, and the interleaving of similarly time stamped supporting data by software tools.
- Data values should default to human readable formats. This means:
    1. Primitives (e.g., integers, floating point numbers, strings, arrays) should be human readable and annotated with type information to facilitate type recognition by software.
    2. Opaque binary data should be annotated with the associated DEM reference (e.g., object class and attribute name) to at minimum provide context to a human reader.
- Structural homogeneity should be maintained, with common XML attributes providing the mechanism to differentiate event types rather than unique XML elements.

4. Discussion

4.1 Extensibility

Format extensibility must be handled with great care. On the one hand, building extensibility into the format allows for logging of data not produced by contemporary simulations or anticipated by authors. On the other hand, extensibility of core elements could be a fatal flaw in a format intended for long-term archival storage of data, especially if the specific extensions are not documented or fail to follow a prescribed, predictable pattern.

Following the suggested model in this paper, format extensibility limited to the archive metadata gains the benefits of future-proofing the format without the danger of breaking backward compatibility.

4.2 Sponsor view

Any adoption of a future data logging DIF standard will require economic as well as technical solutions. It is suggested that any customer or sponsor of a simulation should require that simulation data be logged and saved in a standardized format. This should be part of the project deliverables. In this way data logger and AAR developers and vendors would be incentivized to implement import/export support for the standard data logging DIF.

It should be noted that the authors are not suggesting that native, run-time support be required, as the proposed format has been conceptualized for transparent data interchange and long time archival of data logs, not runtime efficiency.

The OneSAF AAR architecture proposes a possible solution for the gradual adoption of a data logging DIF standard. While OneSAF internally utilizes an optimized runtime database for data collection, it also provides for the export of archival data for independent storage. The archival data is generated by a component called the Simulation Output Repository (SOR) utilizing a schema mapping the runtime database schema to a target XML format. The System Abstraction Layer (SAL) architecture into which the mapping schema is loaded provides an example, and possibly a concrete target, for the adoption of a data logging DIF standard without disruption of existing optimized runtime capture mechanisms [1].

5. Conclusions

This paper has described the need for a data logging interchange and archival format. Such a format would be a substantial improvement on how we develop and use simulations and how we can reuse data both in the short and long term.

An initial set of requirements has been presented, including some notes on possible solutions. While the authors are currently evaluating implementations of many of the proposed solutions, successful development and adoption of a data logging DIF standard will require input from and participation by the larger simulation community.

Further, future efforts within the SISO Distributed Debrief Control Protocol (DDCP) Study Group [16] may overlap with the efforts of any data logging DIF study group, so extensive coordination between the two groups is highly recommended. The different requirement in performance versus reuse must me clearly understood in this case.

The SISO Federation Agreement Template PDG (FEAT) work is also highly relevant to a DIF format, in particular to improve the interpretation of logged data.

The authors strongly believe that a data interchange format for simulation data would be highly valuable to the simulation community. SISO would be the right forum for the development of such a standard starting with a Study Group to further discuss the requirements.

References

[1] Moller, B., Antelius, F., van den Berg, T., Jansen, R. *Scalable and Embeddable Data Logging for Live, Virtual and Constructive Simulation: HLA, Link 16, DIS and more*. Proceedings of the 2011 European Simulation Interoperability Workshop. 11E-SIW-016.

[2] Brunton, R., et al, "Live-Virtual-Constructive Architecture Roadmap Implementation, Common Capabilities – Common Data Storage Formats Progress Report", NSAD-R-2011-022, February 2011.

[3] Pitch Recorder: http://www.pitch.se/products/recorder

[4] MAK Data Logger: www.mak.com/products/datalogger.php

[5] Garnsey, M., Boyd E., Green, K., Kennedy, D. Liu, J. Schow, G., Smith, S., Wahrenberger, D. (1995). "DIS logger interchange format proposed standard." 12th Workshop on Standards and the Interoperability of Distributed Simulations: Volume 1: Position Papers, 249-253.

[6] Meliza, L. L. (1996). *Standardizing Army After Action Review Systems (STAARS) Research Report.* U.S. Army Research Institute. U.S. Army Research Institute.

[7] IEEE: "IEEE 1278, Distributed Interactive Simulation (DIS)", www.ieee.org.

[8] IEEE: "IEEE 1516-2010, High Level Architecture (HLA)", www.ieee.org, August 2010.

[9] Test and Training Enabling Architecture, https://www.tena-sda.org/

[10] Java Archive (JAR) File Format: http://download.oracle.com/javase/1.3/docs/guide/jar/jar.html

[11] OSGI Specification: http://www.osgi.org/Specifications/HomePage

[12] KMZ Files: https://code.google.com/apis/kml/documentation/kmzarchives.html

[13] IEEE: "IEEE 1516.2-2000 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification", www.ieee.org

[14] Base Object Models: http://www.boms.info/

[15] M&S COI (MSC) Discovery Metadata Specification (*MSC-DMS*): http://*www.msco.mil/mscdms.html*

[16] SISO Distributed Debrief Control Protocol (DDCP) Study Group, http://www.sisostds.org/StandardsActivities/StudyGroups/DDCPSGDistributedDebriefControlProtocol.aspx.

Author Biographies

BJÖRN MÖLLER is the vice president and co-founder of Pitch, the leading supplier of tools for HLA 1516 and HLA 1.3. He leads the strategic development of Pitch HLA products. He serves on several HLA standards and working groups and has a wide international contact network in simulation interoperability. He has twenty years of experience in high-tech R&D companies, with an international profile in areas such as modeling and simulation, artificial intelligence and Web-based collaboration. Björn Möller holds an M.Sc. in Computer Science and Technology after studies at Linköping University, Sweden, and Imperial College, London. He is currently serving as the vice chairman of the SISO HLA Evolved Product Support Group.

FREDRIK ANTELIUS is a Lead Developer at Pitch and is a major contributor to several commercial HLA products. He holds an M.Sc. in Computer Science and Technology from Linköping University, Sweden.

RYAN BRUNTON is a member of the Senior Professional Staff at the Johns Hopkins University Applied Physics Laboratory. He received his B.S. in Computer Science from the University of California, San Diego in 2001. Mr. Brunton has extensive experience in simulation, enterprise

architecture, data mining, and web technologies. He currently has a patent pending on a unique application of machine learning to the analysis of domain expert effectiveness. He is a member of Tau Beta Pi, ACM, and SISO.

TOM VAN DEN BERG is scientist in the Modeling, Simulation and Gaming department at TNO, The Netherlands. He holds an M.Sc. degree in Mathematics and Computing Science from Delft Technical University. His research area includes simulation systems engineering, distributed simulation architectures and concept development & experimentation.

REMCO WITBERG is scientist in the Modeling, Simulation and Gaming department at TNO, The Netherlands. He holds an M.Sc. degree in Applied Mathematics from the University of Twente. His research area includes (distributed) simulation, exercise analysis, and automated evaluation in training systems.