

1.3 An Exploration of Trainer Filtering Approaches

An Exploration of Trainer Filtering Approaches

Patrick Hester, Andreas Tolk and Sandeep Gadi
Old Dominion University

ptheater@odu.edu atolk@odu.edu sgadi@cs.odu.edu

Quinn Carver and Philippe Roland
Referentia Systems Inc.

qcarver@referentia.com prolland@referentia.com

Abstract. Simulator operators face a twofold entity management problem during Live-Virtual-Constructive (LVC) training events. They first must filter potentially hundreds of thousands of simulation entities in order to determine which elements are necessary for optimal trainee comprehension. Secondly, they must manage the number of entities entering the simulation from those present in the object model in order to limit the computational burden on the simulation system and prevent unnecessary entities from entering the simulation. This paper focuses on the first filtering stage and describes a novel approach to entity filtering undertaken to maximize trainee awareness and learning. The feasibility of this novel approach is demonstrated on a case study and limitations to the proposed approach and future work are discussed.

1.0 INTRODUCTION TO FILTERING

Whenever the source system in a simulation environment provides more information than needed in the target system, information reduction is needed. There are three basic forms of information reduction:

- **Masking:** the filter defines a subset of the available information that is relevant. This is done by reducing the selected information to a subset of the available information, normally by only looking at a subset of all available properties. Properties that are not useful for the target system are simply not passed through the filter. This subset can be reduced further by limiting the allowed value domain of given properties. Masking does not change the information; it simply cuts properties and property values that are of no interest to the target system.
- **Aggregating:** the filter aggregates several lower level properties into one aggregated property for the target system. Computing the mean value or adding up individual numbers to a sum are typical examples. It should be pointed out that in the process of

aggregation, information gets lost, and it is not reversible.

- **Transforming:** although not contributing to information reduction, filters often support the transformation of information. Transformation is mapping properties of the source systems to equivalent properties of the source system using a reversible function on the value domains. An example is the mapping of country codes to country names.

These concepts are not new, but well understood in information technology. Singhal and Zyada (1999) introduced similar concepts to define interest management as "limiting the amount of information passed across a communications interface to the information of interest for a certain user perspective at that moment in time." Morse et al. (2004) applied these ideas to interest management for web services.

The 1516-2010 IEEE (2010) Standard for Modeling and Simulation High Level Architecture (HLA) defines Data Distribution Management (DDM) functionality that needs to be provided by the Runtime Infrastructure (RTI). DDM in the HLA takes the form of range specifications for generic dimensions and is neutral regarding the definition or

meaning of the dimensions. This approach implements the masking of information. Aggregation and transformation are within the responsibility of the participating federate. DDM, however, allows individual federates to specify individual filters through which they can receive their information. It should be pointed out that the DDM specifications differ significantly between the various versions of HLA (1.3NG, IEEE 1516, IEEE 1516 Evolved) and also between different implementations of these standards. As such, DDM is standardized, but the community has not embraced a particular solution so far.

This paper explores filtering and DDM in the context of a particular problem, discussed in the following section, along with solution requirements. Following is a discussion on the developed solution. Finally, some conclusions and recommendations for future work are provided.

2.0 PROBLEM DISCUSSION

In the particular DDM problem addressed in this study the challenge is to select from thousands of constructive simulated entities and dozens of live entities that interact in a training domain those being relevant to a trainee within a virtual trainer whose graphical capabilities are limited to display only a subset, typically less than 100 entities. On the one hand side, the reason for this need for information reduction can be the limitation of the simulator due to the technical nature of the display. On the other hand side, the reason can be the need for special training, like reducing overload for new recruits by displaying too many options or the focus on a particular set of targets. In both cases, the selection of the targets to be displayed should be configurable by the trainer based on his constraints.

Among the constraints for the architecture considerations are the following:

- All pieces of information needed to display the chosen entities – or an aggregation thereof – need to be

extracted from the training domain (Fleet Synthetic Training (FST) Simulation Domain).

- Only those pieces of information needed to display the chosen entities – or an aggregation thereof – should be extracted from the training domain (FST Simulation Domain).
- Aggregation of information and masking should be done at the earliest possible point, which is when it can be assured that nobody else needs the original pieces of information any longer.
- The FST Simulation Domain should not need to be modified for the solution.

The Joint Live Virtual Constructive (LVC) Data Translator (JLVCDT), also known to as JBUS, is a potential solution to provide these masking and aggregating services. JBUS has originally been developed by the USJFCOM to enable the easy integration and data translation between joint simulation system. JBUS was developed to serve as federation bridging utility, providing federation to federation connectivity, federation to external protocol connectivity, and data filtering between protocols and federations. The US Navy developed NCTE and NASMP federate object model support for JBUS to support FST. This JBUS version currently JBUS filtering utilities contains several built-in filters, reducing the burden of filter development. Filter options within JBUS are currently configured via check boxes within the JBUS GUI for filters.

An initial approach to provide more dynamic filtering envisioned a solution where JBUS filter options could be configured remotely. Filter command messages interpreted by JBUS would be sent as payload in an already existing command message. The values for the infrastructure of choice can be changed by XML messages that are used to update the respective selections corresponding to commands available in the GUI.

The definition of an XML schema for command and control of JBUS was one of the first design activities.

A Training Aware Common Operational Picture (TACOP) will provide tools for monitoring real-time data in a large scale LVC training environment. The TACOP system will be built as a filtering framework, filter controls, and an IOS interaction interface. The following section discusses the authors' solution for developing a filtering approach.

3.0 DISCUSSION

3.1 Proposed Solution

Given the constraints proposed for TACOP, the architecture shown in Figure 1 was developed by the authors.

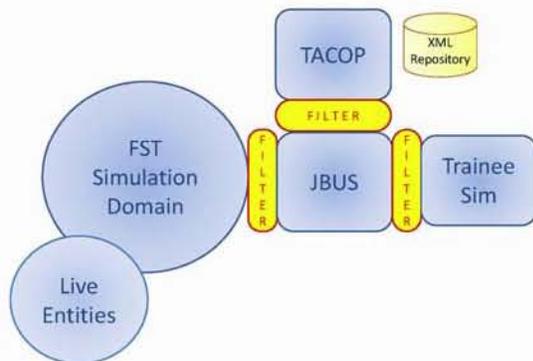


Figure 1: Initial Proposed System Architecture

The assumptions for this recommended point of view can be summarized as follows:

- The JBUS object model (JOM) itself can be interpreted as the master filter, as only elements captured in the JOM can be communicated using JBUS. This is a technical assumption.
- The FST Simulation Domain is outside of the control of the trainer. It simply provides the operational context and all relevant data in support of various training objectives. All activities the

trainer is interested in for his task are embedded into a broader operational context within this domain. This is an operational assumption.

- The trainer only needs a subset of all of the information that is available in the FST Simulation Domain. The information required is training objective specific, e.g., the same scenario can be used to support air operations against submarines as well as against convoy operations. The subset of the overall scenario the trainer needs to see is different for both cases. This is an operational assumption.
- The trainer needs to see all information displayed for the trainee as well as additional information he needs to choose the next configuration to be selected for the trainee. Therefore, the trainee picture must be a real subset of the trainer picture. This is an operational assumption.

This motivated recommendations for the design of three filters:

- Filter One (between FST and JBUS) is configured to support the training objective. It filters out all elements that were not relevant for the current training objective.
- Filter Two (between JBUS and TACOP) is configured to optimize the information presentation for the trainer. Based on current situation, this filter selects a subset of everything that passed filter one and that shall be displayed for the trainer. This filter can be used for various purposes, such as filtering out detailed information needed for 3D high resolution displays for the trainee, but that are not needed for the TACOP display (technical optimization); or to avoid cognitive overload the number of

displayed units can be limited to a maximal number close to the training area (cognitive optimization).

- Filter Three (between JBUS and Trainee) is configured to display the subset of information the trainee needs to see for optimal training purposes. It makes sense to assume that this is a subset of what the trainer sees (although this is not necessary).

All three filters are needed, and all three filters need to be configured in support of technical optimization for engineers, operational optimization for military users, and cognitive optimization for psychologists supporting the training.

3.2 Implemented Solution

The TACOP system will be built as a filtering framework, filter controls, and an IOS interaction interface. The filtering framework will accept and respond to commands and execute the actual filtering. The filter controls will present the capability and flexibility of the framework as an intuitive GUI that makes training a simpler, more manageable task for the instructor operator. TACOP will also develop an extensible protocol for communicating filtering commands and responses of the IOS to the filtering framework through the HLA network.

The goal was to then be able to build a generalized schema to accommodate the addition of new filters in the future. To build the schema, we used Liquid XML Studio 2010. Filters were categorized into filter types and each of these filters contains a name, id, parameters and description as tags. After generalizing the schema, the next step was to show the interactions between the filtering framework and IOS. For this we developed use cases to start with and chose JAXB (Oracle, 2003) to work with the xml communication between the filtering framework (backend) and IOS (frontend).

Our team decided to use JAXB, since it's a straightforward software tool in java which allows java developers to access and process XML. After generating the required classes in JAXB, we developed a frontend (IOS) and backend (filtering framework) and we now try to establish communication between these two as a prototype to the communication between the Filtering network and IOS. This simplified architecture is shown in Figure 2 below, showing also how the simplified version has been derived from the original concepts

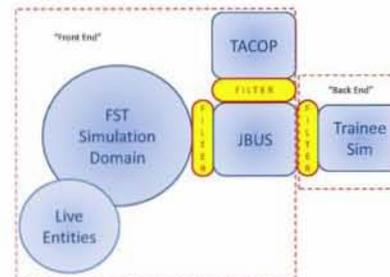


Figure 2: Simplified System Architecture and Filters

The authors utilized the single filter approach (between the front end and back end) in order to demonstrate a proof of concept approach to filtering to ensure the TACOP requirements could be met. The communication is carried through xml messages being exchanged between the frontend and the backend, where we have used marshalling and unmarshalling of xml. To briefly describe XML concepts, we have to know about XML data binding. XML data binding refers to the process of representing the information in an XML document as an object in computer memory.

This allows applications to access the data in the XML from the object rather than using the DOM or SAX to retrieve the data from a direct representation of the XML itself. An XML data binder accomplishes this by automatically creating a mapping between elements of the XML schema of the document we wish to bind and members of a class to be represented in memory. When this process is applied to convert a XML

document to an object, it is called unmarshalling. The reverse process, to serialize an object as XML, is called marshalling. After everything is done as described above you should get the hierarchy in eclipse (which we are using) as shown in Figure 3.

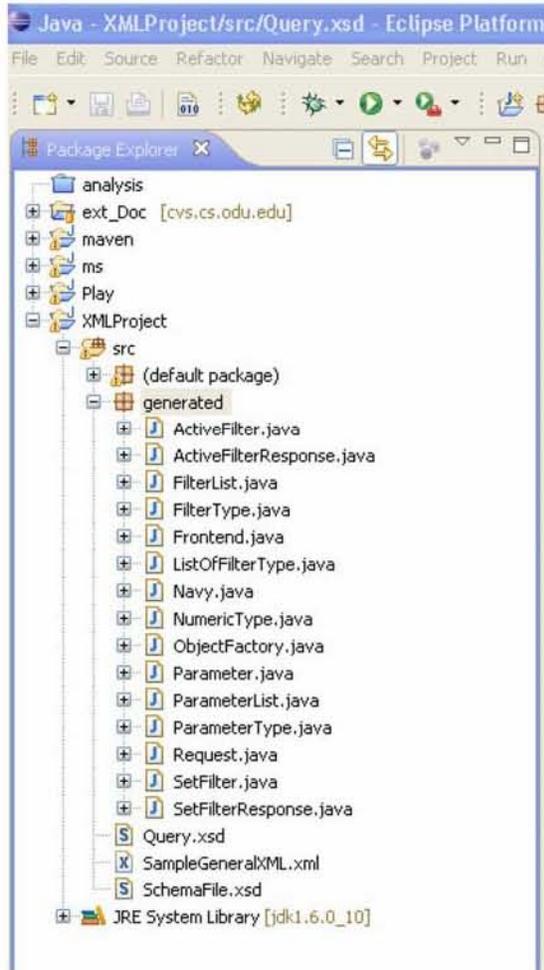


Figure 3: Hierarchy After Generating the Classes Using XJC

The next step was to develop a set of use cases between the frontend and backend, such as the frontend making queries like active filters, specific filter parameters, values and the comparators to filter the entities that the user is looking for. A state machine tool was then developed using JAXB and eclipse based on the developed use cases.

The advantage of this approach in comparison with DDM is that the recommended solution is fully and dynamically configurable during runtime. In addition, predefined solutions can be stored as XML files that can be distributed between various users, supporting the FST user community.

Finally, future work will develop the backend filtering engine using DROOLS (or JBoss Rules which is a Java based rule engine) (see Browne, 2009). DROOLS will be used to compare facts about objects in the simulation against user configured filter rules which are added to a Knowledge Based Session.

During follow-on work the proposed three system architecture will also be revisited in efforts to design a TACOP which functions in a less distributed, API-like format.

4.0 CONCLUSION(S)

The paper discusses the requirements and development of an architecture and associated prototype for the TACOP. The prototype provided for the communication between IOS and filtering framework where the communication is carried out through XML messages. With this prototype we have proposed a solution to select from the thousands of simulated entities to display only those entities which fit user requirements, helping to alleviate a common problem experienced by trainers during LVC training events.

5.0 REFERENCES

- 1) Bizub, W., D. Bryan, E. Harvey (2006). The Joint Live Virtual Constructive Data Translator Framework – Interoperability for a Seamless Joint Training Environment. *Proceedings of the 2006 NATO RTO MSG Symposium, Report RTO-MP-MSG-045*, Rome, Italy, October, paper MP-MSG-045-09-USA
- 2) Browne, Paul (2009). *JBoss Drools Business Rules* (1st ed.), Packt Publishing, pp. 304, ISBN 1847196063.
- 3) IEEE (2010). *1516-2010 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules*.

4) Oracle (2003, Feb.). *Java Architecture for XML Binding (JAXB) Bindings Schema for JAXB*. Retrieved from <http://java.sun.com/xml/ns/jaxb/>.

5) Morse, K.L., R. Brunton, J.M. Pullen, P. McAndrews, A. Tolk, J.A. Muguira (2004). An Architecture for Web Services Based Interest Management in Real Time Distributed Simulation, *Proceedings of the 8th IEEE International Symposium on Distributed Simulation and Real Time Applications*, pp. 108-115, October 21-23, Budapest, Hungary.

6) S. Singhal and M. Zyda (1999). *Networked Virtual Environments: Design and Implementation*, ACM Press.