Shtolo – Converting STEP Short Listings to Annotated Listings

Don Libes

Factory Automation Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract

A STEP Application Protocol includes an AIM EXPRESS Annotated Listing. An Annotated Listing is created by combining the Short Listing and any objects from STEP Integrated Resource Parts that are referenced from the Short Listing either directly or indirectly. A number of transformations are performed on the resulting model, which is then formatted and printed.

In the past, this process has been carried out by hand. This document describes Shtolo, a tool to automate this process. Shtolo conforms to EXPRESS Part 11 and The Supplementary Directives for the Drafting and Presentation of ISO 10303 allowing direct inclusion of the result into an Application Protocol specification.

Shtolo relies on the NIST EXPRESS Toolkit and the NIST EXPRESS Pretty Printer.

Keywords: EXPRESS; presentation; pretty print, Exppp; PDES; STEP; Supplementary Directives

Background

The PDES (Product Data Exchange using STEP) activity is the United States' effort in support of the Standard for the Exchange of Product Model Data (STEP). STEP is an emerging international standard for the interchange of product data between various vendors' CAD/CAM systems and other manufacturing-related software [1]. A National PDES Testbed has been established at the National Institute of Standards and Technology (NIST) to provide testing and validation facilities for the emerging standard. The Testbed is funded by the Department of Defense Continuous Acquisition and Life-Cycle Support (CALS) Program.

As part of the testing effort, NIST is charged with providing software for manipulating STEP data. Provided in the form of tools and toolkits for building new tools, the software is research-oriented and evolving. This document is one of a set of reports [2-12] which describe various aspects of the software.

Introduction

STEP Application Protocols (APs) describe product data to be exchanged and the meaning of that data in a particular industrial context (e.g., associative drafting [13]). Each AP includes an Application Interpreted Model (AIM) which is a mapping of STEP Integrated Resource (IR) Parts to meet the Application Reference Model (ARM) which is another part of each AP.

The AIM is documented by an EXPRESS information model [14] known as the Short Listing. The Short Listing consists of references to Integrated Resource elements and definitions of all new elements and constraints added during interpretation (i.e., development of the AIM). For instance, rules may be defined to place additional constraints on Integrated Resource elements.

An AP also contains an expanded form of the Short Listing, known as the Annotated Listing. The Annotated Listing includes the complete documentation of the AIM. This includes textual descriptions of all of the Resource elements used and other constructs added during interpretation, as well as the EXPRESS definitions of these elements. The concern of this document is the generation of the EXPRESS component of the Annotated Listing. From now on, all references to the Annotated Listing refer only to the EXPRESS model.

Shtolo is a tool to automate the generation of the Annotated Listing. Shtolo reads the Short Listing and based on this, selects any necessary IRs for inclusion. The model is processed and an Annotated Listing is generated in a form suitable for direct inclusion into an AP specification. This document describes Shtolo.

Using Shtolo

Using Shtolo locally is relatively trivial. The following assumes that it is correctly installed. See §Obtaining Shtolo on page 8 for more information on obtaining and installing Shtolo.

Shtolo is invoked as “shtolo” with an argument of the file name containing the Short Listing.

```
% shtolo z.exp
```

Shtolo prints any errors encountered in processing the file. If there are no errors, Shtolo describes the output file. In this example, the Short Listing schema is called shell_based_wireframe_aic and thus, Shtolo places the Annotated Listing in a file by that name with an “.exp” suffix.

```
shtolo: writing schema file shell_based_wireframe_aic.exp
```

A final message is printed that summarizes if any errors were encountered.

```
No errors in input
```

As with all the NIST EXPRESS tools, Shtolo sets its exit status to 0 if there are no errors. If errors are detected, the exit status is set to 1.

Shtolo may also be used remotely in which case it does not need to be installed locally. See [15] for more information.

An Example

A Short Listing is characterized by a large number of USE and REFERENCE statements which refer to IRs, and a small number of object definitions (e.g., entities) and constraints (e.g., rules).

For example, the following (very short) Short Listing is typical in the ratio of references to definitions. The schema uses objects from two schema and defines nothing new.

```
SCHEMA goodies_aic;

USE FROM a_schema;

USE FROM another_schema
   (stuff_entity,
    more_stuff,
    my_stuff);

END_SCHEMA;
```

The referenced schemas are shown below.

```
SCHEMA another_schema;

ENTITY stuff_entity;
  attr: stuff_type;
END_ENTITY;

ENTITY my_stuff;
END_ENTITY;

ENTITY their_stuff;
END_ENTITY;

ENTITY more_stuff;
END_ENTITY;

ENTITY our_stuff;
END_ENTITY;

TYPE stuff_type = SELECT
   (my_stuff,
    our_stuff,
    their_stuff);
END_TYPE;

END_SCHEMA;

SCHEMA a_schema;
TYPE a_type = integer;
END_TYPE;
END_SCHEMA;
```

An Annotated Listing is an (almost) semantically identical model but without the USE and REFERENCE statements. Instead referenced objects are brought directly into the schema. Information describing the originating schema of each object is discarded. A single new schema is created within which all objects are defined. Objects which are not directly referenced through the objects in the Short Listing are discarded. Some pruning and other rewriting takes place as well. In the example shown here, the select type is rewritten to omit the objects that are originally declared but not used in the final schema. A special comment is added to the final listing. The presence of this comment allows Shtolo to overwrite the file without worrying about deleting something hand-written.

Other details about what is written in the Annotated Listing are described in the remainder of this document.

```
(* This file was generated by exppp (an EXPRESS Pretty Printer)
written at the National Institute of Standards and Technology
by Don Libes, February 19, 1993.

WARNING: If you modify this file and want to save the changes,
delete this comment block or else the file will be rewritten
the next time exppp processes this schema. *)

SCHEMA goodies_aic;

  TYPE a_type = INTEGER;
  END_TYPE; -- a_type

  TYPE stuff_type = SELECT
    (my_stuff);
  END_TYPE; -- stuff_type

  ENTITY more_stuff;
  END_ENTITY; -- more_stuff

  ENTITY my_stuff;
  END_ENTITY; -- my_stuff

  ENTITY stuff_entity;
      attr : stuff_type;
  END_ENTITY; -- stuff_entity

END_SCHEMA; -- goodies_aic
```

Shtolo – Processing Overview

Shtolo can be viewed as a sequence of several steps.

1 Reading the Short Listing and any referenced IRs

Shtolo reads the Short Listing by using the NIST EXPRESS Toolkit. The toolkit reads in the Short Listing, storing it in internal memory in a form where the semantics are directly accessible.

The Toolkit resolves references (USE and REFERENCE) to external schemas. In particular, any referenced IRs are also loaded into the model. If there are any unresolved references or any other semantic errors, the toolkit notes this so that Shtolo can halt processing.

If toolkit processing is successful, Shtolo begins the next step.

2 Creating a new model representing the Annotated Listing

An empty model is created. Objects are copied from the model representing the Short Listing to the new model. The objects are found by traversing the entire model. A second pass is made to prune objects that have been copied but are not actually used.

§Creating a New Model Representing the Annotated Listing on page 5 describes this step in more detail.

Once the new model has been built, Shtolo begins the last step.

3 Formatting and printing the Annotated Listing

The last step is for the newly created model to be formatted and printed. The NIST EXPRESS Pretty Printer (Exppp) Toolkit [16] traverses the objects in the model, printing them to an output file.

The Supplementary Directives [17] are followed. For example, objects are indented properly and grouped according to type and alphabetized.

The resulting Annotated Listing consists of a file containing a single schema. The schema is named identically to the original Short Listing schema. The file is named after the schema name with the suffix ".exp".

For example, if the Short Listing schema was named "shell_based_wireframe_aic", the Annotated Listing is stored in the file "shell_based_wireframe_aic.exp". The Exppp Toolkit takes precautions to avoid overwriting files that it did not construct.

Further information on this section are presented in [16].

Creating a New Model Representing the Annotated Listing

Clause 11.4 of the EXPRESS Language Reference Manual (LRM) describes the objects which are visible with respect to USE and REFERENCE statements. This accurately describes the objects which must appear in the Annotated Listing. A number of other transformations are also made, which will be described later.

Clause 11.4 is represented as a set of rules describing the outcome rather than an algorithm. Our algorithm consists of four steps.

1 Creation

An empty model is created. A single schema is created within the model. This model and schema will eventually describe the Annotated Listing.

2 Adding and Rewriting

All objects that appear in the Short Listing (i.e., not through USE or REFERENCE) are added to the new schema. All objects referenced through USE clauses are added to the new schema.

If any objects in the new schema are defined in terms of objects not present in the new schema, the objects not present are added to the new schema. This process recurses until there are no objects in the new schema that are not defined in terms of other objects in the new schema. This process involves stepping through the complete definition of every object. For example, every scope is searched. Every statement in every scope is searched. Every expression in every statement in every scope is searched.

In some cases, no searching is done. For example, the subtypes of entities are not examined for definitions. This conforms to section 4.11 of the EXPRESS LRM.

As expressions are examined, any literal strings are examined for potential rewriting. In particular, substrings that reference schemas are rewritten to describe the new schema name. See §Rewriting String Literals on page 7 for more information.

3 Pruning

A number of objects are added to the new schema in the searching phase which have to be removed. For example, select types can be referenced but not all the select items may be used. In this case, the unused items are deleted. Without this pruning, the items themselves will appear in the Annotated Listing even though they are not used. Entity subtypes and rule parameter lists are also pruned for the same reason.

Entity subtypes require a fair amount of work because the supertype expression may have to be rewritten. For example, removing the entity X from the supertype expression (W AND ONEOF (X, (Y ANDOR Z))) eliminates the need for ONEOF. The resulting expression is (W AND (Y ANDOR Z)). Expressions that collapse entirely are removed. No effort is made to simplify semantic redundancies such as expressions that end up as (W AND W).

Shtolo prints an error if it detects a select type which has had all of its items pruned or a rule which has had all of its parameters pruned. There is no way of syntactically formulating either of these. Indeed, they indicate an obvious type of modelling error since they can only occur if the type or rule is explicitly defined in the Short Listing but is not actually used by anything.

Open Issues

A number of issues remain unresolved or have been solved but in unsatisfying ways. We encourage others to focus on these issues.

Comments

The NIST EXPRESS Toolkit strips comments (or "remarks" as they are called by EXPRESS) while reading in schemas. We have not made any attempt to recover comments. The rationale for this is described at length in [16]. To reiterate, the primary difficulty is with context between objects. Objects are sorted first by group and then alphabetically. It is not clear how to associate comments with objects once this regrouping and sorting has occurred. The comments may not even make sense. Consider a comment such as "(* all objects after this point were added by Meredith *)". Another example, is when the context of a comment is pruned. Since comments have no semantic value in EXPRESS, it is not possible to tell whether the comment is part of that which is pruned or not.

Preserving Case and Case Sensitivity

In EXPRESS, the case of letters in identifier names are not significant. For convenience in string comparisons made internal to the software, all identifiers are referenced as lowercase. The original case is discarded.

On output, all user identifiers are printed in lower case. Users may not appreciate this. It is possible to preserve case by using additional memory to store this information, however this is not currently seen as important (enough).

Rewriting Subtype Expressions

As noted in §Pruning on page 6, supertype expressions may be rewritten due to subtype pruning. Simple rewritings are supported. For example, a binary expression that has had an operand pruned is replaced with its remaining operand. If no operand remains, the expression disappears entirely.

No attempt at more sophisticated simplification is made. For example, a subtype of (A AND A) could obviously be rewritten as A. However, the general case is not as trivial. For example, consider the expression (ONEOF (A AND B), (B AND A)). In the general case, semantic analysis of the expression is required using the algorithms in the EXPRESS LRM. Since this is currently subject to debate, this aspect of processing has been deferred.

Rewriting String Literals

As noted in §Adding and Rewriting on page 5, string literals may be rewritten. The intent is that objects originally defined in other schemas may have references to the schema name itself in the string literal. The following subexpression is a typical example, taken from AP201 [18]:

```
. . . NOT (SIZEOF (USEDIN (a, 'EXPLICIT_DRAUGHTING.' +
                              'DRAUGHTING_APPROVAL_ASSIGNMENT' +
                              'ASSIGNED_APPROVAL'))>0)
```

The expression is testing whether a fully qualified attribute is used in an entity. The attribute is passed as a string which literally includes the schema name. Shtolo replaces the string EXPLICIT_DRAUGHTING with a new string representing the schema name of the Annotated Listing.

This is fraught with peril. No run-time evaluation of expressions is possible in this context. So there is no guaranteed way of locating all schema names. For example, schema names may be built out of component parts or even single letters. In this case, the schema name simply will not resemble a schema name in any form. It is equally possible to misinterpret string literals as schema names when they are not intended to be.

Shtolo uses some simple heuristics that are usually (but not always) successful. Each string is examined for a ".". If a dot is found, the prefix before the first dot is compared against all known schema names. The known schema names are the set of IR schema names that the Short Listing references (directly and indirectly). If the prefix matches a schema name, it is considered to be a schema name and is rewritten.

A number of suggestions were made as to additional heuristics. For example, some AP developers always end schema names with "_SCHEMA", "_AIC", or some other convention. However, these heuristics seem much riskier and are not used.

Future Changes and Enhancements

As should be obvious from the previous section, there are a number of open issues that could stand further research. However, the current users of Shtolo seem to be satisfied with the functionality in this regard of the current operation of Shtolo.

Nonetheless, we plan to make further changes to Shtolo. One example is described here.

Shtolo does not currently record the schema where objects originate. We would like to be able to examine an Annotated Listing which includes this information. We believe this would be helpful to tracking requirements and showing that the original requirements have been maintained or satisfied.

It is not clear what form these additional annotations would be because there is no support for this in EXPRESS, the LRM, or the Supplementary Directives, other than using some ad hoc comments. We will probably try a number of different formats. We are particularly interested in a form that allows the Annotated Listing to be browsed in an intelligent way, such as with WWW [19].

Obtaining Shtolo

Shtolo can be ftp'd from ftp.cme.nist.gov as pub/step/npttools/shtolo.tar.Z. See below for e-mail delivery. Installation is described in the README file. Shtolo depends on the Exppp Toolkit (see below) and the NIST EXPRESS Toolkit.

The Exppp Toolkit depends on the NIST EXPRESS Toolkit. The NIST EXPRESS Toolkit can be ftp'd from ftp.cme.nist.gov as pub/step/npttools/exptk.tar.Z. The associated installation documentation can be ftp'd as pub/step/nptdocs/exptk-obtaining-installing.ps.Z. The Exppp Toolkit can be ftp'd as pub/step/npttools/exppp.tar.Z. Installation is described in the README file.

Any of these files can be sent by e-mail. For example, to request e-mail delivery of the Exppp Toolkit, mail to "nptserver@cme.nist.gov". The contents of the message should be (no subject line), "send pub/step/npttools/exppp.tar.Z". Other files may be requested by modifying the message appropriately.

For Support or More Information

Contact the Factory Automation Systems Division – National PDES Testbed (1-301-975-3386 or **`npt-info@cme.nist.gov`**) for more information about the software in general, or other NIST projects at the National PDES Testbed.

This software is a research prototype, intended to spur development of commercial products. Most of the system is available in source form and you are encouraged to obtain and experiment with it. If you have questions and/or problems, you may send e-mail to the shtolo@cme.nist.gov.

Acknowledgments

This work was funded by the NIST Scientific and Technical Research Services as part of the ARPA Persistent Object Base project, and the Department of Defense Continuous Acquisition and Life-Cycle Support (CALS) Program as part of the Application Protocol Development Environment (APDE) project.

Thanks to Steve Clark for explaining clause 11.4 of the LRM (several times) until it finally penetrated the author's thick head. The author gratefully acknowledges Diane Allen (Northrop Corp), Allison Barnard Feeney, and the rest of the AP 202 Team who were very patient and understanding while trying to use early versions of Shtolo.

Thanks to Jim Fowler, Lisa Phillips, and Diane Allen for significant improvements to the content and style of this paper.

Disclaimers

Trade names and company products are mentioned in the text in order to adequately specify experimental procedures and equipment used. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

References

[1] Mason, H., ed., "Industrial Automation Systems – Product Data Representation and Exchange – Part 1: Overview and Fundamental Principles", Version 9, ISO TC184/SC4/WG PMAG Document N50, December 1991.

[2] Libes, Don, "The NIST EXPRESS Toolkit – Introduction and Overview", NISTIR 5242, National Institute of Standards and Technology, Gaithersburg, MD, September 1, 1993.

[3] Libes, Don, and Fowler, Jim, "The NIST EXPRESS Toolkit – Requirements", NISTIR 5212, National Institute of Standards and Technology, Gaithersburg, MD, June 9, 1993

[4] Libes, Don, "The NIST EXPRESS Toolkit – Design and Implementation", *Proceedings of the Seventh Annual ASME Engineering Database Symposium*, San Diego, CA, August 9-11, 1993.

[5] Libes, Don, and Clark, Steve, "The NIST EXPRESS Toolkit – Lessons Learned", *Proceedings of the 1992 EXPRESS Users' Group (EUG '92) Conference*, Dallas, Texas, October 17-18, 1992.

[6] Libes, Don, "The NIST EXPRESS Toolkit – Obtaining and Installing", NISTIR 5204, NISTIR National Institute of Standards and Technology, Gaithersburg, MD, September 1, 1993.

[7] Libes, Don, "The NIST EXPRESS Toolkit – Using Applications", NISTIR 5206, National Institute of Standards and Technology, Gaithersburg, MD, June 9, 1993

[8] Libes, Don, "The NIST EXPRESS Toolkit – Programmer's Reference", National Institute of Standards and Technology, Gaithersburg, MD, to appear.

[9] Libes, Don, "The NIST EXPRESS Toolkit – Creating Applications", National Institute of Standards and Technology, Gaithersburg, MD, to appear.

[10] Libes, Don, "The NIST EXPRESS Toolkit – Updating Existing Applications", NISTIR 5205, National Institute of Standards and Technology, Gaithersburg, MD, June 9, 1993.

[11] Clark, S.N., "The NIST Working Form for STEP", NISTIR 4351, National Institute of Standards and Technology, Gaithersburg, MD, November 1990.

[12] Clark, S.N., “NIST STEP Working Form Programmer’s Reference”, NISTIR 4353, National Institute of Standards and Technology, Gaithersburg, MD, November, 1990.

[13] ISO/CDC 10303-202, Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 202: Application Protocol: Associative Draughting, International Organization for Standardization, February 2, 1993.

[14] Spiby, P., ed., “ISO 10303 Industrial Automation Systems – Product Data Representation and Exchange – Part 11: Description Methods: The EXPRESS Language Reference Manual”, ISO DIS 10303-11:1992(E), July 15, 1992.

[15] Libes, Don, “The NIST EXPRESS Server: Usage and Implementation”, to appear.

[16] Libes, Don, “Exppp: An EXPRESS Pretty-Printer”, NISTIR 5292, Gaithersburg, MD, October 8, 1993.

[17] Shaw, Nigel, “Supplementary directives for the drafting and presentation of ISO 10303 Version 1.0”, ISO TC 184/SC 4 Editing N-22 Editing Committee, June 1993.

[18] ISO/DIS 10303-201, Industrial Automation Systems and Integration – Production Data Representation and Exchange – Part 201: Application Protocol: Explicit Draughting, TC 184/SC4 N219, September 10, 1993.

[19] T.J. Berners-Lee, R. Cailliau, J-F Groff, B. Pollermann, CERN, “World-Wide Web: The Information Universe”, published in “Electronic Networking: Research, Applications and Policy”, Vol. 2, No 1, pp. 52-58, Meckler Publishing, Westport, CT, USA, Spring 1992.