The SC4 Short Names Registry[*][†]

Joshua Lubell
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Building 220, Room A127
Gaithersburg MD 20899 USA
Email: lubell@cme.nist.gov

Sixth International EXPRESS User Group Conference

September 30, 1996

Abstract

This paper describes a software environment recently implemented for maintaining a registry of unique short names for each of the entity data types within each of the EXPRESS schemas in the ISO TC184/SC4 standards. The new environment replaces an inefficient application and has already provided time savings for both the registry's administrator and editors of SC4 standards. It is also improving the quality of SC4 standards. The new SC4 Short Names Registry resides on a server at NIST and uses an SQL database engine in conjunction with the NIST EXPRESS Toolkit. Two applications have been developed for accessing the Short Names Registry. An application for end users that is accessible using the World Wide Web allows individuals to view SC4 part summary information and entity long names and short names in a variety of useful formats. An application for database administrators supports the insertion, deletion, and modification of SC4 parts.

Keywords: CGI; World Wide Web; STEP; TC184/SC4; product data exchange; SQL; HTML; Tcl; EXPRESS

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

[†]Development of the SC4 Short Names Registry is funded by the National Institute of Standards and Technology's Systems Integration for Manufacturing Applications (SIMA) program, a part of the US government's High Performance Computing and Communications initiative.

1 Background

TC184/SC4 is a subcommittee of the International Organization for Standardization (ISO) working in the area of representation and exchange of digital product data. The National Institute of Standards and Technology (NIST) serves as the SC4 Secretariat, which means that NIST is responsible for SC4's technical and administrative leadership. The *St*andard for the *E*xchange of *P*roduct Model Data[ISO1] (STEP[1]), the largest SC4 project, is a family of standards describing product data for all applications over a product's expected life cycle. Other families of standards are being developed within the Parts Library (PLIB[2]) and Manufacturing Management Data (MANDATE) projects. A significant implementable portion of SC4 standards is specified using EXPRESS[ISO11], an information modeling language defined in part 11 of STEP. Part 21 of STEP specifies a file-based mechanism for transferring product data represented in EXPRESS from one computing system to another[ISO21]. Implementors of SC4 standards commonly use the term *exchange file* to refer to this mechanism.

NIST's Manufacturing Engineering Laboratory maintains a database of unique short names for each of the entity data types[3] within each of the EXPRESS schemas in the SC4 standards. This database and the applications for accessing it are collectively referred to as the *SC4 Short Names Registry*, or simply the *Short Names Registry*. SC4 standards are required to include a unique mnemonic short name of six characters or less for each EXPRESS entity long name. Part 21 in turn requires that all exchange files containing data populating an SC4 EXPRESS schema must use that schema's short names as specified in the SC4 standard within which the schema is defined.

The idea of having short names for EXPRESS entities originated in the Implementation Specifications working group (WG7) of SC4. Short names were introduced because some SC4 standards have very long names for their EXPRESS entities. WG7 was worried that the use of these names would lead to very large exchange files, as these names may occur very often in the file. Specification of short names for EXPRESS entities gives SC4 standards developers the opportunity to use meaningful long names in their EXPRESS models, but without the penalty of then having to use these long names in the exchange file.

Until recently, adding an SC4 part to the Short Names Registry was a tedious and error-prone process. The database administrator (DBA) first ran a program to extract the schema names and entity long names from the part's EXPRESS. Because this program was buggy, sometimes the result had to be massaged using a text editor. Next the DBA used an antiquated application to enter the name of the file containing the schema names and

[1] Officially ISO 10303—Product Data Representation and Exchange.

[2] Officially ISO 13584—Parts Library Data Representation.

[3] An *entity data type*, as defined in part 11, represents "a class of information defined by common properties." Exchange files contain entity data type instances.

long names, as well as other information about the part, into the database and to generate short names. This application's user interface was clumsy, and it did not validate the EXPRESS syntax. Also, because the database engine was intended for large collections of data, it was a mismatch for the relatively small size of the Short Names Registry[4]. Furthermore, because there was no database client available to users outside of NIST, SC4 part editors had to contact the DBA every time they needed to look at a part's short names. As a result of all these shortcomings, the DBA spent an inordinate amount of time performing updates, fixing errors, and generating reports.

In the spring of 1996, NIST started receiving requests to generate short names for PLIB standards. Since the old Short Names Registry's data model only accommodated STEP, it was necessary to modify the data model and application program to support other SC4 standard families. Because of the old Short Names Registry's aforementioned deficiencies, we decided to build a completely new system rather than to patch the old one.

This new SC4 Short Names Registry has been in operation since August of 1996. It uses mSQL (also known as Mini SQL)[MSQL], a lightweight SQL[ULLMAN] database engine. An application program with a modern user interface and built-in EXPRESS validation is used for database administration and updates. In addition, part editors and implementors of SC4 standards can view part summary information and entity long names and short names in a variety of useful formats using the World Wide Web. As a result of these improvements, the DBA now spends far less time than before processing short name generation requests, maintaining the database, and generating reports. The remainder of this document discusses the design, implementation, and operation of the new SC4 Short Names Registry.

2 Design and Implementation

The Short Names Registry resides on a server at NIST where the mSQL database engine runs as a background process. Data is stored in four main tables. The first table contains information about each ISO SC4 part in the database such as its ISO number (usually 10303), part number, version, date entered, and any additional notes. The second table contains the schemas for each part. The third table contains the entity long names for each schema. The fourth table contains the short names for each entity long name. Since EXPRESS entity names are case insensitive, all long and short names in the database are represented in upper case. All SC4 standards share the same name space for schemas and entities. Thus, no two schemas are allowed to have the same name, and no two entities are allowed to have the same long name. However, entities defined in STEP Integrated

[4]The current database size is about a megabyte and is not projected to get any larger than five megabytes or so.

Resource[5] schemas can be re-used elsewhere in schemas belonging to other SC4 parts.

Figure 1 illustrates the relationships between the four tables. For every entry in the Part Information table, there are entries in the Schemas table for each schema belonging to the part[6]. Thus there is a many-to-one relationship between the Schemas table entries and the Part Information table entries. Similarly, because every schema has multiple entity long names, there is a many-to-one relationship between the Long Names table entries and the Schemas table entries. There is, however, a one-to-many relationship between the Short Names table entries and the Long Names table entries. This is because entities defined in the Integrated Resources can be used in other schemas as well, resulting in the possibility of a long name being in more than one entry in the Long Names table.

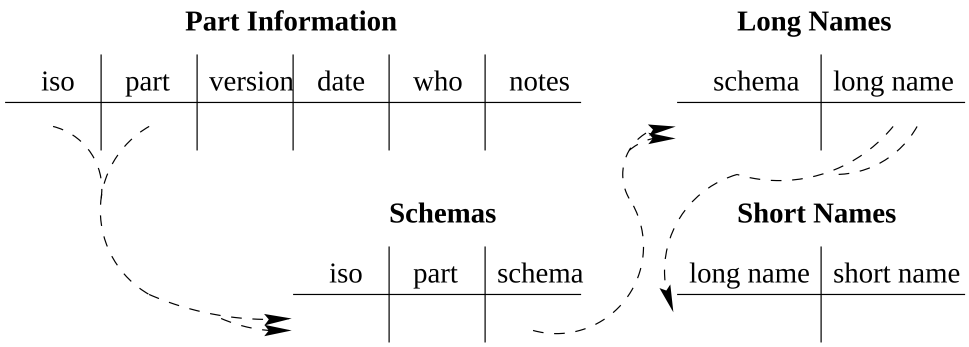

Figure 1: Short Names Registry data model.

Four "working" tables, each corresponding to one of the main tables, are used to store data from a new SC4 part while the part is being added to the database. That way, if an error should occur during database modification, the data in the permanent tables is not corrupted. Prior to the addition of any new part, the working tables are cleared of any leftover data from the previous database insertion operation.

The application for database administration and the web interface for generating reports are implemented using the Tcl/Tk[OUST] scripting language and make use of a Tcl extension for accessing mSQL databases[7]. The database administration application supports the insertion, deletion, and modification of parts. The DBA adds a new part to the database by specifying the part information on a form and specifying the file containing the part's EXPRESS. The EXPRESS is validated, and the schema names and entity long names are extracted. A Tcl extension for interpretive control of the NIST

[5] Integrated Resource standards are used as building blocks for developing other SC4 standards.

[6] Note that the ISO number and part number are both needed to uniquely specify a part.

[7] Tcl extensions with a similar command syntax exist for other SQL database engines as well. Thus, mSQL could be replaced by another SQL database engine with minimal code rewriting.

EXPRESS toolkit[LIBES95] is used to perform the syntax checking and name extraction. Short names are generated using an algorithm that attempts to produce a unique and mnemonic identifier of six characters or less for the entity long name[8].

The web interface is implemented in Tcl and uses the Common Gateway Interface (CGI) standard for interfacing external applications with web servers[BLEE]. CGI scripts, written in Tcl, generate Hypertext Markup Language (HTML) dynamically in response to requests for web pages. These Tcl scripts use a Tcl CGI library[LIBES96B] for reading input from HTML forms and generating reports in HTML for display on the web.

Figure 2 shows the relationship between the database, the database administration application, and the web interface. The SQL Tcl extension, the Tcl extension for the NIST EXPRESS Toolkit, and the CGI Tcl library all form a Tcl binding around the core of the system. The database administration application and the web interface's CGI scripts interact with the SQL database through this Tcl wrapper. The database, Tcl binding, and CGI scripts all reside on a web server. The application for updating the database, intended to be used only by the DBA, is not accessible through the web server, but it is connected to the database and Tcl binding through a local area network. Developers and users of SC4 standards anywhere in the world can read from the database (but not write to it) through the CGI scripts on the web server. Internet access and a web browser is all that is required.

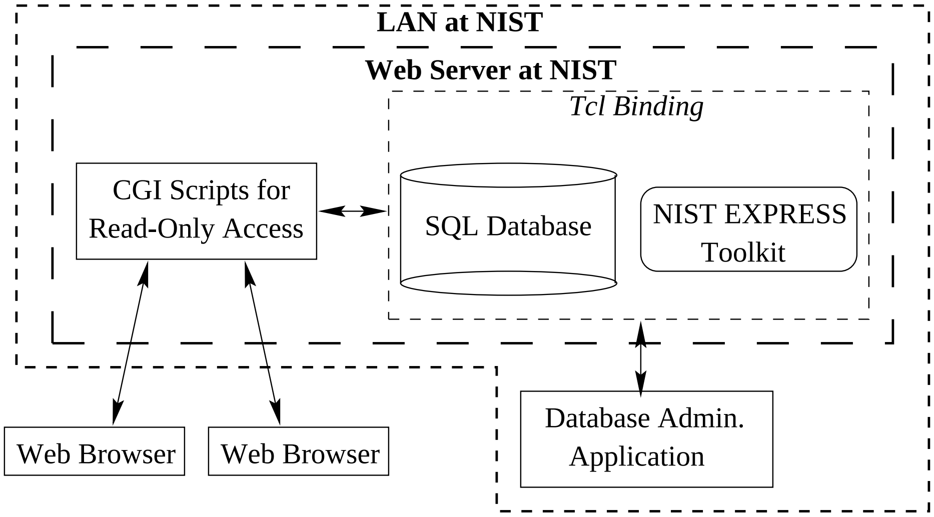

Figure 2: Architecture of the Short Names Registry and its applications.

[8]This algorithm is similar to the one used for the old Short Names Registry. The old algorithm is documented in the old Short Names Registry application's source code.

The short name generation algorithm employs multiple techniques for creating short names. It begins by converting the long name into a string of up to six characters using the following procedure (omitting some of the details):

1. Determine the number of substrings separated by underscore characters in the entity long name,
2. Divide the number of substrings into six to get the number of short name characters per substring (if there are more than six substrings, assume one short name character per substring),
3. For each long name substring (use only the first six substrings if the long name has more than six substrings), create a short name substring using the leftmost characters, but not including any vowels other than the beginning character,
4. Concatenate the short name substrings together to form a short name for the entity long name.

After generating the short name, the algorithm verifies its uniqueness with respect to the short names already in the database. If the generated short name is unique, then the algorithm is finished. Otherwise, the algorithm adopts a brute-force strategy for generating a unique but non-mnemonic short name. This strategy consists of taking the first three alphanumeric characters in the entity long name and trying successive suffixes until the resulting string is unique with respect to the short names already in the database. The algorithm first tries the substrings "0" through "999", a total of 1000 possibilities. If none of these work, the algorithm then tries successive three-letter substrings, a total of 17576 possibilities. Since there are over 18000 potential short names to choose from for each entity long name, it is highly unlikely that this algorithm will ever fail in creating a unique short name. However, if such a failure should ever occur, it would be a simple matter to add additional strategies to the algorithm.

Since entity names in SC4 Express schemas tend to be long and to contain one or more underscore characters, the algorithm should almost always succeed in creating short names that are reasonably mnemonic, given the six character limit imposed by SC4. In fact, the algorithm did not have to resort to brute-force techniques to generate any of the more than 1500 short names currently in the database. Figure 3 provides an example of the short names created by the algorithm for one of the Express schemas in STEP.

Because some SC4 schemas contain hundreds of entities, it can take as long as several minutes to obtain a part's entity long names and short names from the database. In order to avoid such delays, the web interface to the Short Names Registry uses a cache. This cache is simply a directory on the server's file system containing a file of long name and short names corresponding to each schema that has been requested from the database. Whenever a user requests to see a schema's long and short names, or the long and short

Short names of entities for METHOD_DEFINITION_SCHEMA

Entity Name	Short Name
ACTION_METHOD_TO_SELECT_FROM	AMTSF
ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS	AMWAD
ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS_CONSTRAINED	AMWADC
CONCURRENT_ACTION_METHOD	CNACMT
CONTEXT_DEPENDENT_ACTION_METHOD_RELATIONSHIP	CDAMR
CONTEXT_DEPENDENT_ACTION_RELATIONSHIP	CDAR
PROCESS_OR_PROCESS_RELATIONSHIP_EFFECTIVITY	POPRE
RELATIONSHIP_CONDITION	RLTCND
SEQUENTIAL_METHOD	SQNMTH
SERIAL_ACTION_METHOD	SRACMT

Figure 3: Short names for METHOD_DEFINITION_SCHEMA from STEP part 49.

names for an entire SC4 part, the web interface first tries to read the data from the cache. If the data requested is not already in the cache, the database is accessed to service the user's request, and the result is written to the cache. When the DBA deletes an SC4 part from the database, all schema files for that part are deleted from the cache. Caching of long names and short names significantly speeds up response time when generating reports, particularly for EXPRESS schemas containing a large number of entities. The cache takes up fairly little disk space (under 200K for the current database).

3 Operation

The best way to illustrate how the Short Names Registry operates is through sample user sessions. The following subsections show how an end user would generate reports using the web interface and how the DBA would use the database administration application to add a new SC4 part to the database.

3.1 Using the Web Interface to Generate Reports

The entry point to the Short Names Registry's web interface is an HTML document, generated dynamically by a CGI script, containing a form for choosing an SC4 part to browse from a scrolling list. The form, shown in Figure 4, lists all of the SC4 parts currently in the database sorted by ISO number and part number. The CGI script issues an SQL query to the database in order to obtain this information.

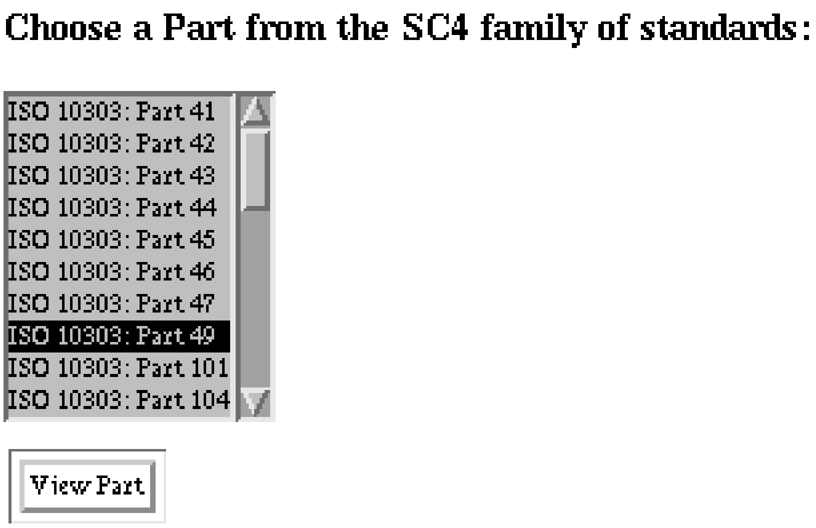

Figure 4: Form for choosing an SC4 part.

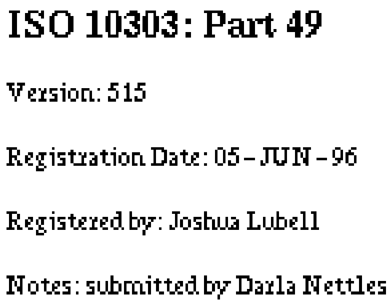

Figure 5: Information for STEP part 49.

Suppose the user chooses to view `ISO 10303:  Part 49` from the scrolling list on the form in Figure 4. This information then gets passed to a second CGI script which uses it to issue database queries to determine STEP part 49's registration information and Express schema names. The CGI script then uses this data to dynamically create an HTML document containing STEP part 49's registration information (Figure 5), as well as two forms for choosing how to view the entity short names.

The first form, shown in Figure 6, allows users to view the entity short names for the entire part in either comma-delimited ASCII format or as Standard Generalized Markup Language (SGML)[GOLD], an international standard for describing the structure of documents for use in text processing applications. Both of these formats are particularly useful to STEP part editors. The comma-delimited ASCII format is easy for a computer program to read and can be used to generate a text file to be included with the part's Express listing diskette, as per SC4's documentation directives[WELL]. The SGML format consists of the part's Short Names Annex, another documentation requirement, tagged using SGML document type definitions (DTDs) designed especially for STEP[PHIL].

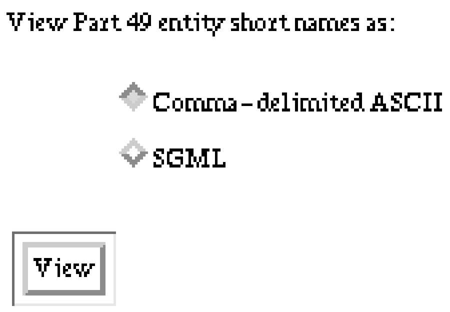

Figure 6: Form for viewing short names as comma-delimited ASCII or SGML.

The STEP DTDs are used in NIST's Application Protocol Development Environment (APDE)[LUB96B], an integrated suite of software tools for accelerating the development and deployment of STEP standards. The APDE uses a central information repository, the AP Information Base (APIB), which contains SGML-tagged STEP documents as well as other information useful to the STEP community. SGML documents in the APIB are indexed for efficient access using an SGML text retrieval engine. A web interface for browsing and searching these SGML documents is available to people developing STEP standards[LUB96A]. STEP Application Protocol[9] and Integrated Resource authors are being encouraged to submit their parts to SC4 as SGML using the STEP DTDs so that their documents can be:

- used in the APDE (and possibly by other SGML applications in the future),

[9] Application Protocols are standards intended to be implemented in software systems. They use Integrated Resources as building blocks.

- shared across multiple computing platforms without being tied to a single proprietary word processor format,
- validated using an SGML parser to help ensure conformance to the documentation guidelines.

The second form, shown in Figure 7, allows users to view the the entity short names for a schema in a human-readable tabular format. Figure 3 shows this format for the entities in `METHOD_DEFINITION_SCHEMA`.

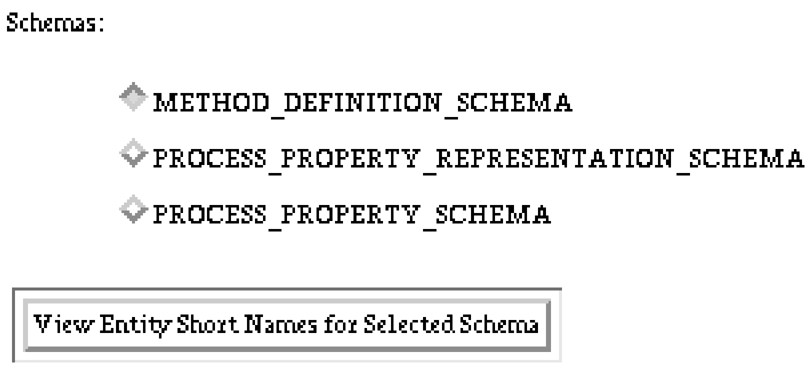

Figure 7: Form for viewing a schema's short names in tabular format.

Users can download reports generated using either of these two forms by using their web browser's "Save" command to save the report to a file. The result should be saved as text rather than HTML so that no HTML tags appear in the saved file.

3.2 Database Administration

The database administration application has as its user interface a series of forms for performing various database operations. The operations currently supported are:

- adding a new SC4 part,
- deleting an SC4 part,
- extracting a part's entity long names without modifying the database[10],
- modifying an SC4 part's registration information.

[10] This is useful when determining whether changes to a part's EXPRESS affect the part's entity long names.

Additional operations will be added to the user interface as needed. One such operation likely to be added soon is modification of individual short names.

Figure 8 shows the form for entering a new SC4 part. In the scenario shown, the DBA has entered information needed to add part 20 from ISO 13584 (PLIB). The " EXPRESS file" field contains the location of the file containing part 20's EXPRESS code. If the DBA presses the "Generate" button, the application adds the part's registration information to the database's working tables. Then it validates the EXPRESS code's syntax using the NIST EXPRESS Toolkit. If the EXPRESS is syntactically invalid, the application displays an error message to the DBA and aborts the operation. If the EXPRESS is valid, then the application uses EXPRESS Toolkit functions to extract the schema names from the EXPRESS file and, for each schema, extract its entity long names and generate short names for the long names using the algorithm discussed in Section 2. The application displays status messages (not shown in Figure 8) throughout the data generation process to let the DBA know what is happening.

Figure 8: User interface for entering a new SC4 part.

Schema names and entity long names are written to the database's working tables as they are extracted from the EXPRESS, and short names are written to the working tables as they get generated. If these extraction and generation processes complete successfully, the working tables' contents are copied to the main tables. If an error occurs, the application displays a modal dialog box containing the error message, and the working table data is not copied. The DBA then has the option of issuing SQL queries using mSQL's terminal monitor program in order to view the contents of the working tables for debugging purposes.

Adding a new part is the most complicated database operation. The other operations the application supports are simpler, but are accomplished using forms similar in style to the one displayed in Figure 8.

4 The Future

Section 3.1 mentioned the APIB, an indexed repository of SC4 parts represented in SGML along with related documentation. Plans are underway to fully integrate the Short Names Registry into the APIB by implementing hypertext linking between the part documents in the APIB and objects in the Short Names Registry. This will enable users of the Short Names Registry web interface to use any part name, schema name or entity name viewable on a web page as a potential entry point into the APIB's browser. For example, a user looking at the table of short names shown in Figure 3 might want to view the definition from STEP part 49 for a particular entity, say `SEQUENTIAL_METHOD`. Suppose the CGI script that generates a schema's short name table in HTML rendered each entity name in the table as an HTML anchor with a reference to the APIB web interface's CGI script for creating an HTML document containing an entity definition[11]. Then the user could click on `SEQUENTIAL_METHOD` in the table of short names, a query would be issued to the APIB to obtain the entity definition, and the SGML-tagged result would be rendered as HTML in the user's web browser. The user could then read the entity description and, if desired, use "backtracking" buttons at the bottom of this HTML page to view the description for the schema containing this entity, the part containing that schema, and so on[LUB96A].

Since the APIB already contains the capabilities required to process queries such as the one just described and display the result as HTML, integration of the Short Names Registry and the APIB should not be difficult. However, the resulting user interface will have to address the following issues:

- Not all parts in the Short Names Registry are in the APIB,
- The part documents in the APIB are under ISO copyright and can only be accessed by developers of ISO standards or people who have purchased the standards.

Therefore, the Short Names Registry's CGI scripts would need to be selective in which objects they link to the APIB. Only those objects whose parts are in the APIB, and which are being requested by people allowed to access ISO-copyrighted documents, should be rendered as HTML anchors.

[11] The reference's URL would have to contain a query string containing the arguments the APIB's text retrieval engine needs to obtain the SGML-tagged data for `SEQUENTIAL_METHOD`'s definition such as document type (Integrated Resource), part number (49), schema name (`METHOD_DEFINITION_SCHEMA`), etc.

Another improvement being considered for the Short Names Registry is a web interface for SC4 editors to submit requests to generate short names for a part (and update the database accordingly). These requests are currently handled using e-mail and often require a dialog between the submitter and the DBA to iron out problems such as mistakes in the EXPRESS and missing registration information. SC4 editors could instead issue their requests using an HTML form analogous to the form the DBA currently uses for adding a new part[12] (Figure 8), but with a field added for the submitter's e-mail address. A CGI script then verifies that no field is left empty and that the input (including the contents of the EXPRESS file) is syntactically correct. If the data entered on the form is complete and correct, the CGI script then sends an e-mail message to the address entered on the form to verify the SC4 editor's identity and an e-mail message notifying the DBA that a request has been issued to update the database. The DBA then waits for an e-mail reply from the SC4 editor before updating the database. This use of e-mail reception as an authentication mechanism provides reasonable assurance to the DBA that the request to generate short names and update the database is legitimate[LIBES96A].

Yet another possible enhancement is a web interface allowing users to generate their own private short names. This capability would be useful for those who would like short names for their EXPRESS but are not interested in registering their short names in the database. These users would merely send their EXPRESS and desired output format (table, comma-delimited ASCII, etc.) to the Short Names Registry web server using an HTML form and would receive their short names back as an HTML document in the format specified.

5 Concluding Remarks

The new SC4 Short Names Registry takes maximum advantage of modern and robust software tools and technologies such as the NIST EXPRESS Toolkit, Tcl/Tk, and the World Wide Web. These tools and technologies greatly reduce the amount of programming required to build software applications. As a result, the total effort spent developing the new Short Names Registry was less than four staff-months. Although it took a relatively short time to implement, the Short Names Registry has already benefited both NIST and the SC4 community. NIST is now less burdened than before with maintenance tasks and requests from SC4 editors. The SC4 editors can now generate reports in the formats that are most useful to them without having to wait for a response from the DBA. Perhaps most important, the built-in EXPRESS validation has caught the kinds of errors that, in the past, had found their way into the initial releases of SC4 standards. Planned enhancements will increase these benefits further.

[12]This requires that both the web server and the submitter's web browser support the uploading of files from the client to the server.

The author thanks this document's NIST reviewers — Mary Mitchell, Don Libes, and Kathleen Tan — for their comments and corrections. The author also wishes to thank Julian Fowler, Sharon Kemmerer, Jan Van Maanan, and Peter Wilson for assistance in understanding SC4's requirement for entity short names, and Newton Breese for help with using the old Short Names Registry application.

The URL for accessing the SC4 Short Names Registry is
http://www.cme.nist.gov/cgi-bin/apde/sc4short.tcl.

6 Author Biography

Joshua Lubell designs and implements software systems for product data exchange applications at the National Institute of Standards and Technology in Gaithersburg, Maryland, USA. His technical interests include SGML, database technology, and Internet computing. His previous experience includes artificial intelligence systems design and prototyping, software development for the building materials industry, and knowledge engineering. He has an M.S. in computer science from the University of Maryland at College Park, where he performed graduate research in diagnostic problem solving, and a B.S. in mathematics from the State University of New York at Binghamton. He lives in Olney, Maryland with his wife and sixteen-month-old son, who has invented his own (undocumented) algorithm for mapping long names to short names.

References

[BLEE] Tim Berners-Lee and Daniel Connolly, *Hypertext Markup Language - 2.0*, Internet Engineering Task Force, HTML Working Group, RFC 1866, November 1995.
URL: ftp://ds.internic.net/rfc/rfc1866.txt

[GOLD] Charles Goldfarb, *The SGML Handbook*, Oxford University Press, 1992.

[ISO1] International Organization for Standardization, *Industrial automation systems and integration—Product data representation and exchange—Part 1: Overview and fundamental principles*, ISO 10303-1, 1994.

[ISO11] International Organization for Standardization, *Industrial automation systems and integration—Product data representation and exchange—Part 11: EXPRESS language reference manual*, ISO 10303-11, 1994.

[ISO21] International Organization for Standardization, *Industrial automation systems and integration—Product data representation and exchange—Part 11:*

Implementation Methods: Clear text encoding of the exchange structure, ISO 10303-21, 1994.

[LIBES95] Don Libes and Steve Clark, *An Object Oriented Tcl-Tk Binding for Interpreted Control of the NIST EXPRESS Toolkit in the NIST STEP Application Protocol Development Environment*, Proceedings of *EUG'95 (EXPRESS User's Group) Conference*, Grenoble, France, October 1995.
URL: http://www.nist.gov/msid

[LIBES96A] Don Libes, *Authentication by Email Reception*, Proceedings of *Fifth System Administration, Networking, and Security Conference (SANS 96)*, Washington DC, May 1996.
URL: http://www.nist.gov/msid

[LIBES96B] Don Libes, *Writing CGI Scripts in Tcl*, Proceedings of *Tcl/Tk Workshop 96*, Monterey, CA, July 10-13, 1996.
URL: http://www.nist.gov/msid

[LUB96A] Joshua Lubell, *The Application Protocol Information Base World Wide Web Gateway*, National Institute of Standards and Technology, NISTIR 5868, July 1996.
URL: http://www.nist.gov/msid

[LUB96B] Joshua Lubell and Lisa Phillips, *SGML Application Development: Choices and Tradeoffs*, Proceedings of *SGML'96*, Boston, MA, November 1996.
URL: http://www.gca.org

[MSQL] Hughes Technologies, *Mini SQL*, version 1.0.16, 1996.
URL: http://Hughes.com.au/

[OUST] John Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, 1994.

[PHIL] Lisa Phillips, *STEP Document Type Definitions: Design and Architecture*, National Institute of Standards and Technology, to appear as a NISTIR, 1996.
URL: http://www.nist.gov/msid

[ULLMAN] Jeffrey Ullman, *Principles of Database and Knowledge-Base Systems*, Computer Science Press, 1988.

[WELL] Joan Wellington, *Supplementary directives for the drafting and presentation of ISO 10303*, ISO TC184/SC4 Technical Report, N432, April 5, 1996.
URL: http://www.nist.gov/sc4/howto/methods/supp_dir/