DRAFT, September 1994

Master's Thesis

# SLIC
# Standard Language for Implementation Conventions

Alex Th. de Jong
dejong@snad.ncsl.nist.gov

Tele-Informatics and Open System Group
University of Twente, The Netherlands

Distributed Applications Research and Technologies Group
Systems and Network Architecture division
National Institute of Standards and Technology, U.S.A.

## 1 Electronic Data Interchange within NIST

NIST has developed an Electronic Data Interchange (EDI) tool called Transaction Set Development System(TSDS), which is able to define and process X12 or EDIFACT interchanges. To define a Transaction Set (or message in EDIFACT) the graphical interface tools of the Definition Environment (DE) can be used. This environment enables a user to fully specify a Transaction Set and verify its compliance with the X12 or EDIFACT standards. To process a Transaction Set, the Operational Environment (OE) is available. The OE consists of specific generated code for Transaction Sets defined within the DE and a generic graphical interface to edit element values. Interchanges can be constructed, sent and received using this tool.

## 2 General problems of EDI Standards

EDI tools available in the marketplace, in general, have functionality similar to the functionality described above. The main differences between the NIST TSDS prototype and the commercially available EDI software are the parser based decoding and database mapping of EDI transaction sets together with a generated Application Programmer Interface (API) (versus conventional table-driven translation). Both the TSDS and commercial products, need to be flexible in order to handle upgrades and changes of the publicly available standards.

Another problem is the wide variety of ways inwhich the standards are used. Within most EDI implementations, standards are adjusted in such a way that they meet specific requirements instead of using the exact standard. EDI product vendors therefore need some kind of mechanism to handle these variations of the standard.

The TSDS uses an intermediate language that can hold an implementation convention. This language (called 'Script') is a simple representation of an EDI transaction set in a machine processible format. The parser together with the language can be used to save and retrieve an implementation convention. Similar mechanisms are used within commercially available products to handle the problem. Unfortunately, every product uses a proprietary technique to solve the problem, which causes incompatibility between the software.

## 3 Standard Language for Implementation Conventions

A Standard Language for Implementation Conventions (SLIC) can reduce a number of problems resulting from the wide variety of changing standards. By providing a standardized implementation convention, EDI vendors only need to provide an interface/parser for this language to update and/or adjust their products.

The language should be flexible and powerful enough to carry different implementations of one standard. A comprehensive constraint syntax should be available to refine an existing standard. The constraints should be context sensitive to account for the fact that within different implementations, segments or elements can be used differently (e.g. a subset of codes can differ in each implementation instance of an identifier data element).

NIST intends to maintain the output of the language (SLIC) as a defacto standard. EDI users will have dial-in (or internet) access to a registry that holds copies of the implementation conventions. To update an EDI product, vendors only need to integrate their code with the publicly available parser and then download the implementation convention. The registry, maintained by NIST and industry, will contain the latest government and industry implementation conventions.

## 4 Project objectives

- Design of a syntax for a language that allows the complete description of EDI implementation conventions in a machine processible format. This phase contains analysis of existing languages and allows industry to give comments.
- Implement a parser for the language that will be used within NIST TSDS. This tool is able to generate code for a parser based decoder according to the language profiles. The parser which will be put in public domain, should be able to function in different software environments, which will enable EDI vendors to use it for their own purposes.
- Integrate parser with the NIST TSDS.

## 5 Schedule

| | | |
|---|---|---|
| September to November 1994 | : | Definition of language (draft version for review) |
| December 1994 | : | Final Language definition |
| January to March 1995 | : | Completion of parser |
| March 1995 | : | Completion of Master's Thesis |
| April to September 1995[1] | : | Integration of parser in the TSDS |

The language definition will take some time in order to get comments from industry, but the first draft should be available for review in November.

## 6 Abbreviations

| | |
|---|---|
| API | Application Programmer's Interface |
| DE | Development Environment (of TSDS) |
| EDI | Electronic Data Interchange |
| NIST | National Institute of Standards and Technology |
| OE | Operational Environment (of TSDS) |
| SLIC | Standard Language for Implementation Conventions |
| TSDS | Transaction Set Development System |

[1]The integration of the parser and defined language within the NIST TSDS will be part of the overall project but not of the Master's Thesis. It is placed here to clarify the use of SLIC.