Transformation Report: The missing Standard for Data Exchange

Roland Eckert[1], Günther Specht[2]
[1]EADS Deutschland GmbH, MT332, 81663 Munich, Germany
Roland.Eckert@m.eads.net
[2]University of Ulm, Department of Databases and Informationsystems, 89069 Ulm, Germany
Specht@informatik.uni-ulm.de

Abstract. The data exchange with STEP ISO 10303 is state of the art, but it is still a fundamental problem to guarantee a given quality of service to integrated operational and informational applications. In STEP there are defined descriptive methods, data specifications, implementation resources and conformance testing, but there is nothing to document how the data is processed. A success report of the mapped data from the source to the target tool is missing. In this paper we introduce a Transformation Report for documenting the data transformation from the source to the target tool. With this report the trustworthiness of the received data can be significantly improved by documenting the data loss, semantic and syntactic errors. With the information in the report it should be possible to infer the proper value to define rules that fix the data after it has been determined to be incorrect or to find a suitable data integrations strategy into a target tool or repository. The intention of the paper is to suggest a standardised Transformation Report, that can be automatically processed and that contains all information for an automated reconciliation process.

1. Introduction

The number of available heterogeneous data sources increases daily. Companies exchange and share information across the country and the world. This has created an increased demand for automated data translation. The primary cause for concern with data translation is to be unaware of what happens during the data transformation. Only restricted techniques are available to handle the challenges of inconsistencies, heterogeneity of data and quality of data.

When the data of such a tool is exported and imported into a target tool, its structure is altered so that semantic concepts in the source schema are represented using the target model's preferred mechanisms for denoting them. Error detection in the target tool is a difficult task if the new delivered data contains many special terms, symbols, formulas, or conventions whose syntactic contributions cannot be established without a complete understanding of the delivered data. This paper presents a Transformation Report that documents the translation of the data from the source to the target. The report is an effective aid for identifying the majority of errors during the data integration of the delivered data into the target system.

Some industrial driven standards for the data exchange are provided by following standards:

- ISO 9506: Manufacturing Message Specification (MMS)
- ISO 10303: Product Data Representation and Exchange (STEP), [ISO, 1994, Kemmerer, 1999]
- ISO 13584: Parts Library (P-Lib)
- ISO 15531: Manufacturing management data exchange (MANDATE)
- ISO 15926: Integration of life-cycle data for oil and gas production facilities (POSC/CAESAR)
- ISO 18629: Process Specification Language (PSL)
- ISO 18876: Integration of Industrial Data for Exchange, Access and Sharing (IIDEAS)
- ISO 9735 Electronic data interchange for administration, commerce and transport (EDIFACT) [ISO 9735, 1990]

These standards ease the data exchange. They suggest a common syntax, a common semantic and a common process for data exchange. Also in these standards it is necessary to translate the transferred data from the source representation to the target representation. All existing standards ignore the fact that, during the data translation, errors can occur or data can get lost. These standards offer no solution for tracing the translation process of the data.

If the user loses 1% of the information without knowing which part, he has to check the whole model. This additional time expenditure reduces the benefit of an automated data exchange dramatically. The data loss is a result of the missing semantic and technical interoperability and also the partially unknown data model of the source tool. These reasons reduce the trustworthiness of the received data significantly.

2. Data Transformation

Errors come up from the different semantics of data schema (semantic issue) of the data sources that we take into account. It is only possible to map (e.g. entities) from the source to the target tool [Fig.2-1], were an equal or suitable structure, at least for a subset of the stored data is available. The optimal cases are equal data models (Equality). The worst case is, were the data models are disjointed (Inequality). One class is the restriction; the target schema is more specific than the source representation. The class generalization is the reverse of the restriction. In the daily use the data models are between equality and inequality (Overlap). The distortion is the generalisation for some information and restriction for other information at the same time.

The established approach for data transformation and integration ignores the data processing before the data integration and looses important information. The general approach ignores also, that the source and the target database can use different naming conventions for identifier. The intension of the strategy and architecture of the suggested Transformation Report and Acknowledgment in this paper is to report all data processing results as early as possible. At every step were the exported data are processed a report is derived. So the error propagation is reduced to an minimum. This reporting starts already at the source until the target and includes also a feed-back, the acknowledgment.

The reporting algorithm [Fig. 2-1.] is divided into two parts, the 'Transformation Report' (R0, R1, R2, R2*) and the 'Acknowledgement' (A1*, A1). The generation of the Transformation Report starts at the source and ends at the target tool. The Acknowledgement' describes which data were actually imported into the repository.
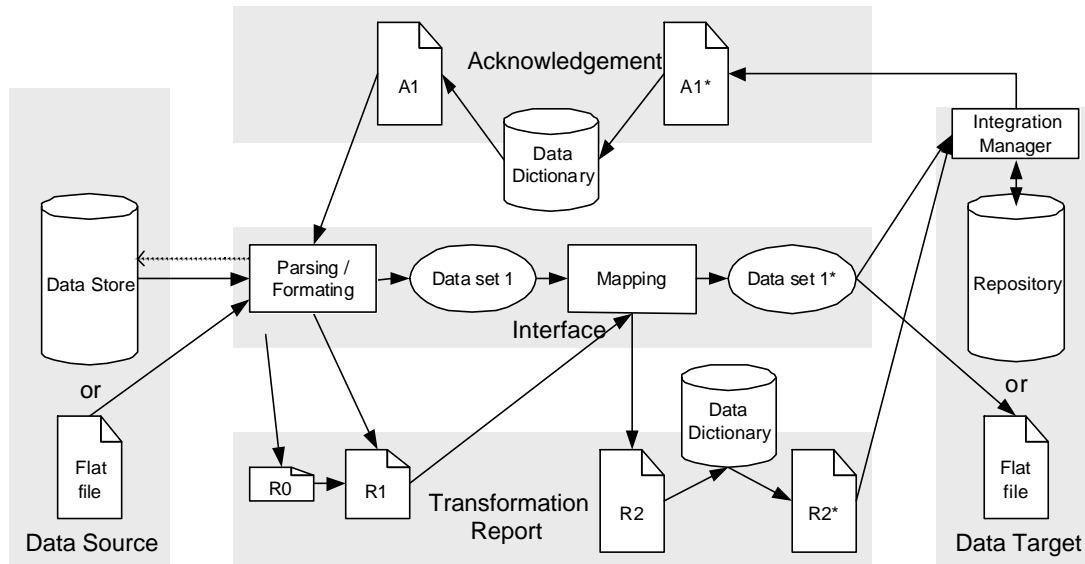
Fig. 2-1 Architecture for gaining the Transformation Report and the Acknowledgement

The transformation process of the data can be divided into a set of processing stages connected by pathways indicating the flow of information from one processing stage to another. The data source and the data target can be proprietary stored data from a tool. Than there is no Integration Manager. The data can also be stored in a repository defined with a standardised data model (e.g. STEP ISO 10303-AP233) or a single file, called flat file, formatted in a suitable format like e.g. ISO 10303-P21 (ASCII), or ISO 10303-P28 (XML) that is e.g. ISO 10303-AP233 conform.

The exporting of the data from the data source starts with the definition of the data set which has to be exported. After having such candidates for export, the first task to be performed is the parsing and formatting section of the data from the data source into an intermediate structure 'data set 1'. This intermediate structure will probably be based on a defined data schema(e.g. application protocol). The next task is to map from the structure of the data schema to the data structure of the destination system. This data set is called 'data set 1*' to signify that it is the same data semantic with different representation. The "Integration Manager" level selects an integration strategy for every entity. An entity can be completely ignored, defined as an initial version, or merged with a version already existing in the repository.

The first available information of the Transformation Report 'R0' contains predefined configuration information of the data source, data target the interface, and Organisation Information. The initial Report 'R0' is extend by a listing of all entities that were exported 'R1'. During the parsing and formatting phase errors, e.g. by referential integrity, syntax errors were discovered. These phenomena are also listed in the Transformation Report. During the mapping phase the entities are separated where it is not possible to map them to the schema of the destina-

tion. Here also semantic errors are discovered and added to the transformation report 'R2'. The suggested technique assumes that source and destination are using different naming conventions for the stored entities. So the entities are translated from one naming convention to another via the central 'Data Dictionary' [Lomax, 1977]. The result of the renaming of the entity identifier is the report 'R2*'. This report 'R2*' together with the 'Data Set 1*' is presented to the Integration Manager.

The delivered data contains several versions for single entities and the data generating source tools use different naming conventions for the entities. For configuration management purposes it is necessary to use an unambiguous identifier for every entity by every involved tool through the life cycle. A data dictionary could fulfil this task. It is an organized list of all the data elements that are pertinent to the system, with precise, rigorous definitions so that both the source and target tool have a common understanding of all inputs, outputs, components of stores. The initiative "Product Life Cycle Support" (http://www.plcsinc.org/) offers an identification mechanism that could be sufficient. These identifiers are organized in the "Data Dictionary".

Data integration into the repository is performed by the integration manager e.g. by overwriting existing entities in the simplest case and generates an 'Acknowledgement' A1*. It documents which entities and which attached files were actually imported. Because of different naming conventions it is necessary to translate A1* to the original naming convention via the data dictionary. The result will be the Acknowledgement A1 that is presented to the loading section inside the parsing / formatting section and a new data export is started. More common is, that the Acknowledgment is presented to an responsible person (engineer) and he corrects the data and starts the data export manually.

The receiver of any message sends a feedback information, "Acknowledgement" about the result of the import process. In case of a successful importing, an acknowledgement message is send to the sender, in the error case, a message send with information about the problems detected. The Acknowledgement has to manage the following conditions:

OK   The Data file successfully imported
Wrong   An error is occurred during import
Wait   The data file is waiting to be processed
Pending   The data fail is waiting to be processed again

It is possible to use the same structure of the Transformation Report for the Acknowledgement report. The only difference is, that there is only the Acknowledgment report and no more data is attached.

Two requirements for the described interface [Fig. 2-1] are the neutrality and the stability of the interfaces:
The first requirement is that the interface only transforms the data from one representation to another representation without correcting the transformed data. In a real world scenario, data is distributed and stored in different tools and repositories. Every tools has his tool specific interfaces. If every interface uses his own strategy for making the data conformant with the data model of the target (e.g. STEP ISO 10303-AP233), we can not guarantee that all data are processed by the same rules. The same data could be interpreted by two different interfaces in two different ways.

The second requirement is, that the interface is robust, also when it discovers a error in the processed data. If the interface discovers an error in the data, like data quality [Redman, 1997; English, 1999; Loshin, 2001; Wang, 2000; Olson, 2003], semantic error, e.g. it denotes the error in the Transformation Report and goes on with the translation.

3. Instantiation of a Transformation Report

For choosing a suitable integration strategy information from different processing steps during data transformation are necessary. Possible errors are documented at the location (during parsing, mapping, …), where they actually occur, the error propagation is reduced to a minimum. In the report the information on the transformation is available in different granularity and from different points of time so the decisions are governed by multiple measures of merit and performance. The used architecture is very pragmatic with a high usability and feasibility.

In table 3.1 the format of a transformation report is specified. In the left column the content of the meta information is listed. The column in the middle is a reference to the figure 2-1 Architecture for gaining a Transformation Report and in the right column there are comments, describing the use of the described section.

Table 3-1 Format Spezifikation of a Transformation Report (caption below)

Type of Information	Source of Information from [Fig 2]				Section
Data Source	R0				Header
Data Target	R0				
Interface Type	R0				
Organisation Sending	R0				
Organisation Recipient	R0				
Configuration Element	R0				
Summary Information			R2*		
Source Unit of Functions		R1			Repeating Section
Target UoF		R1			
[%] of mapped Tables		R1			
Source Tables	R0	R1			Repeating Section
Target Tables		R2	R2*		
[%] of mapped Attributes		R2			
Source tool entity ID	R1				
Type of Data	R1				Definition Section
Target Tool Entity ID		R2	R2*		Repeating Section
Error Type	R1	R2	R2*		Section
Referenced Files	R1				Repeating Section
Error Type	R1				
Business Errors	R1	R2	R2*		Repeating Section
Semantic errors in Line	R1	R2	R2*		
Attribute No.	R1	R2	R2*		Repeating Section
Error Type	R1	R2	R2*		
Syntax errors in Line	R1				
Attribute No.			R2*		Repeating Section
Error Type	R1				
Warnings in Line	R1	R2	R2*		Repeating Section
Attribute No.	R1	R2	R2*		
Warning Type	R1	R2	R2*		

Table 3-1 Format Spezifikation of a Transformation Report

3.1 The Header Section

The header contains administration and technical information on the sender and receiver of data. This information is available before the parsing and formatting activities starts and is marked with R0. Only the Summary Information in the header section is generated, after the whole report is available.
The following attributes for in the header section are proposed:

Data Source
    design_tool_name
    design_tool_version
    schema_identifier
    understandability

Data Target
    design_tool_name
    design_tool_version
    schema_identifier

Interface Type
    implementation Level
    interface_version
    preprocessor_version
    report_describes_interface
    transform_direction

Organisation Sending
    country
    electronic_mail_address   e-mail address of project contact for data exchange
    internal_location
    post_box
    postcode
    region
    street
    street_number
    technical_contact_name   e-mail address of technical contact for data exchange
    telefax_number
    telefon_number
    town

Organisation Recipient
    country
    electronic_mail_address   e-mail address of project contact for data exchange
    internal_location
    post_box
    postcode
    region
    street
    street_number
    technical_contact_name   e-mail address of technical contact for data exchange
    telefax_number
    telefon_number
    town

Configuration Element (of the exported data)
    authorisation
    checked_out
    comment   Free text comment for sender
    contract
    contract_id   Identifier for project or contract under which data exchange takes place
    data_exchange_state
    data_time_stamp_of_report
    described_system
    description
    digital_signature
    file_change_indicator   Indicates if this STEP file contains the new item definition or an update to that item definition
    live_cycle_state
    maturity_stage
    message_identifier   Unique identifier for this exchange message
    object_id
    physical-file-name   Physical name of the data containing file
    revision
    security_classification   Security classification of data containing file (in data containing file record), aggregate security classification of all files
    sequence_number
    substitute_identifier
    superseeded
    system_code
    system_construct
    titel
    type_manufacture
    version

The attribute "Implementation_Level" is an indicator for the trust ability of the data. It is a scale factor indicating, how much data cases of the interface have been tested. It is also an indicator for the maturity of the interface. This factor seems obscure, but is useful, because in much cases the documentation of the underlying data

model of the source tools is not publicly available. So it is necessary to develop an interface by trial and error [Eckert, 2003].

An other quality measurement is the "Understandability". It describes in which level of detail the target system can interpret the delivered data. This measures if it is possible to map the data from the source to the target tool on a direct way target structure or if aiding structures are necessary. The "Understandability" is also an indicator for the compatibility of the source and target data model.

The "Implementation Level" and the "Understandability" are relevant for the data integration strategy and give an indicator about the trustworthiness of the delivered data.

3.2 The Definition Section

Units of Functions (UoF) are a collection of application objects and their relationships that defines one or more concepts within the application context such that removal of any component would render the concepts incomplete or ambiguous [ISO, 1994]. A new trend in the standardisation of data models (e.g. in ISO 10303) is to modularise the so called Application Protocols in sub data models, that are compatible with other models from other domains. The modules are like Units of Functions. In the same concept modules of a data model can be used. This cluster is the first indicator for the size of data loss. It indicated which module or UoF has no or a restricted representation in the target data model.

Tables are a structured collection of data within a database. At its simplest, a table is a grid with columns ("fields") and rows. This section gives detailed information about the transformed data and indicating where data losses occurred.

(Source tool) Entity ID is an identifier. It is a character or group of characters used to identify or name an item of data and possibly to indicate certain properties of that data (ISO 2382/4). This is section provides detailed information about the transformed data.

In the Referenced Files sections the files are listed, that were intended to send. This list is compared with the files that actually were attached.

Business Errors are operating rule/policy that are agreed by the involved organisations and the transferred data has to comply with, e.g. invalid authority, work authorisation is missing, invalid area type, authority is already closed, …

The likeliest causes for Semantic Errors are that the detailed structure of the definition doesn't correspond to what is allowed by the specialization in use, or that the definition is inconsistent. It has to assure, that data sent in an exchange message have the same meaning in the sending environment as the receiving environment after import. The documentation of the Semantic Errors is already during the testing and development of the interfaces very useful. It improves the quality of the interface especially if it is necessary to develop it by trial and error.

Syntax Errors occur frequently, e.g. if the user have not filled correctly the source database with data. In literature this error is described as missing "Data Quality". If the characters in the file don't correspond to a term (taking account of the operators currently in force). This section also assures, that errors within the mapping processor in import or export due to wrong syntax of the data-files are prevented.

Warnings could be a syntactical change like splitting of a structure into two parts or replacing an integer into a real one. A reason for a warning could be also a semantic change like the conversion of a breakdown structure into a plain text representation for e.g. MS Word. A structural change of the data is also documented into the warnings like the transformation of a 3D representation into a 2D representation, or the hierarchical data representation of a tool into a flat data representation.

4. Discussion

The Transformation Report reduces the number of soft information, that are only based on the selectors judgment and experience. The number of hard information, based on scientific principles is increased.

The limitation of the report is e.g. certain prediction hold (e.g. the sum of expense in each department is less than or equal to the department budget). The transformation report can not control project internal requirements.

It is very difficult to detect problems of syntactical mismatch, e.g. indented word "fare", erroneous word "fair",… Both words are syntactic correct, but with a wrong semantic. Error detection by a consistency checker on database level is difficult if the data contains many special terms, symbols, formulas, or conventions whose syntax contribution cannot be established without a complex understanding of the text. e.g.

- The '$' value is only allowed for optional attributes
- The value is not compatible with the corresponding SELECT type, or is not correctly encoded.

5. Conclusion

The quality of data exchange can be significantly improved when the results of the individual steps of the transformation process are well documented. With the transformation report the user can gain a clear understanding of the data itself and, secondly, properly focused information to help determine a suitable data integration strategy for use in a repository. It provides the data integration instance with previously unknown facts and features that can be used for an enhanced integration algorithm:

- Trustworthiness of the delivered data
- Documentation of data loss
- Bill of delivered data
- Minimised error propagation
- History of data
- Constraint violations
- Handshaking function for data integration

The Transformation Report traces data exchange actions and is a precondition for data integration strategies to improve quality of exchanged information. The described report is not tool specific and is a practical way for documenting the results of the single data transformation steps. Therefore it is suggested that it should be included into the framework of ISO standards, e.g. "Standard for the Exchange of product model data (STEP ISO 10303) [ISO, 1994; Kemmerer, 1999]. The Transformation Report could be the missing intersection between the existing application protocols and database schemas. The structure of the transformation report is generic, so it is easy to use the report without updating it for transactions in different domains. It does not require special encoding, but is able to use standard encoding like e.g. XML, [XML, 2000], (ISO 10303-28) or ASCII (ISO 10303-21).

6. Reference

[Eckert, 2003] R. Eckert, G. Johansson, Experiences from the use and development of ISO 10303-233 Interfaces in the systems Engineering Domain, ICE 2003, June 2003, S. 501-508

[English, 1999] English, L.P., Improving data Warehouse and Business Information Quality, John Wiley & Sons, New York, 1999

[ISO 9735, 1990] ISO 9735, Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules, (1990)

[ISO, 1994] ISO, ISO 10303-1, Industrial automation systems and integration - Product Data Representation and Exchange - Part 1: Overview and fundamental principles, 1994

[Kemmerer, 1999], Kemmerer, S. (ed.), STEP - The Grand Experience, NIST Special publica-tion 939, 1999

[Lomax, 1977] Lomax, J.D, Data Dictonary Systems. Rochelle Park, N.J.: NCC Publications, 1977

[Loshin, 2001] David Loshin, Enterprise Knowledge Management: The Data Quality Approach, Paperback 350 pages (January 2001), Publisher: Morgan Kaufmann; ISBN: 0124558402

[Olson, 2003] Jack E. Olson, Data Quality: The Accuracy Dimension, Paperback 300 pages (9 January, 2003), Publisher: Morgan Kaufmann; ISBN: 1558608915

[Redman, 1997] Thomas C. Redman, Data Quality for the Information Age (Computer Science Library), Print on Demand (Hardcover) 320 pages (January 1997), Publisher: Artech House; ISBN: 0890068836

[Wang, 2000] Richard Y. Wang, Mostapha Ziad (Editor), Yang W. Lee (Editor), Data Quality (The Kluwer International Series on Advances in Database Systems), Hardcover 176 pages (November 2000), Publisher: Kluwer Academic Publishers; ISBN: 0792372158

[XML, 2000] Extensible Markup Language (XML) 1.0 (Second Edition) W3C recommendation 6 October 2000, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler