NGOSS Compatibility Overview Paper

Metabula Limited, May 2005

Intentionally Left Blank

Introduction 4
NGOSS Compliance Testing Strategy 4
Scenario 4
The Approach 6
Extensions and Wrappers 6
Overlap analysis 6
Matching 8
The Process 9
1. Import the SID information model 10
2. Define the SIS information model 10
3. Exposing the information models 11
4. Normalizing models to produce a "flattened" representation 12
5. Define transformation "rules" 13
6. Transform the information models 14
7. Create matching criteria 14
8. Generate match candidates 17
9. Select successful candidates 18
10. Tag the associated entities in the SID and SIS models 18
11. Create SID extension for all "unmatched" SIS entities 19
12. Export definitions for comparison and use by other tools 20
The Products 21
Summary 22

Introduction

Metabula is a specialist provider of intelligent information management, integration and consolidation solutions. These solutions help to limit and reduce errors, promote consistency and increase efficiency thereby reducing project costs and times.

In particular, Metabula has specific competences in object-oriented representations, data modelling and mapping. This applies to standard and industry modelling techniques such as UML, ISO STEP representations and AP-models e.g. Oil / Gas, Telecoms, Shipping, Defence, Aerospace.

These capabilities have been applied to the telecoms industry where Metabula has focused upon the representation and management of the NGOSS SID model and comparison with client specific SIS models.

The ability to compare SID and SIS information models allows NGOSS compatibility testing. to be performed. However in addition Metabula is able to generate a complete representation of the SIS in the language of the SID. This is achieved by matching the "same" entities in the SID and SIS and then generating extensions to the SID to handle those SIS entities not yet supported in the SID.

Metabula believes that this provides an extremely powerful solution for those clients wanting to understand, manage, and compare their business information model (an SIS) and the industry standard NGOSS SID model.

NGOSS Compliance Testing Strategy

The focus of the NGOSS Compliance Testing Strategy Technical Specification (TM50A) is to specify what information should collected by a client and how it should be analysed to generate a set of compliance metrics. TM50A does not consider how information should be collected by a client. In the arena of data modelling, this "how" frequently equates to "manual labour".

One of the NGOSS compliance metrics is generated by comparing the client's information models with the SID ABEs.

The objective of this SID compliance metric is to quantify the degree of commonality between:

- the Shared Information Specification (SIS) provided by the client
- the SID ABEs provided by TMF.

A key part of the compliance process is the SID Entity test (NGCTSID02). This test identifies which SID entities are present in the SIS. The client must submit a mapping specification table for this test that lists the matches between SID and SIS class-attribute pairs.

Scenario

The particular scenario chosen utilises the SID Phase 5 model, and the ITU-T X.790 as the SIS model

The ITU-T X.790 "deals with the management of malfunctions ("troubles") in systems and communication networks. It lists trouble administration functions of fault management that facilitate interaction between customer and service provider in the case of a trouble."[1]

[1] http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/6_HPOVUAWS/Papers/langer_nerb.pdf

The ITU-T X.790 model "defines a report format that allows the customer to track the trouble resolution process within the service provider. The report format is modeled using an abstract super-class called Trouble Report (TR) and two derived classes called Telecommunication Trouble Report (TTR) and Provider Trouble Report (PTR)."

The Approach

The approach undertaken by Metabula is based upon the following main stages

1. Detailed analysis of the overlap between the SID and SIS information models.
2. Identification of the "same" entities in the SID and SIS through matching
3. Linking of the SIS entities to their matched SID entities and creation of SID extensions

Once all the stages have been completed then the resulting system can be used in a variety of different ways such as to generate the compatibility mapping specifications as defined in TM50A, or even possibly to present instance data according to these models.

Each of these stages of the process is considered in more detail later, however a couple of generic aspects are considered first.

Extensions and Wrappers

The standard mechanism to enhance an industry standard model to support specific business details is through the use of Wrappers and Extensions.

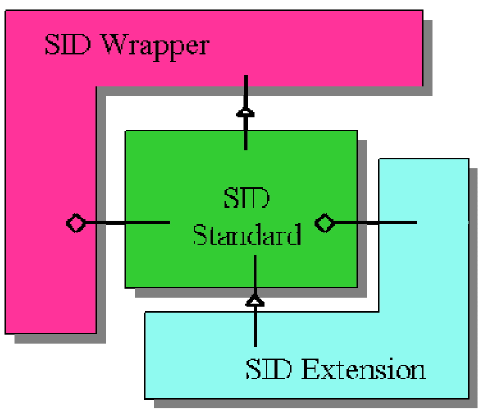

Figure 1 SID Extensions and Wrappers

An extension or wrapper is used to extend the standard model; an example of an extension would be to represent a specialisation of an entity in the standard; an example of a wrapper would be to add a property to all entities of a particular type in the standard.

Overlap analysis

The analysis of the SID and SIS information models will have to categorise the entities into three sets:

- Entities in the SID but not "identified" within the SIS (Set X)
- Entities in the SIS but not "identified" within the SID (Set Y)
- Entities which are in some way identified within the SIS and SID (Set Z)

These three sets can, unsurprisingly be presented as in Figure 2.

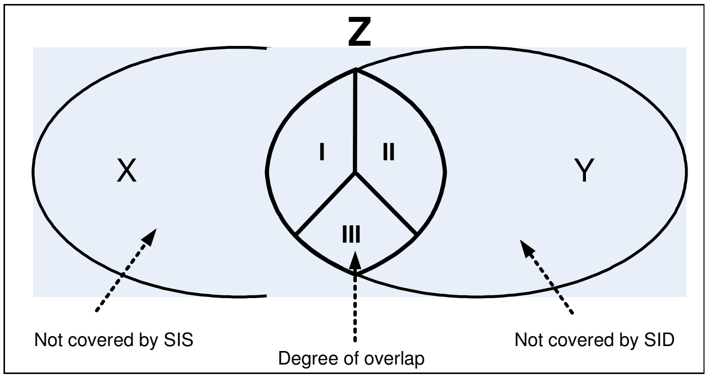

Figure 2 Categories of model overlap

The assumption has been made that the SIS and SID are information models rather than implementation or data models as the comparison must be made between semantically equivalent models.

The entities of interest for comparing the SIS and SID are sets Y and Z. It is unlikely that any SIS will cover all the SID domains; it is more to be expected that an SIS will relate to a few SID domains.

The items in Set Y can be handled fairly easily – they are those entities with no "overlap" between the SID and SIS. A SID extension entity representing the SIS entity can be created and a link created.

The items in Set Z are those which overlap to some extent and it is in the area of the overlap where there is the possibility or potential for "matching" entities from within the SIS and entities from within the SID.

The strategy is to identify everything in the SIS as being in Set Y or Z, where items in Set Y are those SIS entities which are deemed to have no overlap with the SID.

There are various degrees of overlap which can be identified in Set Z. An example is the following three categories:

I. Members match completely on class, attribute and relationship
II. Members match completely on class and attribute but not on relationship
III. Members match on class but not completely on attribute or relationship.

However there are a number of points to note including

- Not all the attributes in the SID are mandatory, in that some are optional.
- Matching on relationships is probably rather difficult due to foreign key handling, in addition to the difficulty in handling cardinality
- Relationship conformance is not included in the SID conformance criteria as it is assumed that fundamental relationships already exist between the entities
- It is assumed currently that containerisation is not used in the tests for matching and conformance selection. This is something that may be considered for the future.
- It is likely at this stage that no distinction (weighting) will be applied between relationship types –
    - generalization
    - specialization
    - composition
    - aggregation

- association

Thus the activity of performing the overlap analysis will generate the Y and Z sets of data which can then be used to complete the process. The current solution focuses upon matching by class characteristics but will be extended shortly to include attributes and relationships.

Matching

The activity of matching is applying rules, techniques and tools to a set of data to attempt to identify “candidates” for being considered to be the “same” item.

The data is often manipulated from its original format to include phonetic and other representations, these fields there then used as “match key” fields. A particular example of this is standardisation to reflect naming conventions in the SID.

Additional “scorer” fields such as number of attributes on the entity, name etc. are then combined to generate a score value. The “scores” for each “match keyed” item are then compared and candidate entity matches are identified. These can then be manually verified and refined if required before being used as actual entity matches.

The Process

The process of generating the solution requires the following steps –

1. Importing the SID information model
2. Defining/Importing the SIS information model
3. Exposing the information models
    a. Testing the SIS information model against data source samples
4. Normalizing models to produce a "flattened" representation
5. Define transformation "rules" to remove noise and standardize it to the SID naming conventions
6. Transform the information models
7. Creating the criteria for matching the transformed SIS model against the normalized SID model.
8. Generating the list of potential match candidates.
9. Selection of those candidates with required degree of matching confidence
10. Tag the associated entities in the SID and SIS models
11. Create SID extension for all "unmatched" SIS entities
12. Export definitions for comparison and use by other tools

The process is "mapped" against the key data structures in Figure 3 and described in more detail in the following sections.

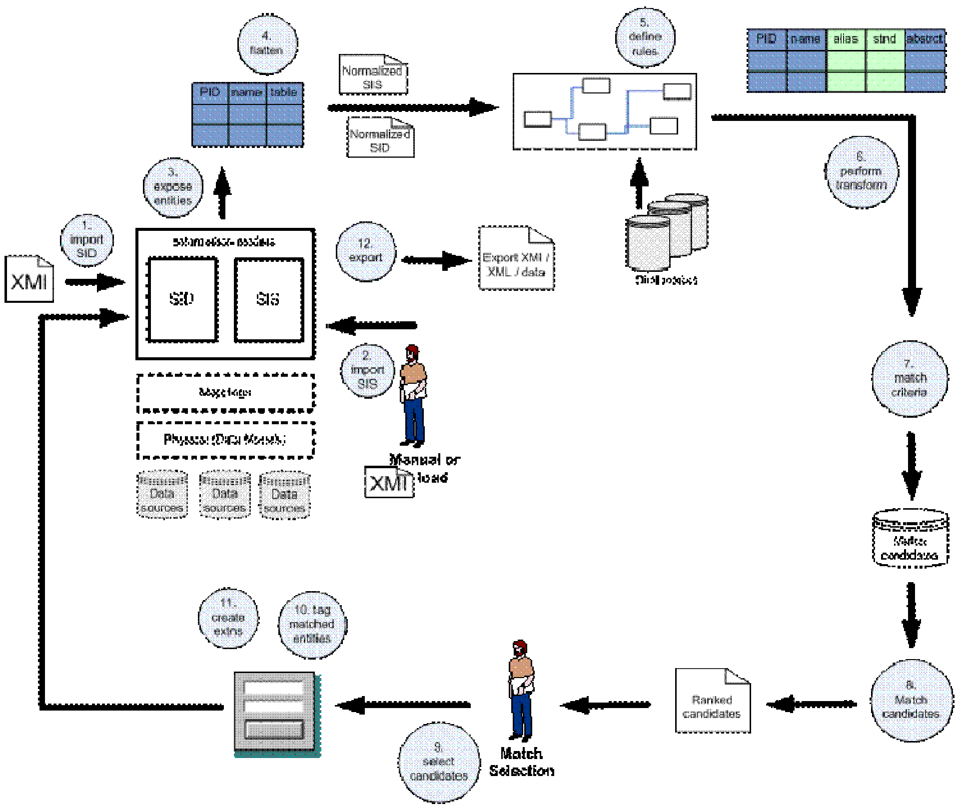

Figure 3 The mapping process

1. Import the SID information model

The first stage is to import the SID Aggregate Business Entities (ABEs) into the system using the Phase 5 XMI file.

The model is loaded into the system's Schema Management tool, Q-Schema as shown in Figure 4.

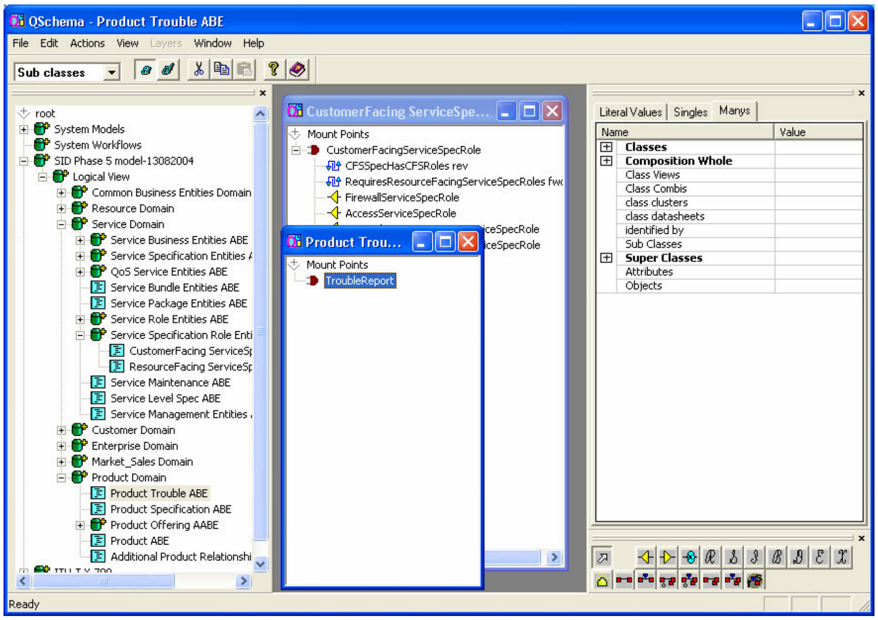

Figure 4 Q-Schema with SID Model

It should be noted however that the system will typically pre-loaded with the SID information model before the start of any project.

2. Define the SIS information model

This stage creates the client's information model i.e. the SIS model within the system. The client may already have an explicit information model or may need assistance in creating one from a set of data (or implementation) models.

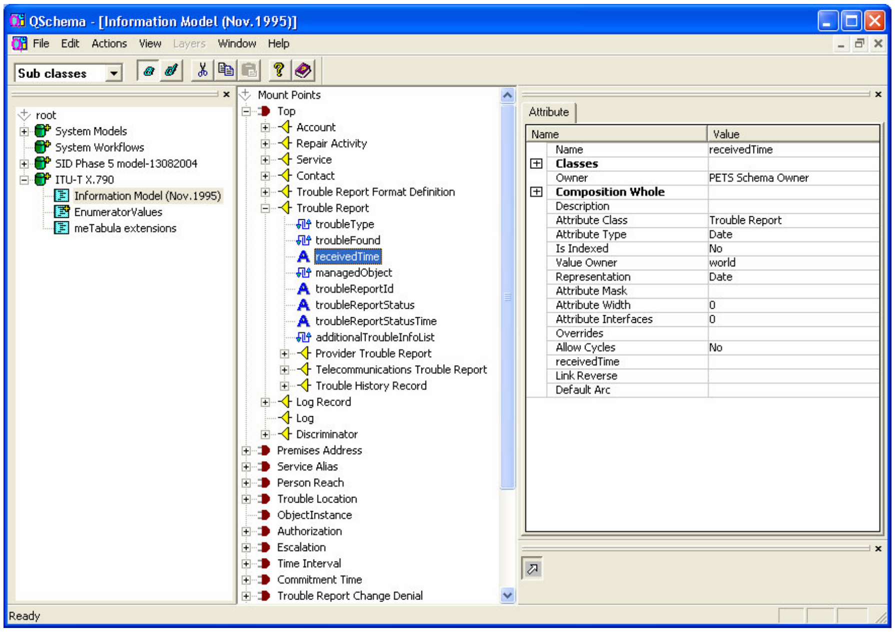

Figure 5 Q-Schema with SIS Model

The SIS model can then be loaded directly into the system if it can be represented as an XMI file. The tool used is Q-Schema (see Figure 5). It supports import from a variety of XMI formats corresponding to different versions of UML.

The alternative is to manually create the model in the system (using Q-Schema). This is achieved through the use of the toolbox which allows for creation of individual entities, attributes, models etc.

3. Exposing the information models

The full models loaded into the system are not required for the comparison and extensions. Therefore only those entities required are exposed for evaluation. This is performed using the DataBridge tool.

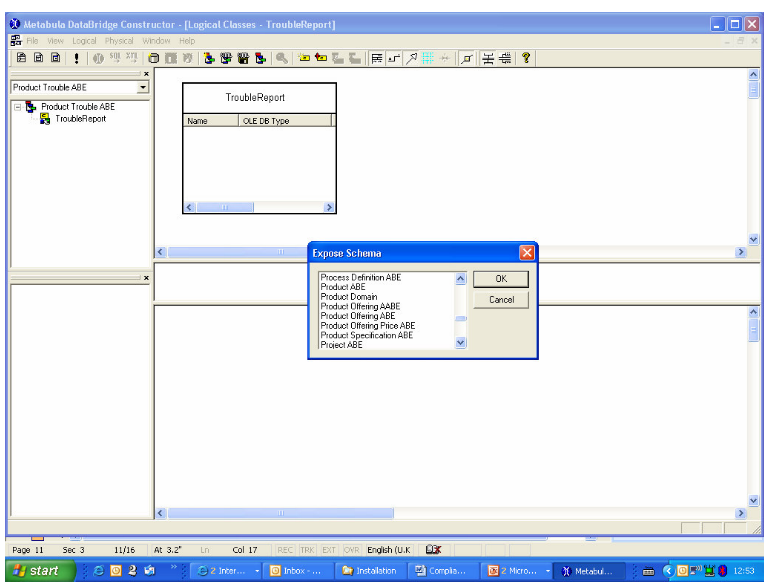

Figure 6 DataBridge Constructor UI

The DataBridge system itself exposes two interfaces – OLEDB and XML which allows access to the models, and any data.

3a. Test the SIS information model

As an optional step, it is also possible to utilise the DataBridge system to link the SIS information model to the underlying system data models.

This is conceptually represented in the three panes in the Constructor view which are

a. Top Pane – SIS model
b. Bottom Pane – data models
c. Middle Pane – mappings between the data and information models

The data models are loaded directly by connecting to the relevant source. The mappings are defined manually using rules. The result is the ability to review a set of sample data from the source systems through the relevant Information Model.

4. Normalizing models to produce a "flattened" representation

This stage extracts the SID and SIS information models from DataBridge into a normalized format. This is achieved by using the OLEDB interface to access a schema-oriented representation of the information held within DataBridge.

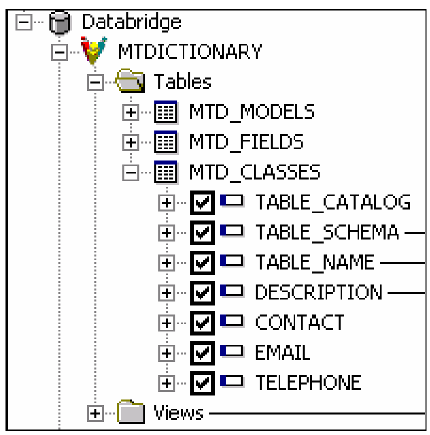

Figure 7 Normalized data access

This flattened data representation is shown in Figure 8, and will then be populated and used for the comparison and matching by adaptation and mapping.

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	DESCRIPTION	CONTACT
metabula	Entities	AddressContactMediumRole	16491	Administrator
metabula	Entities	AnyEntity	16492	Administrator
metabula	Entities	BoundingVolumeRepresentation	16493	Administrator
metabula	Entities	Representation	16494	Administrator
metabula	Entities	LocationIdentification	16495	Administrator
metabula	Entities	Place	16496	Administrator
metabula	Entities	PlacePartyRoleAssoc	16497	Administrator
metabula	GeographicAddress	GeographicAddress	16528	Administrator
metabula	GeographicAddress	AbstractGeographicAddress	16529	Administrator
metabula	GeographicAddress	PropertyAddressAssociation	16530	Administrator
metabula	GeographicAddress	PropertySubAddress	16531	Administrator
metabula	GeographicAddress	GeographicSubAddress	16532	Administrator
metabula	GeographicAddress	UrbanPropertyAddress	16533	Administrator
metabula	GeographicAddress	UrbanPropertySubAddress	16534	Administrator
metabula	GeographicAddress	PostalDeliveryAddress	16535	Administrator
metabula	GeographicAddress	LogicalAddress	16536	Administrator
metabula	GeographicAddress	AmericanPropertyAddress	16537	Administrator

Figure 8 Normalized data values

5. Define transformation "rules"

A three-phase process is applied to transform the native SIS entities (nSIS) to transformed SIS entities (tSIS). This same transformation process is also applied to the SID entities to ensure commonality of language for comparison and matching.

The three phases decompose the name into component parts, remove noise and standardize the values to the SID naming conventions. This is handled by a data quality cleansing product, Datanomic Clean and in more detail the phases are –

1. Atomisation – decompose composite name structures down into one name substructures.
    - Typically SIS classes would attempt to be mapped to SID classes, SIS attributes to SID attributes, however SIS attributes would also attempt to be mapped to SID classes
2. Standardize the nSIS data by applying SID naming conventions e.g. no spaces, hyphens, underscores.
    - This "Standardization" creates consistency of content between the models at a syntactic level.
3. Apply dictionary lookups to identify known synonyms to nSIS entities
    - e.g. noOf -> numberOf, and id -> identifier.
    - Dictionaries create consistency of content between the models at a more semantic level.
    - It is important that these dictionaries are actively grown over time as new SIS entities are analysed.

Datanomic Clean (a product resold by Metabula) transforms source entities via network of processors (as in Figure 9) to one or many targets. These processors provide the parsing, standardization and lookup functionality needed for the three phases of the SIS transformation process.

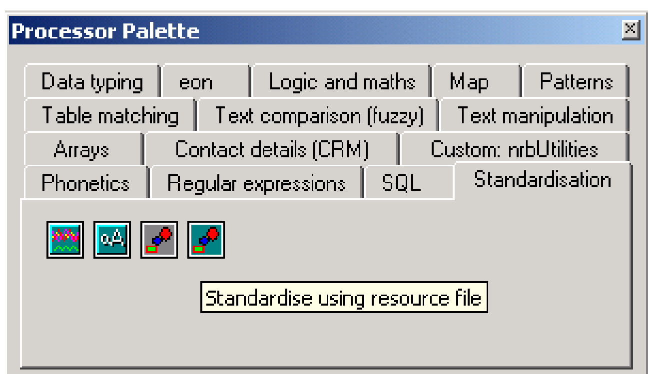

Figure 9 Processor Palette

The processors are applied individually or sequentially from the source fields to achieve the required target results. This is shown in Figure 10 where a number of processors have been applied

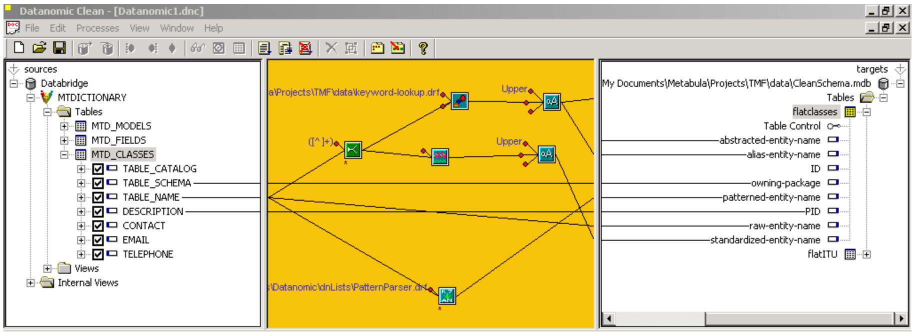

Figure 10 Define transformation rules

6. Transform the information models

This stage executes the transformations defined in the previous stage. The output is a fully populated data set which has been standardised and de-noised with aliases generated and any dictionary lookups performed.

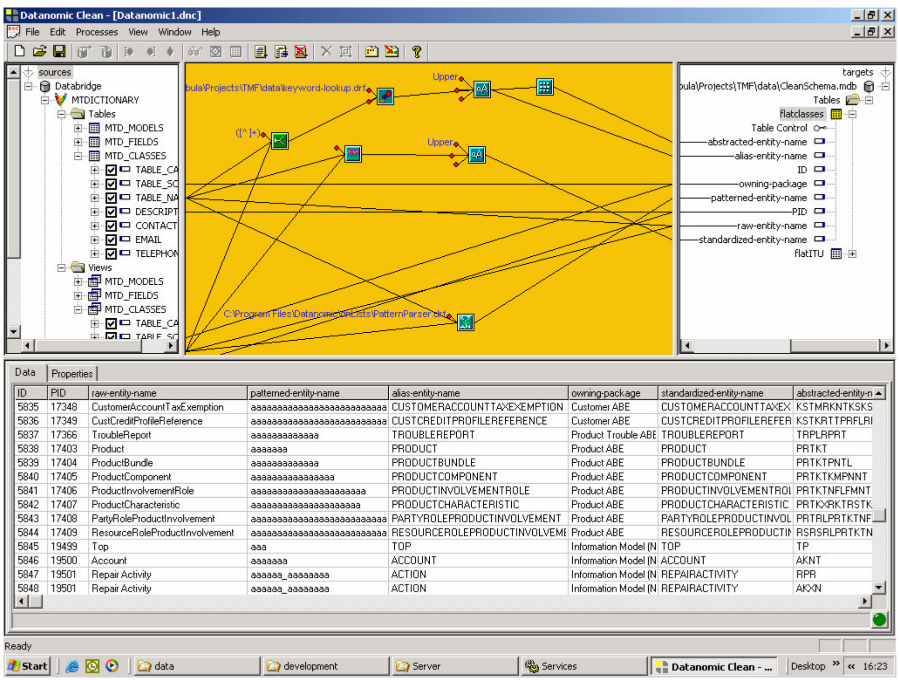

Figure 11 Transformed data

7. Create matching criteria

This step requires the creation of the criteria for matching the transformed SIS model against the normalized SID model.

The two input sources, nSID and tSIS are compared by applying comparing and matching capabilities contained within the DataInspector component of the system (see Figure 12).

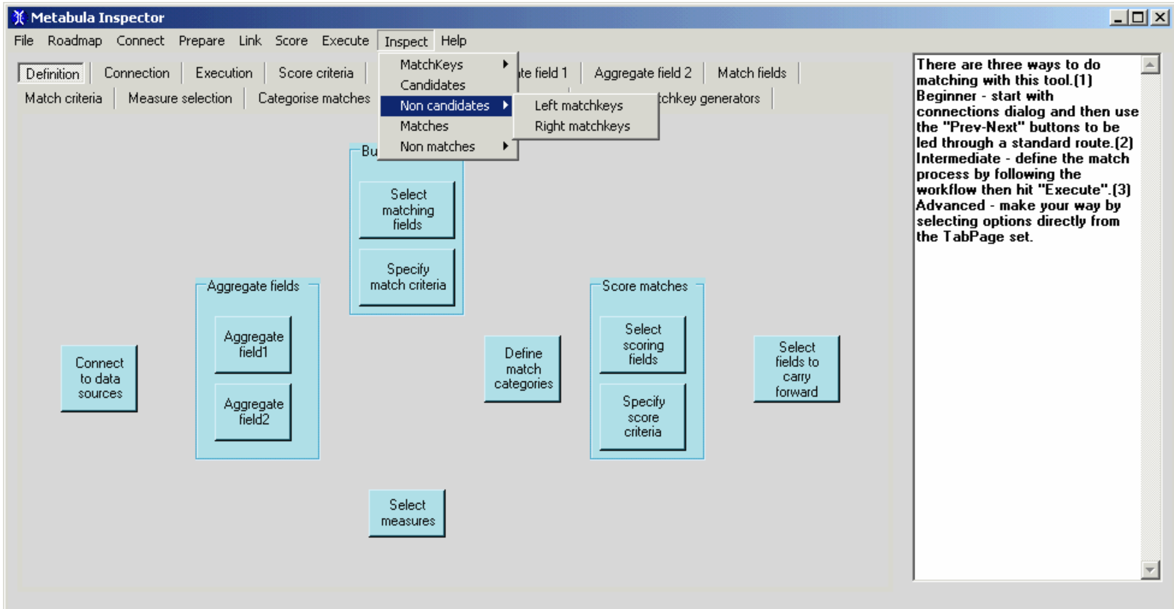

Figure 12 DataInspector UI

A three phase match process is applied –

i. Generation of (abstract) matchkey values from one or more entity fields
ii. Matching of entities on matchkey values
iii. Scoring of entities on non matchkey values

Matchkey values will be constructed from model-class pairs.
Three types of match algorithm will be applied – exact, phonetic and n-gram. Matches will be scored using a simple edit measure. The match process will be applied at the class, attribute and relationship level.

i. Generation of matchkey values

The matchkeys are defined by selecting a collection of data columns from the SIS and SID representations as being match keys.

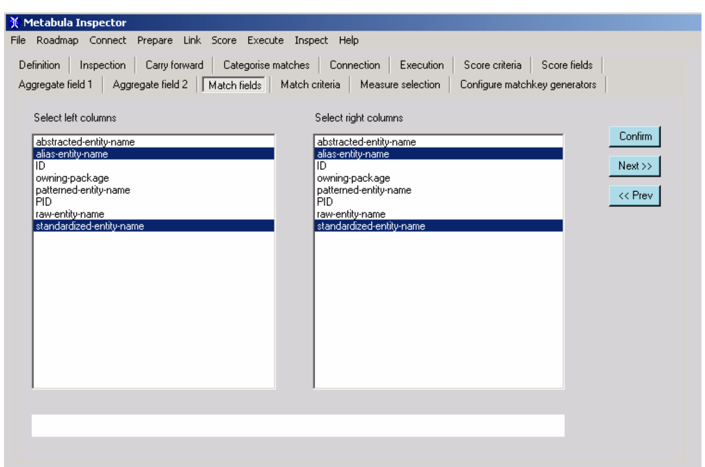

Figure 13 Match key fields

The criteria for matching is then defined based upon the selected fields.

ii. Matching

The matching process itself defines the categories of matching based upon the matchkey fields selected. In this case there are defined to be four levels of matching which vary from Complete if both fields match, to None if neither field match.

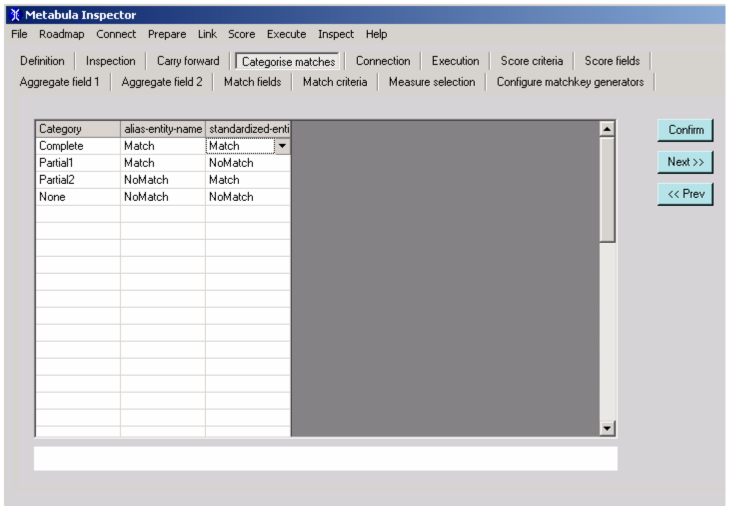

Figure 14 Matching

The candidates generated by the matching process are divided into these four categories.

iii. Scoring

The scoring of each candidate is performed by comparing values for non-matchkey fields held within the structures. As shown in Figure 15, the raw-entity-name has been used for comparing the nSID (left hand) and tSIS (right hand) model values.

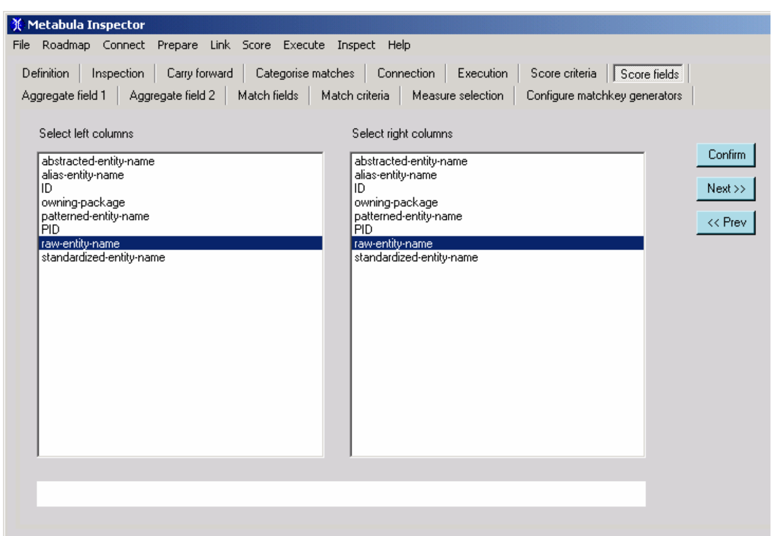

Figure 15 Scoring fields

The actual scoring criteria used are as defined in Figure 16. In this case the EditDistance metric is used to identify how different the candidates are in character terms.

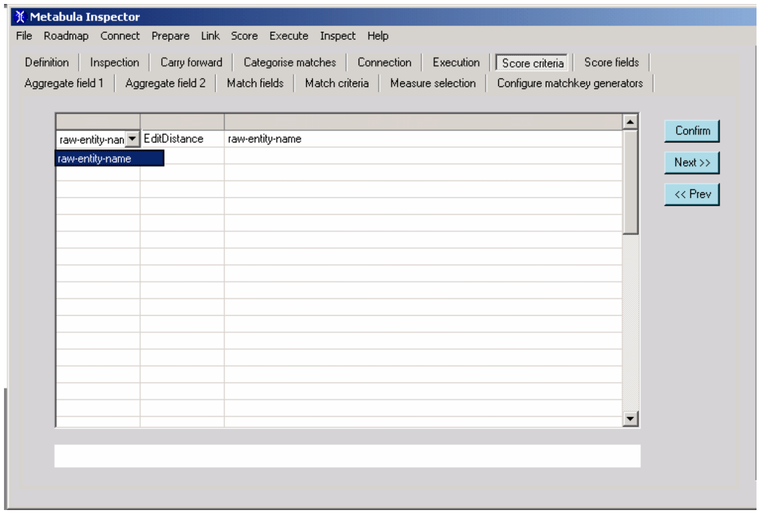

Figure 16 Scoring criteria

8. Generate match candidates

This stage generates the various candidates for matches between the SID and the SIS information models by executing the inspection.

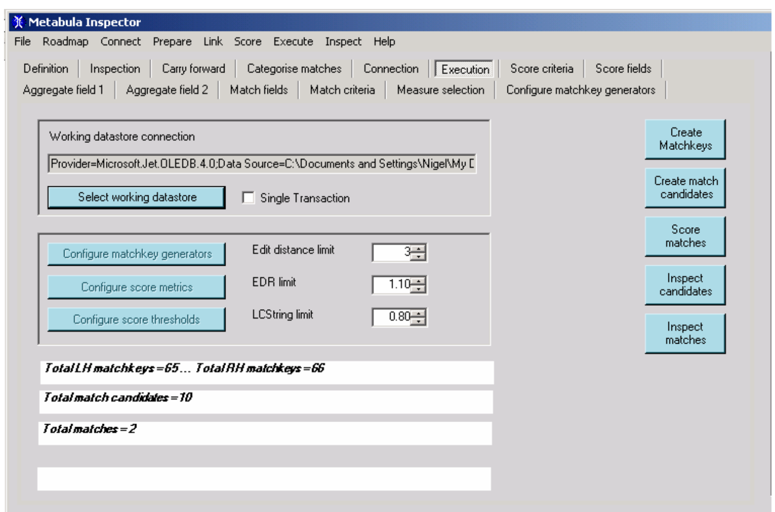

Figure 17 Execute inspection

The match candidates are created by running the matching criteria defined in the previous step and the candidates are ranked by score.

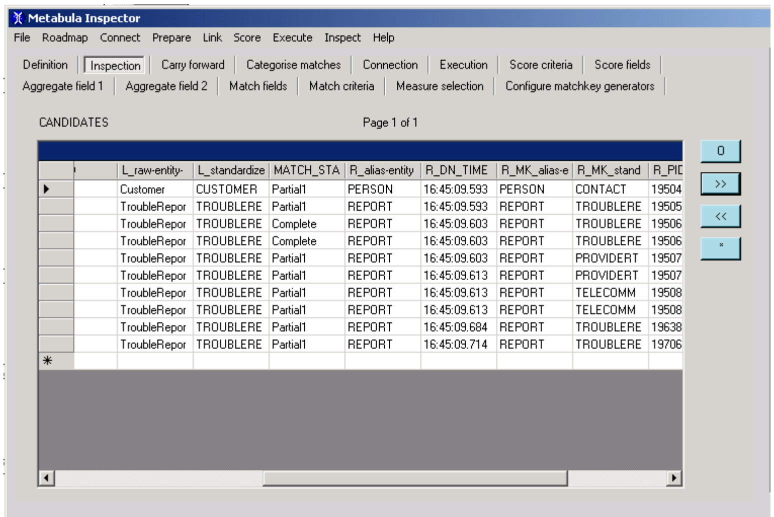

Figure 18 Candidate matches

In this example, as shown in Figure 18, there are some partial and some complete match candidates generated from the process.

9. Select successful candidates

The successful candidates are chosen to be those that attain a certain degree of confidence rating (i.e. scoring) in the matching comparison as shown in Figure 19.

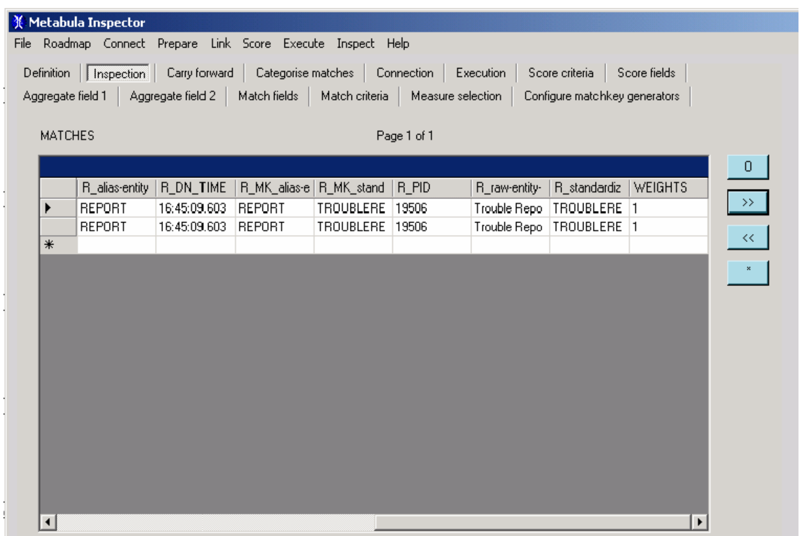

Figure 19 Matched candidates

10. Tag the associated entities in the SID and SIS models

The next stage is to tag those SIS entities which are matched to SID entities. A relationship SIS-SID is added to all the SIS entities

In our example, the SIS Trouble Report entity is the only one found to “match” an entity in the SID. Therefore the SIS Trouble Report and SID TroubleReport are linked.

11. Create SID extension for all "unmatched" SIS entities

The process defined within the system is to generate a SID Extensions model for each of the entities in the SIS model that are not directly matched into the SID.

In this example the only entity matched was the Trouble Report, therefore all other entities are added into the ITU-T X.790 Extensions Model.

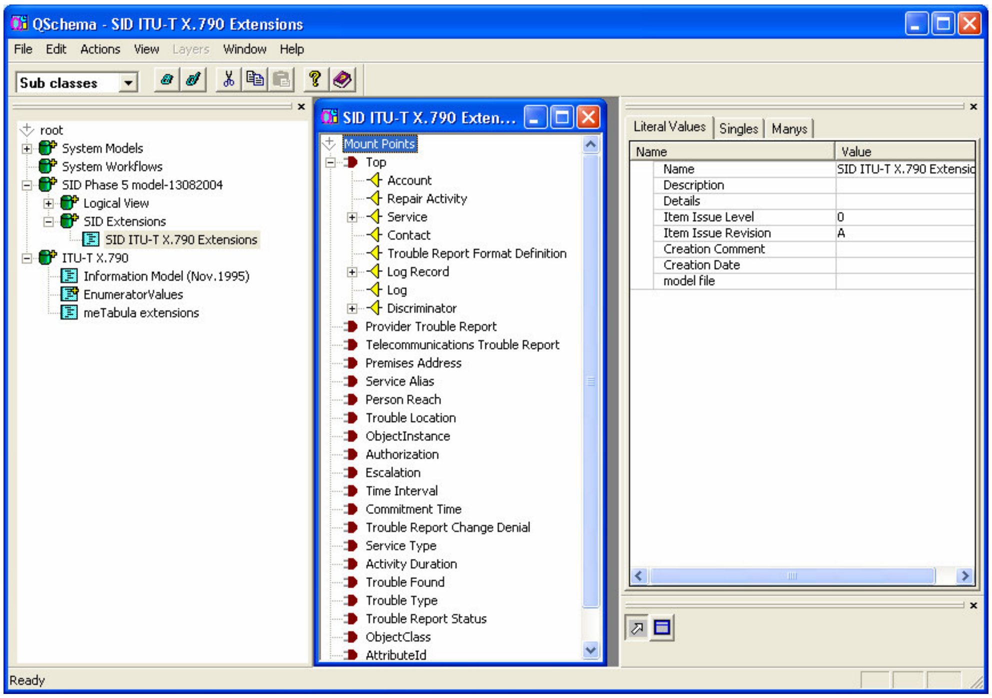

Figure 20 Extensions Model

The interesting points to note are that the mount points definition within the model has been changed. The Trouble Report node under Top has disappeared and this means that all of the sub-classes of Trouble Report now seek new mount points – most of them now become root mount points.

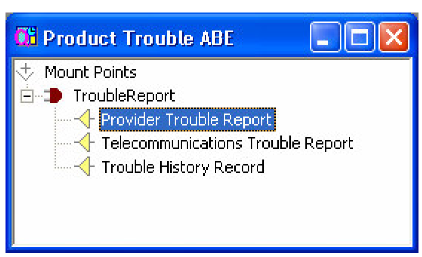

Figure 21 Sub-types of TroubleReport

In addition, as part of linking the extensions into the SID model, the sub-entities of the SIS Trouble Report are "added-in-to" the sub-entities relationship off the SID TroubleReport. This only becomes visible when the ability to explore outside of the bounds of the current model is enabled i.e. in Figure 21 the Provide Trouble Report is not part of the SID model, and only appears when model-free exploration is allowed.

12. Export definitions for comparison and use by other tools

The process within the Metabula solution is now complete. The final step is to export the results generated for subsequent use in compatibility tools or subsequent processes.

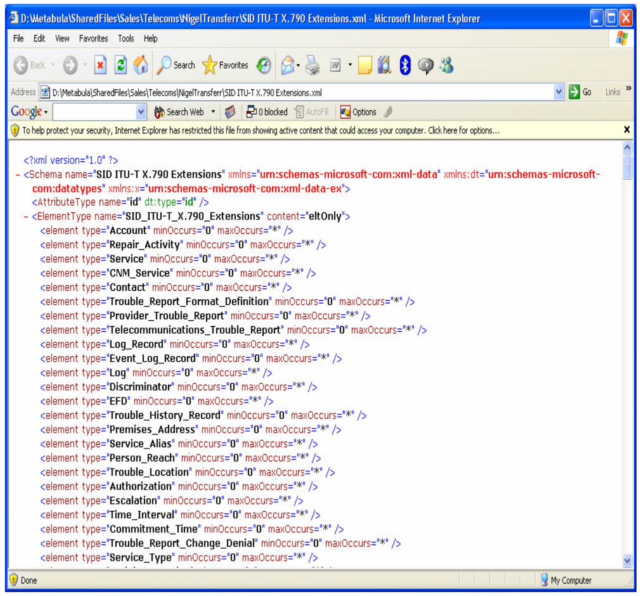

Figure 22 Export XML

The data exported depends upon the required usage, be it for comparison, be it for use in other applications etc. The format of the export can be defined also but is expected to be XML-based with the schema depending upon the requirements.

The Products

The Metabula solution utilises a variety of Metabula's products. The DataBridge product is used as the information model repository and is able to present the views on the SID, SIS, and SID extensions models. These views are then used by the DataClean and DataAudit products to perform the overlap analysis and proposed matching with the results being loaded back into the DataBridge product.

However the DataBridge product can also act as an EII engine that is driven by three sets of model data –

- data models – defining the schema and location of external data sources
- information models – defining the entities to be exposed as "views"
- maps – linkages between data and information models.

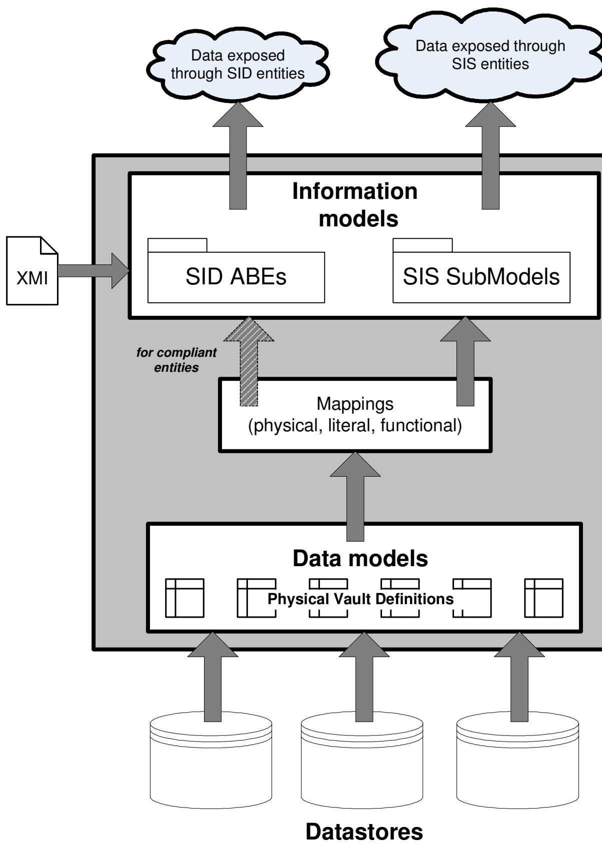

Figure 23 Information model repository

This means that if the data models and maps have been linked to real external data sources then DataBridge will be able to present actual "instance" data "as" SID or SIS instances according to their particular entity type. This is represented in Figure 23 by the data "clouds".

Summary

MeTabula believes that the process described could provide significant benefit in the area of NGOSS compatibility testing. The ability to automate and electronically document the process is extremely powerful, but it also means that the resulting information models can be used to present SIS information according to the SID models as well as make the models available for use in other tools and subsequent phases.