**Data Enrichment and Validation Step**

1. Introduction

The data enrichment and validation procedure (DEAV) is generally an ETL step which

1. Data enrichment and validation – input and output

The data enrichment and validation procedure (DEAV) requires as input the following information:

1. **InputTable:** A reference to a table which contains the input data which will be enriched by the procedure. Note that this table will be used as output for the enrichment, that is, the input table will be updated by adding the additional columns provided by the enrichment.
2. **ErrorTable:** A reference to a table which will contain the list of errors which occurred during the enrichment step.
3. **Action:** A conventional name specifying which enrichment/validation procedure is requested.
4. **[Additional parameters depending on the requested actions]:** Everyenrichment/validation step has its own set of specific parameters. It is necessary to specify them, in order to make the enrichment and validation procedure work properly.

At the end of the process, the required additional columns will be added to the input table. If errors/warnings were detected (either by the enrichment step or by the validation), then they will be inserted into the table specified in the second input parameter.

1. Enrichment/validation for Business Rules

The Business Rules Engine (BRE) needs more and more sophisticated checks, in order to perform a suitable and useful validation of the reported data. This leads to extend the engine in order to accommodate new validations which have a complex structure. In particular, some validations steps of the engine require having:

* External/computed additional information, in order to assess the correctness of the data more precisely.
* The possibility to delegate the validation to a less structured system if the validation is too complex to be managed by the engine (e.g. a check which cannot be expressed by the engine syntax).

The enrichment and validation step achieves these two points in a single procedure callable directly within the Business Rules Engine. More precisely, the DEAV procedure is wrapped inside another process whose implementation is Business Rules specific. In particular, a mapping and a post-processing phase are added to the generic DEAV procedure explained in Section 2.

* 1. Mapping

For technical reasons, the Business Rules Engine has to use a unique macro to run all the DEAV procedures. This leads to a macro which allows setting which DEAV procedure is required and the related set of required parameters (e.g. the FoodEx2 validation requires to specify the column containing the FoodEx2 codes to validate).

Therefore, a pre-processing phase which maps the interface parameters to these values to the real DEAV input parameters is required. In particular, the interface calls another macro which specifically runs the required DEAV.

* 1. Post-processing

The DEAV procedure is a generic validation procedure which is not intended to be used just inside the BRE. In fact, the format of the DEAV errors is not compatible with the one needed by the BRE, because encapsulating the BRE logic in it would lead to a weak piece of code which is not reusable outside of the BR domain. This implies the existence of a post-processing step, which maps the DEAV output to a format which is suitable for the BRE. In particular, the post-processing creates the conventional error table which is used inside the BRE (named FAILED\_RECORDS), in order to simulate the behaviour of a standard business rule and thus minimizing the effort to adapt the BRE to run the DEAV processes.

Furthermore, the post-processing phase is also in charge of copying the enriched columns (if any) to ORACLE, in order to make them visible to ETL processes.

* 1. Summary schema