Master Data Management (MDM)

SI Workstream Project

Exported on Jun 2, 2020 2:45 PM

Table of Contents

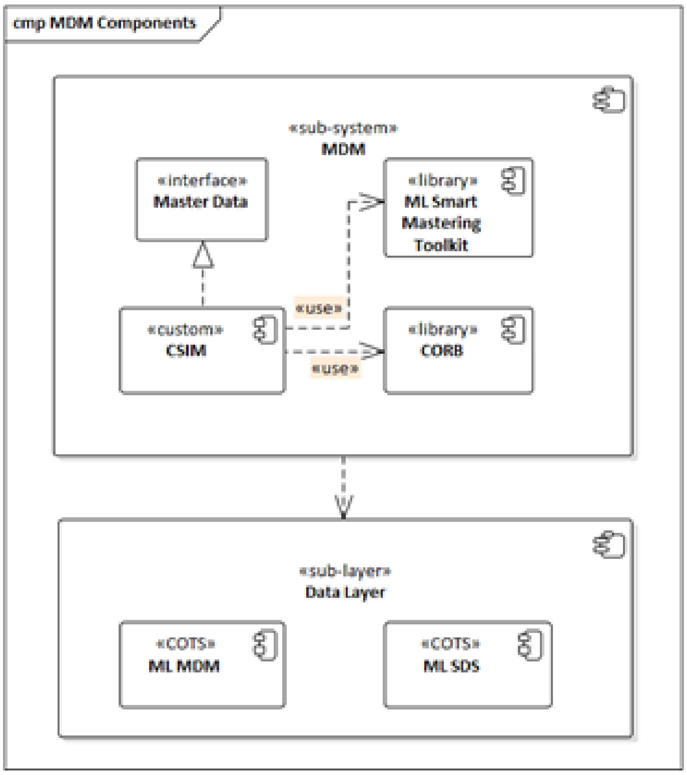
The Master Data Management (MDM) (or Data Mastering) is a method used to define and manage the critical data of an organization to provide a single point of authoritative reference to it. The data that is mastered may include reference data- the set of permissible values, and the analytical data that supports decision-making. This section describes the process for data migration from SMR into MDM as well as the tools utilized. As part of this step, multiple source entity records are matched and merged. The section describes process flow with detailed steps. The following are some examples of the steps employed in this phase:

- Determine potential matches for each entity
- Merge/unmerge records based on a threshold
- Manage potential matches
- Define fallout management approach

The CSIM is the process where multiple source entity records are matched and merged. During this process, data across all sources are matched and merged, data is survived, and a preferred record is identified to assist in the creation of an entity profile and subsequently a consolidated view of a member.

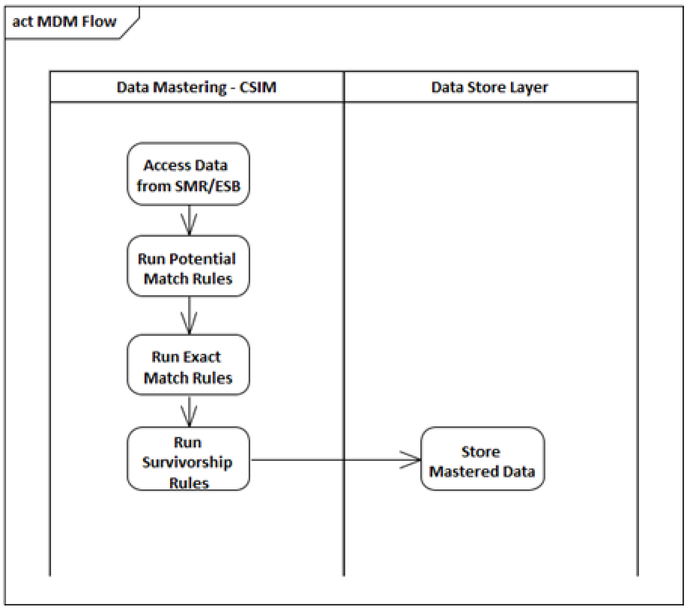
The following figure shows the logical components of MDM.

MDM Components



The overall process flow is described in the figure below.

MDM Process Flow


act MDM Flow
Data Mastering - CSIM
Data Store Layer
Access Data from SMR/ESB
Run Potential Match Rules
Run Exact Match Rules
Run Survivorship Rules
Store Mastered Data

1 Step 1: Determine potential matches for each entity

The first step in the de-duplication of entity records within MDM is to identify potential matches. Based on the tool used this could be broken down into a two-step process. The steps are:

- **Candidate List Builder**: In step 1 (Candidate List Builder), a few Specific Criteria would have to be identified that use a direct search on the entire set of SIM results across sources in the database. This process starts with one document in the SIM results and using the criteria finds all the other SIM documents that fulfill at least one of the criteria among them. This is done to reduce the number of compare operations that require lifting the documents from the disk. The candidate list builder rules are meant to cast a wider net across all the source records. These rules should follow a few key best practices:
- The rules should use match on one or more of the critical data elements. For example, fuzzy name or DOB or SSN match can be a candidate list builder rule for a person entity.
- The rules should be loose enough that no SIM entity that can match the incoming entity is left out. For example, a fuzzy name match alone can be used as a candidate list builder rule. However, SSN match should not be used as a candidate list builder rule alone unless SSN is the only matching criteria for a Client entity.
- The rules should be tight enough to not bring such a large result set, that the performance is impacted. For example, Address State or Zip Code alone may be too wide a criterion. It should be used with perhaps a DOB in conjunction.
- **Identify Potential Match:** In step 2, a set of rules will be identified that use a few fields in a match combination on the candidate list of SIM documents that have been identified in step 1. If there is more than one rule identified for this step, there would be an order in which these rules are applied on the list. Every document returned in the candidate list is compared to determine one or many potential merge groups. These potential merge groups then can be passed on for forming an entity using the survivorship rules. All these rules are typically associated with a match score and the combined match score can be compared with the merge/unmerge threshold to determine whether the potential matches are sent to automatic merge or a queue for data stewards to review.

2 Step 2: Merge/Unmerge

Once the identification of the potential matches is done and a match score is calculated for a group of source records, the match score is compared with the merge thresholds. Comparing the match scores with the merge thresholds may result in three scenarios that are automatically handled by the CSIM module:

- **Automatic Merge**: If the match score for a match group is above the “Automatic Merge Threshold”, then all the source records in the match group are combined together and a new CSIM/MDM document is created. This new document is assigned a new unique identifier that becomes the enterprise entity identifier for the entity. To derive values to survive from the source records, a set of survivorship rules are applied. If a conflict exists where even after all the survivorship rules are applied, then the matched pair is considered fallout and would be handled through the fallout management process.
- **Discard Match Group**: If the match score for a match group is below the “Non-match Threshold,” then all the source records in the match group are discarded as a match of each other and no further processing would be done on those candidate records.
- **Automatic Unmerge**: If because of an update of a source record, and after running the match rules, the match scores of the source records are determined to be below the “Non-match Threshold,” then an existing entity can be required to be unmerged and it is broken down into the source records and the matching rules are applied to each of the source records once again to recreate new match groups.

3 Step 3: Manage potential matches

A potential match is defined as a set of source records that belong to a match group once the match rules have been applied to them. If the match score for a match group is between the “Automatic Merge Threshold” and the “Non-match Threshold,” then the match group is marked for review by the data stewards.

A data steward can potentially mark the match group as ready for merge, then the survivorship rules would be applied, and a merged record is created.

If a data steward determines that the match group is not ready for merge, then the match group is discarded, and no further processing is done on them.

While the incremental Change data capture process is in progress, the source records and match groups that are part of the data stewardship review queue can be handled in two different ways:

- **Continuous Updates**: This way the data stewardship review queue is constantly updated as new records are ingested or updates are made on the existing source records, as every new record and every update can change the potential match groups. This could mean that while the data steward is reviewing a match, the data may have updated in effect nullifying the review process altogether.
- **Locked Updates**: This way the source records and match groups in the data stewardship review queue are locked and no further updates would be accepted into the MDM until the process is resolved. This could mean that the MDM would not remain current with all the updates from the source system and all updates to source record are queued up until the issue is resolved.

4 Fallout Management

Within the context of MDM, the fallout can be defined as the records that cannot be processed by the rules defined within the MDM. One of the essential features of MDM is to be able to identify the records that get match scores that meet a threshold but are lower than the threshold for merging and marked for human review. Based on these identifications, the data stewards are expected to perform data improvement tasks at the source systems.

- **Not enough data/No minimum data set**: If a source record arrives at MDM, which does not contain a set of valid values defined in the “Minimum Data Set,” it becomes impossible for MDM to apply the matching rules. This can lead to issues with data quality within the MDM. Thus, every source record is checked before ingesting into MDM for the availability of valid values for “Minimum Data Set” elements. If the record does not pass this check, it is considered fallout.
- **Conflicting data values**: Once a match group has been identified for an automatic match, the survivorship rules are applied on the data to combine the source record values into a master entity record. If there are not enough rules identified to mitigate a conflict that exists in the data of the records within a match group, then the entire match group is marked as fallout.

Once fallout has been identified, they are handled in one of the following ways:

- If fallouts are related to data issues and source data needs to be corrected, the fallout records are added to a fallout queue for the data stewards and data administrators to review and update the data in the source system themselves.
- If fallouts are related to systemic issues and policy around the data or rules within the MDM needs changing, a change request would need to be raised and a new MDM rule would be added, or an existing MDM rule would be updated. This would mean that once the change is applied, all the records within MDM would need to be re-evaluated using the new rules. Depending on the magnitude of the change, it can be applied in two different ways:
- As a trickle-down update, where the new rules are applied only when an existing entity is touched by MDM.
- As a total refresh update, where the system halts processing of new records and the entire MDM data is re-evaluated with the new set of rules and re-synched with the source systems.

5 Tools for data migration from SMR into MDM

The MDM is developed using the Smart Mastering libraries provided by MarkLogic and is contained within the MarkLogic database. The data in MDM and the SMR are stored in separate instances of the MarkLogic database. Hence, data migration is required to move the data from SMR into the MDM. The tools required for the data migration process are as follows:

Smart Mastering Framework

MarkLogic provided Smart Mastering Framework include a set of RESTful API extensions. The MDM core API (MAC) consists of REST API endpoints to invoke mastering and merging. These APIs are configuration driven to take parameters as input for the API. The API suite also includes services to retrieve history for documents or individual properties within the merged documents. Some of the key APIs provided by this suite are:

- Match – This API identifies the list of documents matching the given document.
- Merge – This API saves or provides a preview of a merge document, combining two or more other documents. The delete option on the same API will unmerge a previously merged document, restoring the original documents.
- Match-And-Merge – This API provides the convenience of calling both the match and merge in a single API.
- History - Smart Mastering Core tracks the match and merge history of a document, as well as its provenance.
- Notifications – Notifications API identify the potential matches but did not score high enough to automatically merge. These are then presented to the data stewards for manual merge.

In addition to these APIs, there are other utilities and miscellaneous APIs focusing on nonfunctional aspects of the smart mastering library like statistics, dictionary, etc.

MDM XQuery Libraries

The XQuery libraries are structured to separate the API from the implementation. The APIs acts as a facade to external consumers and does not undergo drastic changes. However, the XQuery APIs continue to evolve to make the MDM operations efficient and robust. These XQuery libraries are persisted inside the MarkLogic's modules database. The modules database is an auxiliary database that is used to store executable XQuery, JavaScript, and REST code. These libraries are loaded inside MarkLogic with execute permissions. The loading operations along with setting security privileges are handled via Gradle scripts. Some of the key XQuery libraries used by the Smart Mastering Core are:

- Matcher.xqy – This module provides functions to store, retrieve, delete, and list match options; find potential matches for a document; and to store, retrieve, delete, and list match blocks.
- Merging.xqy – This module provides functions to build (preview), save, or remove merged documents and to store, retrieve, delete, and list merge options.
- Process-records.xqy – This module provides two functions to run through both matching and merging for a particular document.
- Match-And-Merge-Trigger.xqy – This module implements a trigger to process matching and merging any time a new document is inserted into the content collection.

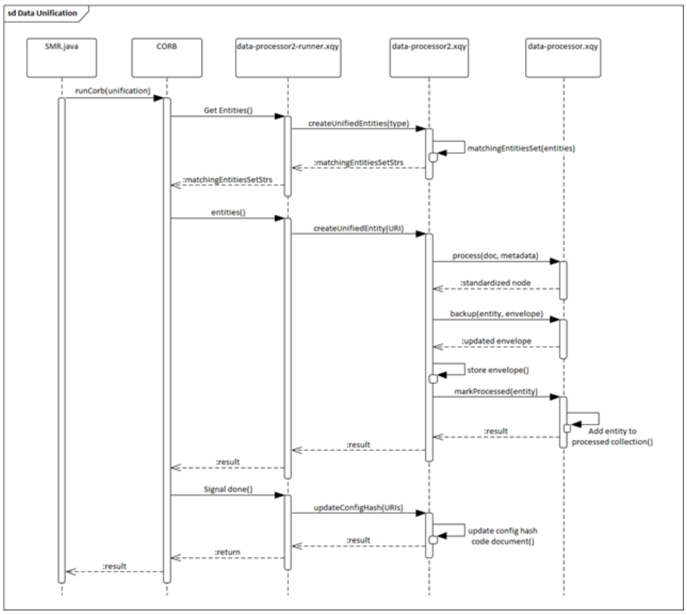
The MDM function in the SI Platform is achieved by executing the following custom XQuery programs. These programs are executed by invoking a configurable Java program called SMR with “Unification” as the parameter. The program follows the following steps:

1. java invokes a CORB function that integrates the different parts of the MDM.
2. The CORB program determines the potential matches.
3. For each entity retrieved, the program attaches the metadata and creates the entity if the document is not existing already.

4. If the document exists, it performs the unmerge and then merge based on the new matches.

The sequence of method calls is provided in the following figure.

CSIM Sequence Diagram


sd Data Unification
SMR.java
CORB
data-processor2-runner.xqy
data-processor2.xqy
data-processor.xqy
runCorb(unification)
Get Entities()
createUnifiedEntities(type)
matchingEntitiesSet(entities)
:matchingEntitiesSetStrs
:matchingEntitiesSetStrs
entities()
createUnifiedEntity(URI)
process(doc, metadata)
:standardized node
backup(entity, envelope)
:updated envelope
store envelope()
markProcessed(entity)
Add entity to
processed collection()
:result
:result
:result
Signal done()
updateConfigHash(URIs)
update config hash
code document()
:result
:return
:result