Paper 129-2011

How TV 2 Improves Data Quality and Facilitates a Unified Customer View for Optimizing the Customer Experience

Terje Vatle, SAS Institute, Oslo, Norway

Svein Ola Gundhus, TV 2 ASA, Oslo, Norway

ABSTRACT

Norway's largest commercial television broadcaster, TV 2, wanted to improve the customer experience by better targeting customer interests in their one-to-one campaigns. They were challenged with having multiple data sources with different customer definitions, and they were not able to get a unified customer view showing interests, usage patterns, and purchases for each customer. In this presentation, TV 2 will share how the use of DataFlux® and SAS® have given them a single customer view that they can use for analyzing customer behavior to better understand customer needs. In addition, they will share how this knowledge will be used for targeted one-to-one campaigns and share thoughts on some of the legal and business ethical considerations they are facing.

INTRODUCTION

Being a television broadcaster TV 2's media content is categorized into news, sport and entertainment, which in turn is distributed multi-platform through digital broadcasting and cable distribution on the Web, Web TV and mobile services such as apps. Web TV allows for both live and on-demand viewing.

The introduction of new services and products on new platforms provides TV 2 with access to customer databases in addition to traditional TV metering. This is a paradigm shift for TV 2's ability to gain customer insight which is considered a key tool to help strengthening the competitive edge by supporting the development of new products and help positioning the media house for the next shift in distribution technology. However, the customer databases exist under specialized legacy-systems with different customer definitions which makes the consolidation difficult.

Since there is no exact correspondence between customers in the various data sources for example due to misspellings in fields such as names and addresses, the idea was to extend the capabilities of traditional data integration in SAS® with advanced fuzzy-matching provided with DataFlux® (a SAS® company).

This paper proposes a pattern and key experience points on how to apply DataFlux® together with SAS® to create a enterprise wide unified customer view. Note that the pattern can also be applied to other areas where we need a unified view for example products, materials, offerings and client relations.

There are four main sections of this paper: Technology brief, generic pattern, case study and conclusion. The technology brief is a short introduction to main data quality components of DataFlux® and SAS® and how we call DataFlux® jobs from SAS DI Studio®. The generic pattern is a step-by-step tutorial from designing the business rules to implementing each type of job in SAS DI Studio® and DataFlux dfPower Studio®. In the case study we apply the generic pattern on example rules of matching logic and elaborate how to meet requirements such as for example traceability between design and implementation. Our experience from using the pattern, including benefits and possible future improvements is summed up in the conclusion.

TECHNOLOGY BRIEF

DataFlux® provides a mean for measuring the degree of similarity between two data elements. We can compare data elements in a variety of contexts assisted by DataFlux® tools that use a comprehensive data quality knowledge base. This knowledge base is supported in many locales since the interpretation of a context is language dependent. For example, there are different sets of names in English and Norwegian so subsequently typical misspellings and abbreviations will also differ.

The required data quality software from SAS® and DataFlux® is part of some of the enterprise software bundles for SAS 9.2 Intelligence Platform®.

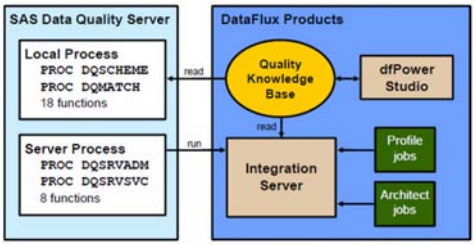


Figure 1: Functional overview of SAS Data Quality Server Software

In context of this paper we will investigate Architect jobs being deployed on DataFlux Integration Server® and consumed as services by SAS Data Integration Studio® as part of the SAS Data Quality Server®.

DATAFLUX® QUALITY SOFTWARE

- Quality Knowledge Base (QKB): Contains locale-specific data type definitions that DataFlux® uses to parse, standardize, match and process data.
- dfPower Studio®: Suite of DataFlux® software, contains i.e. dfPower Architect which enables us to create jobs and services for profiling and data quality to be deployed on the DataFlux Integration Server®.
- DataFlux Integration Server®: Service-oriented architecture application providing full access to all DataFlux® functionality from within SAS. It is highly scalable since complex data quality operations are run in a server environment close to where the data resides. Converting batch jobs into services enable data cleansing operations to be executed real-time by any Web service enabled application.

SAS® DATA QUALITY SOFTWARE

- Quality Knowledge Base (QKB): This QKB is provided by DataFlux® and enables SAS® to perform certain data quality operations without calling DataFlux® jobs or services.
- SAS Data Quality Server: Provides language elements enabling the developer to analyze and cleanse data in SAS® by using the local QKB. It also gives us a SAS® programming interface to DataFlux Integration Server®.

MAKING DATAFLUX® AND SAS® WORK TOGETHER

Se the following configuration check-list, please refer to SAS® documentation for a detailed configuration of DataFlux® and SAS® applications.

1. Use the DataFlux Integration Server Manager® to Configure the DataFlux Integration Server® with options such as server name, port number and timeout.
2. Define an HTTP server in SAS Management Console® representing the DataFlux Integration Server®.
3. Create jobs and services in DataFlux Architect®.
4. Use the DataFlux Integration Server Manager® to upload DataFlux® jobs and services and thereby exposing them to SAS®.
5. Configure the Data Quality options in SAS Data Integration Studio® in order to access the DataFlux Integration Server®.
6. Use the built-in nodes in SAS Data Integration Studio® to call a selected DataFlux® job or service as part of the SAS® job flow.
7. Monitor the execution of DataFlux® jobs and services in the DataFlux Integration Server Manager®.

GENERIC PATTERN

PRECONDITION: ANALYSIS AND DESIGN

Capturing Business Rules

We start out defining a set of business rules to identify the customers that, despite having different technical IDs in the data sources, indeed refer to the same individual. Ideally, the implementation of the rules will help our business to see the unique set of individuals that are our customers.

Simple business rules for a single data source could be:

1. *Two customers are the same individual if name and address are the same.*
2. *Two customers are the same individual if name and phone number are the same.*

3. *If two customers are identified as the same individual then store the newest information available from each customer on that individual.*

The following questions can be considered:

- What does it mean for two information elements to be the same? For example, we would assume phone numbers should be matched more accurately than names.
- Is it sufficient if only one single business rule applies or must all business rules apply at the same time?
- Would some of the business rules produce a more reliable match than others?
- How do we identify the newest information among the customers that match? Should we check for any potential glitches after such a consolidation? For instance if we consolidate two customers and see a difference in gender it might be indicating that the match was wrong.
- Can we be provided an example set of customers helping us to confirm or reject the logic of the business rules? This could later be used for unit testing.

In case we have several data sources we must first define business rules for each data source secondly we define rules across all source systems. See also Step 2: DataFlux®: Matching and Consolidation.

Analyzing Relevant Data Fields

In order to analyze the information elements that are referred to in the business rules you can apply the DataFlux Profile Configurator® and Profile Viewer® available as part of dfPower Studio®. These provide access to frequency distributions, pattern frequency distributions, percent missing/empty values, min/max values and more. For example it would be helpful to identify that certain fields have overall higher levels of data quality or better accuracy when we use these fields for comparison. For example it might be that email addresses are verified as part of the customer registration process whereas for example addresses are often missing or misspelled.

Note that in a real-world situation the criteria for matching two customers are likely to consist of several rules and must be tested thoroughly on a representative set of customers in order to verify the correctness and completeness of the rules.

Designing Matching Rules

Matching rules should be expressed in a way which removes potential ambiguity in the business rules. For example:

Matching:

1. Name (middle accuracy, *do not* standardize)
   AND Address (middle accuracy, standardize)

   OR

2. Name (middle accuracy, *do not* standardize)
   AND Phone number (high accuracy, standardize)

Consolidation:

- Within each group of customers identified as a match: For each field keep the version where field value is different from null and the updated-timestamp for that record is max among the group.

STEP 1: SAS DI STUDIO®: PREPARING DATA AND CALLING DATAFLUX

In overview the pattern consists of three steps after analysis and design is completed:

1. SAS DI Studio®: Preparing Data and Calling DataFlux
2. DataFlux®: Matching and Consolidation
3. SAS DI Studio®: Populating Enterprise Data Model

Step 1.1: Preparing Data in SAS

As a performance measure we limit the number of input columns to the minimum needed for the processing of the matching rules in DataFlux®. It is typically the data fields mentioned in the rules as well as IDs and date fields indicating update times. This initial step also holds any standardization in SAS® on field types and field formats, and the possibility to filter the input customers for testing purposes.

Step 1.2: Calling DataFlux Jobs from SAS DI Studio®

The prepared data sets from SAS® are used as input for calling jobs in DataFlux® as shown in the following sequence diagram.

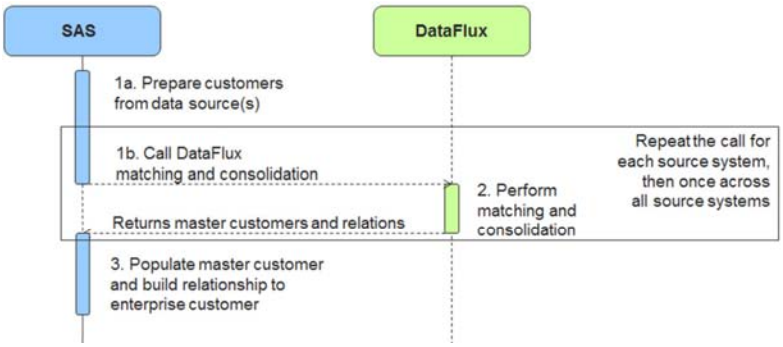


Figure 2: Calling DataFlux® from SAS®

The results from the matching and consolidation in DataFlux® is a set of unique customers referred to as master customers. In addition we get the relationship from the master customers to the original input set of customers which enables us to tell exactly which customers are identified as duplicates.

If we have multiple data sources, we let SAS® call DataFlux® once for each consecutive data source making it flexible to future extensions. Then, we append the resulting set of master customers from each data source into a single table and perform the matching and consolidation once across all data sources. This produces a single set of enterprise wide master customers.

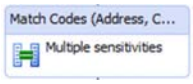
STEP 2: DATAFLUX®: MATCHING AND CONSOLIDATION

We define matching as the process of grouping customers that are identical based on each matching rule. This is explained in chapter Step 2.1: Matching in DataFlux®. Consolidation is referred to as the process of combining the results from all matching rules and constructing the master customer, see Step 2.2: Consolidation in DataFlux®.

Step 2.1: Matching in DataFlux®

Note that data standardization is part of the match code generation algorithm so extra standardization should not be necessary.

Step 2.1.1 Creating Match Codes



DataFlux® provides fuzzy-matching technology that can process a textual string such as a person's name and generate a match code which is an interpretation of the concentrated meaning of the words specific to the locale and the subject of your match. Each locale has their own set of matching definitions specialized on subjects such as names, addresses, phone numbers and email addresses.

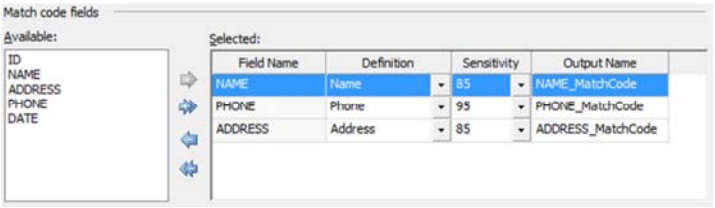


Figure 3: Setting properties for match codes

When the same name is written in different ways, for example as a result of manual tipping at the customer registration desk, the idea is that the generated match codes will be identical. Then if identical names were the only criteria for saying two customers are the same individual, you could group by the corresponding match code in order to identify the unique set of individuals.

For each match code generation in DataFlux® the developer must decide on an appropriate level of sensitivity for the matching algorithm. A higher sensitivity yields a more exact match code and will therefore require a higher degree of similarity between to field values before we can say that two values are the same. Lower sensitivity will on the other hand make us accept a lower degree of similarity. 100% sensitivity means an exact match and is typically used for phone numbers or email addresses. In Figure 4 we see match codes generated using three different sensitivities on an address field.

ID	ADDRESS	ADDRESS_MatchCode_70	ADDRESS_MatchCode_85	ADDRESS_MatchCode_95
1	9171 Town Center Dr	-ZIZ~LP$$$$$$$$	-ZIZ$~LPJ$$$$$$	-ZIZ$~LPJ~$$$$$
2	9171 Town senter drive	-ZIZ~LP$$$$$$$$	-ZIZ$~LP4$$$$$$	-ZIZ$~LP4P$$$$$
3	9171 Town cntr	-ZIZ~LP$$$$$$$$	-ZIZ$~LPJ$$$$$$	-ZIZ$~LPJ~$$$$$

Figure 4: Generated match codes with different sensitivities

Here we can see that input data despite being spelled slightly different generates an identical match code for all three records on the lowest sensitivity. When the sensitivity increases so does the precision yielding only two records with identical match codes.

The DataFlux® algorithm creating the match codes is in short the following:

1. Normalization (i.e. setting all letters capital and applying regular expressions)
2. Removing noise (i.e. removing semantically unimportant information)
3. Standardization (i.e. common way of writing for example a street name)
4. Phonetics (i.e. spelling and sound-like algorithms)

Fine-tuning sensitivities is a subject to experience as well as quality of the information you are matching. A tip is to use default 85% sensitivity for i.e. names and street addresses, and 95% or exact for email addresses and phone numbers. A step-by-step approach to tuning:

1. For each field you are comparing produce a limited subset of data where you know which field values should be identified as a match.
2. Create for example a DataFlux Architect® job where you branch the input data in two parallel flows. In the first flow generate match codes with standard sensitivities. In the second flow alter just one single sensitivity. Let the two branches meet in a Cluster Diff node to compare the differences between the clusters.
3. Run the job and identify which of the combined sensitivities yielded the best results. Once again, alter one single sensitivity in the second flow, repeat the run and compare the results. Continue until the optimal combination of sensitivities between the fields is found. Note that the sensitivity is available in steps of 5%.

Step 2.1.2: Applying Match Rules Through Clustering

After the match codes have been generated for each data field used in the matching rules we implement the rules by using the clustering abilities in DataFlux®. It means we specify on which fields we would like to group our entities.

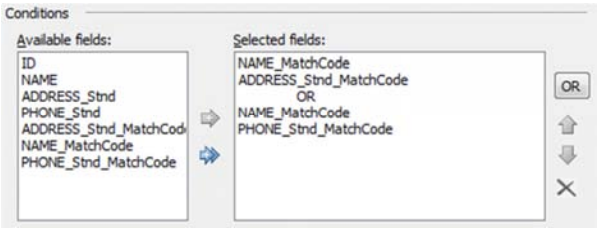


Figure 5: Clustering - implementing matching rules

The clustering produces a numeric value for each row where identical numbers indicate that the corresponding records are regarded as belonging to the same group and thus the same individual. In Figure 5 we implement the matching rules 1 and 2 in a single clustering node. The two matching rules are separated with “OR” indicating that both rules are weighted equally. Note that future releases of DataFlux are planned to provide functionality to specify individual weighting of matching rules.

Step 2.1.4 Alternative Approaches to a Job Flow

As shown in Figure 6 we identify at least two approaches to creating a job in DataFlux that performs standardization, generates match codes and clusters based on matching rules.

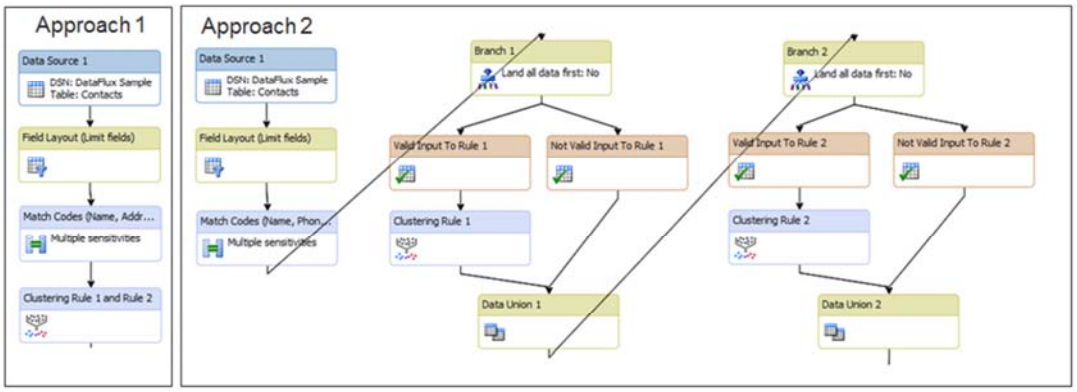


Figure 6: Two approaches to matching

Approach 1 puts all matching rules into one single clustering node. This approach is preferable when:

- The matching rules only consist of a chain of positive terms on the form *(A and B) or (X and Y)*.
- There is no need to differentiate between customers matching on a single matching rule and on those matching on more than one rule.
- It is not required to see the effect of a single matching rule on the final outcome, in stead all matching rules are always considered combined.

ID	NAME	NAME_MatchCode	PHONE	PHONE_MatchCode	ADDRESS	ADDRESS_MatchCode	DATE	CLUSTER_ID
1	James E. Briggs	MYF$$$$$$$C&B_4$$$$	450-157-0772	$$$$$$$$XXXS50Z5I0IIH$	19 East Broad Street	Z-$$$MY~$$$$$$$	14.05.1998 00:00:	0
3	Mr James Brigs	MYF$$$$$$$C&B_4$$$$	950-886-6346	$$$-50$$XXXDD66KS6$$$$	19 E Broad St	Z-$$$MY~$$$$$$$	17.03.1998 00:00:	0
4	Jim Briggs	MYF$$$$$$$C&B_4$$$$	717-977-1810	$$$IZI$$XXX-IIZDZ0$$$$	19 E. BROAD ST.	Z-$$$MY~$$$$$$$	09.12.1997 00:00:	0
10	Bob Brauer	MYLY$$$$$$$M@M$$$$$$	323-198-3282	$$$$$$$$XXXXHKZ-DKHDH$	6512 Six Forks Road - 404B	65ZH$6GY3S0SM$$	04.05.1998 00:00:	1
11267	Bob Brauer	MYLY$$$$$$$M@M$$$$$$	(Null)	(Null)	6512 Six Forks #404B	65ZH$6GY3S0SM$$	12.11.1997 00:00:	1
9054	Robert Brauer	MYLY$$$$$$$M@M$$$$$$	(Null)	(Null)	6512 Six Frks Road Ste 404B	65ZH$6GY3S0SM$$	12.01.1998 00:00:	1

Figure 7: Sample result from approach 1

In Figure 7 we see a subset of the results from approach 1 where CLUSTER_ID indicates that three by three customers are regarded the same master customer according to matching rule 1 and 2. If we had several matching rules it could be difficult to deduct which rule contributed to the final cluster. Note that with approach 1 you should consider using the option “Generate null match codes for blank values” in the match code generation to avoid empty fields from being matched.

Approach 2 lets us process each matching rule sequentially and therefore yield a cluster for each matching rule we apply. This approach is preferable when:

- We want to limit the processing of a validation rule to those customers that adhere to a specific validation rule. Validation rules requiring the input fields not being null is similar to using “Generate null match codes for blank values” in the match code generation.
- We need to investigate the correlation between a single matching rule and the final outcome. For example if we need to indicate two customers matching on more than one rule as a more reliable match.
- There is a complex matching rule not supported OOTB in DataFlux® for example matching rules containing negative terms such as *(A and not B) or (not X and Y)*. In this case the clustering node could be exchanged with custom code.

ID	NAME	NAME_MatchCode	ADDRESS	ADDRESS_MatchCode	PHONE	PHONE_MatchCode	DATE	CLUSTER_1_ID	CLUSTER_2_ID
1	James E. Briggs	MYF$$$$$$$C&B_4$$$$	19 East Broad Street	Z-$$$MY~$$$$$$$	450-157-0772	$$$$$$$$XXXS50Z5I0IIH$	14.05.1998 00:00:	0	0
3	Mr James Brigs	MYF$$$$$$$C&B_4$$$$	19 E Broad St	Z-$$$MY~$$$$$$$	950-886-6346	$$$-50$$XXXDD66KS6$$$$	17.03.1998 00:00:	0	1
4	Jim Briggs	MYF$$$$$$$C&B_4$$$$	19 E. BROAD ST.	Z-$$$MY~$$$$$$$	717-977-1810	$$$IZI$$XXX-IIZDZ0$$$$	09.12.1997 00:00:	0	2
10	Bob Brauer	MYLY$$$$$$$M@M$$$$$$	6512 Six Forks Road - 404B	65ZH$6GY3S0SM$$	323-198-3282	$$$$$$$$XXXXHKZ-DKHDH$	04.05.1998 00:00:	1	3
11267	Bob Brauer	MYLY$$$$$$$M@M$$$$$$	6512 Six Forks #404B	65ZH$6GY3S0SM$$	(Null)	(Null)	12.11.1997 00:00:	1	(Null)
9054	Robert Brauer	MYLY$$$$$$$M@M$$$$$$	6512 Six Frks Road Ste 404B	65ZH$6GY3S0SM$$	(Null)	(Null)	12.01.1998 00:00:	1	(Null)

Figure 8: Sample result from approach 2

Figure 8 shows the exact same data as with approach 1 in Figure 7 only with two distinct CLUSTER_IDs, one for each matching rule. Thereby we can decide how we want to combine the results from the two matching rules. The combination is done as part of the consolidation process.

Step 2.2: Consolidation in DataFlux®

The two alternative job flows for matching require different handling during consolidation.

Step 2.2.1 Preparing IDs

In both cases the primary key must be converted to an integer, either provided by the sequencer or explicitly converted to an integer as part of custom code. Also, any null-values of a cluster ID should be set to the value of the sequencer. The sequencer’s initial value should be set well above the maximum expected numeric ID from the data sources. If this is set too low it can be interpreted as a match with another record having the same cluster ID.

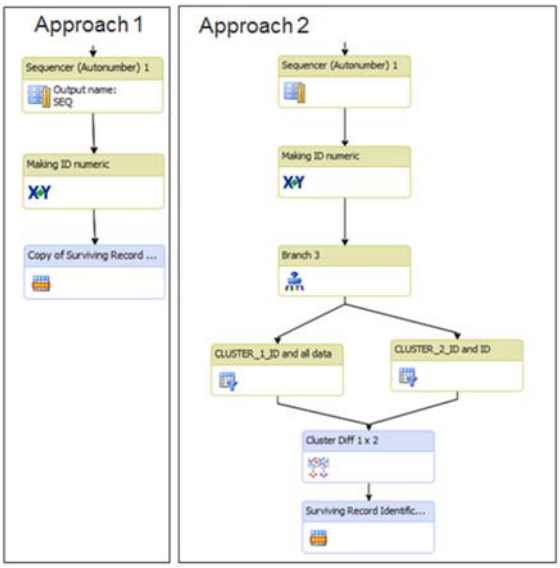


Figure 9: Consolidation job flows

For example, if we had 5 matching rules we would do cluster diff four times: 1x2, 12x3, 123x4 and 1234x5 continuously increasing the final super cluster. We use the same Cluster Diff node as for tuning match code sensitivities.

Step 2.2.1 Multi Cluster (if approach 2 is chosen for matching)

Having applied multiple matching rules the question is how to identify the greater combined set of customers across all clusters. DataFlux® provides the Cluster Diff node which helps us comparing two clusters at a time and find the differences between them. In the example above we compare the outcome of cluster 1 with cluster 2 to produce a super cluster 1x2. If there were further clusters the super cluster 1x2 would be compared with cluster 3 to produce super cluster 12x3 and so on until all clusters are covered.

Step 2.2.2: Surviving Record Identification

Record rules determine which of the customers within a group will be basis for the master customer. The order of the customers is important since if no record rule apply then simply the first record in the group will be selected. Field rules decide how to select field values from the different customers in the same group now being combined into one master customer. For example you may want to generate a new row for the master customer and use the most up-to-date non-null field values that occur among the customers within the group.

Step 2.3: Repeat the Matching and Consolidating for Each Data Source

In case there are multiple data sources we repeat the matching and consolidation as described in 2.1 and 2.2 until all data sources are covered.

Step 2.4 Repeat the Matching and Consolidation Across all Data Sources

In case there are multiple data sources we repeat the matching and consolidation as described in 2.1 and 2.2 once across all data sources. Final output from DataFlux® should contain:

a) In case of only one data source
   - Set of master customers
   - Set of customers in the data source with relation to the master customers

b) In case of multiple data sources
   - Set of customers of each data source with relation to the master customer of that data source
   - Set of master customers from each data source with relation to enterprise master customers
   - Set of enterprise master customers (master customers across all systems)

STEP 3: SAS DI STUDIO®: POPULATING THE ENTERPRISE DATA MODEL

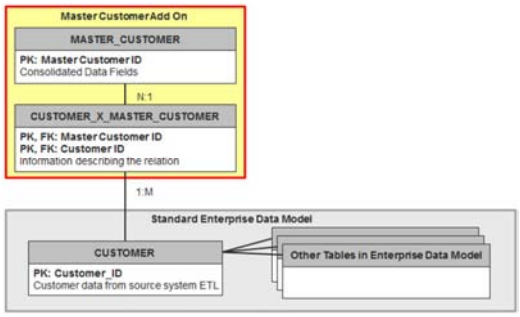


Figure 10: Extended Enterprise Data Model

Taking one step back, the results from DataFlux® must be fitted into our enterprise data model. This model contains enterprise wide entities such as customer, subscription, purchase, product, organization and payment. Each entity is populated from different tables in the data source so the enterprise customer entity in general contains a record for each customer record in the data sources with an enterprise wide customer ID. This model is extended with the master customer entity with a many-to-many relationship to the customer.

This design of decoupling the master customer from the customer provides a series of benefits:

- It simplifies the population of entities that relate to customers such as subscription and payments, since they can still relate to the customers in the data sources with a fixed rule of creating a enterprise wide unique customer IDs. These IDs don't have to be updated on historic records in case the matching and consolidation in DataFlux® finds a new set of master customers.
- We can introduce the master customer onto an existing data model or postpone it to a later project phase.
- We expect business rules to be subject to frequent change and tuning as soon as we see the results from the matching process. Decoupling enables us able to keep track of changes in the relations between customers and master customers i.e. against a version number on the business rules.
- By experience, implementing and testing matching rules in DataFlux® is a complex process. Decoupling isolates the master customer thus making it more efficient to debug and correct errors.

Note that several events that can change the set of master customers produced by the matching in DataFlux®:

- The most obvious reason is a change in matching rules. It can produce a new set of master customers without change in input customer data simply because we alter the ways of identifying duplicate customers.
- Another reason is new customers are added or existing customers either updated or removed in the data sources. If for example customer B is the only connection point between the two customers A and C the previous master customer could consist of A, B and C. If B is removed or updated it might not provide the transitive connection between A and C anymore and A and C might therefore no longer be part of the same master customer.

Step 3.1: Joining in Additional Information

As the program flow returns to SAS® we map the results from DataFlux® to our enterprise data model. We can enrich data i.e. with lookups such as for postal codes, or adding fields previously removed. Note that adding back fields can only be done with information where no surviving record analysis is needed i.e. we can simply use the value from any of the customers that compose the master customer.

Step 3.2: Building the Relationship Between Customer and Master Customer

We need to establish the relationships from the customer to the master customer. In the case of a single data source we have this relationship directly from the results of the matching and consolidation. However, having more than one data source we promote the master customers from each data source to the matching and consolidation across all data sources, producing a new set of enterprise master customers. In the latter case we build the relationship in two steps, starting from enterprise master customers via master customers to the customers of a single data source.

BUSINESS CASE: 2 DATA SOURCES WITH TOTALLY 3,6 MILL CUSTOMERS

PROBLEM STATEMENT

TV 2 has several business units with separate sources to customer information:

- Web TV streaming services (referred to as Sumo): Approximately 100.000 customer records
- Mobile services (referred to as Mobil): Approximately 3,5 million customer records
- www-communities such as for special interests like weight clubs and motor clubs
- Conceptual Facebook groups controlled by TV 2 such as X-Factor, Premier League soccer and Tour de France cycling.

A unified customer view covering all business units will help TV 2 gain a complete 360 degree view on their unique set of customers which will be valuable for analysis and one-to-one communication.

Client Requirements

A. Results from the data quality solution must be fitted seamlessly into the enterprise data model
B. The data quality solution must run efficiently as part of the SAS® nightly batch
C. The matching rules must be versioned and traceable from design document to actual relationship between customers and master customers
D. It should be easy to extend the solution with new data sources
E. It should be easy to change, add or remove matching rules
F. It should be possible to target the master customer for campaign communications

Technical Challenges

1. How do we efficiently test the results of the matching and consolidation?

2. On what detail can we trace the matching rules?
3. Will the total runtime of the data quality solution be acceptable?

APPROACH

Precondition: Analysis and design

We undertook a data quality assessment on the two data sources Sumo and Mobile. For each data source we profiled the potential fields that could be used for identifying duplicate customers, in regards to completeness (% not missing) and quality of content. We saw that several fields could be used for matching rules internally in each of the data sources, but only a few across both systems. In the Mobile data source a fraction of the records had been previously enriched from external data providers and could be more extensively matched both internally and against the Sumo data source. As we specified the sequence of steps in the matching process these enriched mobile customers were handled as a separate logical data source. A design document was created with a versioned section defining the matching steps, need for standardization and matching rules for each step.

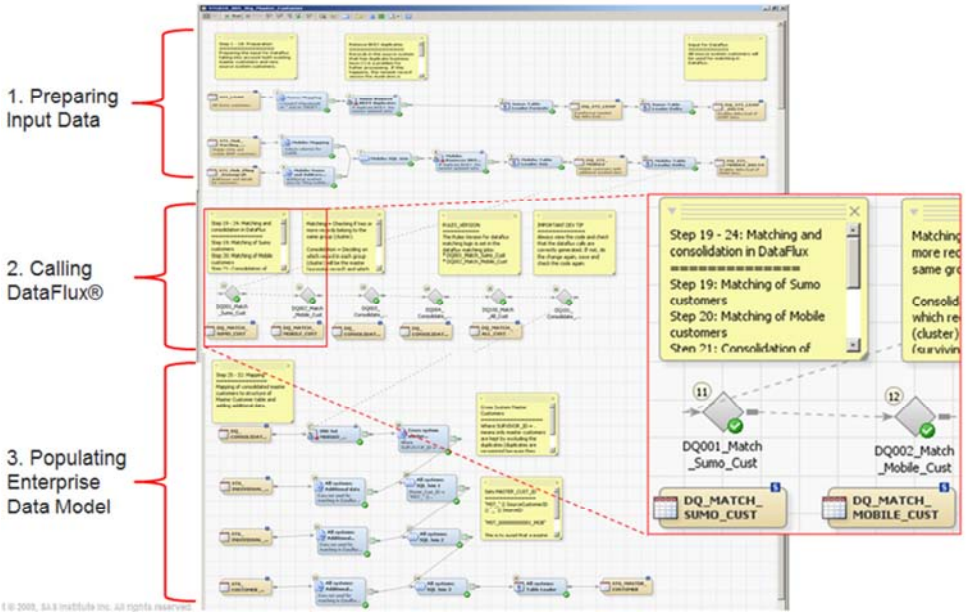
Answering Client Requirements

Requirements	Solution
A. Results from the data quality solution must be fitted seamlessly into the enterprise data model	As shown in Figure 10 in *Step 3: SAS DI Studio®: Populating the Enterprise Data Model* we extend the customer domain model with the Master_Customer table and the relationship table Customer_X_Master_Customer. The latter connects the Master_Customer table with the Customer table containing the customers extracted from each data source.
B. The data quality solution must run as part of the SAS® nightly batch	All DataFlux® operations are called from SAS® code generated from jobs in SAS Data Integration Studio®. These jobs are scheduled to run as part of the nightly batch just like any other job. Since potential error messages from DataFlux® runtime do not stop the processing of the SAS® job special error detection routines are developed scanning relevant log files in order to notify the development team.
C. The matching rules must be versioned and traceable from design document to actual relationship between customers and master customers	The complete set of matching rules is assigned a version number in the design document. Next version is defined to describe the changes from the last version. Each DataFlux® job sets a version field depending on the current version of the implemented rules and this field is kept as part of the tables Master_Customer and Customer_X_Master_Customer. As both tables are subject to Slowly Changing Dimension type 2 we keep the history of changes that might come i.e. as a result of new matching rules.
D. It should be easy to extend the solution with new data sources	We separate into matching and consolidation jobs in DataFlux® for each data source, making it easier to add new data sources in a modular way. Note that it still requires the jobs that performs matching and consolidation across all data sources to be adjusted if new data sources are added or previous data sources are removed.
E. It should be easy to change, add or remove matching rules	Matching rules are laid out sequentially with a free text description of each rule. Changing a rule will not affect the other rules as long as each rule is regarded equal, meaning that a match on rule X is just as good as match on rule Y. Rules can be added by extending the sequence where the order of the rules is insignificant. Rules can be removed by either deleting from the DataFlux® job or leading the dataflow outside a rule. If rules have been added or removed the consolidation job should be revised to assure it's still compatible.
F. It should be possible to target the master customer for campaign communications	A campaign mart is built as a star diagram with the customer entity in the center. A two step process was designed since TV 2 would like each individual customer to accept any potential mapping into a master customer before contacting using the unified view. At first, the campaign mart customer entity is equal with the data source customer, only with a grouping reference to the master customer for manual use of the marked coordinator. The campaign mart has been designed to make the switch from customers to master customers in the future also considering the campaign history that will have to be converted.

Answering Technical Challenges

Challenge	Solution
1. How do we efficiently test the results of the matching and consolidation?	For each matching rule we design a set of source system customers that will match and some that will not. Within the group of customers that match we compare the field values of each customer with the field values being propagated to the master customer as part of the surviving record identification. After having tested each data source we continue testing across all data sources. When loaded into the enterprise data model all customers extracted from the data sources must have a relationship to one and only one master customer. On the other hand, a master customer can be related to one or several customers. Finally, we evaluate random samples of master customers and related customers.
2. On what detail can we trace the matching rules?	At first, we examined the possibility of putting each single matching rule under version control. We quickly realized it would be difficult to maintain multiple active rule versions. Having one single rule version for the complete set of matching rules is considered a good tradeoff which is also easier to understand from a business perspective.
3. Will the total runtime of the data quality solution be acceptable?	Processing approximately 3.6 million customer records with 12 matching rules and populating the enterprise data model takes 1 hour to complete. This was considered sufficient from a performance perspective and is part of the nightly batch process always keeping the unified customer view up to date.

Step 1: SAS DI Studio®: Preparations



Preparations are done in the job which will later populate the table Master_Customer as shown in Figure 11.

We identify the division of the job into preparations, calling jobs in DataFlux Architect® and populating the enterprise data model. The zoomed-in nodes show the built-in capability in SAS Data Integration Studio® to call jobs from DataFlux Architect® that are published on DataFlux Integration Server®.

Figure 11: Master customer job in SAS Data Integration Studio®

Step 2: DataFlux: Matching and Consolidation

We have a total of six jobs in DataFlux Architect® which we deploy on the DataFlux Integration Server®. Four of the jobs were for matching and consolidating within each data source, and two jobs were for matching and consolidating across the data sources. In the following we see an overview of a matching job and a consolidation job in DataFlux®. Details such as node descriptions are blurred since the intention is to show the main elements of the job layout.

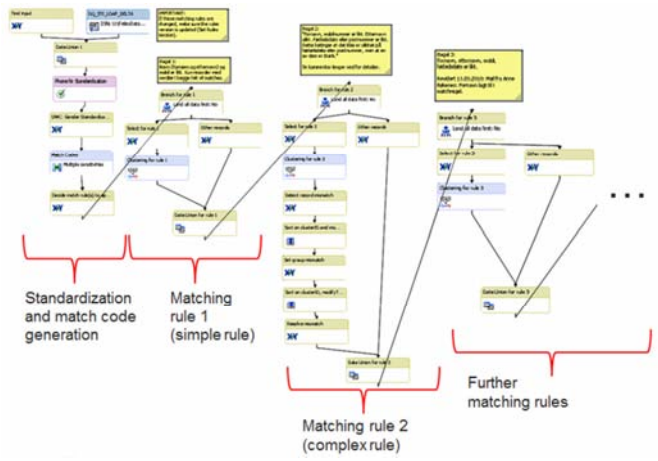


Figure 12: Matching job in DataFlux Architect®

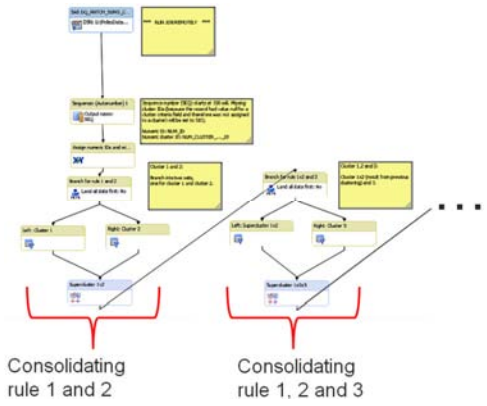


Figure 13: Consolidation job in DataFlux Architect®

Note the sequential processing and varying complexity of each matching rule. We implemented matching job approach 2 to better handle complex matching rules and see the impact from a single matching rule on the output master customers. In the consolidation job we gradually increased the number of rules that are consolidated until all rules are covered.

Step 3: SAS DI Studio®: Populating The Enterprise Data Model

We extended a best-practice enterprise data model referred to as Detailed Data Store (DDS) specialized for media companies. In addition to the job for table Master_Customer there is a job for table Customer_X_Master_Customer which derives the relationships between customers and their master customers taking under consideration the two level hierarchy of master customers which comes from having more than one data source.

CUSTOMER COMMENTS

As a Norwegian-based business TV 2 is obliged to comply with domestic legal requirements. For example, customer profiling across databases is prohibited without the customer being informed and actively providing their consent. Also, customer consent is required to issue commercial communication to non-active customers. Therefore, from a legal perspective, a parallel business process collecting customer consents is vital.

In addition to external requirements TV 2 takes strong internal precautions to protect brand reputation and adhere to strict company policies as for example not consolidating or analyzing on sensitive information such as opinions of individuals.

TV 2 will need to constantly improve data quality processes to ensure compliance with external and internal requirements and TV 2 believes SAS® and DataFlux® provide the best tools for this task.

CONCLUSION

When consolidating different data sources such as legacy systems or after company mergers, there can be a need for advanced data integration. As we have seen in this paper DataFlux® extends SAS® with fuzzy-matching which help us identifying a unified customer view (a unique set of master customers) even when there is no exact correspondence between the customer entities of each data source. Similar operations in SQL would be very difficult to develop and maintain.

The step-by-step pattern along with its considerations can serve as a quick introduction for other similar projects and help develop a best practice. The pattern can be applied to other domains such as for example creating a unified view on products, materials, offerings and client relations.

BENEFITS

- This is a structured approach where DataFlux® runs as part of the scheduled SAS® code.
- The unified customer view can be applied to customer analysis and reporting to see all related subscriptions, purchases, products and payments across systems. When preconditions are met, we can switch the communications from customers of each data source to master customers across all data sources.
- The unified view is refreshed every night as new customers are matched against each other and existing customers. If rules are changed or tuned the results are available next morning and all changes are

traceable back to the rule versions.
- By decoupling the master customer created in DataFlux® from the existing customer entity we enabled the master customer to be refined without having to change any data in other parts of the enterprise data model, making it possible to implement this extension onto any existing data model.

RECOMMENDATIONS

- Start with only a few rules and slowly increase the number and complexity of rules.
- Make sure enough time is estimated for testing as it contributed to about 75% of total development efforts.
- The team should know how to use and configure tools from DataFlux® together with SAS®. They should also be able to lay out a process supporting a continuous process of data quality improvements. A client representative should be dedicated to data source analysis and following up the data quality process.
- Make sure the team is provided access to real-life data from analysis phase and throughout the project.

POSSIBLE ENHANCEMENTS

- Matching and consolidation for each data source could be conducted in parallel and then synchronized before starting the cross system matching and consolidation.
- Generic DataFlux® jobs could be designed where matching rules were configurable in XML for even easier change of rules without altering the code.

ACKNOWLEDGMENTS

Many thanks to TV 2 for letting us publish this pattern and to our colleagues at SAS Institute and DataFlux in Norway for providing feedback and inspiration. Special thanks to Henrik Slettene as one of the early promoters of this paper.

REFERENCES

- SAS® 9.2 Documentation. “What's New in SAS Data Quality Server 9.2”. 2011. Available at: http://support.sas.com/documentation/cdl/en/dqclref/63101/HTML/default/viewer.htm#dqclrefwhatsnew902.htm
- Stander, Jeff. 2007. Proceeding at SAS Forum Poland. “Case Study: Implementation of Data Quality using SAS Data Integration with DataFlux Integration Server”. SAS Institute. USA. Available at: http://www.sas.com/offices/europe/poland/events/forumtech/present/a5.pdf
- Hazejager, Wilbram. 2010. Proceeding 327-2010 at SAS Global Forum 2010. “What’s New in DataFlux® dfPower® Studio and DataFlux Integration Server Software?” DataFlux Corporation. Germany. Available at: http://support.sas.com/resources/papers/proceedings10/327-2010.pdf
- Howard, Philip. Stanley, Nigel. 2010. White Paper. “Single Customer View in Financial Services”. United Kingdom. Bloor Research. Available at: http://www.dataflux.com/Login.aspx?ReturnUrl=%2fResources%2fDataFlux-Resources%2fWhite-Paper%2fSingle-Customer-View-in-Financial-Services.aspx
- Snyder, Brian. 2009. “DataFlux dfPowerStudio Quick Tips, Tricks, & Best Practices”. Information Architecture Services, NEOS. Available at: http://neosllc.com/PDF/NEOS_DataFluxTipsAndBestPractices.pdf

CONTACT INFORMATION

Name:	Terje Vatle
Enterprise:	SAS Institute AS
Address:	Parkveien 55, P.O. Box 2666, 0203 Oslo, Norway
City, State ZIP:	N-0256 OSLO
Work Phone:	+ 47 23 08 30 50
E-mail:	terje.vatle@nor.sas.com
Web:	www.sas.com/no