Building the Single Customer View in a Data Warehouse

Dr. David Cattrall

Dr. David Cattrall is director of BSS Refresh, a consultancy specializing in architecture of business support systems including data warehousing, business intelligence, customer relationship management, and e-commerce. dave@bss-refresh.co.uk

ABSTRACT

Any organization that operates multiple channels of customer communication faces the challenge of consolidating data to fully understand customer behavior. This article presents a pragmatic approach to building a single customer view in a data warehouse to resolve this problem.

INTRODUCTION

Increasingly, organizations are offering multiple channels of communication to reach customers such as Web advertising and portals, online and retail stores, and call centers. These channels are valuable sources of customer information, but the supporting operational systems can quickly become disparate silos of customer data.

Effective customer relationship management (CRM) demands a complete and accurate view of each customer. A *single customer view* is an aggregated and holistic representation of the data known by an organization about its customers, based on all interactions and transactions. Such a view offers the ability to analyze customer behavior to better personalize interaction, improve service, and increase sales.

Many enterprises spend significant time and money implementing systems yet struggle to achieve the single customer view. As illustrated in Figure 1, a marketing database will not record sales or service transactions, a CRM service system will not capture customer purchases, and a data warehouse of purchases will

not relate buying behavior to service requests. Fundamentally, any solution that addresses only a limited number of channels, business processes, and operational systems will provide only a partial view of customer activity.

This article describes an alternative approach to building a true single customer view across multiple channels based on a data warehouse. I present a conceptual model; show how it can be represented in a data warehouse following a Kimball, Inmon, or hybrid architecture; describe the ETL mechanism required; and discuss the business intelligence (BI) capabilities it facilitates.

CONCEPTUAL MODEL

Over time, each customer will embark on one or more "customer journeys" by way of interaction with various communication channels. Each interaction affords an opportunity to accumulate further information about that customer. To illustrate the model, we'll use the following series of customer interactions:

1. Customer first visits the organization's website, researches products, and is uniquely identified by a Web browser cookie.
2. Customer then visits a retail store and purchases a product, Prod01. The customer is identified by the payment card used for the purchase.
3. Customer visits the online store and purchases another product, Prod02. The customer data captured is the last name, first name, email address (for purchase confirmation), and home address (house number, street, city).
4. Customer registers on the Web portal and uses self-service to answer a question about Prod01.

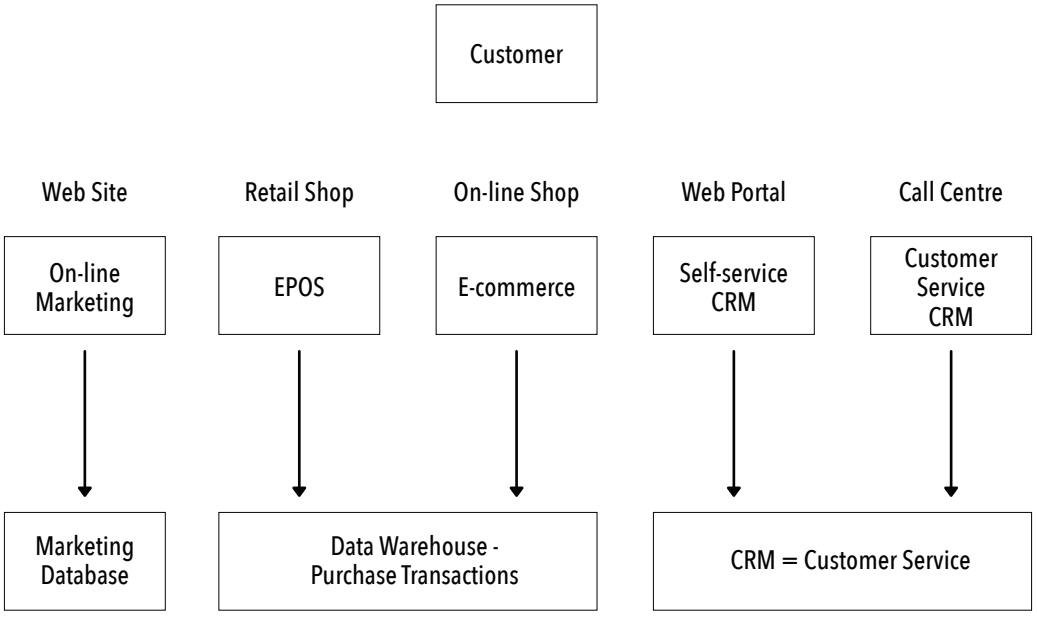

Figure 1. The multiple systems involved in customer relationship management.

5. Customer telephones the call center to discuss Prod02. The customer's telephone number (Phone01) and address (street and ZIP code) are captured as part of product registration.

Figure 2 illustrates the data captured at each interaction, broken down into three sections. Channel Transaction shows each customer interaction, its channel—website, online or retail store, Web portal, or call center—and a

Channel Transaction						
	No.	001	002	003	004	005
	Channel	Web Site	Retail Shop	Online Shop	Web Portal	Call Centre
	Date	2017-01-01	2017-02-01	2017-03-01	2017-04-01	2017-05-01
	Time	19:00	10:00	20:00	8:00	12:00
	Trans Type	Web visit	Retail purchase	Online purchase	Self-service	Service request

Channel Customer						
	No.	001	002	003	004	005
	Cookie	Cookie01		Cookie01	Cookie01	
	Last Name			Jones	Jones	Jones
	First Name			John	John	John
	Email			jj@example.com	jj@example.com	
	Number					Phone01
	Payment Card		Card01	Card01		
	Product		Prod01	Prod02	Prod01	
	Address			1 Red Rd, Liverpool		1 Red Rd L1 2XY

SCV Customer						
	No.	001	002	003	004	005
	Cookie	**Cookie01**		Cookie01	Cookie01	Cookie01
	Last Name			**Jones**	Jones	Jones
	First Name			**John**	John	John
	Email			**jj@example.com**	jj@example.com	jj@example.com
	Number					**Phone01**
	Payment Card		Card01	Card01		Card01
	Product		**Prod01**	Prod01, **Prod02**	Prod01, Prod02	Prod01, Prod02
	Validate Address			**1 Red Rd, Liverpool, L1 2XY**	1 Red Rd, Liverpool, L1 2XY	1 Red Rd, Liverpool, L1 2XY

Figure 2: The accumulation of data in the single customer view over time.

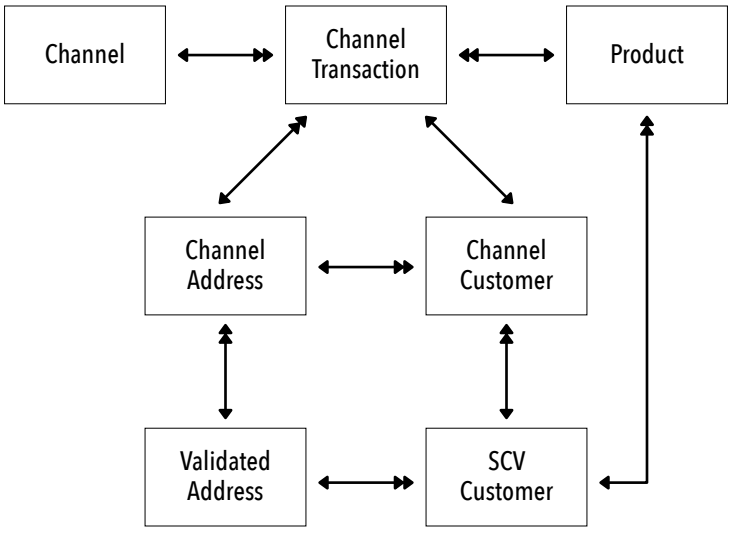

Figure 3: Entity-relationship diagram model for the conceptual single customer view.

timestamp. Channel Customer shows the data captured during each transaction, reflecting the partial customer view seen in that channel during that transaction. SCV Customer represents the single customer view as it is initiated and refined, with the information gained from each interaction shown in bold.

Note that the full picture of the customer only begins to form at the third transaction, when the information provided by the cookie from the first transaction can be combined with the purchase information provided in the second and third transactions and supplemented by validation processes (such as for the shipping address). As more information is collected, the single customer view has more data points available for further consolidation and validation.

Figure 3 presents an entity-relationship diagram (ERD) data model for the conceptual single customer view. In addition to Channel Transaction, Channel Customer, and SCV Customer as described above, the ERD contains the following entities:

- **Channel:** The channel through which the customer interaction occurs (e.g., website, Web portal, online or retail store, call center).
- **Product:** The product referenced in the Channel Customer, if any. For example, the product purchased in a purchase transaction or referenced in a service request.
- **Channel address:** The customer address captured in a Channel Transaction.
- **Validated address:** A complete version of the customer's address, capturing all relevant attributes. This is created by applying industry-standard address validation software to the Channel Address.

✓ = unique customer identifier

Channel		Web Site	Online Shop	Retail Shop	Web Portal	Call Center
Operational System		Online Marketing	E-commerce	EPOS	Customer Self-service	Customer Service
Customer Attribute	Cookie	✓				
	Last Name	•	•	•	•	•
	First Name	•	•	•	•	•
	Email	✓	•		✓	•
	Phone number				•	✓
	Payment card		•	✓		
	Home address		•		•	

Figure 4: Customer attributes referenced across channels and operational systems.

Our conceptual model creates two levels of customer data. The transactional level provides a view of customer transactions as captured through the entities Channel Transaction, Channel, Product, Time, and Date. The single customer view provides the holistic view of each customer, which is layered on top of the transactional level. It includes the entities SCV Customer and Validated Address.

Figure 4 shows the customer attributes of interest to the organization and illustrates the variation in attributes captured across channels and associated operational systems. Channel Customer and SCV Customer both contain all the customer attributes shown.

DATA WAREHOUSE REPRESENTATION

It is necessary to consider how the SCV conceptual model can be represented in a data warehouse following an Inmon, Kimball, or hybrid architecture, and the potential benefits of each approach.

Inmon

At the start of data warehouse design, Inmon (2005) advocates construction of a high-level data model as an entity-relationship diagram (ERD) and a data item set (DIS), which together identify the main entities and attributes in the data warehouse. The ERD and DIS would be based on the conceptual models detailed in Figures 3 and 4.

Inmon defined a data warehouse as a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions. Our single customer view is subject-oriented and based on the concepts of Customer, Product, and Transaction (Channel Transaction), which Inmon recognizes are common subject types for data warehouses. It is integrated by capturing data from multiple, disparate customer-facing operational systems that support each channel.

Inmon notes that, within the operational environment, data within systems is updated as a

matter of course. In contrast, a data warehouse is nonvolatile. Data is uploaded once in a static snapshot and, when subsequent changes occur, a new snapshot record is written. This allows the data warehouse to maintain historical activities and events as a sequence of snapshots. Each snapshot captures data that is accurate a given moment in time (time-variant). To support this, records often contain time stamps.

Within our scheme, each Channel Transaction is time-variant containing date and time stamps. The associated Channel Customer represents a snapshot of the customer at the time of that transaction.

The actual single view of the customer, SCV Customer, is accumulated from a series of time-stamped snapshots captured from the Channel Transaction and Channel Customer entities. To ensure it remains nonvolatile, I'll introduce the concept of survivor and nonsurvivor records. For a given customer, the *survivor* represents the current snapshot and the *nonsurvivors* represent all previous snapshots. To allow us to implement this, SCV Customer includes these metadata attributes:

- **Successor:** If this record is a survivor, this attribute is NULL. If this record is a nonsurvivor, this attribute contains the unique identifier of the survivor record that succeeded it.
- **Start date and start time:** Time stamp from when this record became active.
- **End date and end time:** Time stamp on which this record ceased to be active and became a nonsurvivor.

An Inmon-style implementation involves a central enterprise data warehouse in which the single customer view is constructed. This contains all the entities and attributes identified in the ERD and DIS. It is populated through ingestion of data from the operational system for each channel.

In a top-down manner, the data warehouse can be used to derive data marts, each of which provides a view of the customer for one or more specific channels or departments (e.g., a customer service department perspective for the Web portal and call-center channels).

Kimball

Kimball advocates implementation based on the concept of *dimensional modeling*. This divides the data warehouse into "fact" tables, which represent measurements captured by the organization's business processes, and "dimension" tables, which provide context (Kimball and Ross, 2013; Kimball Group, 2013). Kimball et al. (2008) recognize that the most common fact tables are transactional with the grain of one row per transaction.

Each customer interaction with a channel is represented as a Channel Transaction, which forms a fact table. For those transactions that involve the purchase of a product, the natural measurement is the price paid. Other transactions may not contain any measurement but simply denote a channel interaction occurred. These are referred to as "factless" fact tables (Kimball and Ross, 2013; Kimball Group, 2013).

The Channel Transaction fact table is associated with a number of dimensions, namely the other entities identified in the conceptual ERD model:

- **Time and date:** Capture a date and time stamp for each transaction. In practice, there may be a number of transactions occurring at the same date and time.
- **Product:** The product involved in the transaction (if any).
- **Channel:** The communication channel through which the transaction occurred.
- **Channel address and validated address:** Represent the address provided in the transaction and the validated version of that address, respectively.
- **Channel customer:** Represents a partial view of that customer as seen at the time of that transaction in that channel.
- **SCV customer:** The single customer view, which is updated as customer data is accumulated.

Kimball and Ross (2013) recognize that real-world dimension data can change over time and identify three strategies for handling such "slowly changing dimensions." This dimension uses type 2—whenever a channel transaction introduces additional customer data, a new row is added to the dimension table. That row becomes the new survivor and the previous rows for that customer become nonsurvivors. To support this, in the same manner as the Inmon implementation, this table includes metadata attributes: Successor, Start Date, Start Time, End Date, and End Time.

In practice, an organization may commence by building data marts based around channels and associated operational systems. To allow separate data marts to be plugged together to incrementally construct the enterprise data warehouse (EDW), Kimball advocates data marts use "conformed" dimensions and facts (Kimball and Ross, 2013). Figure 5 presents a data warehouse bus matrix for the single customer view showing the business processes supported by each channel and operational system, and the associated dimensions for each department. This can be used to create the bus architecture for the EDW.

Thus, the Kimball approach facilitates the planning and construction of a separate data mart for each department/channel and associated operational system. Each data mart would provide a partial view of the customer for that channel.

The integration of those data marts into an EDW could then be achieved incrementally to form the single customer view across all departments of the organization.

Hybrid

The hybrid data warehouse structure can contain both normalized snowflakes (in the style of Inmon) and de-normalized star schemas (in the style of Kimball). Such an architecture aims to accommodate the large volumes of historical data required in the enterprise data warehouse while also performing efficiently for the online analytical processing (OLAP) queries typically done in data marts. In principle, supporting both EDW and data mart applications in one database should reduce the cost of implementation.

Dimensions are snowflaked when redundant attributes are moved to separate tables and linked back to the original table. When snowflaking involves a dimension referencing a subdimension, the term *outrigger* is used (Kimball and Caserta, 2004).

A hybrid architecture could adopt Kimball-style facts and dimensions to represent the transactional level of the conceptual model. This would facilitate efficient OLAP queries for customer transactions.

The architecture could use normalized tables to represent the SCV level of the conceptual model and provide the accumulated view of customers and the products owned.

ETL MECHANISM

Introduction

Regardless of whether an Inmon, Kimball, or hybrid architecture is being used, successful implementation of the single customer view depends on a sophisticated ETL mechanism. At a high level, the processing steps required are:

1. Extract data for each customer interaction from the corresponding operational system.

Department	Channel	System	Business Process	Dimension							
				Channel	Product	Channel Customer	SCV Customer	Channel Address	Validated Address	Date	Time
Marketing	Website	Online Marketing	Browse website	•		•	•			•	•
			Search for product	•	•	•				•	•
Sales	Online Shop	E-commerce	Purchase product	•	•	•	•	•	•	•	•
			Review product	•	•	•	•	•	•	•	•
	Retail Shop	EPOS	Purchase product	•	•	•	•			•	•
			Review product	•	•	•	•			•	•
Service	Web Portal	Self-service CRM	Create service request	•		•	•			•	•
			Read product article	•	•	•	•			•	•
	Call Center	Customer Service CRM	Request service	•		•	•	•	•	•	•
			Request product information	•	•	•	•	•	•	•	•

Figure 5: Data warehouse bus matrix for single customer view.

2. Transform that data. For each customer transaction:

   a. Create transaction records

   b. Identify existing SCV records

   c. Create the new, updated SCV record

3. Load records from that data model into the data warehouse.

Step two illustrates the additional steps over conventional ETL processing required to form the single customer view, which is discussed in the sections below.

Create Transaction Records

For each customer interaction with a channel, ETL is required to extract the appropriate transaction data from the corresponding operational system and transform it for the required data warehouse representation. To do this, it creates the following records:

- **Channel transaction:** This is formed from the date, time, channel, and product attributes captured in that transaction.
- **Channel customer:** This is formed from the customer data fields in that transaction, as identified in Figure 4.
- **Channel address:** If customer address attributes were included in the transaction, ETL uses them to form this record.

A customer's geographic address could be captured through multiple channels and systems. In practice, the attributes used to represent address lines are likely to vary between systems (see Figure 2). This can make address processing problematic. In particular, address comparison can be difficult because variants of the same address may not be detected.

Address validation software provides functionality to compare a given address against a reference database containing accurate address information and, if it can be matched, return the complete address as a standardized set of fields. A number of industry vendors offer address validation as a cloud-based service. (Note: This can be contrasted with address verification software, which determines whether a given payment card is registered at a given address.)

ETL applies address validation software to the Channel Address record to form a Validated Address record. The resulting Channel Transaction, Channel Customer, and Validated Address records are referenced in the following steps.

Identify Existing Single Customer View

For a given channel transaction, the ETL is required to identify *all* active records from the single customer view that match the customer who made that transaction. Ideally there would only be one such record. However, as illustrated in Figure 4, not all customer attributes are referenced across all channels and the same customer could be identified through different attributes in different channels. Consequently, within our single customer view, there could be multiple records that relate to the same customer but have not yet been identified as such.

To identify all customer records that relate to this transaction, ETL uses a set of data-matching rules based on the nature of the business, the specific channels, and the supporting

operational systems. Examples of customer data-matching rules include:

- Cookie and email
- First name, last name, and email
- First name, last name, and validated address
- First name, last name, and telephone number
- First name, last name, and payment card
- Email and payment card
- Cookie and payment card
- Payment card and validated address

It should be noted that ETL uses the Validated Address, rather than the Channel Address, to improve effectiveness of address matching.

Thus, for each Channel Customer record and associated Validated Address, the ETL mechanism uses the above matching rules to identify the active SCV Customer record that represents the existing single view of that customer.

Create New Single Customer View

Once the existing SCV records have been identified, ETL is required to create a new single view for the customer. The new SCV Customer record merges field values from the existing SCV Customer records and includes any additional customer attributes captured in the Channel Customer record. Its address attribute references the Validated Address derived from the transaction.

The existing SCV Customer records become nonsurvivors with end date/time stamps set to the date/time of the Channel Transaction and identify the new SCV Customer record as their successor.

BUSINESS INTELLIGENCE FOR THE SINGLE CUSTOMER VIEW

As discussed, the conceptual model contains two levels of data: transactional and single customer view. Regardless of the implementation architecture (Inmon, Kimball, or hybrid), the transactional level will be associated with data marts. The single customer view, across the whole organization, exists in the enterprise data warehouse. Once the data warehouse has been implemented, BI tools can process the constituent data to answer key questions.

The transactional level can be used to answer questions relating to specific transactions within specified time frames and the products involved, such as

- Which products were bought through each channel over the last week?
- How many products in total were sold through each channel in the last month?
- What was the total sales value for each channel over the last quarter?

However, the two levels arguably provide the greatest BI benefit when combined to allow answers to questions that relate individual transactions to the single customer view. This can be achieved through the cross-references between entities in the model. Each Channel Transaction references a Channel Customer, which references an SCV Customer. If that

SCV Customer is a nonsurvivor, then it references its successor, which may in turn reference another successor.

By navigating that chain of references, it is possible to identify the survivor SCV Customer, which represents the most recent single view of that customer. This allows us to answer questions such as

- How many customers who bought product X also bought product Y?
- How many individual customers raised service requests for product X last year?
- What is the total value of specific customers in terms of products purchased?
- What is the total cost of service for specific customers?

Before selecting the data warehouse architecture and BI tools, it is useful to consider the nature of the questions that the organization is concerned with at both the transactional and SCV levels. The choice of data warehouse architecture can impact the manner in which those questions will be answered. It may be considered desirable to answer certain questions through specialized data marts that reflect specific departments, business processes, or channels. Similarly, the single customer view could be tailored to fit the needs of the organization.

The choice of BI tools will determine the manner in which customer data can be analyzed to support business decision making and, ultimately, make the organization more profitable. Organizations may also want to use the valuable SCV data that can be provided by the data warehouse to directly improve the performance of operational systems. To do this, organizations are increasingly deploying "feedback" interfaces from the data warehouse to the operational systems (Cattrall, 2016). By pushing the SCV data to the communication channels, an organization can directly improve personalization, promote sales, and improve service.

CONCLUSION

The importance of the single customer view is increasingly being recognized by organizations. At the same time, many organizations are encountering challenges in creating that view.

This article presented a conceptual model for constructing the single customer view that is appropriate for any organization offering customers multiple communication channels supported by disparate operational systems.

The manner in which that model could be represented in a data warehouse following an Inmon, Kimball, or hybrid architecture was described along with key elements of the requisite ETL mechanism.

Once it has been constructed, the single customer view offers significant rewards from a BI perspective. Consequently, this is something organizations will increasingly wish to embrace. ●

REFERENCES

Cattrall, David [2016]. "The Data Warehouse Feedback Interface," *Business Intelligence Journal*, Vol. 21, No. 3, pp. 10–19.

Inmon, William H. [2005]. *Building the Data Warehouse*, 4th ed., Wiley.

Kimball, Ralph, and Joe Caserta [2004]. *The Data Warehouse ETL Toolkit*, Wiley.

Kimball Group [2013]. "Kimball Dimensional Modeling Techniques," available at http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques.

Kimball, Ralph, and Margy Ross [2013]. *The Data Warehouse Toolkit*, 3rd ed., Wiley.

Kimball, Ralph, Margy Ross, Warren Thornwaite, Joy Mundy, and Bob Becker [2008]. *The Data Warehouse Lifecycle Toolkit*, 2nd ed., Wiley.