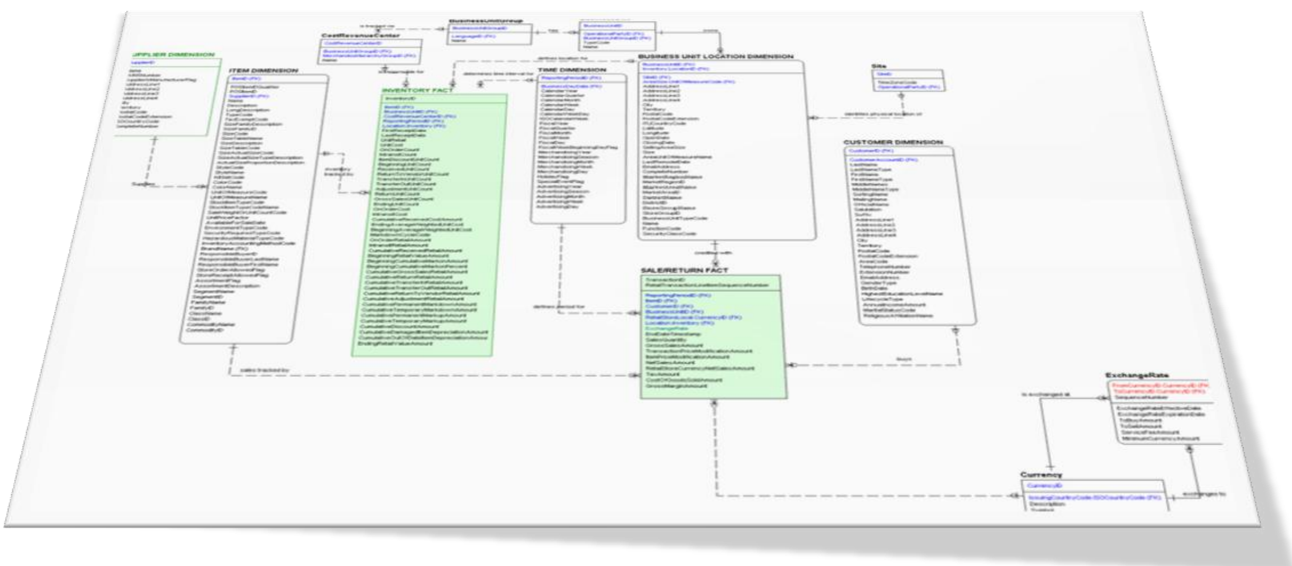
ARTS

Association for Retail Technology Standards

of the NATIONAL RETAIL FEDERATION



ARTS Data Model Committee
Retail Data Warehouse Model

July 2011

Release 2.0

Document Change Log			
Version	Date	Author	Remarks
2.0	11/29/2010	M. Gregory	• Initial updates for additions in version 2
2.1	12/6/2010	M. Gregory	• Revisions based on work team feedback in meeting on 12/3 include: • Removal of Promotional Analysis view due to lack of data points; reversion to Store Sales view to reflect the absence of labor and other store operations data • Revision of comparable sales to reflect net sales basis • Addition of Inventory related KPI discussions based on work team direction
2.2	12/15/2010	L. Naeyaert	• Updates to section 4 through 4.2
2.3	12/19/2010	M. Gregory	• Adjusted business metrics definitions to reflect nomenclature in the model provided by Tom Sterling
2.4	12/19/2010	L. Naeyaert	• Updates to the Entity and Dimension sections to reflect the new model
2.5	12/20/2010	M. Gregory	• Additional updates to business metrics to reflect the model nomenclature
2.6	12/24/2010	M. Gregory	• Accept changes and remove comments that are no longer relevant
2.7	5/1/2011	T. Sterling	• Re-architected content and presented in the format seen here, including substantial work onintegration with the ODM
2.8	7/07/2011	T. Sterling	• Review and make corrections to diagrams, spelling errors and exhibit reference numbers.

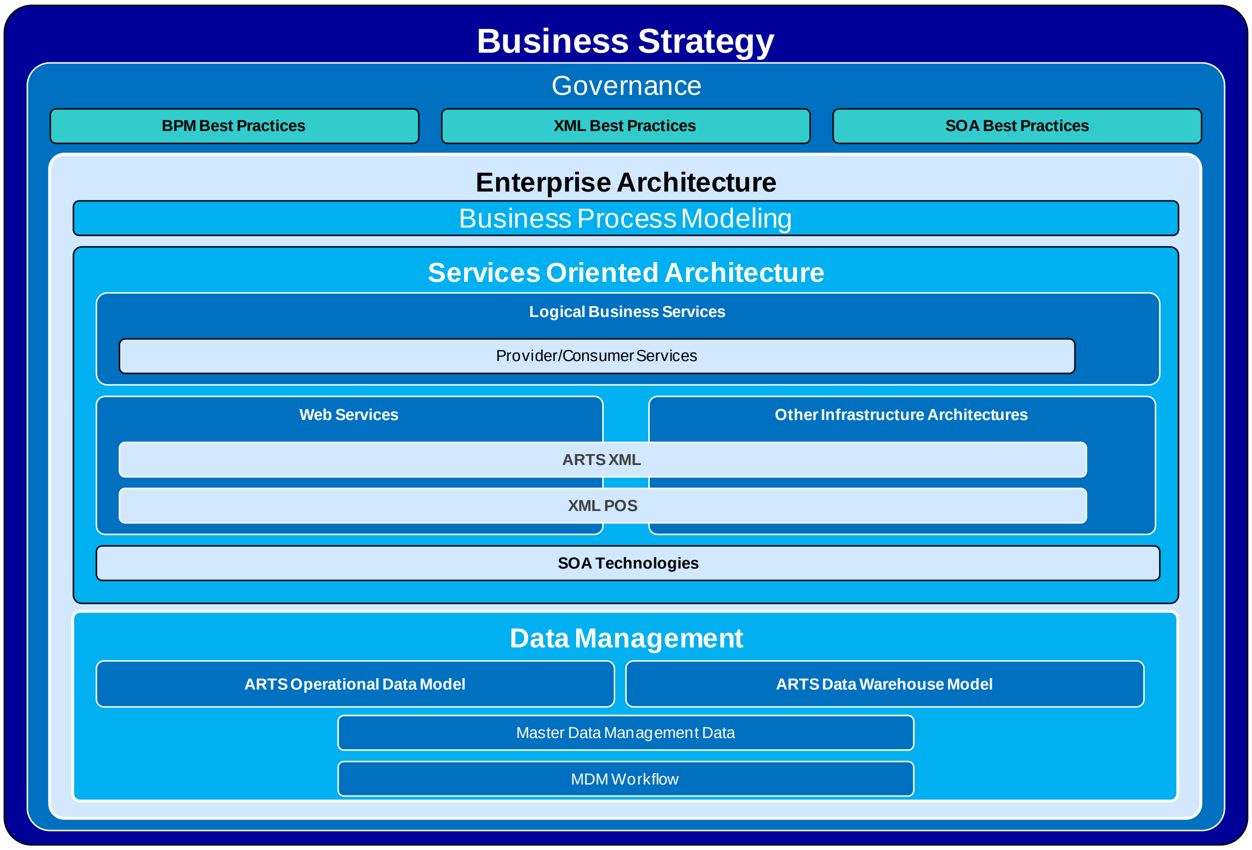
Contents

1 Introduction

1.1 Document Purpose

This document describes the ARTS Data Warehouse Model (DWM) Release 2 and basic retail performance measures. It complements the entity relationship model and dictionary presented in the appendices. The ARTS DWM is part of an integrated system of retail technical standards

Figure 1: ARTS Technology Standards Context



The ARTS DWM and ARTS Operational Data Model (ODM) are the two model-based products in the Data Management collection. These two data models are complementary and very tightly integrated. A major portion of this document discusses how the ODM – DWM integration is represented.

1.2 Relationship between ARTS Data Warehouse & Operational Data Models

The DWM is a dimensional extension to the ARTS Operational Data Model (version 6.1). It is a reference model, not a standard model. The difference is that a reference model provides guidelines and examples for retailers to use in creating a data model while a standard model defines specific entities, attributes, relationships and definitions that must be satisfied by a data model. Standards have compliance criteria while reference models supply examples.

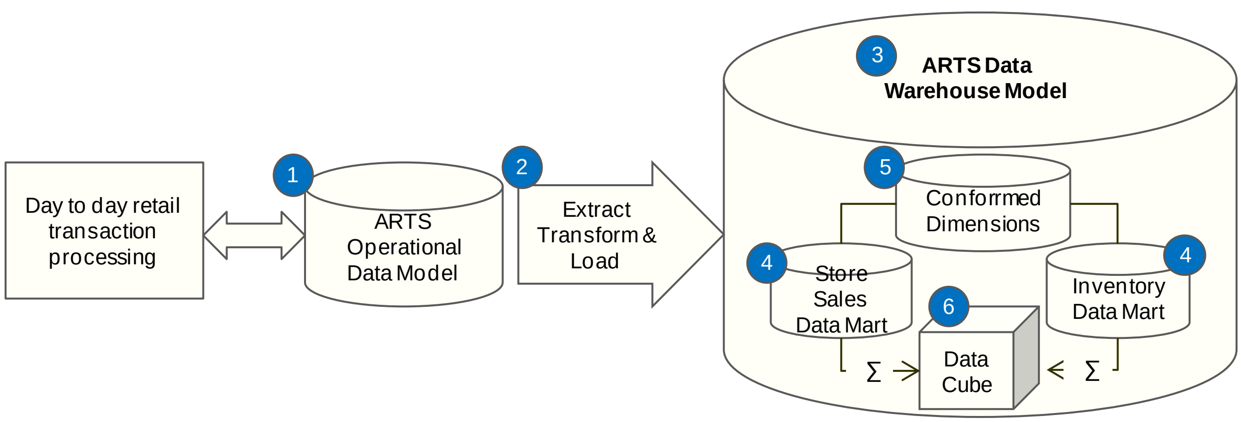
Table 1 lists other salient differences between the ARTS ODM and DWM.

Table 1: ODM Versus DWM Characteristics

ARTS Operational Data Model (ODM)	ARTS Data Warehouse Model (DWM)
Standard Model	Reference Model
Designed around third normal form	Designed around star schema which denormalizes data structure
Supplies relational data structure to support on-line transaction processing	Supplies dimensional data structure to support business information selection, aggregation and analysis
Optimized for speed and data integrity when creating and updating data	Optimized for end-user accessibility and comprehension when reading data for reporting and analysis
Data access is through transactional application programs	Data access is through a wide variety of analytical toolsets
Data structure abstract and complex with emphasis on data integrity and create/update efficiency	Data structure simplified with focus on end-user comprehension and accessibility
Abstract design principles based on Information Engineering and IDEF1x relational data modeling methods	Concrete design based on dimensional data design
Large model composed of 663 entities and 4,000+ attributes	Simple model composed of12 entities and 245 attributes 245 attributes

While there are distinct differences between the ODM and DWM data structures and organization, they share data content. Data content refers to the actual values of the data attributes that make up the data models. Figure 2 illustrates the relationship between the ARTS ODM and DWM and introduces some key data warehouse concepts.

Figure 2: Relationship Between ARTS ODM and DWM



Looking from left to right, the day to day transactions made up of sales, receiving, inventory adjustments, promotion set up and maintenance and all of the things retailers do every day are captured in a database build on the ❶ ARTS Operational Data Model. The ARTS Operational Data Model is designed to support transaction processing applications. It provides pre-defined operational reports for monitoring and control of day to day activities. It does not support ad-hoc analytical reporting. A ❷ periodic extract-transform-load (ETL) process is run to re-factor the third normal relational structure of the ODM in to a ❸ simplified dimensional structure designed around data warehouse principles – the

ARTS Data Warehouse Model. The ARTS Data Warehouse Model in release 2 is composed of two data marts. A ❹ data mart is a small set of data structures that include facts and contextual data organized around a set of business questions. In Release 2, the business questions address store sales and retail inventory.

The ARTS DWM ties its data marts together using ❺ conformed dimensions. Conformed dimensions are entities that define the context for data mart facts. By context, we mean they identify the "who, what, where, when, how and why" needed to make raw facts meaningful to the business. As the ARTS DWM evolves, more and more data marts (collections of facts) will emerge. Conformed dimensions are the key piece of the DWM that keeps all of the disparate facts organized and accessible using a consistent context. Data warehouses (which are enterprise wide assets) are distinguished from data marts (which are more narrowly focused departmental assets) by their rigorous implementation of conformed dimensions.

Conformed dimensions are possible because the DWM is built on the foundation provided by the ARTS Operational Data Model which is a third normal form relational model.

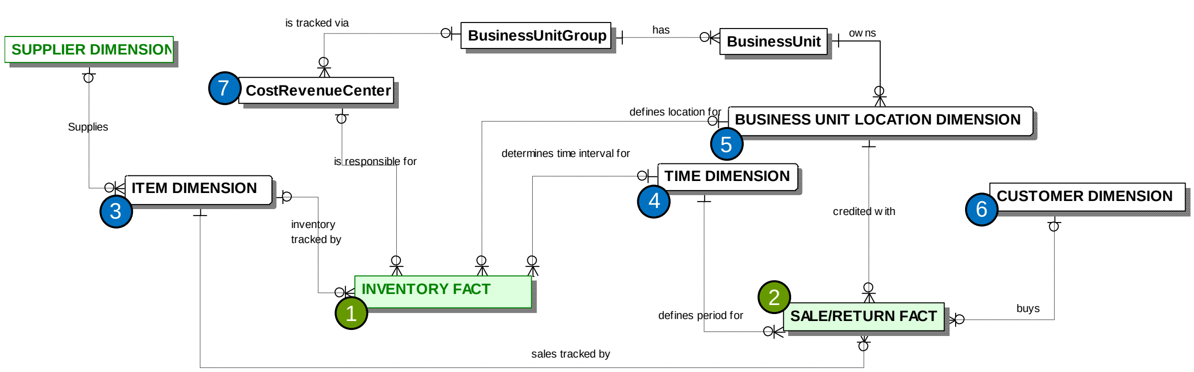
The ❻ data cube(s) represent pre-staged summary information and may be used to improve data warehouse performance. They are not explicitly modeled in Release 2 of the data model. They are included in the diagram to illustrate their place in the overall ARTS DWM scheme.

In addition to identifying, defining and describing the data structures that make up the ARTS DWM, this document also presents a set of sample retail performance measures that draw the data for their formulas from the data marts. The purpose is to demonstrate how to leverage the information once it is stored in the data warehouse.

1.3 ARTS Data Warehouse Model Contents

Release 2 is a basic data warehouse model made up of two data marts with shared dimensional entity types (conformed dimensions). Figure 3 presents a high, entity-level overview of the ARTS DWM.

Figure 3: ARTS Data Warehouse Entity Level Overview



The light green entity types are the facts. Facts represent counts, balances, point in time quantities, monetary valuations and other data that may be aggregated and processed to yield useful information for decision makers.

➊Inventory facts include item counts, inventory balances, receipt counts, inventory retail and cost valuations for the retail enterprise. ➋Sale return facts include store-level item sales counts, extended retail values, extended cost values and similar information. Both fact entities share dimensional entities (shown as white boxes). As noted earlier, the use of common entities to tell where a fact takes place, when a fact takes place, who is involved in that fact, what items are involved in a fact and why the fact occurred are the components that make up conformant dimensions.

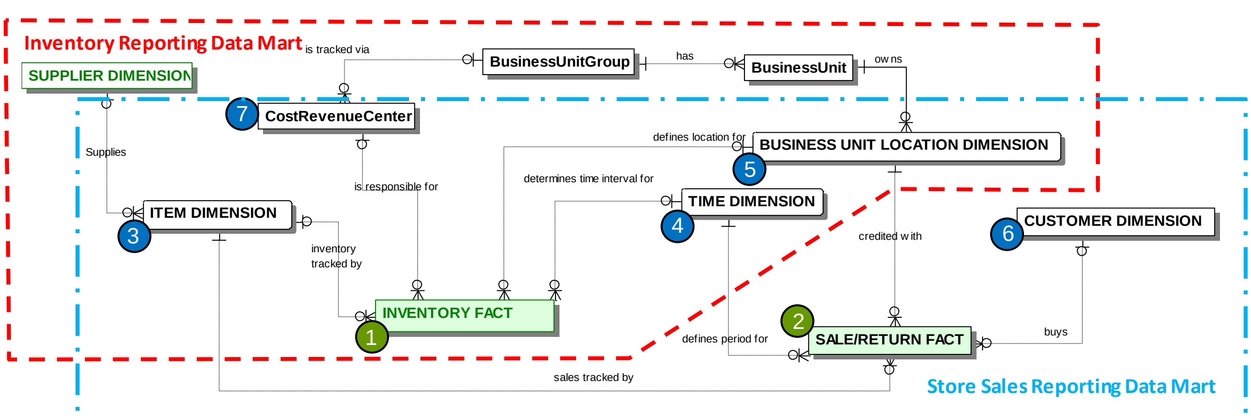
The dimensional entity types (also called dimensions) include:

- Item Dimension➌ – Identifies, defines, describes and classifies the products and services referenced in the Inventory Fact and Sale/Return Fact entity types;
- Time Dimension ➍ - Identifies, defines, describes and classifies the time periods and specific points in time that tell when the transaction recording a fact occurred;
- Business Unit Location Dimension ➎ - identifies, defines, describes and classifies the location and retail organization where a transaction recording a fact occurred;
- Customer Dimension ➏ - identifies a person or organization who is buying or returning a product or service from/to the retail store.

The other dimensional entities are covered later in this document. They serve to extend the data warehouse model and facilitate mapping back to the ODM. The goal of this section is to develop a basic understanding of the Release 2 Data Warehouse Model.

Figure 4 shows how the inventory reporting and store sales reporting data marts share dimension entity types. It also highlights the way data marts cluster around fact entity types.

Figure 4: Illustrating Data Marts within ARTS DWM



1.4 ARTS Data Warehouse Model Assumptions and Constraints

Developing an enterprise-wide data warehouse model is a very large initiative. An incremental strategy is being applied to the ARTS DWM. Each increment reflects a bounded work product which means each has a set of assumptions and constraints that limit what it will and will not provide. These assumptions and constraints are not permanent restrictions on the ultimate scope of the DWM, they represent boundaries needed to deliver a useful increment of the ultimate project.

1.4.1 Overall Assumptions & Constraints

The following assumptions and constraints reflect the basic principles guiding the DWM effort:

- The DWM will be derivable from the ARTS Operational Data Model. A source to target mapping from the ARTS ODM to the ARTS DWM will be included as a deliverable;
- The DWM will incorporate only the underlying data entities, attributes and relationships required to support analytic reporting and analysis;
- Where necessary the DWM may introduce new attributes intended to make implementing reporting and analysis data structures easier; and
- The DWM will provide concrete rather than abstract information structures and definitions.

1.4.2 Store Sales Reporting Assumptions & Constraints

The following assumptions apply to the Store Sales and Reporting data mart:

- It captures basic sales and return transactions which include: sales quantity, gross sales amount, price modification amount, net sales amount, and retail store local currency net sales amount;
- It incorporates only sales and return transaction line items and represents sales as positive numbers and returns as negative numbers for line item quantities and monetary amounts;
- It assumes that all transactions are completed at the point of sale as part of a single session;
- It assumes that all items recorded on the transaction line item are sold from the store's stock;
- It does not address promotions beyond recording price modification results reflected in the transaction line item;
- It identifies the individual customer making a purchase or return and does not associate that customer with a household or group;
- It assumes that each transaction is completed by one and only one point of sale operator;
- It uses the corporate currency of the retail enterprise for all monetary attribute types in the data warehouse except for those explicitly named retail store currency which shall use the national currency for the country where the store is located;
- It includes the exchange rate used to transform retail store currency into the corporate currency at the time a transaction is recorded (to facilitate reporting in either the store's local currency or the corporation's currency);

- The most granular identification of products and services in a transaction is at the retailer's item level. Item level is defined here as the most detailed level a retailer chooses to report and analyze sales and may not be at a Universal Product Code or European Article Number level;
- The items sold are pre-packaged products sold in units of "each" – i.e. this release does not address variable weight/volume bulk items sold by the pound, quart, etc. and
- It does not address transaction voids, sales audit and other transactions that deal with exceptions.

1.4.3 Inventory Reporting Assumptions & Constraints

The following assumptions and constraints apply to the Inventory Reporting Data Mart:

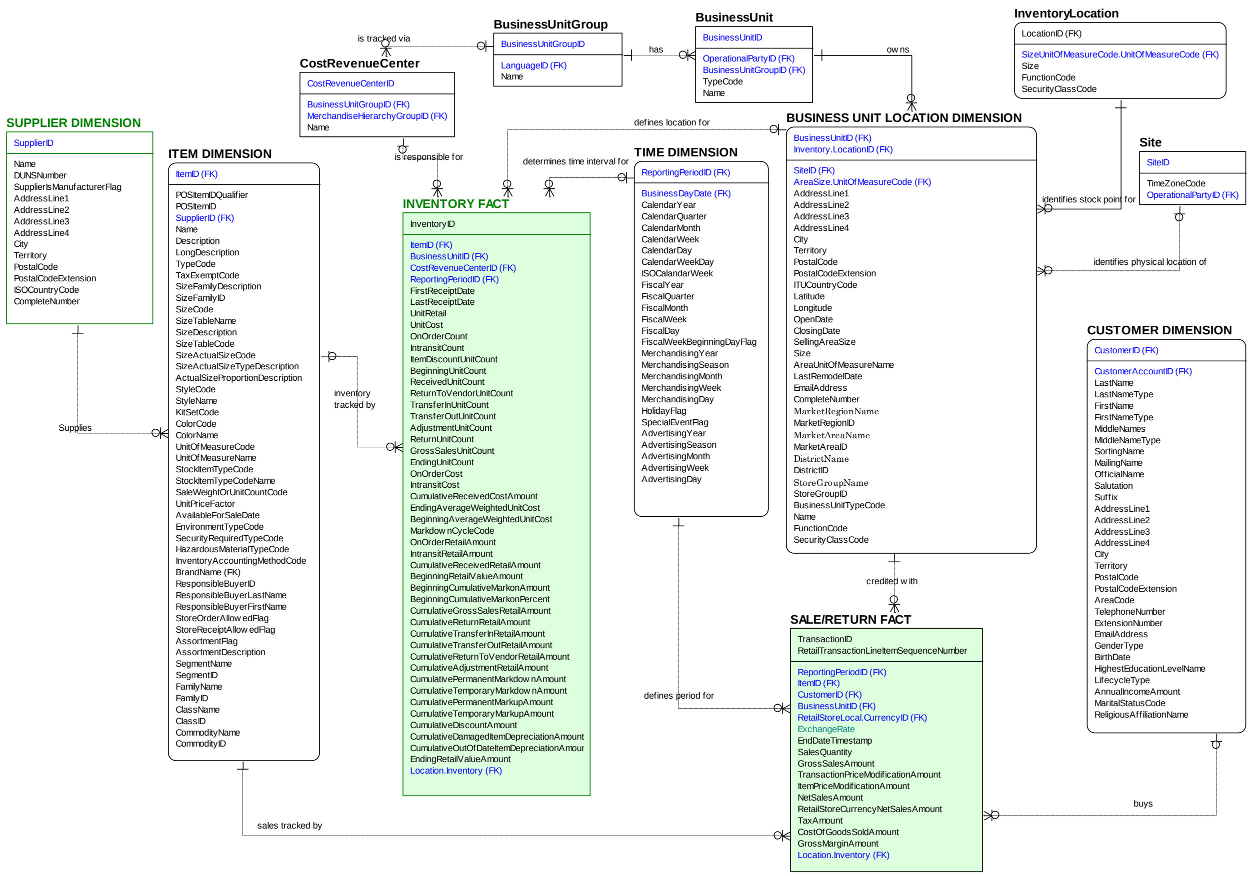
- It adds aggregated data based on non-sales inventory transactions which include: orders, purchases, receipts, physical inventory and other inventory adjustments, permanent markdowns (and markups);
- Assumes cost can be read and used without performing calculations to weigh costs over time (to derive an assigned cost and margin) These costs are typically calculated by a stock ledger at item level. No calculations to weight cost over time are included;
- Assumes that cost is read from the ODM and that it is a weighted average cost. For data warehouse purposes, this is a "given" cost for a reporting period;
- Does not address costs that are incurred at the purchase order level or any level above item;
- Does not include the customer dimension and therefore will not support customer margin/profitability at this time; and
- Assumes items in inventory are pre-packaged products sold in units of "each" – i.e. this release does not address variable weight/volume bulk items sold by the pound, quart, etc.

This list of assumptions and constraints provides the reader with a realistic understanding of what is and is not supported in Release 2 of the DWM. As noted, the boundaries set by these assumptions and constraints are temporary and will be lifted incrementally as the model evolves.

2 ARTS Data Warehouse Model: Detailed Description

Figure 5 shows an attribute-level view of the ARTS DWM. As in earlier illustrations, the light green boxes represent the fact entity types. The white boxes represent dimension entity types.

Figure 5: Fully Attributed View of ARTS DWM



2.1 Fact Entity Type Descriptions

Figure 4 shows the two data marts and the fact entity types that form the nucleus of each one. Fact entities contain business measures including counts, monetary values and other quantities that are aggregated and processed for reporting and analysis. There are three basic varieties of facts represented in the fact entities of a data mart/data warehouse. These include:

- Transactional facts;
- Periodic Snapshot facts; and
- Accumulating Snapshot facts.

2.1.1 Transaction Facts

Transaction facts are fine grained representation of individual line items in a sale, receiving, inventory count, price change or other business transaction. Transaction facts are data points that measure an aspect of business activity at a specific instant in time. In a retail context the point in time is usually represented by a timestamp. The Sale/Return Fact entity type is composed of transaction facts. Transaction facts are almost always additive facts. Additive facts can be accumulated across one or more dimensions and yield useful information.

2.1.2 Periodic Snapshot Facts

Periodic Snapshot facts capture a beginning or ending quantity that reflects the state of business at a point in time. Beginning and ending on hand balances are examples of periodic snapshot facts. Periodic snapshot facts aggregate transaction facts. So the difference between a beginning on-hand balance and an ending on-hand balance is explained by diving into transactional fact detail which shows the incremental changes that caused the beginning balance to change to the end of period balance. The Inventory Fact entity type contains a number of periodic snapshot facts.

2.1.3 Accumulating Snapshots

Accumulating Snapshots facts show the successive state changes of business entities over a period of time with a clearly defined beginning and ending point. For example, a customer order accumulating snapshot fact entity will have attributes to capture each process step completion date and time along with other quantitative facts. The other quantitative facts may include employee hours used, costs incurred, packaging materials consumed, etc. Accumulating snapshot entity types always tell a story (in numbers) with a beginning, middle and end.

2.1.4 Additive, Semi Additive and Non-additive Fact Attributes

There are three distinct kinds of fact attributes. These include:

- Additive facts;
- Semi-additive facts; and
- Non-additive facts.

2.1.4.1 Additive Facts

Additive attributes may be summed. Examples include transaction sales, receipts, returns, etc. Additive attributes may be summarized across all dimensions and yield meaningful information.

2.1.4.2 Semi-additive Facts

Semi-additive facts may be summed over some, but not all dimensions. For example, an end of period inventory on-hand balance may be summed up and make sense by business unit location. It would not make sense to sum the period end inventory on-hand balance over the time dimension. Instead of a summation it may make more sense to average the end of period on-hand inventory balances. Many of the facts in the Inventory Fact entity are semi-additive facts.

2.1.4.3 Non-additive Facts

Non-additive facts represent categories of things. Non-additive facts cannot be summed or averaged. They are strictly speaking not quantities. They may be counted and used in doing frequency analyses. For example the **MarkdownCycleCode**[1] fact classifies an item based on how many permanent markdowns have been taken. This attribute is a non-additive fact used to classify an item. It cannot be added or averaged. It can be counted. Accordingly a merchant can count the number of products in their 3rd markdown.

2.1.5 Data Mart/Data Warehouse Grain

The term grain refers to the level of detail a given fact entity type represents. The transaction fact entity types are fine grained which means they are the most detailed level of information provided in the data warehouse. Typically, transaction facts represent transaction-line item level data. If you consider a retail sales receipt, each line item on the receipt is represented by a row in a sales transaction fact table.

Data can be aggregated into larger sets of intermediate grain data. For example, the fine grained line item level sales facts may be aggregated into transaction level facts that summarize item SKU-level sales into a single transaction-level sales figure. Dimensional tables provide the attributes used to develop a large number of different intermediate grain summary data.

When the term grain is used, it is referring to the level of detail. This concept will be important when introducing the different fact attributes for the Store Sales Reporting and Inventory Reporting data marts.

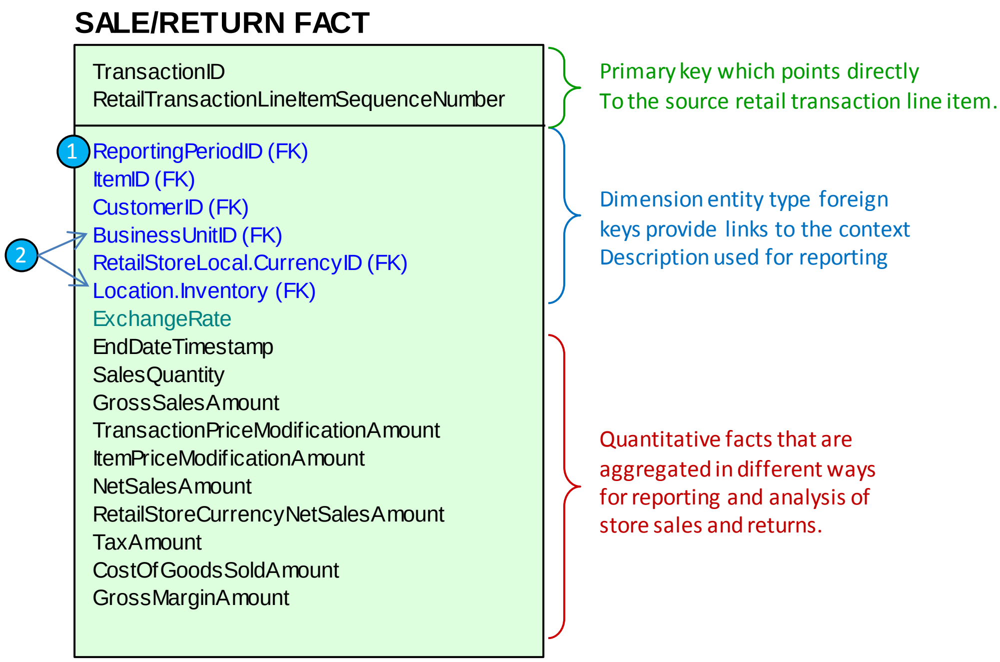
---

[1] Markdown cycle code is a retailer defined value that identifies which markdown this item is in. This definition assumes that a retailer has a defined retail price lifecycle that incorporates a sequence of markdown stages. For example a fashion retailer may establish 4 markdown cycles for a class of women's apparel which might include - initial markdown, second markdown, third markdown, clearance rack and markdown. This provides a quick snapshot of an item's retail price life cycle stage.

2.2 Sale/Return Fact Entity Type

Figure 6 presents the Sale/Return Fact entity type. This entity captures transaction-line item sales point in time data at the store level.

Figure 6: Sale/Return Entity Type



The attributes above the horizontal line in Figure 6 identify an instance of a **SALES/RETURN FACT** entity type. The blue attributes, shown as foreign keys, point to the dimension entity type instances which provide item, location, time and customer contextual information for the transaction line item.

Release 2 of the warehouse model release introduces some changes to the dimension attributes:

❶ The **BusinessDayDate** attribute is replaced by **ReportingPeriodId**. This change is required to ensure that inventory facts (particularly period balances and retail/cost valuations) are aligned with common time periods.

❷ **RetailStoreID** attribute is replaced by two attributes, **BusinessUnitID** and Inventory (which is the role name for **Inventory.Location**). Because inventory is addressed at an enterprise level, not a store level, the location dimension has to be modified. In the ODM, a **RetailStore** is a type of **BusinessUnit**. So, a **RetailStoreID** is a **BusinessUnitID**. Inventory may, at a retailers option, be stored in more than one identifiable place in a store. **Inventory.Location** accommodates this requirement. The **SALE/RETURN FACT** entity is still representing store-level sales. All these changes do is to enable the sharing of the location dimensions (conformed dimensions) which, as noted earlier, is a key feature of data warehouse models.

The fact attributes are the *additive fact attributes* used to measure and report business activity. The following attributes of SALE/RETURN FACT are "additive fact attributes":

- SalesQuantity;
- GrossSalesAmount;
- TransactionPriceModificationAmount;
- ItemPriceModificationAmount;
- NetSalesAmount;
- RetailStoreCurrencyNetSalesAmount;
- TaxAmount;
- CostOfGoodsSoldAmount; and
- GrossMarginAmount;

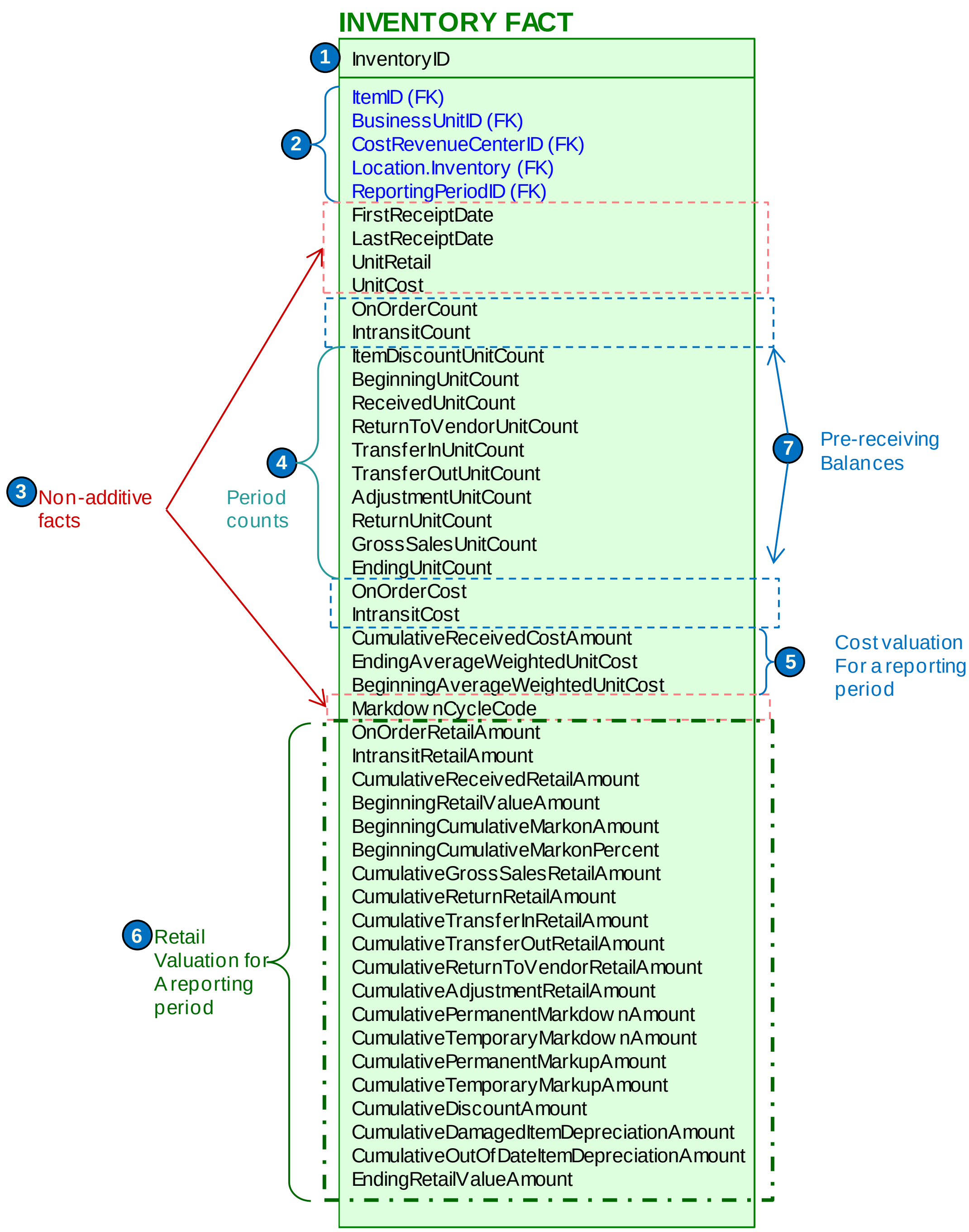
Exchange rate is a special type of *non-additive fact*. The exchange rate is a ratio of a retailer's corporate currency (the currency it uses at a consolidated enterprise level) and local store currency (the currency used at a local store that is located in a different country than the enterprise corporate headquarters). Exchange rates fluctuate and may vary throughout a business day. Each transaction could have a different exchange rate and building it in here ensures that sales and return can be reported using corporate and local store currencies. Exchange rates may be used in calculations, but they will not be summed like the other additive attributes.

The *grain* of this fact table is at the sales transaction line item level. The identification of retail stock keeping units (SKUs) occurs at this level at the point of sale. Only the retailer's internal item identifier is used in the fact table. Other item identifiers, like Universal Product Codes (UPC) or European Article Numbers (EAN) may be accessed using the item dimension.

2.2.1 Inventory Fact Entity Type

Figure 7 presents the **Inventory Fact** entity type. The scope of this fact entity is much broader than the Sale/Return Fact entity. The inventory fact instances recorded in this entity are at an enterprise-level. So, unlike the store-based facts this entity addresses inventory at all locations in a retail enterprise.

Figure 7: Inventory Fact Entity



The formal definition of the **Inventory Fact** entity type is:

> A snapshot fact that shows period end counts and value balances along with cumulative counts and values which summarize individual inventory transactions. The snap shot Inventory Facts represent periodic summarized inventory activity.
>
> The individual atomic inventory transactions that when summed result in these balances and summarized counts and values reside in the operational data model.
>
> This fact table is by item, supplier, business unit, cost revenue center, inventory location and reporting period. Retailers will have to carefully examine how they want to synchronize the relationship between cost revenue centers and inventory locations. They both tie an inventory fact to a place and an organizational entity. This requires defining business rules and procedures to ensure that these two attributes are consistently associated together for extracting and transforming cost values, retail values and counts.

As discussed earlier, the Inventory Fact entity is a periodic snapshot fact entity type. Accordingly, it provides a collection of period-level counts and monetary balances. The period-level counts and monetary amounts are put into a business context through the ❷ foreign key relationships to the dimension entity types.

Each instance of an Inventory Fact entity is identified by an ❶ **InventoryID** attribute. This attribute serves only as a unique, non-significant identifier. It is not used in queries or reporting which will use the dimension foreign key attributes.

The ❸ non-additive facts provide additional data for categorizing inventory facts based on time, markdown state and unit cost/retail value.

The ❹ count attributes represent a period unit count summary. There is a **BeginningUnitCount**, a set of intermediate counts (received count, return to vendor count, gross sales unit count, etc.) that add to and subtract from the beginning balance and yield an **EndingUnitCount**. The beginning and ending counts reflect the balance of units on hand at the beginning and end of a reporting period. The unit count figures in between represent period summed counts from inventory and sales transactions. In addition to these inventory counts there are two sets of pre-receiving balances❼. These reflect an on-going count of items on order and items in transit as well as cost amounts.

The relationship between the inventory counts, beginning and ending balances and pre-receiving balances 3is illustrated in Figure 8.

Figure 8: Inventory Counts &Pre-receiving Balances

SKU 27867676 Sample Item A												
		Pre-receiving Balances		**Period Start**	**Inventory Control Transaction-based Facts**							**Period End**
Reporting Period	Inventory Control Document	On Order	In Transit	BeginningUnitCount	ReceivedUnitCount	ReturnToVendorUnitCount	TransferInUnitCount	TransferOutUnitCount	AdjustmentUnitCount	ReturnUnitCount	GrossSaleUnitCount	EndingUnitCount
1	Period Start			1000								1000
	PO	500										1000
	Intransit	-250	250									1000
	Receiving Doc.		-250		250							1250
	Adjustment								-3			1247
	Sales Journal								-3	4	977	271
	Period End											271
	Period End Fact	250	0	1000	250	0	0	0	-6	4	977	271

Figure 8 shows the period start and period end on-hand balances for a sample SKU. In between it lists the various inventory control documents that affect the quantity on hand balance (the EndingUnitCount column). The inventory control transaction based facts reflect summed transaction unit counts for the period. They are not cumulative totals that carry from one period to the next.

The Pre-receiving Balances column lists an ongoing count of on order counts and in transit counts. These reflect different states of items before arriving at the retailer's receiving dock. As shown, purchase orders (PO) add to the **OnOrderCount**. In transit (which means the merchandise is in the possession of the carrier) subtracts the quantity shipped from the **OnOrderCount** balance and adds it to the **InTransitCount**. When the product is received by the retailer, its count is subtracted from the **InTransitCount** balance and posted as an inventory receipt (which adds to the quantity on hand balance). The pre-receiving balances are similar to the period **BeginningUnitCount** and period **EndingUnitCount** in that they reflect an arithmetic balance of inventory in a given state. For pre-receiving balances inventory is in an "inbound" state but not actually received. For inventory control the goods are physically received by the retailer and in a "retailer controlled" state.

The ❺ cost valuations for a reporting period include cumulative extended costs (unit cost x quantity) and beginning and ending average unit costs for the **ReportingPeriodID**. The average unit costs *are read from the ARTS ODM*.

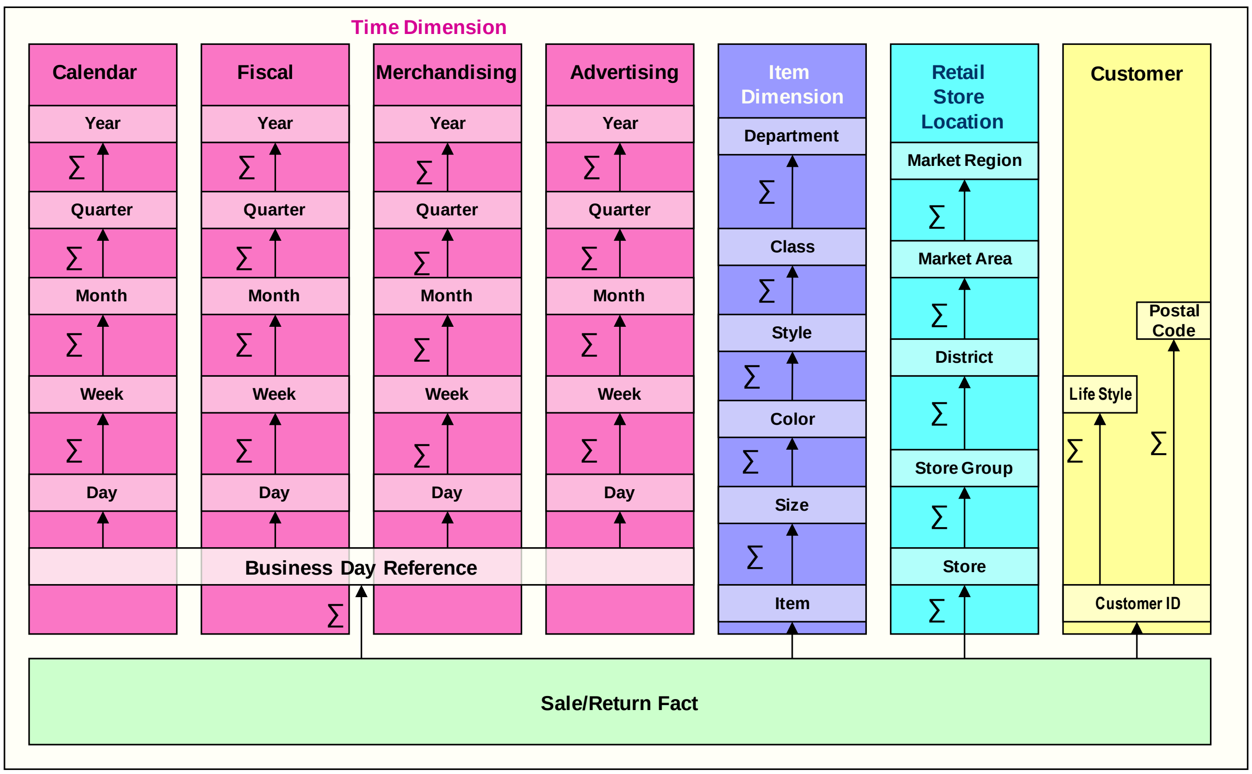
The ❻ retail valuations for a **ReportingPeriodID** are, like the costs, read from the ARTS ODM. The complexity of how costs and retail values are derived at an item level are assumed to be handled in the transactional applications and stored in the ARTS ODM. The data warehouse inventory data mart is simply aggregating the item-level cost and retail information. For Release 2 of the ARTS DWM, the different methods and interpretations of how to assign retail and cost values are delegated to the ARTS ODM. The DWM simply reads the data supplied by the ODM (which will be covered in Section 3).

2.3 Dimensional Entity Type Descriptions

Dimension entity types, as illustrated in Figure 4 place the quantitative data in fact entities into a business context. By identifying and describing items, time periods, business units, suppliers, customers, etc., dimensional entities answer key questions like: what? when? where? who? how? and why?

Dimension entities also serve as a basis for classifying and summarizing facts into data that is useful to decision makers. Figure 9 illustrates the role dimensional entities play in identifying and defining hierarchy levels used to summarize sale/return and inventory facts.

Figure 9: Dimensional Entity Types and Aggregation Levels



Each dimensional entity type defined in the ARTS DWM has a group of attributes that represent the hierarchy points illustrated in Figure 9.

For example, the Item Dimension is composed of six hierarchy levels (including item). The highest level is department. The department level is made up of class groupings, the class groupings are made up of style groupings, the style groupings are made up of color groupings, the color groupings are composed of size groupings and at the lowest level size groupings are composed of individual items. The hierarchies are derived from the ARTS ODM through a process that yields a contiguous, ordered set of keys that are used to summarize detailed facts. The process of summarizing from fact detail level into a hierarchy level is called aggregation. The process of taking a hierarchy level summary and breaking it into lower level hierarchy points (down to the fact level) is called drill down. Aggregation and drill down reporting and analytic reporting functions use these hierarchy keys. Later in this document, the process used to derive these keys is discussed in detail. Like all of the fact and descriptive attributes, hierarchy points must be mapped back to the ARTS ODM.

The Release 2 version of the ARTS DWM provides a basic set of hierarchies. Model implementers may extend this model by incorporating additional hierarchy keys in the dimensions. Future releases of the ARTS DWM will incorporate new types of hierarchy keys – all directly mapped to the ODM.

2.3.1 Item Dimension

The item dimension identifies, defines, describes and classifies the things that a retailer sells.

Figure 10: Item Dimension Entity Type

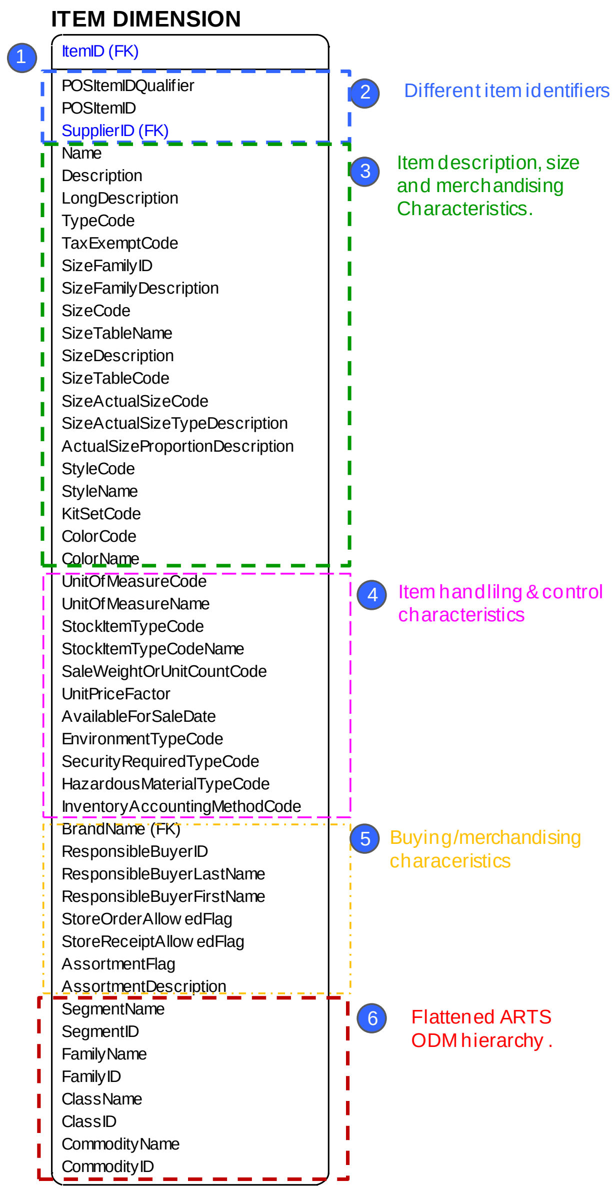


Figure 10 presents an annotated entity. The types of information about an item will vary from retailer to retailer. As discussed in the introduction, the ARTS DWM is a reference model precisely because a grocery retailer's item dimension will contain different item characteristics than a fashion retailer or hardware retailer. The ODM has a large number of descriptive attributes to support a broad range of different retail segments. Retailers may modify the owned attributes shown here as long as they can be mapped to the ODM.

Item attributes can be organized into the kinds of characteristic groups shown in the diagram. The ❶ **ItemID** attribute is the retailer's internal item ID and is copied directly from the ARTS ODM **Item** entity type. Retailers have to deal with a number of different schemes for identifying items. The ARTS ODM establishes a globally unique item identifier that serve as a token for the retailer. Some retailers refer to these unique item ID's as stock keeping units (SKUs). **Item** ALWAYS represents the lowest grain level for tracking item movement. As modeled, the Item Dimension's **ItemID** is identical to the ODM Item **ItemID**. This guarantees consistent identification of an item between the ODM and DWM.

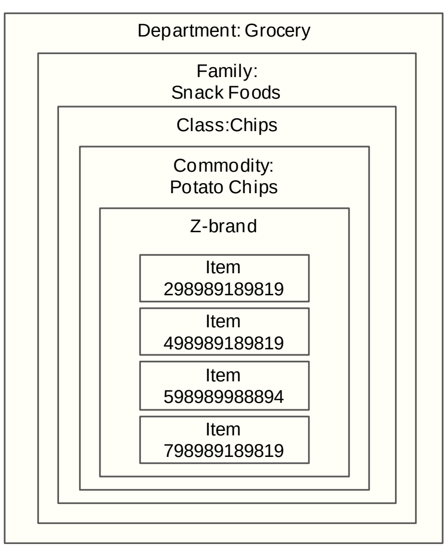
The Item Dimension entity contains ❷ additional item identifiers as owned attributes. In this reference model there are POS item identifiers (scan codes) and supplier item identifiers. Retailers may elect to add additional item identifiers as long as they adopt a single enterprise wide **ItemID** for the primary key and as long as those addition identifiers are supported in the ODM.

A number of ❸ item descriptive characteristics are supported in the reference model. All of these are mapped directly from the ODM. There are a number of additional descriptive item attributes in the ODM that can be added to or replace those shown in Figure 10.

Item handling and control attributes ❹ and buying and merchandising attributes ❺ may be added or changed as long as they can be traced to the ODM.

The attributes discussed up to this point serve as ways to filter inventory and sales fact data based on item characteristics. The next group of attributes defines a retail classification hierarchy ❻ that is used to summarize inventory and sales facts by merchandise hierarchy levels. Figure 11 illustrates the merchandise structure built into the Item Dimension entity type with sample data. The grouping of hierarchy points and the report show how the aggregation/drill down aspect of the merchandise hierarchy works.

Figure 11: Example Product Hierarchy



Sample Merchandise Hierachy Summary Report

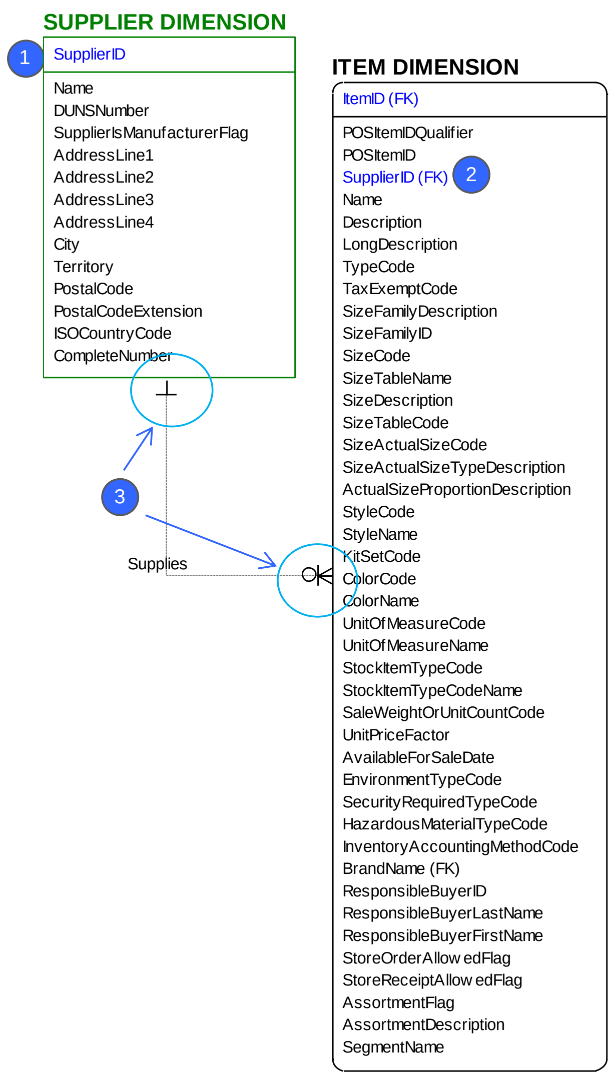
Merchandise Structure	Item	Units	Unit Price	Sales
Grocery				
Snack Foods				
Potato Chips				
Z-Brand				
	298989189819	398	0.89	354.22
	498989189819	346	0.89	307.94
	598989988894	107	0.79	84.53
	798989189819	487	0.99	482.13
Z-Brand Subtotal		1,338		1,228.82
X-Brand				
	389487567676	478	0.69	329.82
	120987368874	488	0.79	385.52
	39287655368	100	0.89	89.00
X-Brand Subtotal		1,066		804.34
Potato Chips Subtotal		2,404		2,033.16
Snack Foods Subtotal		2,404		2,033.16
Grocery Subtotal		2,404		2,033.16

This approach to modeling hierarchies is used throughout all of the ARTS DWM dimension tables. Likewise, the underlying ARTS ODM uses a basic pattern to modeling hierarchies.

2.3.2 Supplier Dimension

The **Supplier Dimension** entity ❶ identifies and provides basic contact information about the manufacturer or wholesaler that supplies the item. As modeled, the **Supplier Dimension** is related to the **Item Dimension** entity through a foreign key ❷, not to a fact entity. Also, the relationship is limited to one **SupplierID** per **ItemID** ❸. This is a limitation imposed to simplify the Release 2 DWM. Adding multiple supplier capability requires adding **SupplierID** to the primary key of the **Item Dimension** entity type which would complicate the rest of the model. The best way to overcome this constraint is to develop a supplier-item data mart that is tailored to monitoring and controlling item movement by supplier-item as part of a future DWM release.

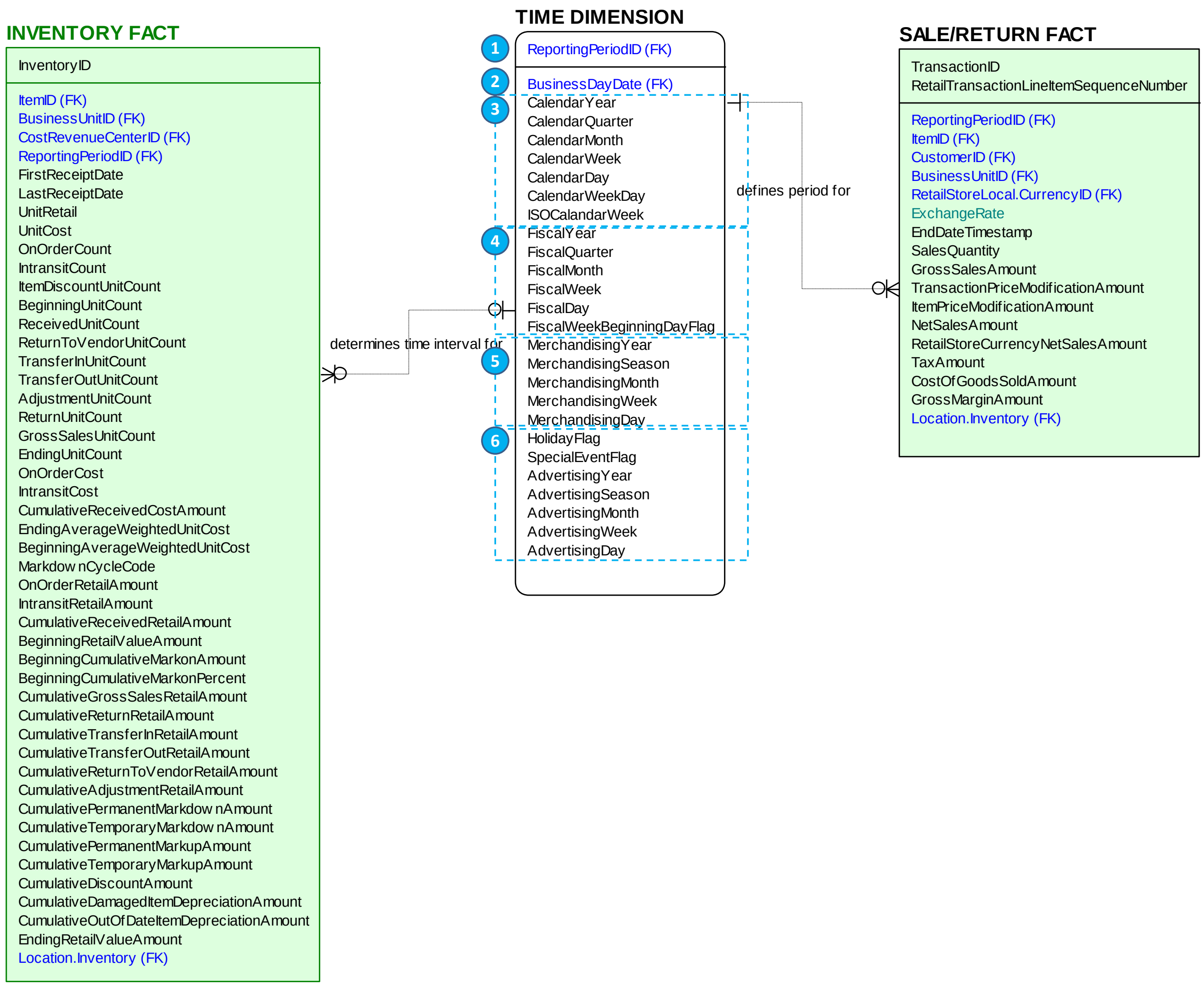
Figure 12: Supplier Dimension Entity Type



2.3.3 Time Dimension

The time dimension defines the time intervals used to assign business activity to one or more calendars.

Figure 13: Time Dimension Entity Type



The **Time Dimension** entity type places Inventory and Sale/Return facts into a reporting period. It defines the time interval that a fact is assigned to for reporting and analysis purposes. The reporting period is identified by a **ReportingPeriodID**. A reporting period may represent a range of different time periods from one business day to a year. *In Release 2 of the ARTS DWM, the* ***ReportingPeriod*** *is assumed to correspond to a business day*. This assumption is required to ensure that time periods used for sale/return facts and inventory facts are conformant. It does not prevent a retailer from summarizing business day facts into weeks, months or other higher-level reporting periods. It is required because the common denominator across the different ODM source entities used to populate the DWM fact tables is reporting period.

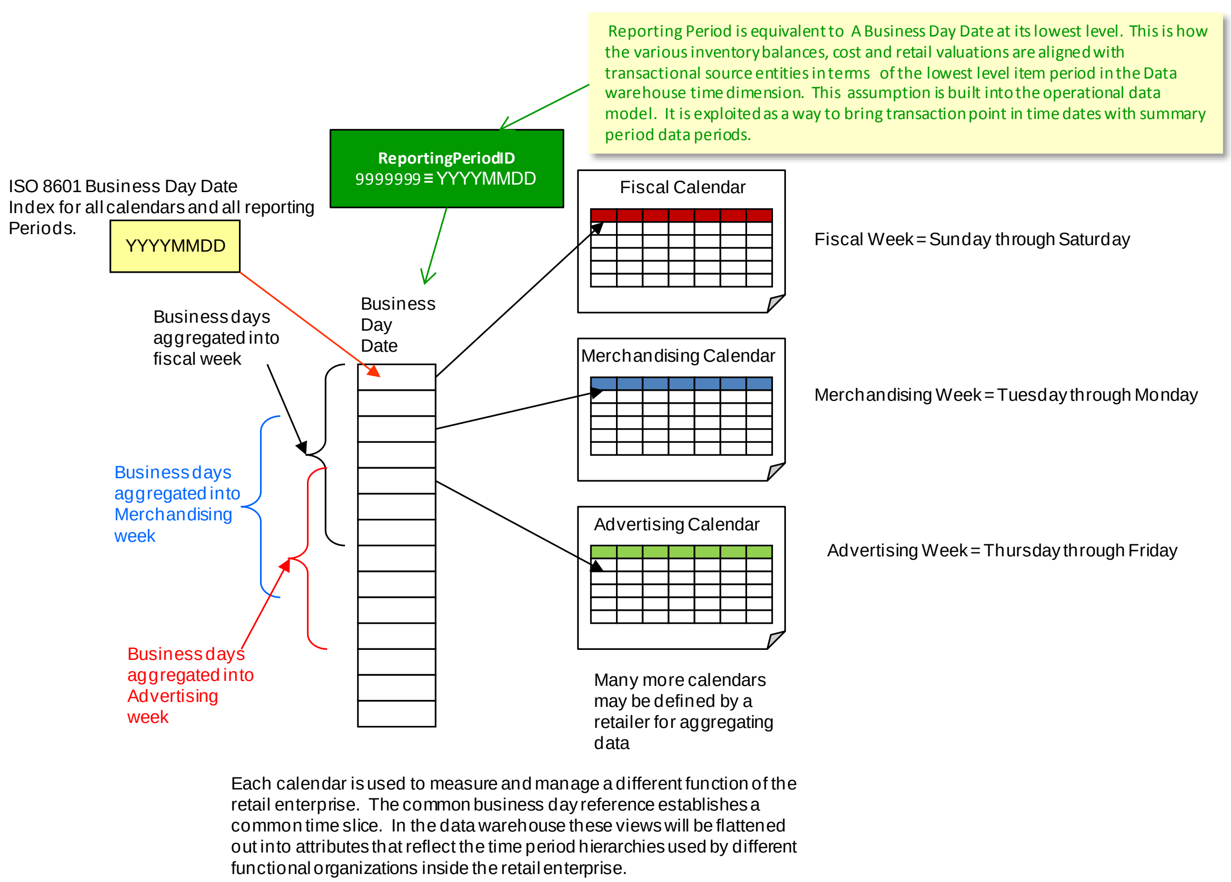
ReportingPeriodID ❶ is the primary key of the Time Dimension entity type. **ReportingPeriodID** is a key used to uniquely identify each instance of a business day. The **ReportingPeriodID** always corresponds to a **BusinessDayDate**. The BusinessDayDate attribute physically corresponds to a time stamp data type.

The **CalendarYear**, **CalendarMonth** and **CalendarDay** of the Gregorian baseline group ❸ can be concatenated into a ten character ISO 8601 format ("YYYY-MM-DD") as required for reporting purposes.

The reporting period (along with the facts linked to it through foreign keys) may be grafted on to a number of different calendars. The ARTS ODM allows a retailer to create multiple calendars to support different temporal views of its business. In Release 2 of the DWM several calendars are presented for illustration purposes. These include the baseline Gregorian calendar ❸, a fiscal calendar ❹, a merchandising calendar ❺ and an advertising calendar ❻. These calendars may be extended (based on the calendar types reflected in the ARTS ODM).

Figure 14 shows how the **Time Dimension** entity is mapped to different calendars. Because every fact to assigned to a **ReportingPeriodID**, they can be summarized and classified into a number of different calendar periods and still remain conformant at the detail level.

Figure 14: Illustration of ARTS DWM Time Dimension and Alternative Calendars



2.3.4 Business Unit Location Dimension

The Business Unit Location Dimension places inventory and sale/return facts into a location within the retailer and assigns the facts to an organizational unit within the retailer. This dual purpose entity reflects a store-orientation based in the ODM. In future releases both the ODM and DWM will be less store-centric.

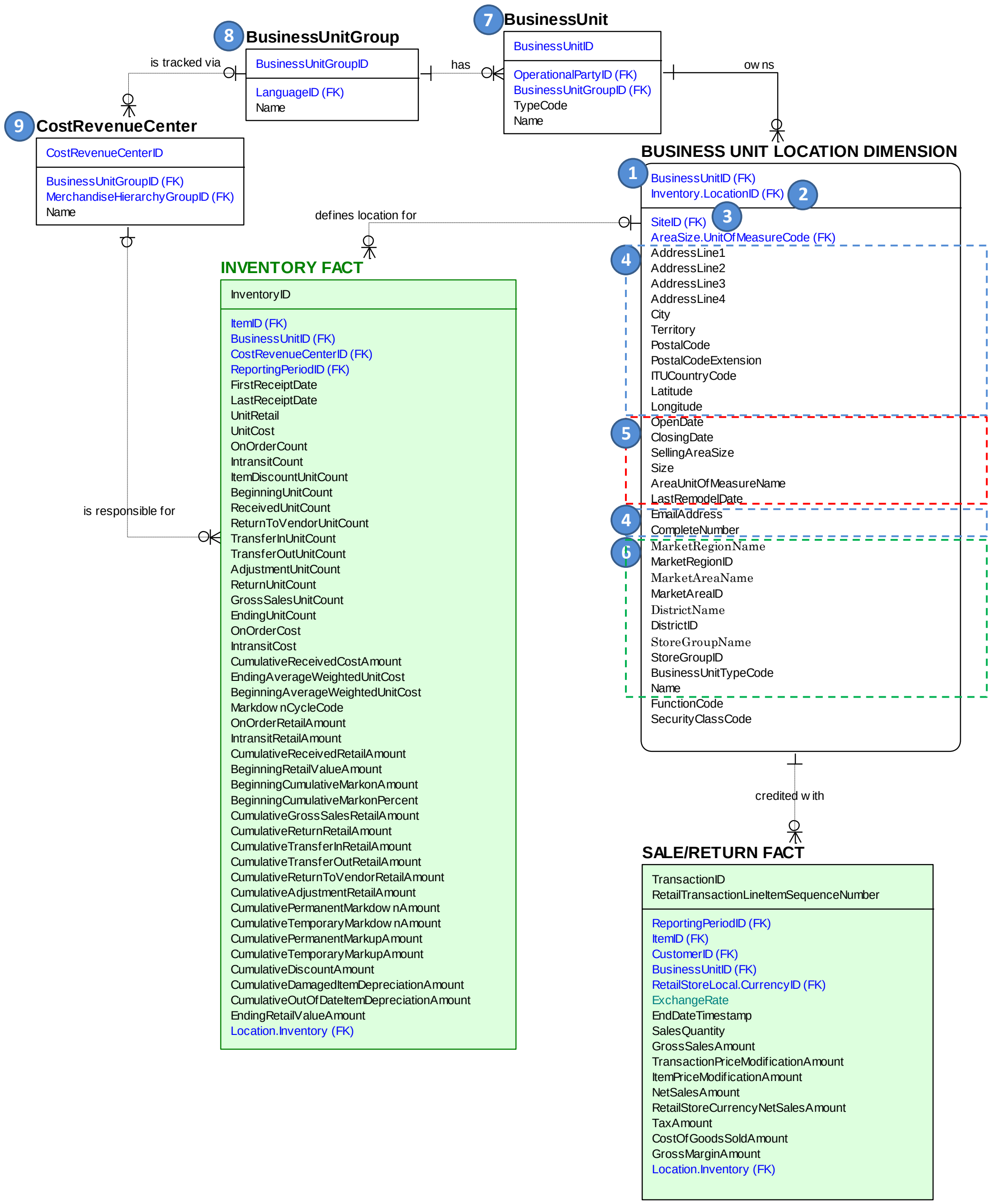
A **Business Unit Location Dimension** entity is an identified, named, described physical or virtual place at which a retailer conducts business which may include selling merchandise and services, stocking merchandise, receiving merchandise, processing merchandise, transporting merchandise, business finance and administration, manufacturing, cross docking and other activities

Examples of Business Unit Locations include:

- Retail Stores ("bricks and mortar");
- Distribution Centers;
- Warehouses;
- Administration Center;
- Manufacturing Plant;
- Web Site (URL); and
- Catalog.

Business units are hybrid entities that identify and describe places and geographically based organizational responsibilities. As currently modeled, the Business Unit Location Dimension retains the store-orientation of the ARTS ODM. In the Release 2 DWM, the **Business Unit Location Dimension** is supplemented by **BusinessUnitGroup** and **BusinessUnit** entity types. A **BusinessUnit** is a specific retail store, distribution center, warehouse, administrative office or other location that is involved in retail operations. A **BusinessUnit** entity, as retail moves from a store-orientation to a channel orientation can also identify and describe a web site, catalog, kiosk or other virtual place where retail business is conducted.

Figure 15: Business Unit Location Entity Type



A **BusinessUnitGroup** ⑧is a set of related **BusinessUnit** entities. Examples of **BusinessUnitGroup** entity types include store groups, warehouse regions, etc. **BusinesUnitGroup** entity type have dependent **CostRevenueCenter** entity types ⑨ which designate financial accounts , that own merchandise and are used to track the financial performance of the retail enterprise. For DWM Release 2, the **BusinessUnitGroup** is included because it provides the logical join path between a **Business Unit Dimension** Entity, a **BusinessUnit** entity and the **CostRevenueCenter** entity. The **CostRevenuCenter** entity type is important because a number of retail and cost valuation source ODM entities assign retail and cost valuations of inventory to cost revenue centers not to **BusinessUnit** entities. This collection of entities and rules that define how they are populated provide the mechanism for enforcing conformance across the heterogeneous operational model entities used to populate the Inventory Fact entity type.

The Business Unit Location Dimension entity is identified by a **BusinessUnitID** ❶and an **InventoryLocationID** attribute ❷. The business unit designates a store, warehouse, etc. The **InventoryLocationID** designates a physical place within a business unit that holds inventory. Inventory locations may be stock rooms, cold storage areas, freezers, gun rooms or any other point of inventory control within a business unit. A **SiteID** ❸ designates a physical geographic location for a business unit. The **Business Unit Location Dimension** contains a number of address, phone and related contact information ❹. Retail selling area, store operation dates and times and other operational information about a business unit are included ❺.

Like the item dimension, the **Business Unit Location** dimension has a set of hierarchical levels formatted as contiguous keys ❻. These keys provide the basis for performing aggregation and drill down reporting. The hierarchy point keys are derived from the ODM location hierarchy.

The **Business Unit Location** dimension entity descriptive attributes and hierarchy level attributes may be expanded as required as long as the new attributes can be mapped to source ODM attributes.

3 ARTS Operational Data Model – Data Warehouse Model Mapping

This part of the narrative describes how the entities and attributes that make up the ARTS Data Warehouse Model are derived from the ARTS Operational Data Model. As discussed earlier, this is a requirement to make sure that all ARTS data model work products are consistent. For retailers and application developers that adopt the ARTS ODM, this section represents an extract-transform-roadmap that can be used to save time and money populating a data warehouse.

3.1 Mapping Fact Entity Data

The most complex mapping rules for Release 2 of the DWM is between the ODM and DWM **Inventory Fact** entity and attributes.

3.1.1 Inventory Fact ODM to DWM Mapping

The counts, retail values, cost values and balances for the Inventory Fact entity come from a variety of different operational data model entities and attributes. These include:

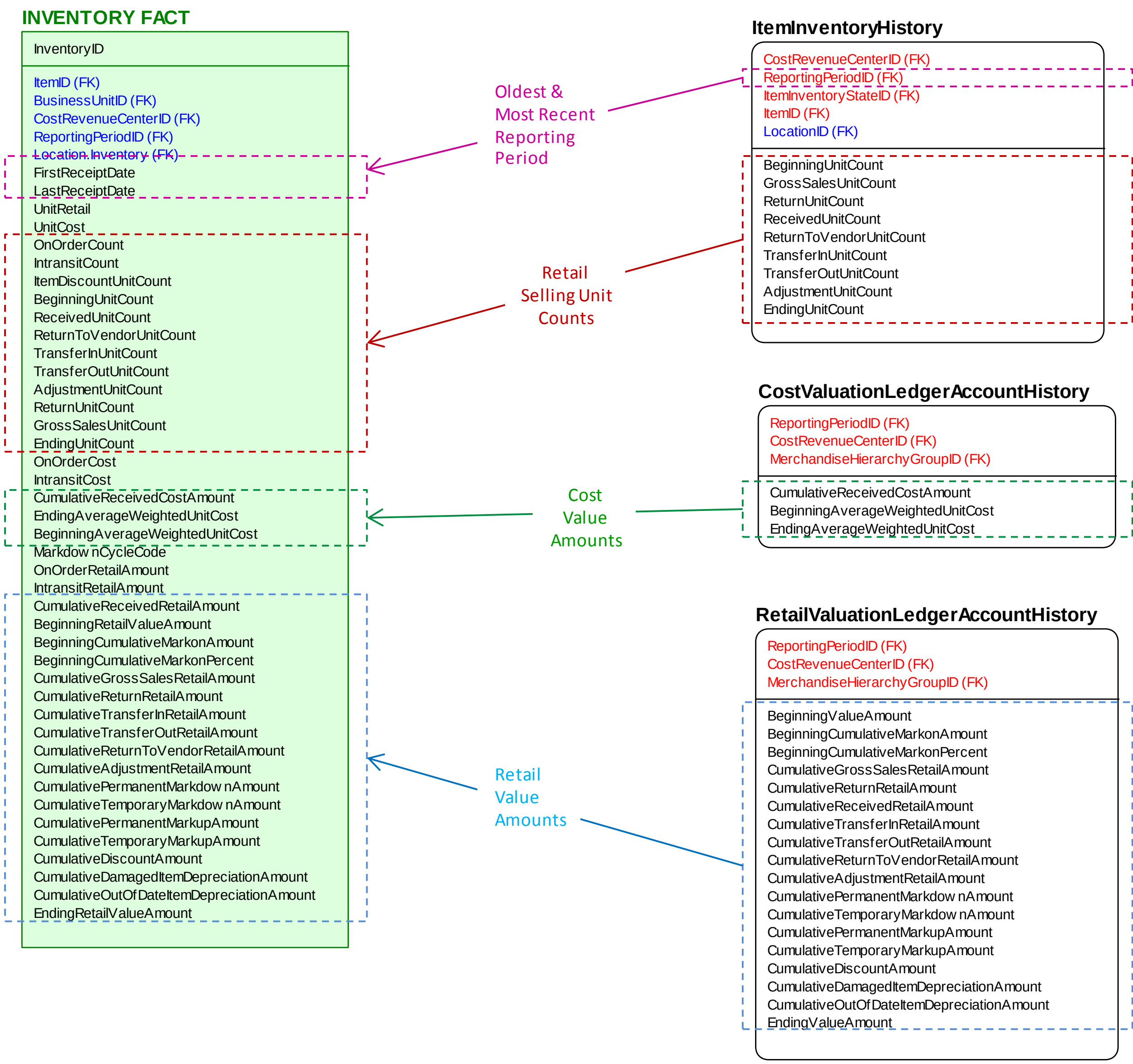
- Transactional entity types which capture the original atomic-level data created during a sale, receipt, inventory adjustment or any other transaction-based business activity; and
- Aggregated entity types which are ODM summaries of transactional detail and include balances, cumulative summaries of activity and other aggregations derived from transaction entity types.

The mapping logic defined in the Inventory Fact Entity Mapping from the ODM depends on several essential assumptions and assertions. These include the assumptions that:

- All facts are assigned to a **ReportingPeriod** that is equivalent to one business day;
- The cumulative data presented in the **Inventory Fact** entity is the cumulative count or value for that period not a running summary;
- The lowest level **MerchandiseHierarchyID** always corresponds to **ItemID** on a one-to-one basis (this is how the ARTS ODM merchandise hierarchy is designed but it is important that it its implemented with this assumption in place; and
- The **BusinessUnitGroupID**, **BusinessUnitID**, **CostRevenueCenterID** and **Inventory.LocationID** represent a hierarchy. This is not enforced within the ARTS operational data model but is required to align inventory facts and sale/return facts for aggregation.

Figure 16 shows the operational entities used as the source for the Inventory Fact entity's owned attributes. Unlike the Sale/Return attributes, the inventory draws its counts, balances, retail and cost valuations from a wide variety of tables.

Figure 16: Mapping Model For Inventory Facts

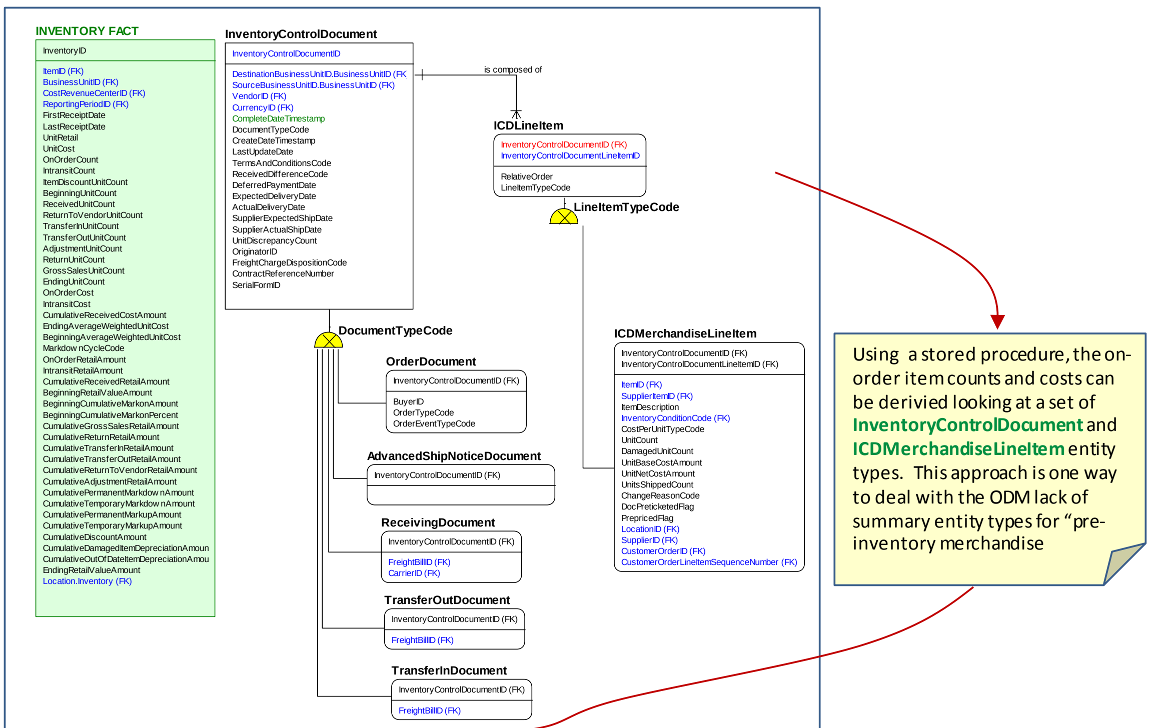


Notice that the attributes are grouped into distinct types of facts. Also notice that each attribute in a group is drawn from the same operational data model entity as its cohorts. This pattern is possible because the ARTS ODM provides several aggregation entities used for reporting. This is also why earlier in this document we mentioned that retail valuations, cost valuations, average unit costs, etc. are read from the ODM.

All of the mappings illustrated in Figure 16 reflect inventory that has been received by the retailer. There are several quantities that reflect inventory that is on order or in-transit. These counts require a more complicated mapping process that accesses order and advanced ship notice inventory control document (ICD) entities for a reporting period. In-transit and order item unit cost figures are taken from the **ICDMerchandiseLineItem UnitNetCostAmount** for Release 2 of the DWM for the reporting period. These figures (from different documents) have to be aggregated into a reporting period. Furthermore, a purchase order may specify a single item to be delivered to multiple retailer locations at different costs.

For now, this complexity has to be addressed procedurally by aggregating all item counts and costs in a purchase order and averaging their cost, aggregating item counts and averaging costs across multiple purchase order documents, and matching intra-period receipts that reduce the on-order summary for the item. This complexity points to a need to create additional ODM purchase order, in-transit and supporting reference entities. These additional summary entities are necessary to summarize and handle item counts, cost and retail valuations related to transactions that record a commitment to buy but do not reflect the physical possession of items. For Release 2 of the DWM, implementers will have to create a procedural solution to populate the on order and in-transit values. Figure 17 shows one way this approach may work.

Figure 17: Illustration of Handling On-order Item Counts and Costs



Item	ICD	Action	Date	Qty	Unit Cost	Ext Cost	On Order	Received	On Order	On Order Balance	On Order Bal	On Order Avg Cost
Item 1293 BOX	PO A	P	1/5/2010	100	27.99	2,799.00	100	0	2,799.00	100	2,799.00	$ 27.99
	REC	R	2/3/2010	100	27.99	2,799.00	(100)	100	(2,799.00)	-	-	$ -
	PO B	P	3/1/2010	150	24.99	3,748.50	150	0	3,748.50	150	3,748.50	$ 24.99
	PO C	P	4/12/1010	200	22.99	4,598.00	200	0	4,598.00	350	8,346.50	$ 23.85
	REC	R	5/1/2010	100	22.99	2,299.00	(100)	100	(2,299.00)	250	6,047.50	$ 24.19
	REC	R	7/1/2010	100	22.99	2,299.00	(100)	100	(2,299.00)	150	3,748.50	$ 24.99
	PO D	P	12/5/2010	500	17.99	8,995.00	500	0	8,995.00	650	12,743.50	$ 19.61
	REC	R	12/15/2010	300	17.99	5,397.00	(300)	300	(5,397.00)	350	7,346.50	$ 20.99
	REC	R	1/7/2011	200	17.99	3,598.00	(200)	200	(3,598.00)	150	3,748.50	$ 24.99

The derivation of an on-order count and cost requires procedural logic to read through open PO's for each item and calculate the OnOrderBalance (Units and monetary value). An averaging approach (here a running average) can be used to establish an average On Order Unit cost. This solution can be extended to address in-transit ICD's (Advanced Ship Notices) and to handle other actions like return to vendor. These "other actions" are methods for reducing the on-order balance.

This table is a rough sketch, not a formally modeled solution. The intent is that each line in this table represents an ICD and ICD merchandise line item for each stock item. The query represents an item view of the ICD (purchase order) and its line items. It rolls everything into an item value in this example. The dates represent transformation states for on-order counts. Reporting periods (business days) falling between these dates use the last transform on-order balance in this example

Accessing a retail unit price for determining on order and in-transit retail valuation is a more complicated process. Item selling prices are carried in several entities. CURRENT retail prices are carried in the **ItemSellingPrices** entity type. Because these retail prices are current, are subject to change on a daily basis (or potentially a more frequent basis). For our purposes, we are assuming that each inventory fact reporting period is a business day -- so part of the daily ETL process will read the current day's selling price from the **ItemSellingPrices** entity and use it to set the Inventory Fact entity **UnitRetail** value and as part of a calculation to assign a retail value to on order and in-transit items.

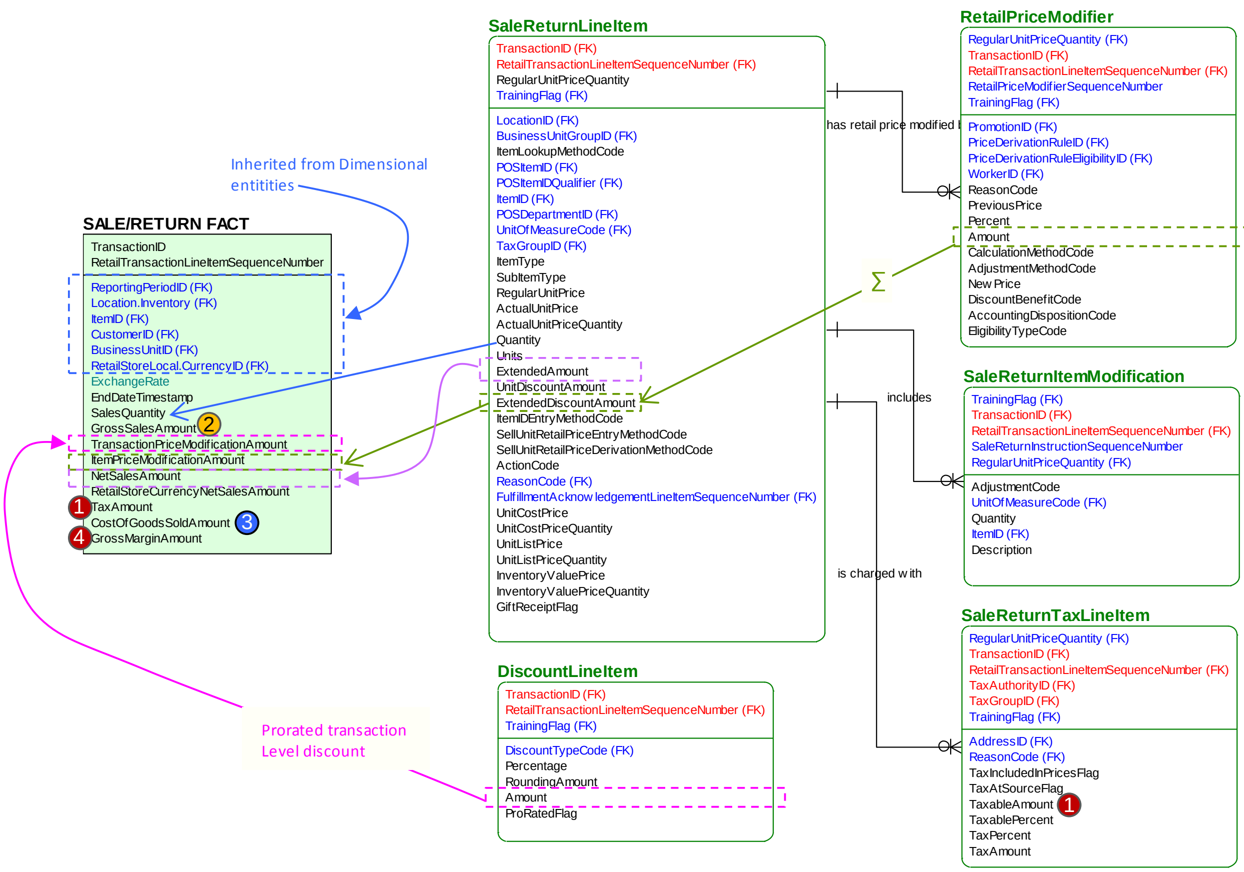
One limitation of this approach is that retail price is qualified only by item. So, the retail valuation of on order and in-transit items does not reflect geographic or organization retail price variations. This limitation does not apply to the other retail and cost valuations because they are assigned by **MerchandiseHierarchyGroupID** (which, at its lowest level corresponds to an item) and **CostRevenueCenterID** (which can support pricing variability within the retail enterprise).

Appendix B: ARTS Data Warehouse Model Release 2 Entity-Attribute Mapping List presents a detailed DWM – ODM transformation map. It explains what steps are involved in extracting, transforming and loading operational data into the data warehouse model.

3.1.2 Sale/Return Fact ODM to DWM Mapping

The Sale/Return fact entity is a simplified store-level sales reporting data collection. The mapping process required to trace its additive attributes back to the operational data model are less complicated than the process required for Inventory. All of the fact counts, monetary values and ratios are drawn from sales transaction entities as illustrated in Figure 18.

Figure 18: Sale/Return Fact ODM to DWM Mapping Model



2 GrossSalesAmount = RegularUnitPrice * (Quantity / RegularUnitPriceQuantity)

3 CostOfGoodsSoldAmount = UnitCostPrice * (Quantity/UnitCostPriceQuantity)

4 GrossMarginAmount = GrossSalesAmount - CostOfGoodsSoldAmount

The SALE/RETURN FACT entity is populated from ODM sales transaction related entities. There are some basic derivations illustrated in the formulas in the lower left area of the diagram.

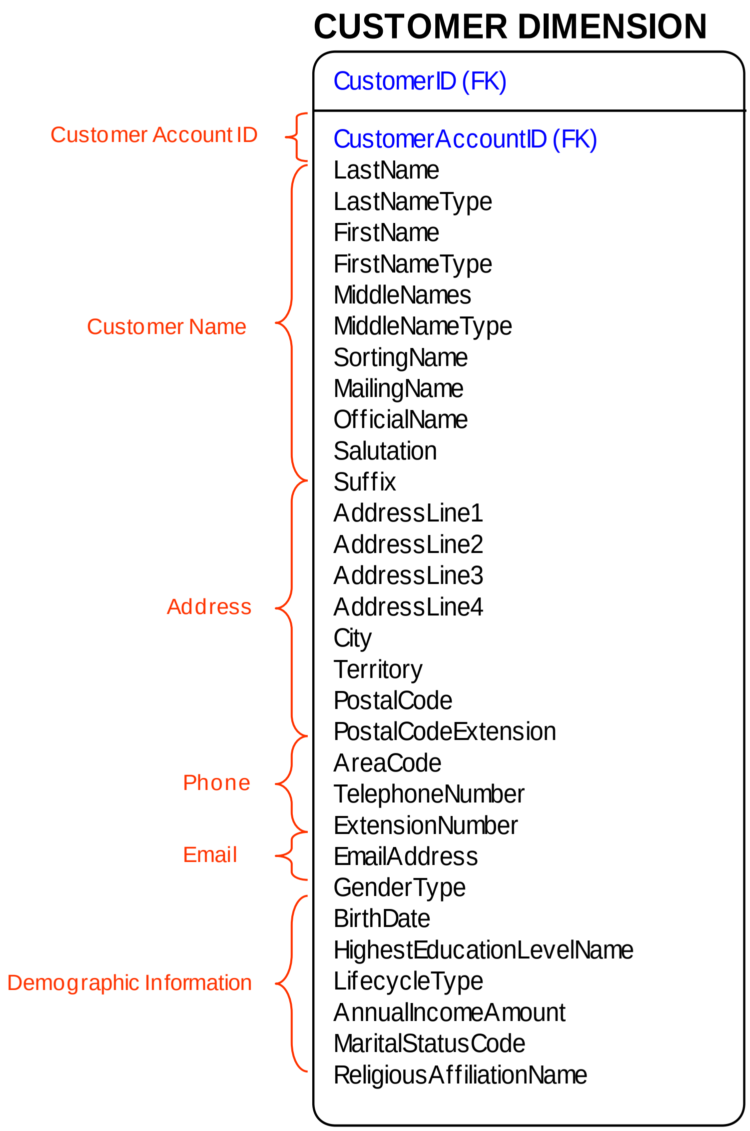
3.2 Mapping Dimension Entity Data

The mapping of dimension entity types reflects a flattening process which simplifies each entity to make it easier to use for data retrieval and summarization. This initial list of dimensional entities will grow as the DWM evolves. In its current form the DWM dimensions are conformant.

3.2.1 Customer Dimension Entity Mapping

The Customer Dimension Entity in its current form is extremely basic as shown in Figure 19.

Figure 19: Customer Dimension



The attributes shown here reflect the demographic data in the current ARTS data model. Retailers can extend the customer attributes they feel are important indicators of customer purchasing behavior and customer profitability.

The existing source entity types in the ODM currently have only code values for some attributes like **GenderType**, **LifecycleType** and **MaritalStatusCode**. In future releases of the ODM, these codes will be supplemented with names that spell out the values implemented as value lists or generalized lookup entities. Currently that support is not available for use in the DWM. The DWM will be upgraded to carry names/descriptions as soon as the ODM is updated.

The customer dimension represented in Figure 7 points to a customer loyalty account. It does not, in its current state, provide detailed information about cumulative points and rewards. These attributes are important but should be addressed in a separate data mart dedicated to reporting and analyzing a retailer's customer loyalty program. Customer loyalty purchase activity along with the accumulation of points and redemption of rewards requires a differently designed fact table – and it is good practice to create data marts organized around fact tables intended to answer different kinds of business questions. It is important in creating the data marts to use conformed dimensions so reports and queries use consistent reference and master data.

The ARTS Data Model provides the entities, attributes and relationships required to accumulate points and other tokens used to measure customer purchasing activity and their redemption. Also, it provides the data structures to describe the eligibility criteria and rules that define when, where, what and how points are earned and redeemed. A customer loyalty program data mart may be included in a future release.

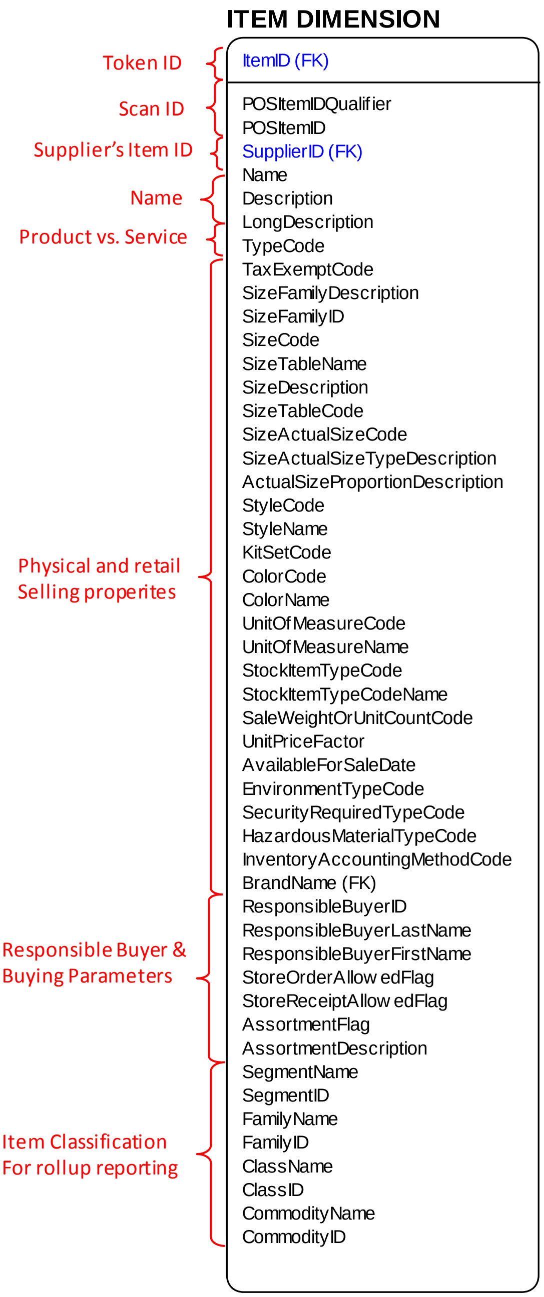
Appendix A lists the attributes and definitions for the customer dimension. The initial version of the data warehouse model provides basic customer identifying information and limited demographic information.

3.2.2 Item Dimension Entity Mapping

The item dimension identifies and describes the products the retailer stocks and sells and the services it sells. An item may be a tangible object or an intangible service. Items are the central focus for measuring sales/return activity. Like the location dimension, each retailer will tailor their item description to meet their specific business requirements.

Item dimensions will tend to be wide – that is having a large number of descriptive attributes associated with them. Figure 20 presents the item dimension for the Sample ARTS Data Warehouse Model.

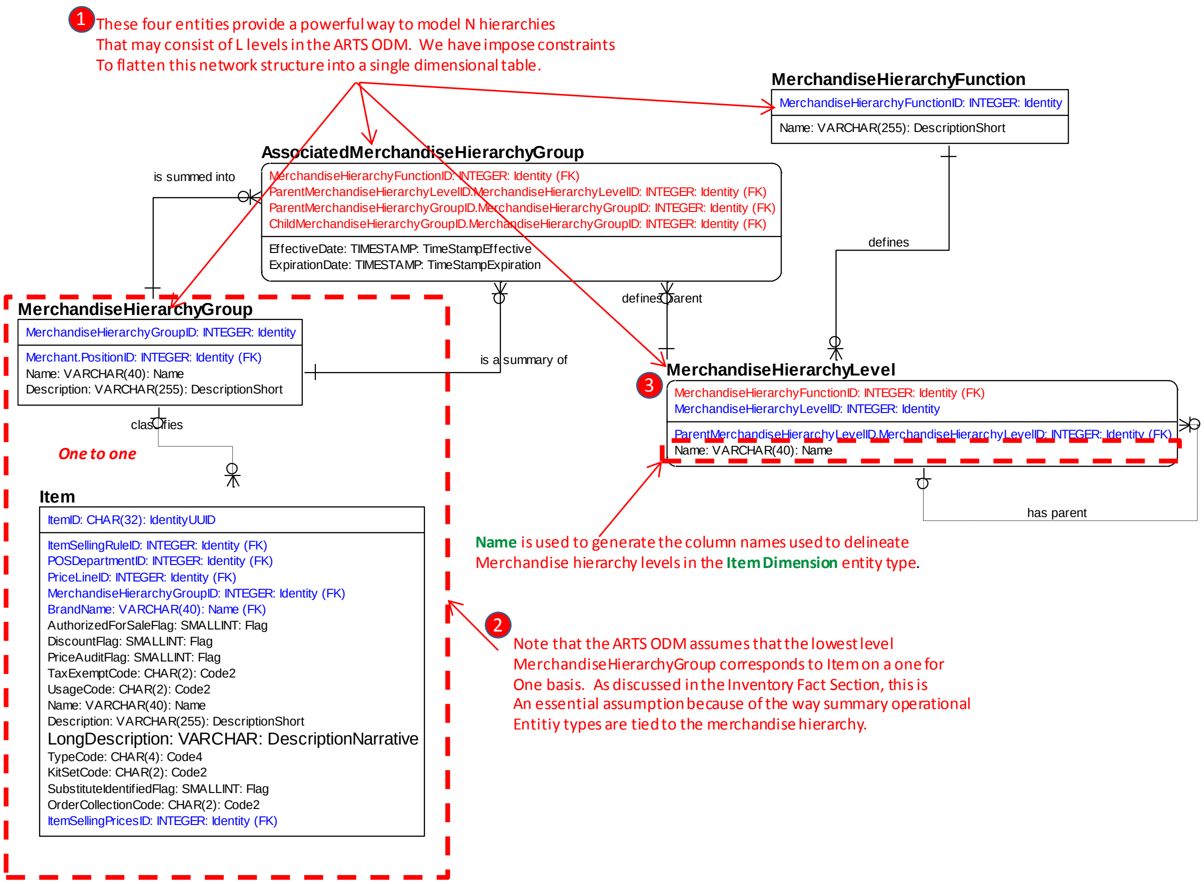
Figure 20: Item Dimension



Item classification attributes are shown at the bottom of the **ITEM DIMENSION** entity type. These, unlike the other attributes, are not directly extracted from the ARTS Data Model. They are derived from the ARTS Data Model representation of a merchandise hierarchy. Segment, family, class and commodity attributes are examples of classification levels a retailer might use to classify and manage its merchandise. The names will change with each retailer. The ARTS Data Model uses a very flexible but abstract set of entities and attributes to classify merchandise. The data warehouse should represent data structures using semantics and a structure that can be used directly by retail decision analysts.

These four levels transform the ARTS hierarchy structures into a "flattened" set of attributes that are easy and more efficient to sort, aggregate and select on.

Figure 21: Illustration of ARTS ODM Merchandise Hierarchy



The entity diagram here is a small excerpt showing how merchandise hierarchies are represented in the ARTS Operational Data Model. The network structure here is powerful and flexible but has to be constrained to create a flattened item dimension useful for building a star schema.

❶ The four entity types pointed to are the basis for hierarchically relating different levels of merchandise classifications as a collection of named hierarchies (**MerchandiseHierarchyFunction**).

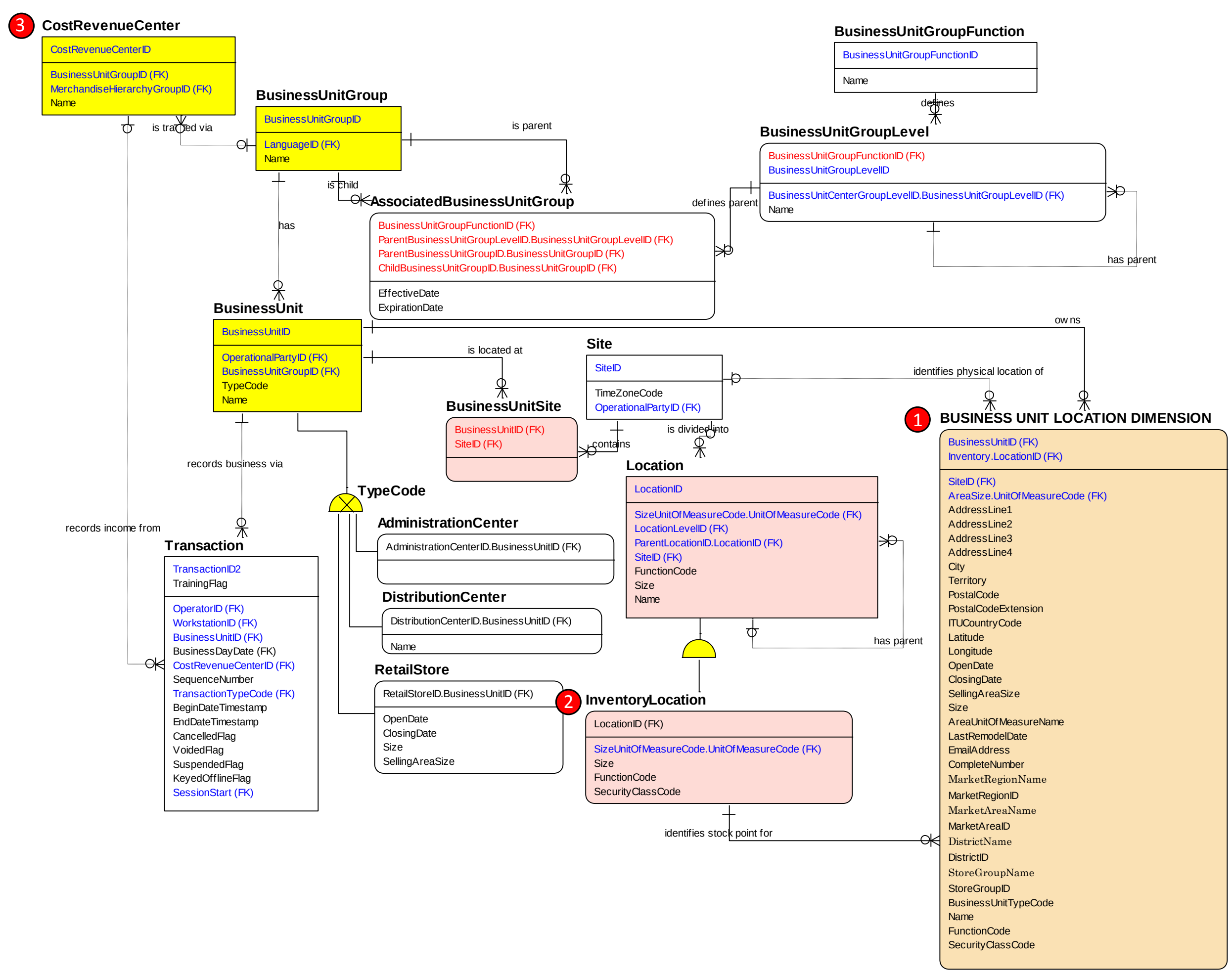
❷ The ARTS model is designed with the assumption that each Item will have a corresponding **MerchandiseHierarchyGroup**, so the lowest "set" of a items in a merchandise hierarchy is the item itself which is a group of one. This assumption is not enforced in the relationship between Item and **MerchandiseHierarchyGroup** (which is why we are emphasizing this assumption). This is a prerequisite for tying most of the summary source entities used to populate the Inventory Fact entity type's balance, retail and cost valuation attributes.

❸ The Name attribute of the **MerchandiseHierarchyLevel** entity type is transformed into a column name in the Item Dimension entity type using dynamic SQL. This reflects another constraint we have imposed in mapping the ARTS ODM to the data warehouse model. The network-oriented abstract structure which provides operational flexibility is not workable as a star schema dimension, and so we flatten it out using the level names as column names in the item dimension entity.

3.2.3 Business Unit Location Entity Mapping

Business unit location, as the name suggests, is actually a hybrid dimension. Business units and locations for inventory reporting purposes are highly correlated in the ARTS Operational Data Model. Like the Item Dimension, this one requires a flattening of the hierarchy. The ARTS ODM structure for business unit hierarchy follows the same network pattern used for the merchandise hierarchy. The strategy for transforming that structure into a set of concatenated keys useful for reporting is the same (albeit with business unit related entities).

Figure 22: Business Unit Location Entity Mapping

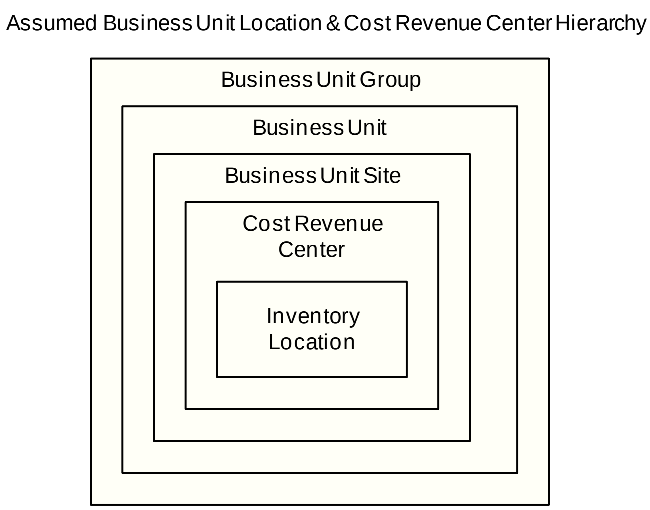


The design of the ARTS ODM required some extensions to the Business Unit Location Dimension. The extensions were required because the two primary retail and cost valuation entity sources referenced **CostRevenueCenterID** can be resolved to business unit and location but only through access to some supporting entities outside the Business Unit Location Dimension. So (as in many ETL scenarios) we added a Cost Revenue Center dimension and entities which enable us to get to business unit and location. The light-yellow entity types ❸ identify business units which represent organizational subdivisions of the retail enterprise. For the inventory reporting purposes we have included **CostRevenueCenter** in with these organizational entities. The pink entity types ❷ represent location entities. They reference physical locations where inventory counts are taken and reported on. The orange entity ❶ is the **Business Unit Location Dimension** of the Inventory Data Warehouse Model.

This entity diagram shows how **BusinessUnit**, **CostRevenueCenter** and **InventoryLocation** are linked together in a way that allows us to properly define where inventory is located and who (from an organization perspective) is responsible for that inventory. Because the **RetailValuationLedgerAccountHistory** and **CostValuationLedgerAccountHistory** use **CostRevenueCenterID** we have to make some very critical assumptions about how a retailer populates these different location and organization context entities. These assumptions are necessary because without them the mapping between the ODM and DWM may not provide useful ways to aggregate inventory data.

The **BusinessUnitGroupID**, **BusinessUnitID**, **BusinessUnitSite**, **CostRevenueCenterID**, **Inventory.LocationID** represent a hierarchy. This is not enforced within the ARTS operational data model, but is required to align inventory facts for aggregation. So when implementing the ARTS Operational Data Model care must to be exercised to ensure that these different entities are aligned. The model, Aligned Location and Organization Dimensions shows how the different entity dependencies should be organized to support inventory aggregation. Figure 23 illustrates the intent of this alignment for aggregation and drill down reporting.

Figure 23: Aligned Location and Organization Dimensions

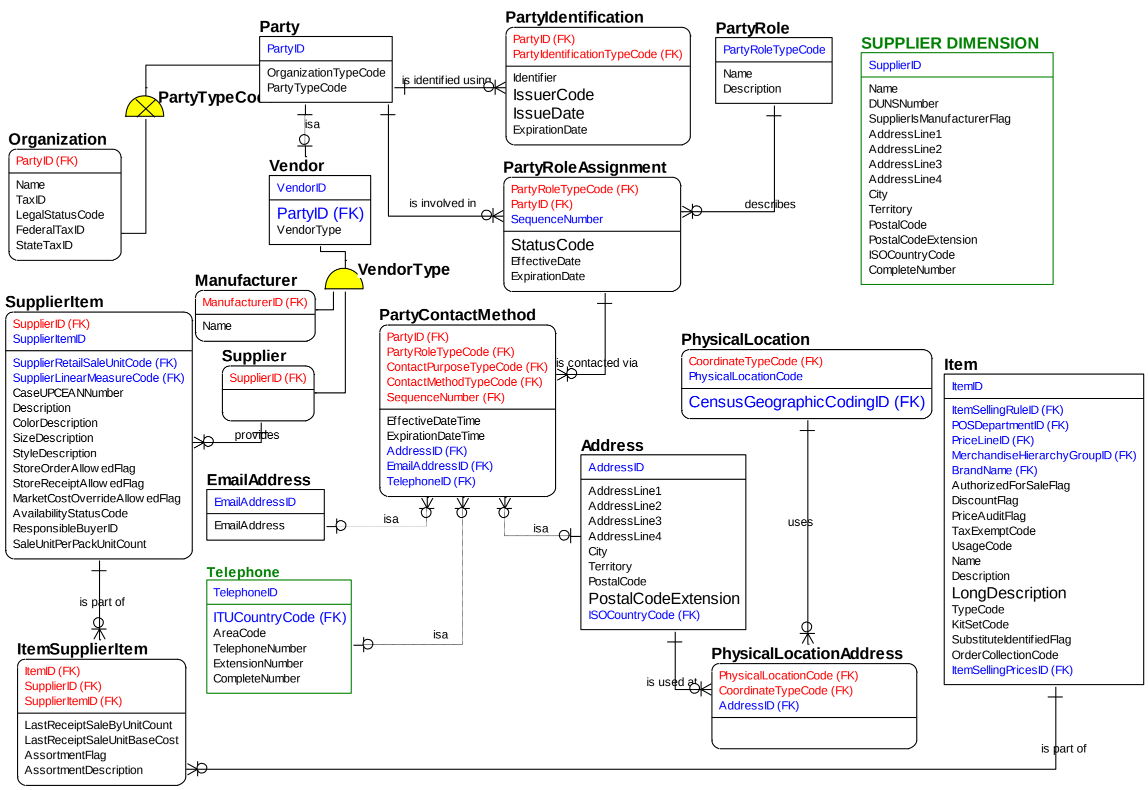


3.2.4 Supplier Dimension Entity Mapping

The Supplier Dimension Entity Mapping from Operational Data Model table lists the mapping sources and rules for supplier information. Within the ARTS Operational Data Model a Supplier entity is a subtype of Vendor. The primary keys are identical. The Supplier entity type is referenced in this mapping table because the Supplier is mapped to Item through **SupplierItem** and **ItemSupplierItem**. There is no direct linkage between Item and Vendor. Because **Supplier** is a subtype of **Vendor**

The vendors that fill the role of suppliers will be directly linked. This same assertion applies to vendors that are suppliers as well as manufacturers. Vendors that are only manufacturers and have no associated Supplier entity have no way of being associated with an Item entity in the current operational data model

Figure 24: Supplier ODM – DWM Mapping



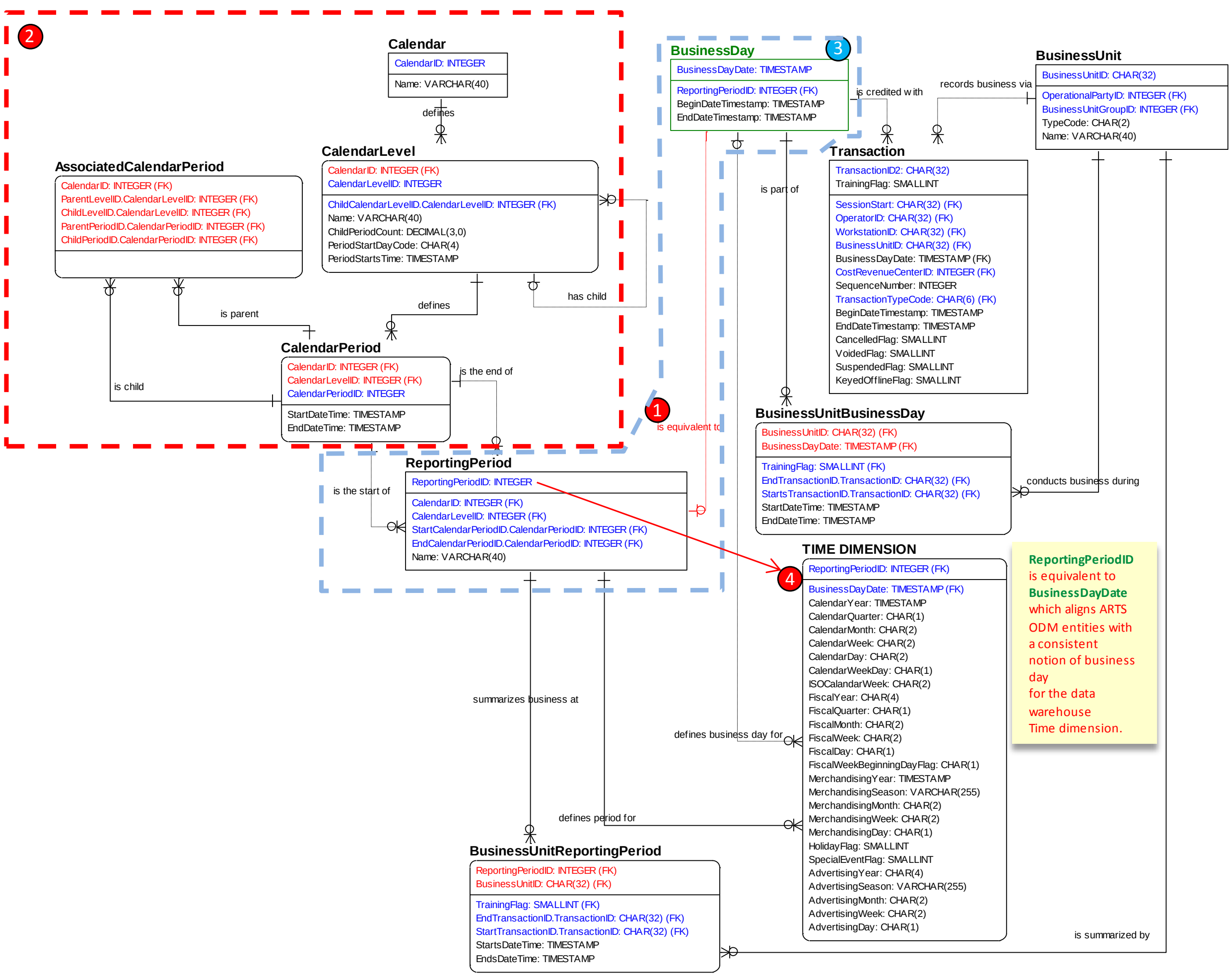
3.2.5 Time Dimension Entity Mapping

The mapping of the time dimension between the ODM and DWM requires reconciling the notion of periods (blocks of contiguous time units bounded by a beginning and ending point in time) with points in time. This reconciliation requirement is introduced by the Inventory Data Warehouse model which extracts both periodic and point-in-time data to populate the Inventory Fact entity type. Fortunately, the ARTS Operational Data Model contains some assumptions that help align periods and points in time.

- The most granular point in time, for data warehouse purposes, is the business day;

- Reporting periods may represent any number of one or more contiguous business days. For the data warehouse, the underlying operational entities that use **ReportingPeriodID** as their period identifier are assumed to be at the business day level. This is to align transactional sources with summary entity sources; and
- This works because the ARTS Operational Data Model maintains an equivalence relationship between a **BusinessDay** entity type and a **ReportingPeriod** entity type. For purposes of supporting the data warehouse, this relationship is mandatory. Without it, the time periods and point in time attributes that place facts in time will not be aligned.

Figure 25: Entity Diagram Showing ODM To DWM Time Dimension Mapping



There are some important observations to be made about this diagram

❶ The equivalence relationship defined here (which is optional) is a requirement for aligning period and point-in-time source data for the Inventory Fact entity.

❷ The power and flexibility of the ARTS ODM allows a retailer to create many calendars and time reporting hierarchies. Like the merchandise and business unit hierarchies these have to have constraints imposed as part of mapping to a dimensional representation. This means placing reasonable constraints on the number of levels (depth) of the hierarchy and choosing which calendars you want to include in the Time Dimension entity type.

❸ **BusinessDay** and the lowest level of **ReportingPeriod** are equivalent. Note that both are foreign keys of the Time Dimension entity type. **ReportingPeriodID** is used as the primary key because it is the predominant attribute referenced in the various operational entities used as the source of counts, balances, cost and retail values used in the Inventory Fact entity type.

❹ The Time Dimension entity type is the "product" of all of this mapping logic. Like its peer dimensional entity types, it provides a flattened hierarchy to allow **ReportingPeriod** - **BusinessDay** bounded quantities, counts and values to be aggregated in different higher-level time periods.

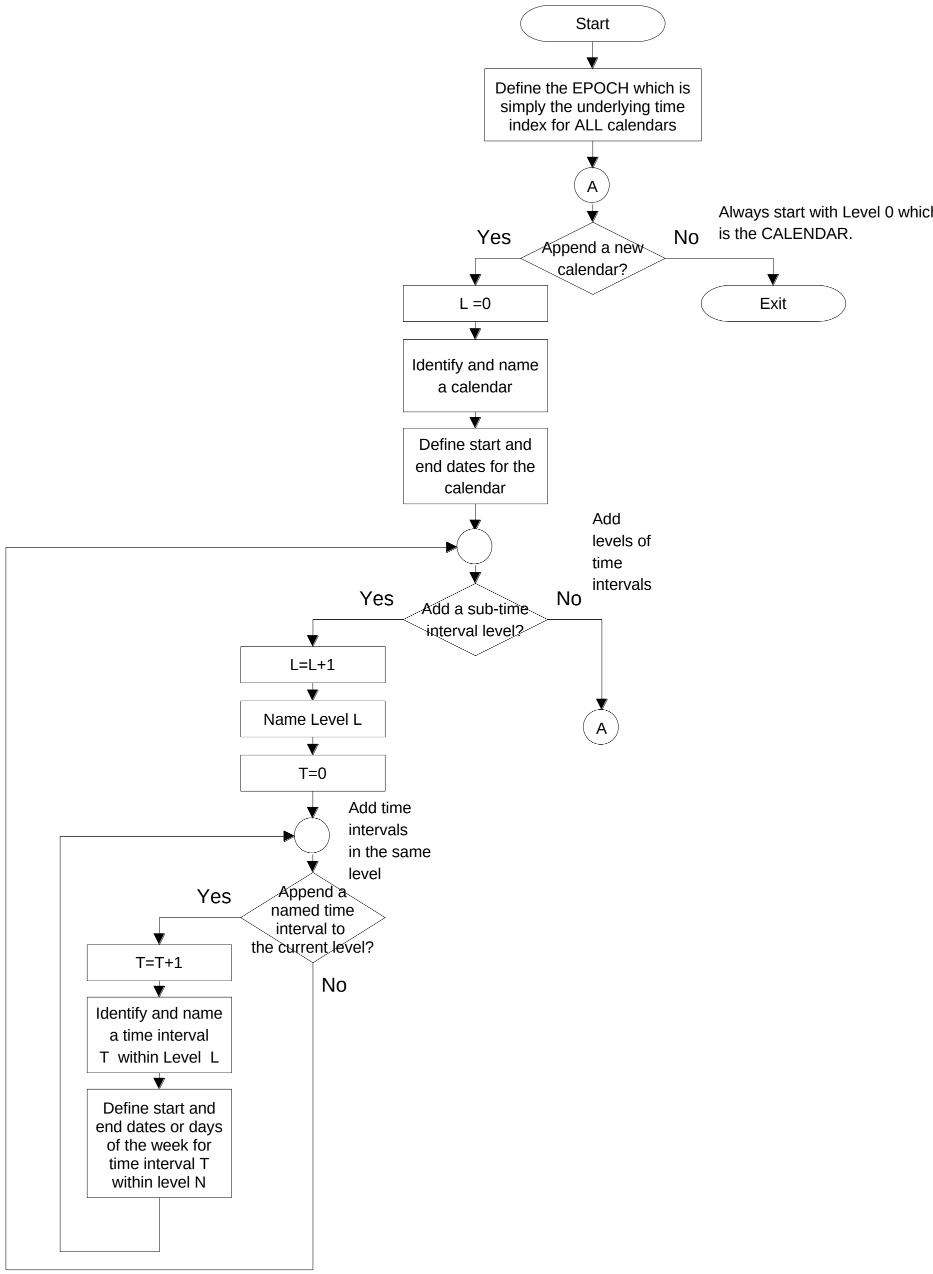
The actual mapping logic for the time dimension is based on a generation of calendar dimensions (discussed a little later, see Figure 26).

Mapping ARTS Operational Data Model time entities, attributes and relationships will follow a different pattern than the one used for location and item classification hierarchies. A major reason for this different pattern is that retail business time periods are driven by reporting requirements. The other dimensions, store location and an item are driven by geography and item physical and selling properties.

Instead of mapping time periods from the ARTS Operational Data Model, a time line will be defined divided into business day references. A business day is a calendar day (a 24 hour period of time starting at 12:00 midnight (00:00:00 in 24 hour format)) and running to 11:59:59 PM (23:59:59 in 24 hour format). For the initial release of the ARTS Sample Data Warehouse model, business day is the lowest level of granularity used for reporting.

The “mapping” process presented here is a derivation process. In effect, the approach is to generate the retail calendars as opposed to refactoring the ARTS Operational Data Model structures. The reason for this approach is that by definition, reporting periods and calendars are driven by the way a retailer wants to aggregate data over time. It is their reporting requirements and preferences that define the calendar as opposed to geography or item characteristics. Retailers that cover large geographical areas will contend with time zones. However these represent local time interval displacements from a base set of retailer calendars. The incorporation of business unit into the time dimension view of the ARTS Operational Data Model allows a retailer to establish named reporting periods for business units over a common set of retail calendars. It also enables the assigning of local, time zone-dependent names to reporting periods.

Figure 26: Notational Calendar Generation for ARTS Data Warehouse Time Dimension



4 Data Aggregation

Data aggregation is any process in which detailed fact data is gathered and expressed in a summary form, for reporting and statistical analysis. A common retail aggregation purpose is to get more information about particular customer groups based on specific variables such as age, profession, or income.

The creation and maintenance of aggregated data can be accomplished in a number of ways. One approach is to create stored aggregations as part of the dimensional data warehouse. The other approach is to use the data warehouse model defined in Figure 8 as the data source for creating OLAP cubes using a proprietary tool. For most retailers interested in doing anything beyond trivial reporting and analysis, an OLAP tool of some kind will most likely be used.

The technical design of an OLAP dimensional database is different from the relational technical design underlying an ARTS operational database built on the ARTS data model. The logical design of an OLAP dimensional database however will closely follow the star pattern already established by the ARTS Sample Data Warehouse model. Data aggregations in an OLAP environment are much easier to code than in a pure SQL environment. Also, because they are using a different technical database design, the OLAP queries perform better than a purely relational database. Even with this improved performance and ease of use, it still is important for retailers to define an aggregation strategy for their data warehouse. This section of the document will discuss some of the issues related to defining a data warehouse aggregation strategy. Like the ARTS operational data model and the data warehouse model, this strategy will be cast as a logical model.

4.1 Data Aggregation: Defining Logical Cubes for Reporting & Analysis

Data aggregations summarize detailed fact data using different combinations of dimensions from the item, store location, customer and time entity to define rollup levels. Aggregations use one or more dimensions to establish selection criteria for detailed facts to be summarized. A one way aggregation uses one dimension. Aggregating item sales (for the whole enterprise) by day is a one way aggregation. A two way aggregation uses two dimensions. For examples store group sales by item class by day (day is not aggregated since it is at the fact level) is a two way aggregation. Store group sales, by item class, by fiscal month - is a three way aggregation. Aggregations may be mixed and matched across dimensions. This means that a data warehouse model can potentially have to create and maintain a large number of aggregates. Figure 9 illustrates the main aggregation levels for the initial ARTS Data Warehouse Model. This represents the number of different rollup levels possible in each dimension. Take the number of rollup levels in each dimension, multiply them times the number of dimensions a data warehouse supports and the result is a count of possible aggregations (pre-summarized, stored data) in the data warehouse.

Figure 27 shows the attributes and a count of possible aggregations based on those attributes in the ARTS Data Warehouse Model.

Figure 27: Aggregation Levels Based on Dimension Attributes

	CUSTOMER	SUPPLIER	ITEM	TIME	BUSINESS UNIT
	CustomerID	SupplierID	ItemID	ReportingPeriod	BusinessUnitID
	PostalCode	ISOCountryCode	SupplierID	CalendarYear	CostRevenueCenterID
	LifeCycleType		SegmentID	CalendarQuarter	Location.Inventory
			FamilyID	CalendarMonth	MarketRegionID
			ClassID	CalendarDay	MarketAreaID
			CommodityID	ISOCalendarWeek	DistrictID
				FiscalYear	StoreGroupID
				FiscalQuarter	BusinessUnitGroupID
				FiscalMonth	
				FiscalWeek	
				FiscalDay	
				FiscalWeekBeginningDayFlag	
				MerchandisingYear	
				MerchandisingSeason	
				MerchandisingMonth	
				MerchandisingWeek	
				MerchandisingDay	
				HolidayFlag	
				SpecialEventFlag	
				AdvertisingYear	
				AdvertisingSeason	
				AdvertisingMonth	
				AdvertisingWeek	
				AdvertisingDay	
Aggregations	3	2	6	24	8

There are 6,192 different 5-way aggregations, 8,064 different 4-way, 2,100 3-way, 378 2-way and 38 1 way aggregations. That's a total of 17,497 possible aggregations using the dimensions listed here.

Clearly some of the 17,497 aggregations will not make sense. But the volume of combinations makes it impractical to specify aggregations as part of this document.

Each aggregation identified in Figure 27, requires its own space and processing time to perform the rollups. In most data warehouse environments, retailers will implement a subset of all possible aggregations. To choose which aggregations to carry and which to roll up from detail on an as needed basis, retailers have to prioritize each candidate aggregation. It is important to define a strategy for defining aggregations, performing pre-aggregation set selection and deciding when older drill down detail data may be rolled up and retained in summary form only.

The prioritization of aggregates requires a careful evaluation of:

- The types of reporting and analysis required to answer the business questions posed by retail decision makers; and
- The volume, range and frequency of different retail transactions identified by item, location, customer and time period.

Each retailer will choose a different aggregation strategy. This is an important point. Data warehouse and the downstream reporting and analysis products can borrow patterns but in the end must be implemented around each retailer's business.

4.2 Data Aggregation Reporting & Analysis Design Drivers

The first and most important part of choosing which aggregations to implement is to define the business reporting and analysis requirements that are to be supported using the data warehouse.

4.2.1 Business Benefit from Using a Data Warehouse

The data warehouse is a utility and may be used to support:

- The exploration and definition of business hypothesis about how retail business results (mostly dealing with sales/returns) are generated – and why;
- The testing of business hypotheses defined in the preceding step – the level of usage of the data warehouse to formulate a hypothesis may be much lower than the use to test that hypothesis; and
- Monitoring of the ongoing retail business to identify actual or potential problems or opportunities based on customer purchasing behavior.

These three high-level use cases represent different ways to employ data warehouse technology as well as related analysis and reporting tools. They provide the "why" behind a retailer's commitment to use a data warehouse.

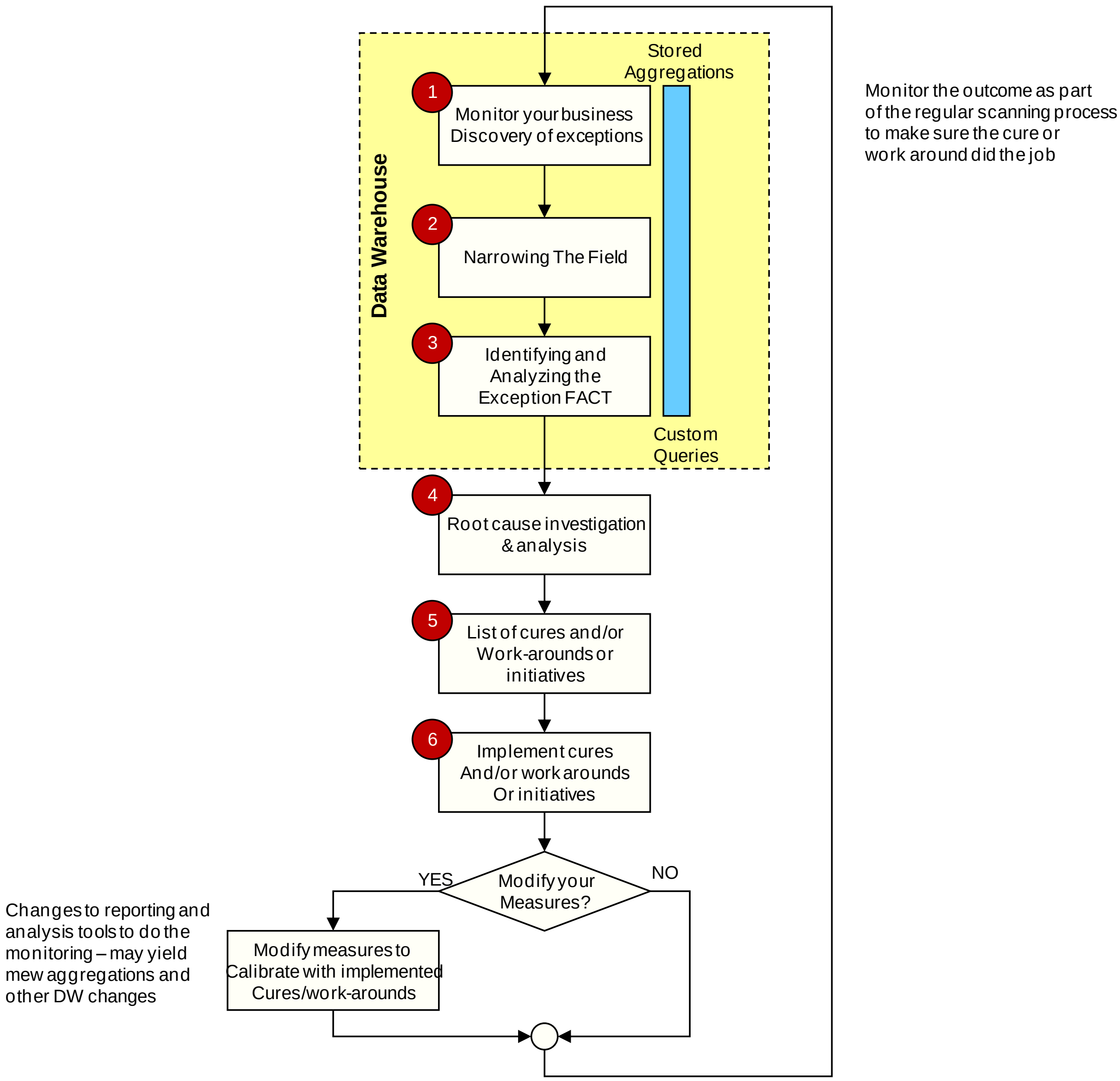
These different use cases are not mutually exclusive. In fact in a classic plan→ do→ check→ act process model, all three of these use cases will be in play for a retailer.

4.2.2 Usage Pattern – How is a Data Warehouse Used

The kinds of questions posed by business users will be colored by the data warehouse use case selected from the three listed here. How the data warehouse will be used to improve the business will follow a pattern similar to the model illustrated in Figure 28. Figure 28 places the data warehouse and supporting tools into a learning system context. The shaded area surrounded by the dashed line is the focus of the data warehouse effort backed by the ARTS model. Also, to the right of the three process blocks, the expected use of stored aggregations and the transition to the use of custom queries is shown using a graduated color scale. This pattern of use is one way a retailer can decide what aggregations make sense to create and store and which should be developed as ad-hoc queries.

The other process blocks outside the data warehouse area are important because they highlight the context for using data warehouse technology. The data warehouse is a tool used to drive business improvement (learning).

Figure 28: Data Warehouse As Part of A Learning System



The ❶ process block in Figure 28 is the area where stored aggregates are most likely to be used with the highest frequency. The aggregates are used to allow decision makers to quickly look over the business activity they are responsible for (including their subordinates areas) and spot exceptions. Exceptions, means any data spike or drop that the decision maker feels is outside the "norm". In addition to the aggregate data, detailed alerts about specific problems or opportunities may be delivered to the decision maker based on business rules defined in the underlying reporting system. Exception reporting will alert the decision maker to go directly to the next step.

Narrowing the field step ❷ in Figure 28 involves a mix of pre-stored and customized aggregations. It will depend on a number of factors including the nature of the problem or opportunity being investigated, the analytic and technical skills of the decision maker and the volume of data that has to be looked at. Narrowing the field is data "detective work". It requires a combination of rigor, discipline and a willingness to look outside the box. Narrowing the field, as the name suggests is about finding which transactions or set of transactions in the fact table are behind the exception condition discovered earlier.

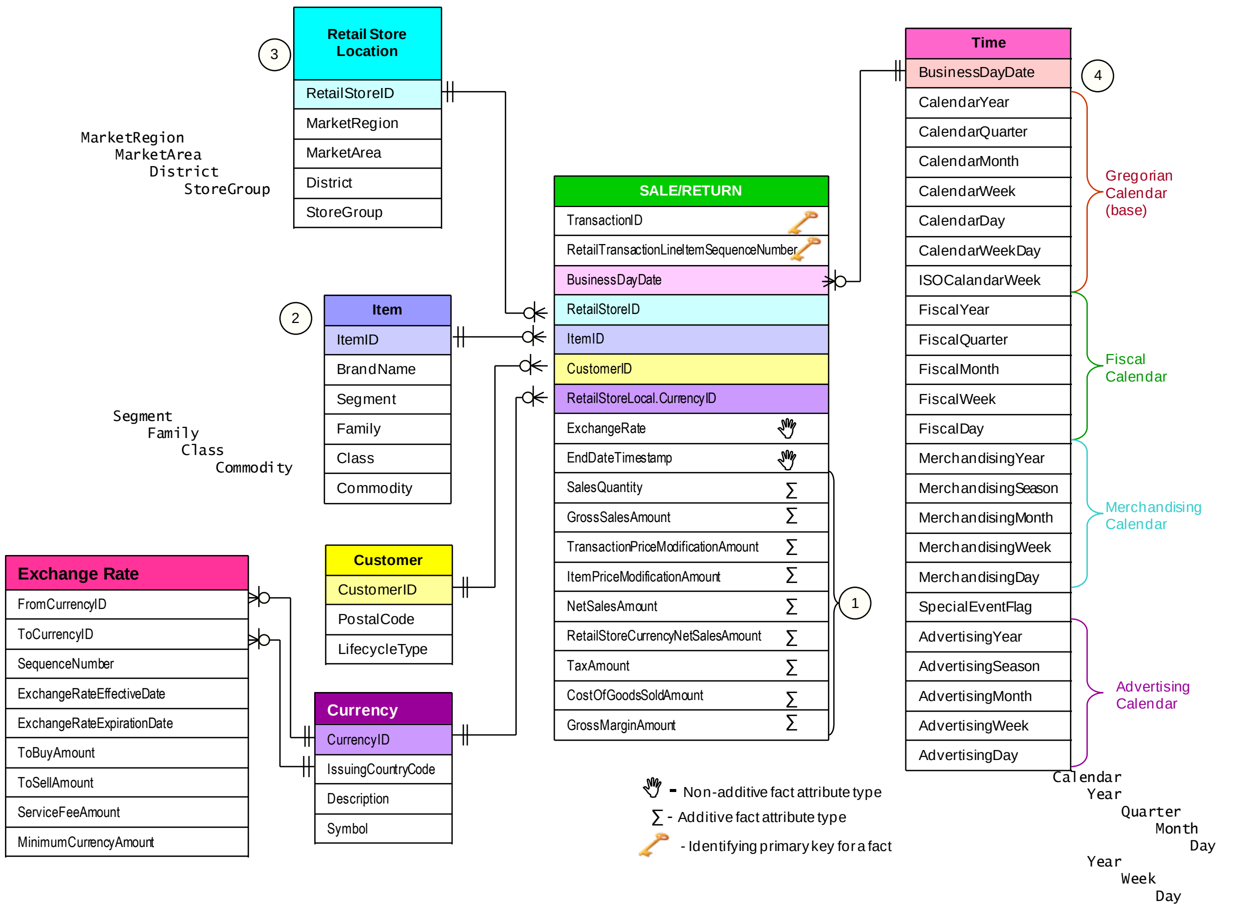
Once the transactions are "discovered", the next step ❸ involves looking at the fact rows of data in detail to figure out why their values are not in line with expectations. This typically requires custom, ad-hoc queries at a low level of detail. Also, it is a part of data warehouse usage that is not likely to be helped by stored aggregations because it is exploratory.

Steps ❹ through ❻ show how the work products extracted from the data warehouse are used to improve the business. One outcome from the improvement action may be modifying the reports and potentially the data warehouse (including changing or adding stored aggregations).

As discussed earlier, the process flow illustrated in Figure 28 applies to all three data warehouse use cases. Also, the process flow applies to On-Line Analytical Processing (OLAP) based as well as relationally based aggregation strategies. The tools and detailed syntax change – the essential process steps do not.

Figure 29 presents a detailed model of the ARTS Data Warehouse Model. It shows all of the attributes, attribute usage, hierarchy levels and key relationships. Its purpose is to provide a backdrop for additional discussion about aggregations and cubes.

Figure 29: Detailed Dimensional Model for Discussion of Aggregations & Cubes



4.2.2.1 What business activity does the retailer want to measure?

The answer to this question will identify the facts required from the SALE/RETURN entity. The choices available in the ARTS Data Warehouse Model are identified by the ① in Figure 29. Notice that each of the attributes pointed to by ① is additive. The values to choose from are measured in monetary units (currency) or retail units.

4.2.2.2 What items or classification of items does a retailer want to measure?

The answer to this question comes in two parts. First the retailer needs to choose what level of item information is needed. The choices available the ARTS Data Warehouse Model are identified by ② in Figure 29. The item classification structure is illustrated by the indented list next to the Item dimension. The retailer may choose a high-level summary or drill down to the item level. This is the first part of the answer. The second part of the answer for time is to define which instances of the item level a retailer wants included in the answer set. For example, a retailer may want to look at only two families of product and drill down to the commodity level for those two families. This second part of the answer establishes a filter to select only the fact rows that meet those criteria.

4.2.2.3 Where does the retailer want to look at the measures defined earlier?

The answer to this question, like for item, comes in two parts. First the retailer needs to choose the location level. The choices available in the ARTS Data Warehouse Model are identified by ③ in Figure 29. The location hierarchy is illustrated by the indented list next to the **RetailStoreLocation** entity type. The retailer may choose to look at business activity at an enterprise level or drill down to an individual store (or anywhere in between). The second part of the answer, like item, is to define which instances of a store, store territory, etc. to be included in the answer set.

4.2.2.4 When does the retailer want to look at the measures (i.e. what period or periods does a retailer want to look at)?

The answer to this question consists of three parts. First, the retailer needs to choose a calendar. The ARTS Data Warehouse Model supports four calendars. The retailer must choose which one(s) is(are) to be used for the answer set he/she is creating from the data warehouse. The calendars are part of the time dimension identified by ④ in Figure 29.

The second part of the answer is to define what time period levels a retailer wants to see. For example, a high-level overview may look at **MerchandisingYear** and **MerchandisingSeason**. A buyer may want a much more granular time frame and drill down to a **MerchandisingWeek**. This second choice defines the level of time period.

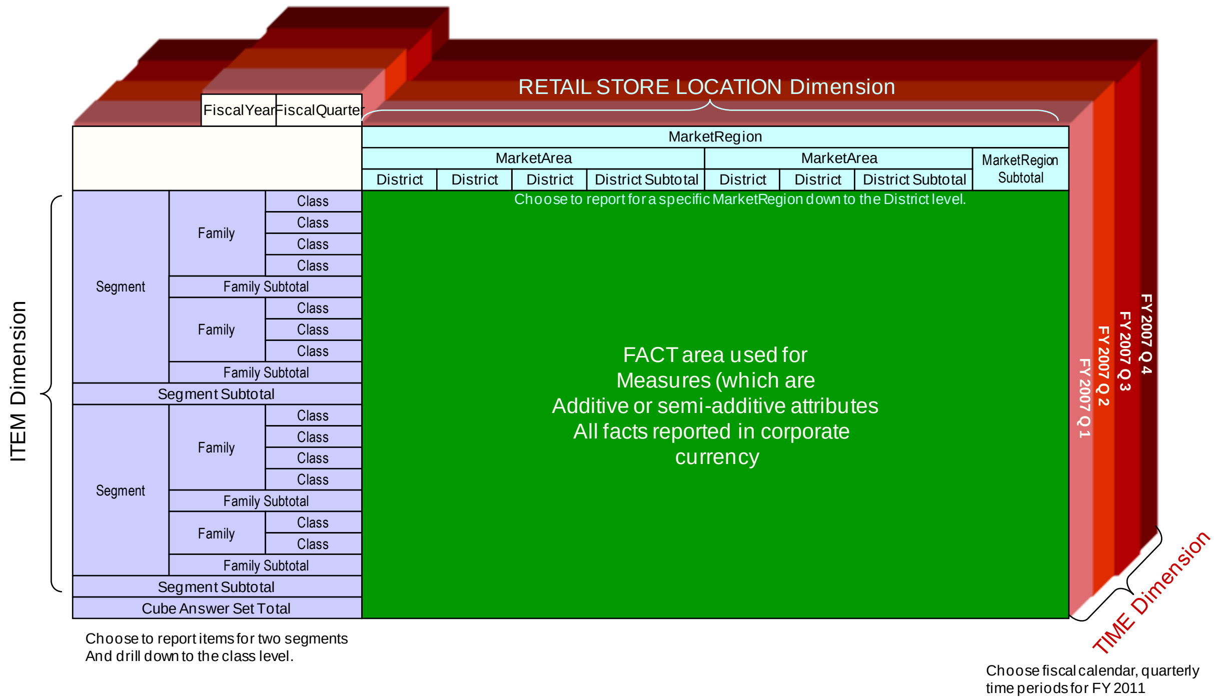
The third part of the answer is to define specific instances of time periods to include in the answer. For example, a high-level view may only want to look at the fourth quarter of a fiscal year. The answer set would omit quarters 1 through 3. So this third part of the answer to “when?” involves defining filters.

The answer set defined through answering these four questions uses three dimensions: item, location and time and one fact table. The term cube comes from this multi-dimensional approach to posing business questions. The process shown here does not include customer. However, customer could be added as another dimension (although it may generate a confusing answer set) or included in a separate aggregation set/cube.

Figure 30 presents one hypothetical cube built from the ARTS Data Warehouse Model. It is one of many possible ways to summarize and report a retailer’s business activity using the model as attributed in Figure 29.

The three dimensions used in this cube include item, retail store location and time. The answer set as specified provides a high-level over view of quarterly net sales revenue and unit sales counts over a 1 year period. As noted earlier, this data structure is similar to what most OLAP tools will create from a data warehouse star schema. This is a very basic example of what can be done

Figure 30: Example of a Data Cube/Stored Aggregation Built on ARTS Data Warehouse Model



The definition of aggregation sets (cubes) is an iterative process. Managers will experiment with different ways of looking at the fact data. Once an answer set is defined that meets the retail decision makers needs, it can be optimized and saved for future use. The primary role of the ARTS Data Warehouse Model is to provide the data structure used by tools and processes to generate answers sets like the one illustrated in Figure 30.

5 ARTS Data Warehouse & Retail Performance Measures

The data warehouse provides the raw factual material used to derive measurements that retailers use to manage their business. The facts represented in the ARTS Data Warehouse Model consist of transaction-level retail sales and discounts, as well as inventory on hand and on order in terms of unit count, retail sales value and item cost value. From these business critical foundational facts a number of key measures can be derived to help retailers improve purchasing, allocation, and operational decisions.

Release 2 of the DWM addresses three retail performance measure categories including:

- Store sales performance measures;
- Merchandising performance measures; and
- Inventory monitoring and control measures.

The DWM supplies the data values used in the formulas for performance measures. The DWM performance measures are organized around answering some basic retail questions. The performance measures addressed in this document are all historical, lagging indicators.

It is important to understand that the formulas presented here use variables that may represent aggregations of detailed facts. The specific criteria for including or omitting detail facts are too complex and varied to specify here. To simplify the presentation, formulas are presented using variables that are tied back to the fact attributes. In practice, the variables will be populated by aggregated values, not singular fact data.

Many of the formulas used to derive performance measures are simple. When combined with different combinations of dimensional attributes, they become quite complex and allow retail decision makers to summarize or drill down by location, organization/responsibility, customer, timer period and item category.

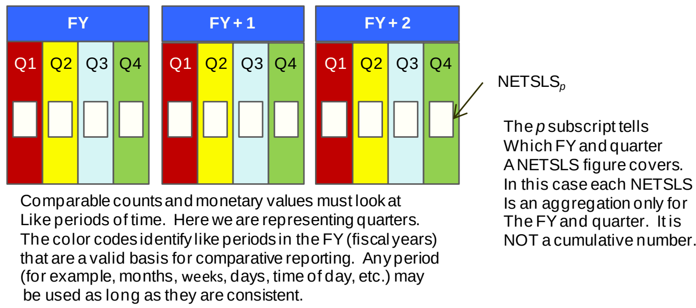
5.1 Basic Selling Measures By Selling Location

5.1.1 Comparable Sales

The comparable sales key performance indicator is a period to period comparison of net sales[2]. The comparable sales key performance indicator is considered to be a measurement of productivity in revenue used to compare sales of retail stores that have been open for a year or more. Historical sales data allows retailers to compare this year's sales in their store to the same period last year. Comparable sales always represent sales for the same relative time period in the years being compared. The time periods compared may be quarters, months, fiscal weeks or even business days as long as they are the same from one year to the next. This concept is illustrated in Figure 31 which uses quarters.

[2] Net sale = Gross Sales - Returns

Figure 31: Comparable Sales By Quarter Illustration



Comparable net sales, as the name suggests, is a measure of periodic change between net sales. The change may be expressed in monetary value, retail unit counts or as a percentage.

1. Calculate net sales for each period as NETSLS(p) = GrossSalesAmount [3]for SALES(p) – GrossSalesAmount for RETURNS(p)
2. Calculate the monetary value of the difference between NETSLS for two like periods as VARIANCE = NETSLS(p+1) – NETSLS(p).
3. Calculate the percentage difference as:

   VARIANCE% = (((NETSLS(p+1) – NETSLS(p))/NETSLS(p))*100

In this example, the GrossSalesAmount (which is coming from the Sale/Return Fact entity) is a summed figure. The rollup dimensions will vary depending on what kind of reporting is required. As noted, there are many different ways to summarize GrossSalesAmt before using it in these formulas.

The same concept reflected in the formulas for comparable net sales can be applied to units which may be called comparable unit net sales. Instead of measuring net sales in monetary terms, retail unit counts are used. Retailers find it useful to look at comparable sales in monetary terms to measure financial performance and retail unit terms to measure operational performance.

Comparable sales are widely used when publically held retailers report results to Wall Street. Comparable sales are considered a key measure of success. A consistent period to period increase in comparable sales indicates that the retailer is growing. A consistent period to period decrease indicates that the retailer is losing ground.

In the ARTS data warehouse model, comparable sales can be broken down and analyzed on a single store basis or a location aggregation level basis. Comparable sales may also be presented by item or some aggregation level of item (using the merchandise classification hierarchy). So from a very elementary collection of facts (i.e. sales) a number of different observations can be made about the

[3] For now we are looking at valuing sales and returns at their gross retail value. The Sale/Return Fact table also provides NetSalesAmount which reflects the subtraction of discounts from the GrossRetailAmount. The naming can be confusing because the net sales figure we are calculating in this example is SALES – RETURNS, not gross retail minus discount. To simplify the explanation we are using the GrossRetailAmount. Returns that reflect the net purchase price paid by a customer are a bit more complex because merchandise prices fluctuate over time as merchandise is put on sale (temporary markdown) or permanently reduce (permanent markdown). So a customer may purchase an item for $5.99 and when they return an item, it may have been marked down to $4.99 requiring the retailer to adjust the retail of the returned item.

retail businesses performance from year to year. The way the comparable sales function arguments are aggregated will depend on the decision makers' frame of reference.

Period to period comparable sales is a deceptively simple calculation. When extended to include different combinations of time frames, selling locations, items (and item hierarchy levels) the simple calculation becomes a powerful way to measure and monitor retail business performance at many levels.

5.1.2 Which stores are leading growth and which stores are lagging?

This question is a variation on comparable sales. Here sales are compared based on selling location, not time period. The purpose is to identify selling locations that are performing well and those that need attention from a retailer's management team.

The central theme of this measure is selling activity by location. Like comparable sales, it can be analyzed using a lot of additional dimensions – too many to cover in detail in this document.

Sales by location are usually net sales which is gross sales minus returns. Sales may be expressed in monetary terms or as retail unit counts. In addition, the gross margin may be reported by location. Gross margin is the retail value of sold items minus their cost. While the other measures reflect selling volume, gross margin expresses comparative profitability (before general and administrative costs).

Sales by location can be compared for one reporting period or over a series of reporting periods as shown in Figure 32

Figure 32: Comparable Location Sales Over Time – Example of Combining Dimensions

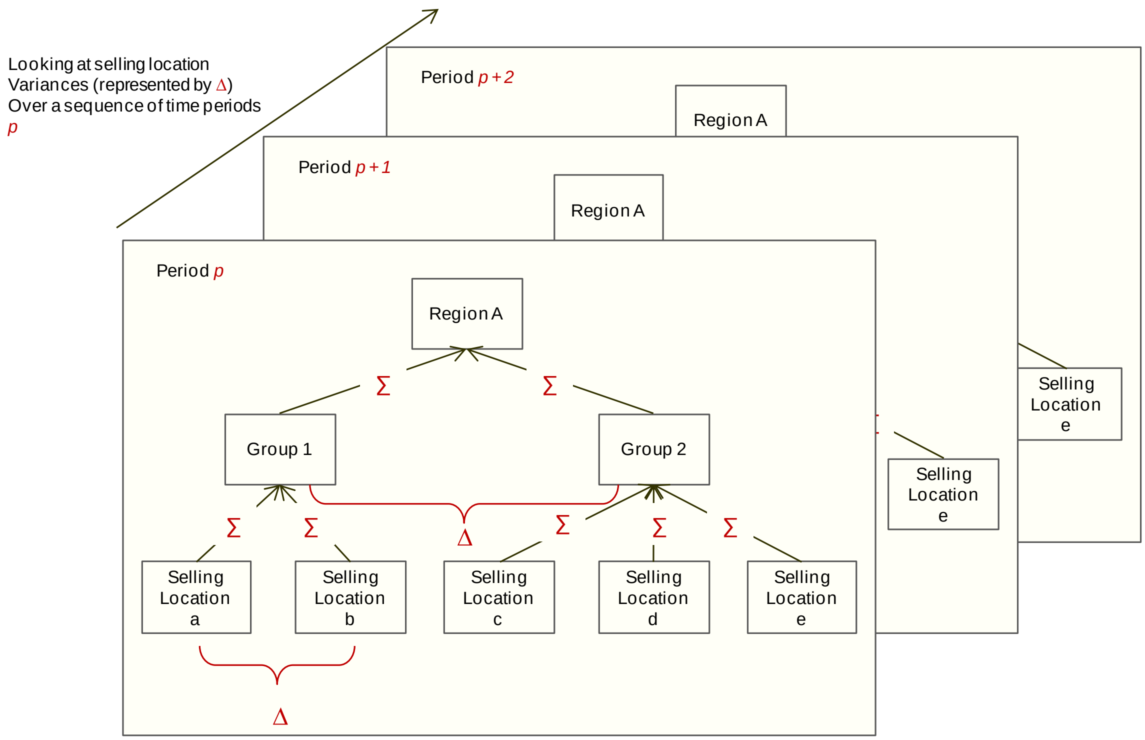


Figure 32 illustrates one way the comparison of sales by selling location (at different aggregation levels) may be combined with the time dimension. This three dimensional view allows retailers to quantify how selling locations perform relative to one another over time. For retailers with stores in a number of different climate zones this becomes very important because sales differences may be a function of the rate at which seasons change. For example snow blower sales in Minnesota are likely to be higher in October than sales in Virginia. This kind of analysis (using very basic measures) helps retailers define selling patterns and with some additional analysis understand why those patterns occur.

5.1.3 Selling Location Utilization

Selling location utilization is another way to compare store performance. It represents sales volume per square meters (or feet) of selling area in a store. Sales volume is typically measured as an aggregate net sales (gross sales less returns after the application of POS discounts) divided by the selling area of a store. Selling area as used here means the area where merchandise is displayed and customers shop. This approach may be modified to include stock room areas as well as selling area if appropriate to a retail business. For this measure sales are typically aggregated to the store (or Business Unit) level. This means summing sales of all inventory locations within a business unit. Sales per Area is calculated as:

Sales per Area = NetSalesAmount / SellingAreaSize

The NetSalesAmount is an aggregation of the NetSalesAmount in the Sale/Return fact entity. The SellingAreaSize is read from the Business Unit Location Dimension entity type. This measure is an indicator of how efficiently the selling area and associated costs are being utilized.

This basic measure can be extended in many ways to look at different item classification levels, individual items, buyer responsibilities, etc. It can also be used with the time dimension to look at selling area utilization over time for seasonal planning purposes.

5.2 Inventory Movement & Velocity Measures

This set of measures is concerned with the rate at which items move through the retail enterprise. The measures may apply to a single store or group of stores. They may apply to a single item or any classification of items. The context is determined by the dimensions used to select data for inclusion in the inventory analysis. In this discussion of inventory measures, we deliberately avoid qualifying measures by item and location dimensions to concentrate on the essential measurement calculations and their association with the ARTS DWM fact entity types.

5.2.1 Inventory Velocity Measurement – Retail Selling Units

Figure 33 illustrates how the Inventory Fact and Sale/Return Fact entity types can be used to support measuring inventory velocity reporting. This illustration uses retail units as its basis. It provides the basis for the cost-based inventory velocity movement discussed later.

Figure 33: UNIT Inventory Velocity View of Inventory Fact Table

	In Flows (from Inventory Fact)				Out Flows (from InventoryFact and Sale/ReturnFact					
ReportingPeriodID	BeginningUnitCount	ReceivedUnitCount	TransferredInUnitCount	ReturnUnitCount	ReturnToVendorUnitCount	TransferOutUnitCount	SalesQuantity	EndingUnitCount	1 Avg Inventory UNITS	2 Inventory Turnover UNITS
1	100	100	10	5	5	5	150	55	77.5	1.9
2	55	100	0	0	5	0	130	20	37.5	3.5
3	20	150	0	0	0	0	130	40	30	4.3
4	40	150	0	2	2	0	125	65	52.5	2.4
5	65	150	0	1	0	0	100	116	90.5	1.1
6	116	100	0	2	0	0	95	123	119.5	0.8
		750	10	10	12	5	730			

Average Inventory (UNITS) for 6 Periods	3	74.14286	units
Inventory Turnover(UNITS) for 6 Periods	4	9.85	times
Periods of invenory onhand in 6th period	5	1.01	periods

The light-green cells headers reflect reporting period data that is taken from the **Inventory Fact** entity and **Sale/Return Fact** entity (either directly or as summed data). There are two calculated attributes by period. The first ❶ is Avg Inventory UNITS. This is the average unit count on hand for the **ReportingPeriodID**. It is calculated as follows:

Avg Inventory Units = (BeginningUnitCount + EndingUnitCount) / 2

The second calculated reporting period attribute is ❷ Inventory Turnover UNITS. Inventory turnover is a count of the number of times the average inventory unit count is sold during the reporting period. It is an important indication of sale velocity – in this instance based on retail units.

Inventory Turnover UNITS = SalesQuantity / Avg Inventory Units

Together, the Avg Inventory Units and Inventory Turnover UNITS tell a retailer how fast an item is moving. This information is in UNITS so it will be different from inventory velocity based on cost. It is a good way to compare item velocity, selling location velocity, period-to-period velocity, etc. Because it is expressed in units, it represents a physical measure of performance.

In addition to intra-period measures it is also useful to derive inter-period information. Figure 33 presents two figures covering the six reporting periods. Here the measures look at a contiguous set of reporting periods. The number of periods can be any number determined by the individual doing the analysis.

The first figure ❸ is the Average Inventory Units for the entire 6 reporting time interval. It is calculated as follows:

Average Inventory Units = (Sum(BeginningUnitCount for the 6 periods) + EndingUniCount for period 6) / (Periods +1)

In this example, the "Periods + 1 figure is 7. This provides an average on-hand inventory for the 6 period interval which is used in the next calculation.

The ❹ Inventory Turnover (UNITS) is a measure of the number of times the 6 period average inventory is sold over the 6 period time interval. It is an important indicator of how well a retailer (or a specific item, location, category, etc.) is performing. A low value may indicate that there is too much inventory. A high number may indicate insufficient inventory. The "right" value depends on the item type, retailer merchandising business rules, market demand and a host of other factors.

The formula for Inventory Turnover (UNITS) is

Inventory Turnover = Sum(SalesQuantity for 6 periods) / (❸Average Inventory Units)

It is possible for inventory on hand balance to go to zero which is a stock-out condition. Inventory on hand balances should not go negative. Instead additional attributes to support back orders should be added. In this model (which as noted earlier is a work in progress) back order attributes are not represented. Future versions (that incorporate ODM support for customer orders) will incorporate backorders.

A useful alternative measure of inventory is weeks supply on-hand. It is a measure of the number of weeks worth of inventory a retailer has to cover average weekly sales over a period of time. In Figure 33 the weeks are replaced by periods for ease of presentation. The ❺ Periods of Inventory On-hand are calculated as follows:

Periods of Inventory On-hand = EndingUnitCount for period 6 / Average (SalesQuantity for periods 1- 6)

The "right" value will depend on supply chain reliability, fluctuations in customer demand, retailer inventory carrying capacity and other factors. The figure illustrated here provides a measure of actual periods on-hand based on historical information derived from the DWM and ODM.

5.2.2 Inventory Velocity Measurement: Cost

This next set of performance measures extends the model shown in Figure 34 by incorporating inventory costs. The basic analytic method is the same, but the inclusion of cost changes the physical frame of reference of retail units to a financial frame of reference. Both perspectives are very critical and complement one another.

The unit costs used to translate the item unit counts in Figure 34Figure 33 are given in the ODM and reflected in the **EndingAverageWeightedCost** for a period. Note that the **EndingAverageWeighted** cost for a period *p* is the **BeginningAverageWeightedUnitCost** for period *p + 1*. The ODM supplies this figure to the DWM as a "given value".

The method of analysis for analyzing inventory velocity based on cost is the same as that used to measure retail UNIT velocity (as illustrated in Figure 33).

Figure 34 translates the analysis of inventory velocity into cost. The light green column headings indicate attributes taken directly from the Inventory Fact entity. The other attributes are simply extended costs (units multiplied by unit cost).

Figure 34: Inventory Cost Velocity Analysis Based on Cost

	In Flows (from Inventory Fact)				Out Flows (from InventoryFact and					
ReportingPeriodID	BeginningCost	CumulativeReceivedCost Amount [1]	1 Transferred In Cost	1 Return Cost	1 Return to Vendor Cost	1 Transfer Out Cost	1 Cost of Goods Sold	2 Ending Cost	3 Average Inventory Cost	4 Inventory Turnover at Cost
1	500.00	500.00	50.00	25.00	25.00	25.00	750.00	275.00	387.50	1.9
2	275.00	500.00	-	-	25.00	-	650.00	100.00	187.50	3.5
3	100.00	720.00	-	-	-	-	638.86	181.14	140.57	4.5
4	181.14	720.00	-	9.76	9.76	-	610.00	291.14	236.14	2.6
5	291.14	720.00	-	4.86	-	-	486.15	529.85	410.50	1.2
6	529.85	500.00	-	9.76	-	-	463.60	576.01	552.93	0.8
		3,660.00	50.00	49.38	59.76	25.00	3,598.61			

Additive Facts

Average Inventory (COST) for 6 periods	5	$ 350.45
Inventory Turnover (COST) for 6 periods		10.27 times 6
Periods of inventory on-hand in 6h period	7	0.96

NOTES 1. CumulativeReceivedCostAmount is the cumulative amount *WITHIN* a period.

The different costs labeled with ❶ are derived by multiplying the unit counts in the same column positions as Figure 33 by the **EndingAverageWeightedCost** read from the **Inventory Fact** entity. The **EndingAverageWeightedCost** value is a calculated value in the *ODM* and is a moving average calculation based on item receipts. The Ending Cost for a **ReportingPeriodID** ❷ is calculated as:

> Ending Cost = (Beginning Cost + CumulativeReceivedCostAmount +Transferred in Cost + Return Cost) – (Return to Vendor Cost + Transfer Out Cost +Cost of Goods Sold)

This calculation is the same used for retail units except the values are expressed as monetary costs. The Average Inventory Cost ❸ for a reporting period is calculated as:

> Average Inventory Cost = (Beginning Cost + Ending Cost) / 2

The Inventory Turnover at Cost ❹ for a reporting period is calculated as:

> Inventory Turnover at Cost = Cost of Goods Sold / Average Inventory Cost

Inventory Turnover is the number of times average inventory is sold for a reporting period. It is the velocity measure that indicates to a retailer how fast merchandise is moving through their inventory. It is a critical measure because inventory turnover (along with gross margin) is the basis for retail profitability. As discussed earlier, these calculations can be applied to different aggregations of items and locations using selection and summarization criteria applied to the dimension entity types in the DWM.

The Average Inventory Cost for 6 Periods ❺ is calculated as:

> Average Inventory Cost for 6 Periods = (Sum (Beginning Cost for 6 periods) + Ending Cost for period 6) / (Periods + 1)

The Inventory Turnover (COST) for 6 Periods ❻ is calculated as:

> Inventory Turnover(COST) for 6 Periods = Sum(Cost of Goods Sold for 6 periods) / Average Inventory Cost for 6 Periods

This figure reflects the number of times the average inventory cost is sold during the six reporting periods. It is an aggregate measure that looks at a block of contiguous reporting periods.

Periods of Inventory on hand in 6th period ❼ measures, in cost, how many periods of inventory are on-hand to meet demand. It is calculated as:

> Periods of Inventory on-hand in 6th period = Ending Cost for period 6 / Average (Cost of Goods Sold for periods 1-6)

This figure may be used to analyze inventory carrying costs over time so that a retailer can examine service level (based on retail units) and cost (based on cost).

The analytic methods presented in Figure 34 can be applied to the retail valuation of inventory. The methodology is similar except for the determination of retail unit price. The derivation of a retail unit price over time is complicated by the application of permanent and point of sale (temporary) markdowns. For decision making about inventory velocity, service levels and similar issues, the unit and cost analysis are more useful to managing inventory levels and cost.

5.2.3 Inventory Service Level Measures

Inventory velocity measures help retailers understand inventory movement and, indirectly service level. This next set of measures provides a more direct measure of service level.

5.2.3.1 Active Inventory Out of Stock Percent by Period

Active inventory out of stock percent measures the portion of a retailer's items that encounter an out of stock condition during a reporting period. Out of stock conditions represent lost sales opportunities and can cause customers to shop at a competitor who has the items they want in stock.

> Active inventory out of stock for a reporting period = Count(ItemID with EndingUnitCount <=0) / Count(ItemID) * 100

This formula makes several assumptions. First all items referenced in the **Inventory Fact** entity are "active" items. Second the stock out condition is signified by a reporting period **EndUnitCount** that is a non-positive number. Third that reporting period refers to a business day. This provides a daily exception snapshot of where stock outs are occurring in a retailer's inventory.

5.2.3.2 Item Out of Stock Period Count

The Item Out of Stock Period Count is a variation of the stock out percentage measure. The Item Out of Stock Period Count tells retailer how many reporting periods (business days) an item out of stock condition persists. This measure is by **ItemID** and by **BusinessUnit** - **InventoryLocation**. The measure may be aggregated to item summary levels and business units as *COUNTS*. This reflects the notion that we are counting out of reporting period inventory states, not item quantities or costs.

> Item Out of Stock Period Count = Count (ReportingPeriodID where Item EndUnitCount <=0) By ItemID, BusinessUnit and InventoryLocation

This measure is a non-additive fact (see section 2.1.4.3 Non-additive Facts). This measure counts the number of days the EndUnitCount for an item at a business unit inventory location is zero or less.

Aggregating this count measure should be done with care. The day count is only useful within the context of a specific item at a specific location. Summing and reporting the information to a business unit group level will hide important exception information. This same "hiding" problem can occur when summarizing individual items to a hierarchy level.

5.2.3.3 Measurements of Inventory Freshness

Inventory freshness measures indicate how recently merchandise has been received and put into salable condition. For fashion items (or any item that is not replenished) this is an important measure of how much of their stock is new. New stock tends to sell at a higher gross margin than older stock. For non-fashion merchandise that is replenished, receiving lead time is important to avoiding stock out conditions.

> Days Since First Received = Julian(Today's Date) – Julian(FirstReceiptDate)
> Days Since Last Received = Julian(Today's Date) – Julian(LastReceiptDate

Today's date is read from the system date. The FirstReceiptData and LastReceiptDate come directly from the Inventory Fact entity. The Julian function is a placeholder to transform the dates so date arithmetic can be performed. Most RDBM's have functions that handle date arithmetic.

Both of these measures are typically supplemented by order and intransit day counts to provide a broader picture of inventory acquisition lead times. These supplemental measures will be addressed in future versions of the DWM

5.3 Gross Margin & Profitability Measures

The measurements discussed up to this point have considered selling volume (in units and monetary terms) and inventory velocity. In this section we will address gross margin and related measures of the difference between selling price and cost of merchandise. Also, the inventory performance measures discussed in and Figure 34 presented unit counts and costs. They did not take into account the retail value of inventory.

All merchandise is assigned a retail price at the time it is purchased. This retail price is the original retail because it is assigned when an item is first brought in for sale. The difference between the original retail price and the cost of an item is its gross margin. Gross margin is basis for retailer profitability. Retailer profitability is managed by designing a business strategy that combines sales volume (units sold) and gross margin (the difference between unit sales price and unit cost).

5.3.1 Gross Margin Return On Investment (GMROI)

Gross margin is the difference between gross sales less returns and the cost of goods sold for a given period. For financial reporting purposes gross margin is reported at the enterprise level. For internal merchandise analysis gross margin can be reported at more detailed levels.

Gross margin when combined with the inventory movement measures discussed earlier introduces a key retail merchandising performance measure – gross margin return on investment. Gross margin return on investment (GMROI) represents a useful combination of gross margin and inventory turns that measures a retailer's merchandising efficiency. The formulas for calculating gross margin return on investment (GMROI) are:

$$GMROI = \frac{\text{Gross Margin}}{\text{Net Sales}} \times \frac{\text{Net Sales}}{\text{Average Inventory at Cost}}$$

$$\text{Gross Margin} = (\text{Gross Sales - Returns}) - \text{Cost of Goods Sold}$$

$$\text{Average Inventory at Cost} = \frac{\text{Beginning Inventory} + \sum \text{Month End InventoryValue}}{13}$$

GMROI is one of several key merchandise performance metrics retailers use to measure profitability. It tells the retailer how many times over a year they get their stock investment returned with a given gross margin. It is a good way to compare and contrast different mixes of gross margin and volume across merchandise categories, retail selling locations, market regions, etc.

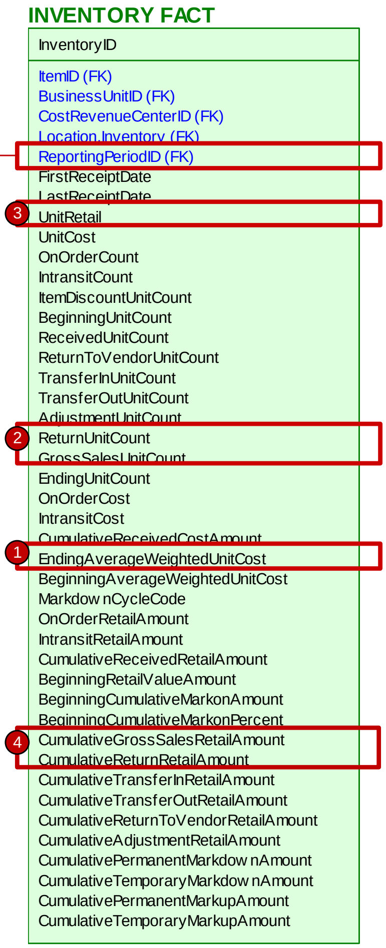
GMROI can be applied at the enterprise level using balance sheet and income statement figures or down to the individual stock keeping unit. Typically retailers use GMROI to analyze performance at some intermediate level (e.g. at the department or class level). The data warehouse structure provides a flexible mechanism to apply GMROI to different levels of merchandise granularity. This allows retailers to compare and contrast merchandise performance at different levels and use GMROI to fine tune their mix of high margin low turnover and low margin high turnover products.

Figure 35 illustrates the calculation of GMROI for 6 periods (months). It builds on the previous example shown in Figure 34. It also shows where the variables used in calculating GMROI originate in the **INVENTORY FACT** entity type.

Figure 35: GMROI Example

GMROI Data Table

Reporting Period	1 Ending Inventory at Cost	2 Net Sale Units	3 Retail Unit Price	4 Net Sales at Retail	5 Gross Margin $	6 Period Beginning Stock To Sales Ratio	7 Gross Margin Percent
0	500.00						
1	275.00	145	$ 8.75	1,268.75	543.75	0.39	42.9%
2	100.00	130	$ 8.75	1,137.50	487.50	0.24	42.9%
3	181.14	130	$ 8.75	1,137.50	498.64	0.09	43.8%
4	291.14	123	$ 7.50	922.50	322.26	0.20	34.9%
5	529.85	99	$ 6.50	643.50	162.21	0.45	25.2%
6	576.01	93	$ 6.25	581.25	127.41	0.91	21.9%
Total	2,453.15	720		5,691.00	2,141.77		
8 Average Inventory at Cost	$ 350.45						
9 GMROI	6.11						



The 6 period time frame used in Figure 34: Inventory Cost Velocity Analysis Based on Cost is used in this example. This example reflects an Item-level analysis. Ending inventory at cost 1 is taken from the **EndingAverageWeightedUnitCost** attribute of the **INVENTORY FACT** entity type. Net Unit Sales 2 is the difference between the **GrossSalesUnitCount** and the **ReturnUnitCount** both from the **INVENTORY FACT** entity type. Retail Unit Price 3 is copied from the **UnitRetail** attribute of the **INVENTORY FACT** entity type. The Net Sales at Retail 4 is the difference between the **CumulativeGrossSalesRetailAmount** and the **CumulativeReturnRetailAmount** for the period both provided in the **INVENTORY FACT** entity type.

Gross Margin 5 is the difference between Net Sales at Retail and the Cost of Goods Sold calculated in Figure 34. Period Beginning Stock to Sales Ratio 6 is calculated by dividing Ending Inventory at Cost by the prior period Net Sales at Retail. The Sales to Stock Ratio tells how much stock a retailer has on hand to support period sales for a given period. Gross Margin 7 percent is the period's Gross Margin divided by Net Sales at Retail. The Average Inventory at Cost 8 is the sum of each period's Ending Inventory at Cost plus the beginning period inventory (Reporting Period 0) divided by the period count plus 1 (in our example 7 periods). The GMROI 9 is calculated by dividing the Gross Margin sum for the 6 periods by Average Inventory at Cost. The GMROI value of 6.11 indicates that this item is a fast moving (which is good) but at risk for stock outs (which is bad). GMROI will vary by retailer as well as from item to item. GMROI and all of the performance measures discussed here need to be calibrated to the retailer's business strategy to be used as key performance indicators.

5.3.2 Retail Price Management

Retail pricing measures identify, define and classify the differences between the price customers pay at the point of sale and the cost of goods sold.

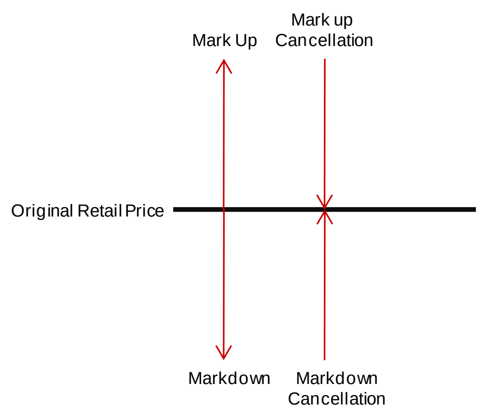
Pricing decisions are complicated and vary by retailer. Pricing strategy is driven by a number of factors including:

- Supply and demand for a product;
- Merchandise type (fad vs. fashion vs. staple)
- Retailer assortment strategy (breadth vs. depth)
- Brand pricing power;
- Merchandise age;
- Vendor sponsored promotions;
- Seasonality of merchandise;
- Competition;

As already discussed, retail profitability is a function of the quantity of merchandise sold and gross margin. In this section we will discuss some basic profitability measures that address retail price and cost from a retail accounting method perspective.

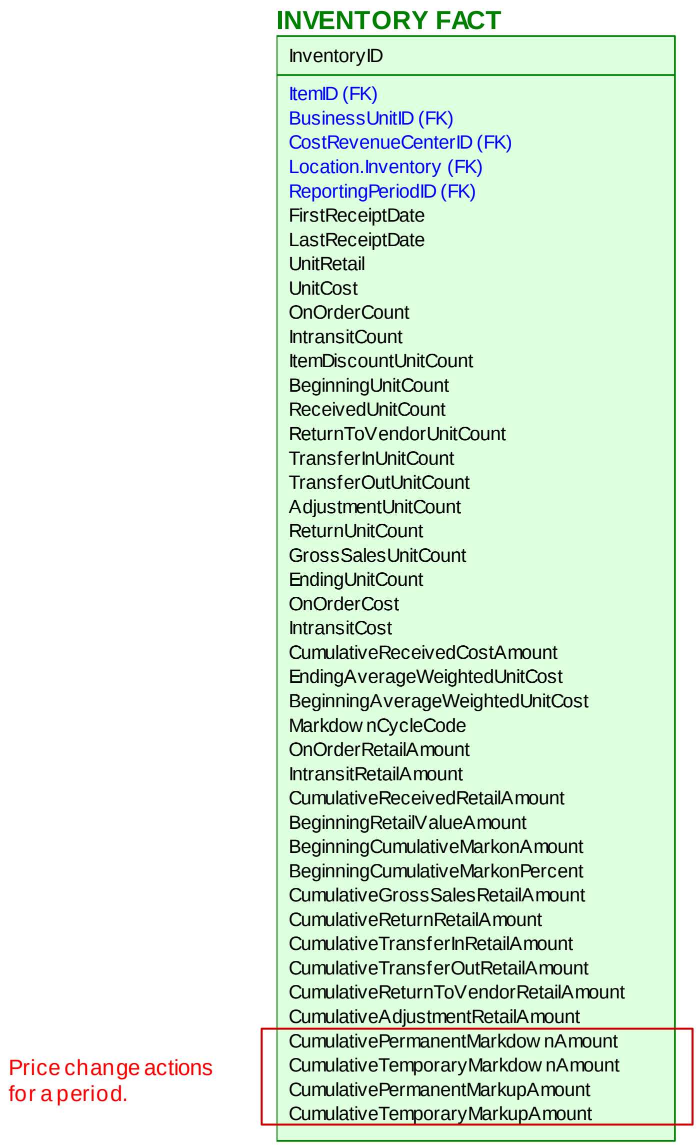
Retailers adjust retail prices through four price change actions as illustrated in Figure 36. All price changes are based from the original retail price.

Figure 36: Price Change Actions



In practice retailers rarely use mark ups except to correct an erroneous original retail price or to provide compare at pricing. Like mark ups, mark up cancellations are rarely used except to correct mistakes or to reduce compare at pricing to the original price. Price changes may be taken permanently which causes a revaluation of inventory or at the point of sale which realizes the changed value only for items sold. Price changes taken at the point of sale are temporary in nature – that is they have a start and ending time. In most cases, a temporary point of sale price change is used to promote the sale of merchandise. In the retail method of accounting, mark ups, mark downs (and their cancellations) have to be tracked. The ITEM FACT table has attributes for temporary and permanent mark ups and markdowns.

Figure 37: ITEM FACT Temporary and Permanent Markup and Markdown Attributes



Markdowns, because they change retail price directly impact a retailer's gross margin. Also, they need to be reflected in the derivation of the cost of goods sold for retailers using the retail method of accounting.

5.3.2.1 Gross Margin Percent

Gross Margin is the difference between the retail price and cost of an item (or aggregation of items) at a particular point in time. Gross margin percent is the portion of total retail price over the cost multiplied by 100.

5.3.2.2 Initial Markup

The initial markup for an item or aggregation of items it the excess of the original retail value for the merchandise over the related cost for that merchandise. The initial markup percent is the excess divided by retail price.

For example Item A has an initial retail of $49.99. It costs $25. The initial markup is $24.99. The initial markup percent is 99%.

The initial markup for an item is set when the item is first created and assigned a retail price.

5.3.2.3 Markup

Markup is the difference between the original retail plus any mark ups and the cost of merchandise. Markup may be applied to individual items or to classes of items.

An item with an initial retail of $49.99, a subsequent mark up of $3.00 and a cost of $25 will have a markup of $27.99. Unlike the initial markup, markup value may vary over time.

5.3.2.4 Cumulative Markup

Cumulative markup for a period represents purchases at retail plus period beginning inventory plus mark ups minus purchases at cost and beginning inventory at cost. Cumulative markup is illustrated as follows:

Table 2: Cumulative Markup Example

Cumulative Markon Example	
Purchases At Retail	$ 350.00
Beginnning Inventory at Retail	225.00
Markups (during period)	10.00
Purchases at Cost	200.00
Beginning Inventory at Cost	100.00
Cumulative Markup	$ 285.00
Cumulative Markup Pct	49%

5.3.2.5 Cumulative Markup Percentage

Cumulative Markup percentage for a period is the cumulative markup value divded by the sum of purchases at retail for the period plus the beginning inventory at retail plus mark ups. The cumulative markup percent is multiplied by the period ending inventory value at retail to derive the estimated cost of inventory for the retail method of accounting.

5.3.2.6 Net Markup

Net markup is cumulative markup less markdowns. Net markup is similar to gross margin, but more limited because it does not reflect purchase discounts and other cost adjustments.

5.3.2.7 Maintained Markup

Over the course of a longer period of time, merchandise prices are changed. Markdowns, sales discounts and shrinkage all result in changes to the retail valuation of retail inventory between the assignment of the initial markup and the last reporting period. The original retail price is not expected to be the "lifetime" value of merchandise. It is however, management's educated guess at the margin required to pay for inventory and operations and generate a reasonable profit. Maintained markup for a period is calculated using the following formula.

$$\text{Maintained Markup} = (\text{Original Retail} - \text{Markdowns}) - \text{Cost of Goods Sold}$$

5.3.2.8 Point of Sale Markdowns

Point of sale markdowns represent temporary or promotional reductions in retail price (retail valuation of inventory) when an item is sold. Only items sold during the promotional period are marked down. Point of sale markdowns are an important tool to help manage the balance between item movement and gross margin.

5.3.2.9 Inventory Shrink

Inventory shrink is the difference between the inventory on-hand value shown in the retailer's stock record and the actual inventory value based on a physical count. Inventory shrink and stock overages (excess inventory) may be caused by bookkeeping errors. Inventory shrink may also be caused by employee theft, shoplifting, damage to merchandise, etc. Shrinkage is not reflected in markup, cumulative markup, or the cost multiplier used to derive inventory cost in the retail method.

5.3.2.10 Retail Method for Inventory Accounting

Figure 38 provides a basic model of how the retail method of accounting works.

Figure 38: Conceptual Model of Retail Method of Accounting

Department or Class Level Inventory Valuation Using Retail Method			
Line	Item	Retail (RTL)	Cost (CST)
1	Beginning Inventory	$ 39,000	$ 22,000
2	Purchases	60,000	35,000
3	Total Inventory Available for Sale (line 1 RTL + line 2 RTL)	99,000	57,000
4	Markups	1,500	-
5	Markup Cancellations	500	-
6	Net Mark up (line 4 RTL - line 5 RTL)	1,000	-
1 7	Total Inventory Available for Sale with Net Markup (line 3 RTL + line 6 RTL)	100,000	57,000
2 8	Cost Complement (line 3 CST ÷ line 7 RTL)		0.570
9	Markdowns	2,400	-
3 10	Markdown Cancellations	400	-
11	Net Markdowns (line 9 RTL - line 10 RTL)	2,000	-
4 12	Sales	88,000	
13	Ending Inventory (line 7 RTL - line 11 RTL -line 12 RTL)	$ 10,000	$ 5,700
14	Ending Inventory Cost (line 12 RTL * line 8 Cost Complement)		

The retail method captures actual inventory cost of a period beginning inventory plus purchases for the period. It also captures the original retail value of beginning inventory and purchases. Next it adds markups, subtracts markup cancellations and arrives at a total inventory available for sale ❶. The Cost Complement ❷ is the calculated by dividing the Cost (CST) Total inventory Available for Sale divided by the Retail (RTL) Total Inventory Available for Sale. The Cost Complement is the percentage of the retail inventory value used to derive the ending inventory cost.

Markdowns and Markdown Cancellations are netted (Net Markdowns) and subtracted from the retail value of the inventory ❸ . Because they change the retail valuation, but not the original purchase value at cost, markdowns and markdown cancellations are not used in calculating the Cost Complement. Markdowns and markdown cancellations occur after merchandise has been purchased so the markup for calculating the cost complement is locked in before markdown actions are taken.

Sales ❹ reflect the major consumption of inventory. The Sales plus Net Markdowns are subtracted from the Total Inventory Available for Sale With Net Markup to yield the Ending Inventory retail value. The Ending Inventory at retail (RTL) is multiplied by the Cost Complement to arrive at the cost of the ending inventory.

Figure 39 expands the example presented in Figure 38 by adding provisions for shrink and sales discounts. Also, it illustrates how cumulative mark on and aggregate gross margin for a department, class or other aggregation of items is calculated.

Figure 39: More Detailed Retail Method Inventory

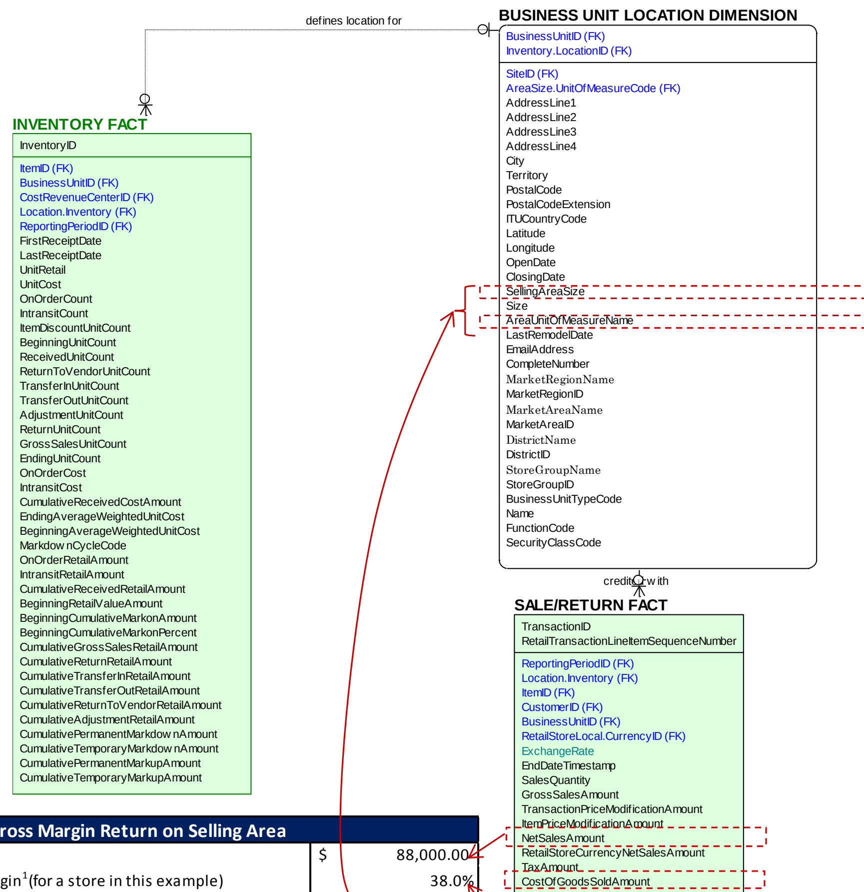
Department or Class Level Inventory Valuation Using Retail Method			
Line	**Item**	**Retail (RTL)**	**Cost (CST)**
1	Beginning Inventory	$ 39,000	$ 22,000
2	Purchases	60,000	35,000
3	Total Inventory Available for Sale (line 1 RTL + line 2 RTL)	99,000	57,000
4	Markups	1,500	-
5	Markup Cancellations	500	-
6	Net Mark up (line 4 RTL - line 5 RTL)	1,000	-
7	Total Inventory Available for Sale with Net Markup (line 3 RTL + line 6 RTL)	100,000	57,000
8	Cost Complement (line 3 CST ÷ line 7 RTL)		0.570
9	Markdowns	2,400	-
10	Markdown Cancellations	400	-
11	Net Markdowns (line 9 RTL - line 10 RTL)	2,000	-
12	Net Sales	88,000	
13	Sales Discounts	2,200	
14	Shrinkage (estimated or actual)	4,500	
15	Total Inventory Consumption	96,700	
16	Ending Inventory (line 7 RTL - line 15 RTL)	$ 3,300	$ 1,881
17	Ending Inventory Cost (line 16 RTL * line 8 CST)		
Markup Calculations			
18	Cumulative Markup (line 7 RTL - line 7 CST)	43,000	
19	Cumulative Markup Percent (line 18 / line 7 RTL)	43%	
20	Maintained Markup (line 7 RTL - line 11 RTL - line 7 CST)	41,000	
21	Maintained Markup Percent (line 20 / line 7 RTL)	41%	
Retail Method Aggregate Gross Margin			
22	Cumulative Markup Percent (line 18)		43.0%
23	Markdown Percent (line 11 / line 7)	2.0%	
24	Sales Discount Percent (line 13 / line 7)	2.2%	
25	Shrinkage Percent (line 14 / line 7)	4.5%	
26	Sum of Retail Reductions (line 23 + line 24 + line 25)	8.7%	
27	Retail Reduction Summary reduced to cost (line 26 * line 8 CST)		5.0%
28	Aggregate Gross Margin for Department or Class (line 22 - line 27)		38.0%

The measurements discussed to this point take into account item movement (both in terms of purchases and sales) item retail value and item cost. There are other measures that apply to how well a retailer is using their selling space and labor.

5.3.2.11 GMROS - Gross Margin Return on Selling Area

Store-based retailers have a large fixed cost in owned and leased sales space. One measure of retail business efficiency looks at gross margin return based on square footage or square meters of selling area. Figure 40 presents a sample model.

Figure 40: Sample Gross Margin Return on Selling Area Model



Gross Margin Return on Selling Area			
1	Assumed Net Sales	$	88,000.00
2	Aggregrate Gross Margin[1] (for a store in this example)		38.0%
3	Margin Value (in monetary units - Line 1 x Line 2)	$	33,476.08
4	Store selling area (square meters)		4500
5	GM per square meter	$	7.44

1 The Aggregate Gross Margin, as shown in Figure 38 reflects markdowns, sales discounts and shrink. This is different from a single item gross margin which is current selling price minus the cost of the item.

$$\text{Aggregate Gross Margin} = 1 - \frac{\sum \text{CostOfGoodsSold}}{\sum \text{NetSalesAmount}} \text{ for a store}$$

The data used to derive the GMROS comes from the SALE/RETURN FACT entity and the BUSINESS UNIT LOCATION DIMENSION entity type. The aggregate gross margin value reflects the cumulative markup as well as markdowns, shrink and discounts (retail reductions). The GMROS is a very useful way to evaluate different retail locations operating efficiency for a period. Its key advantage is that it associates gross margin to selling area which is a major fixed cost for retailers. Because it is looking at gross margin and not sales alone it is a more useful way to compare and contrast store performance.

5.3.2.12 Assortment Analysis and Evaluation

All of the performance measures up to now have considered merchandise as an undifferentiated collection of items. Retailers sell many different kinds of items. The mixture of different items is one of the key ways a retailer has to manage sales volume and pricing.

Assortment management is the art of blending different items (and varieties of the same items), and quantities stocked by each retail selling location (i.e., store and location inside a store) in a way that maximizes the profitability of the selling location. Assortments vary by selling location, season (time of year) and merchandise category.

5.3.2.12.1 Buying Assortment

There are two kinds of assortments represented in the ARTS ODS and DW models. The first kind is a *buying assortment*. This is a defined set of items that:

- Share common characteristics like style, brand, model number;
- Come in a variety of color, size, feature choices;
- Each have a unique SKU (andr EAN/UPC); and
- Are ordered as predefined, pre-counted item collections (which are called assortments).

The buying assortment is explicitly represented in the **ITEM DIMENSION** entity type's **AssortmentFlag** and **AssortmentDescription** attributes. This is how merchandise is ordered and is generally determined by the item manufacturer.

5.3.2.12.2 Presentation/Selling Assortment

The second kind of assortment is the presentation/selling (selling) assortment. This is the way a retailer chooses to organize individual items into collections of individual SKUs with common characteristics as well a color, size, features. The selling assortment may be identical to the buying assortment. Selling assortments:

- Share common characteristics like style, brand, model number and/or merchandise classification;
- Come in a variety of color, size, feature choices;
- Each have a unique SKU (and/or EAN/UPC);
- Are presented to customers as a collection of items from which they may choose one or more that meet their needs

Selling assortments are defined by the retailer. The range of items and depth of color, size and features within each range of items is determined by the retailer. The common characteristics and theme of the item range is determined by the retailer.

Selling assortments are identified using the retailer's merchandise hierarchies. Merchants try to create assortments that maximize the aggregate gross margin of a collection of items. It is a complex process that has to consider:

- Store location;

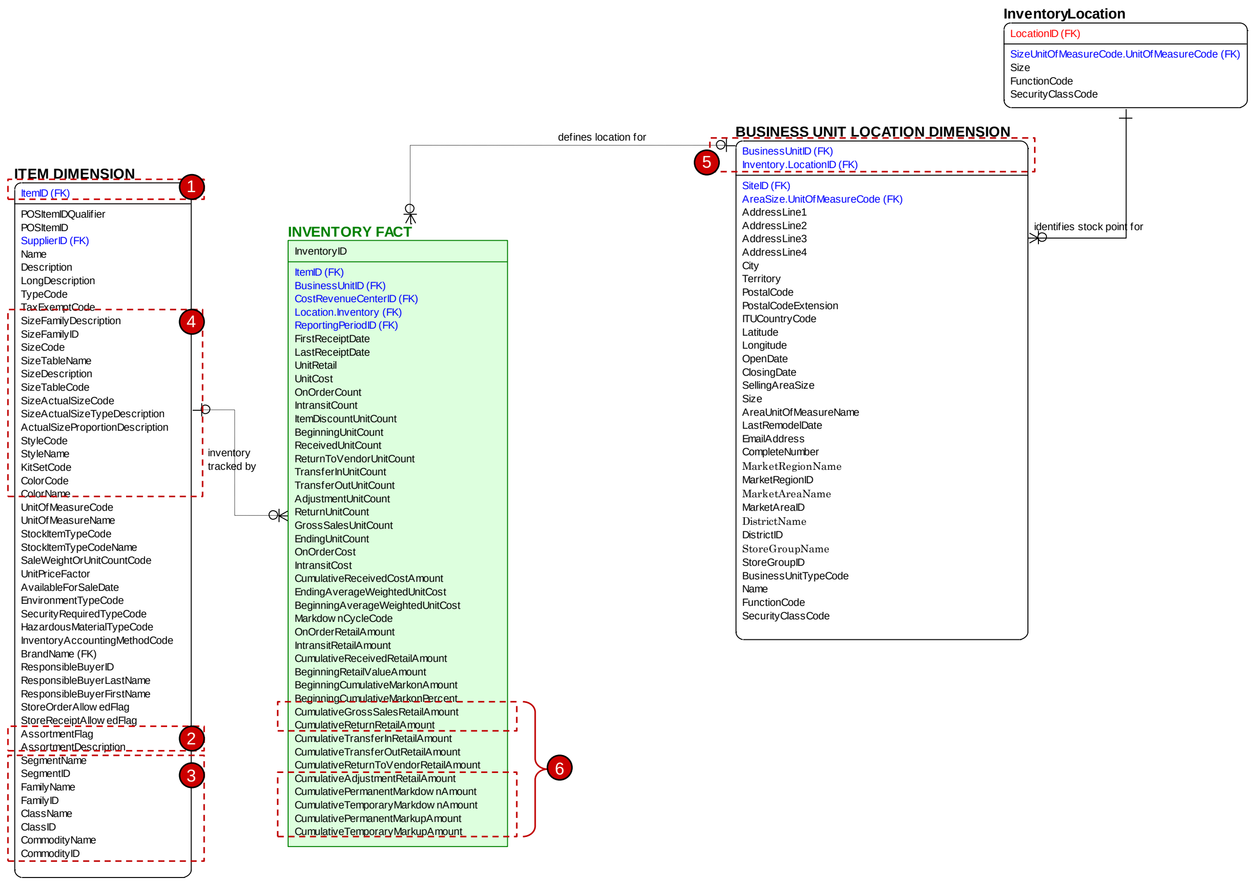
- Local weather and climate;
- Selling season;
- Local shopper demographics and preferences;
- Competition's approach to assortment presentation;
- Interaction between different SKU's within an assortment collection and interaction between assortment collections;
- Relative inventory movement and GMROI of individual items within an assortment;
- Supplier performance (reliability, lead time, service levels, etc.); and
- Retailer's overall value proposition to its customers – the offer it makes to its customers that differentiate it from the competition.

The ARTS DW model provides an open ended set of entities, relationships and attributes for defining selling assortments using one or more of the retailer defined hierarchies that can be defined in the ARTS ODS. The sample DW model provides a single hierarchy linked to each ITEM FACT to simplify model presentation.

Figure 41 presents a subset of entities and attributes from the DW mode and highlights the attributes that are relevant to the identification of assortments.

In terms of **INVENTORY FACT** (the sales, costs and other figures being analyzed), the information is aggregated into the merchandise assortment classifications. Typically, merchants will look at the overall sales behavior (and GMROI) of higher-level classes of items and drill down to look at lower levels that represent leaders (high sales, high GMROI) and laggards (low sales, low GMROI). For data warehouse purposes, it is important to provide the attributes that enable retailers to decide what item characteristics to use to lump individual SKUs into assortment groups and how to identify and track those groups.

Figure 41: ARTS Data Warehouse Model Support for Assortments



Assortment planning deals with identifying and classifying items for buying and selling operations. It begins with individual **ItemID** 1 and **ITEM DIMENSION** entity type instances. Each ITEM DIMENSION entity type has an explicitly assigned (though nullable) **AssortmentFlag** and **AssortmentDescription** 2. Both of these attributes reference a buying assortment.

The merchandise classification attributes 3 serve to identify retailer selling assortments. These assortments can occur as a layered structure. In practice, most assortment planning occurs at the classification level just above Item. That is because assortment planning is about deciding what range of item variations will be offered within that classification. In the **ITEM DIMENSION** entity type these variations are described by the attributes reference by 4.

Assortment planning is integrally tied to where merchandise is sold both in terms of store level and in terms of location within a store. This "where" dimension is represented by the **BusinessUnitID** and **Inventory.LocationID** attributes of the **BUSINESS UNIT LOCATION DIMENSION** entity type 5.

The relevant counts and monetary values for sales, gross margin, etc. are carried in the INVENTORY FACT entity 6. The assortment attributes discussed here can be used in conjunction with the GMROI and other performance measures discussed earlier to develop a comparative analysis of assortments and items in an assortment.

There are a number of software tools to help retailers analyze and optimize assortment plans. They use sales history data to find how different SKU selling volumes interact, customer-item-assortment interaction, item movement variations, etc. The algorithms used by these tools are beyond the immediate scope of this document.

The ARTS Data Warehouse Model is still very much in its early stages of development. There is still a wealth of information that retailers can, with the appropriate analytic methods and tools, extract and use to support decision making. The model in its current form provides basic sales and inventory facts and a basic set of conformant dimensions to establish a business context for those facts.

6 Future Direction for ARTS Data Warehouse Model

The ARTS Data Warehouse model will continue to evolve. Up to this point the primary focus has been on developing the dimensional model. With this second release, the Data Warehouse team has added a more comprehensive discussion of performance measures and how they can be derived from the underlying data mode. This discussion is just scratching the surface. There is a lot more both in terms of data warehouse content and structure and performance measures to be covered.

Future directions that might be considered include:

- Customer analytics related to buying behavior and responses to promotions;
- Vendor performance analysis and reporting;
- Deeper coverage of the items sales, gross margin and movement performance measures introduced in this discussion;
- Retail store and distribution center labor utilization and efficiency;
- Advertising and promotion effectiveness and analysis (likely tied closely to customer analytics); and
- Other ideas and suggestions from the ARTS members.

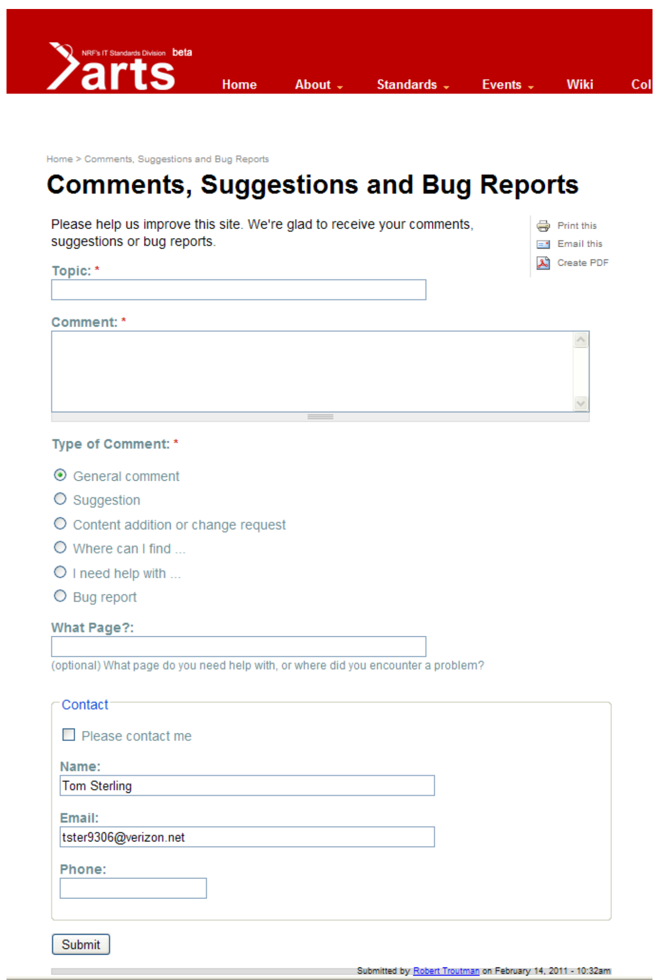
7 Request for Feedback

The ARTS Data Warehouse Model is a work in progress. The ultimate goal is to create an integrated retail enterprise data warehouse model. Our approach is to deliver a series of data marts organized around business subjects determined by the ARTS Board and integrate them through:

- Using conformant dimensions; and
- Rigorous mapping back to the ARTS ODM.

As retailers and retail development teams work through this model, they will identify opportunities to improve the data model. Changes should be submitted by accessing the ARTS Web Site at www.nrf-arts.org, clicking on the “DATA model” button which will take you to the Data Model page. On the Data Model page click on Questions/Comments in the menu at the top of the form. The form can be used to submit questions and comments about the data warehouse model.

Figure 42: ARTS Data Model Questions/Comments Form



The ARTS Data Model Committee is actively seeking feedback from retailers and software developers using the data warehouse model. That feedback is critical to improving the model.