

Universal Data Models for Financial Services

• By Len Silverston •

People often ask me for a universal data model for banking, credit card processing, securities brokerage or other specific aspects of financial services. I reply that there is a universal data model for financial services (see *The Data Model Resource Book, Volume 2*, Wiley, 2001) that provides detailed,

generic data models suitable for all types of financial service enterprises including banks, securities companies, credit unions, brokerage services, credit card companies and many other types of financial services enterprises. Should there be separate data models for each of these specific types of businesses or should there be common data constructs that handle the information needs for any of these types of finance services?

One reason for providing a common financial services model is that enterprises often provide (or may provide in the future) many of these various financial services. Another reason is that various types of financial service organizations usually have very common information needs. In each of these types of businesses, individuals or organizations make financial transactions against their financial accounts. Accounts are governed by both the agreements that are established for the account as well as financial regulations. Financial service products often have features such as the interest rate and functional settings such as when to send statements. This article explores the issue of having specific data models such as a banking or securities brokerage model versus having a generic financial services model. As this article summarizes and expands upon *The Data Model Resource Book, Volume 2*, please refer to this book for a great

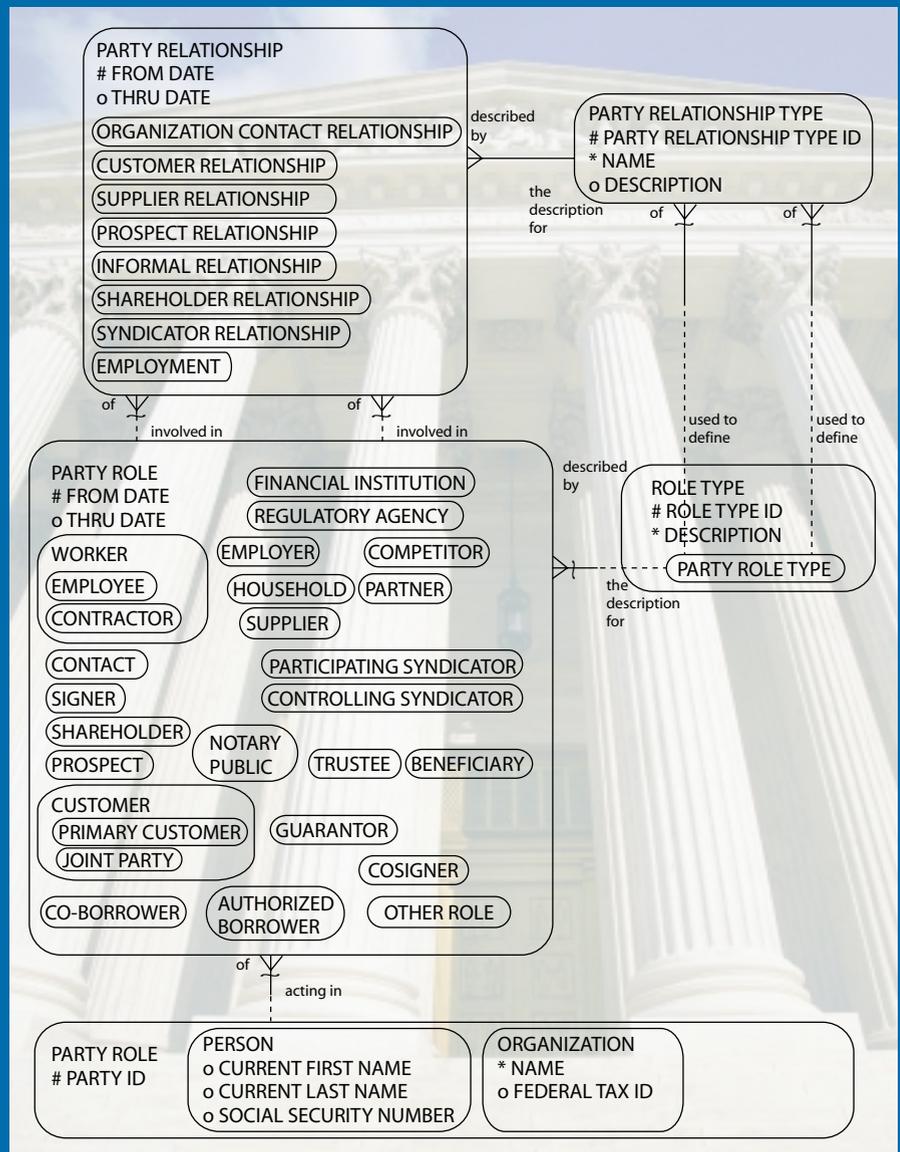


Figure 1: Financial Services Party Roles and Relationships

many more details and explanations behind the models in this article, plus examples of the types of data that would populate these models.

The models provided in this article (and in related universal data model publications) are offered to provide toolkits of common data structures that have been proven to work in real-life enterprises. However, they are not intended to represent the only “right” answer in modeling this type of data. Certainly, there are many options with pros and cons; and the experienced data modeler knows to base the data model on the requirements of the enterprise, using tools such as universal data models when they are applicable.

This article will principally focus on four key data areas within financial services: parties, accounts, agreements and financial products.

Financial Services Parties

Figure 1 provides a party model that can be used for financial service enterprises. As in most other enterprises, there are PERSONS and ORGANIZATIONS, or various PARTIES that may take part in various PARTY ROLES and may be involved in various PARTY RELATIONSHIPS.

The PARTY entity records information about the PERSON or ORGANIZATION independent of the PERSON or ORGANIZATION role or relationship such as name, demographics, SSN or federal tax ID. The PARTY ROLE subtypes would be used to relate information about the party in the context of role, such as the credit rating(s) of a CUSTOMER. Thus, certain types of information are only related to a specific role of a party and not for all parties. Therefore, it is useful and usually practical to maintain information about specific party sub-entities. The PARTY RELATIONSHIP entity maintains information about two parties such as the status or priority of the relationship.

Each PARTY may be acting in many PARTY ROLES, thus identifying the various ways in which people and organizations may interact with the enterprise. Some may argue that

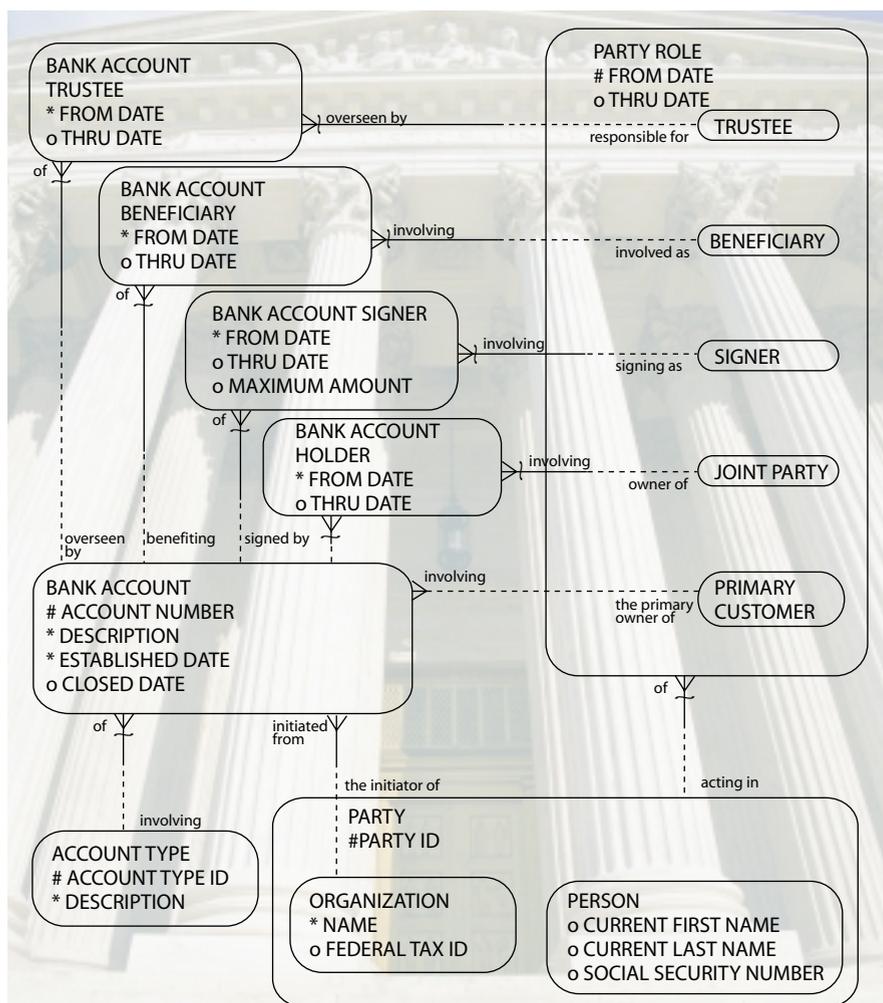


Figure 2: Specific Bank Account Roles Data Model

these party roles are subtypes of a party. Thus, a CUSTOMER could be a subtype of party. However, how can we then show that a party can play many roles? For instance, a party may be acting as a CUSTOMER, a SHAREHOLDER and an EMPLOYEE at the same time. These would then need to be inclusive subtypes of party. Thus, there could be many inclusive subtypes, making the model quite complex. Some data modeling conventions such as Oracle Designer’s subtyping notation of boxes within boxes doesn’t even allow inclusive subtyping, and subtypes are defined as mutually exclusive.

Others may argue that the party role only makes sense within the context of a relationship. Thus, if an organization is a customer, with whom do they have the customer relationship? There may be two or more customer relationships such as a relation-

ship between the customer and one bank and another relationship that the customer has with another bank. However, regardless of the number of relationships involved, there is still information about the customer (for instance, the customer credit rating) that may be related to the customer (assuming the enterprise only stores the credit rating for customers) and should not be recorded redundantly for each relationship. Additionally, some party roles may exist without a corresponding party relationship; for example, a person may play the role of NOTARY PUBLIC, thus further establishing the need for a PARTY ROLE independent of the relationships involved.

The ROLE TYPE accommodates any number of various roles, allowing additional roles to be added over time. The subtypes within PARTY ROLE are shown to illustrate the various

types of roles that may exist for a party, and both the logical data model and physical design should be customized depending upon the individual needs of the enterprise. Alternate ways of modeling this information are to either just include the PARTY ROLE TYPE or just to include the PARTY ROLE subtypes; however, both are shown in Figure 1 for illustrative purposes. The PARTY RELATIONSHIP TYPE allows categorization of the various types of relationships that may exist. The relationship from the PARTY RELATIONSHIP TYPE to the PARTY ROLE TYPE provides a mechanism to maintain business rules about what types of roles may exist for what types of relationships. For example, it may define that an “employment relationship” is a relationship between an “employee” role and an “employer” role.

Note that many of the PARTY

ROLE subtypes shown apply to many types of financial service organizations. For instance, banks, credit unions, security brokerage companies and credit card companies all typically have the PARTY ROLES of PRIMARY CUSTOMER, JOINT OWNER, FINANCIAL INSTITUTION, REGULATORY AGENCY, SIGNER, GUARANTOR, TRUSTEE and NOTARY PUBLIC as well as many of the very generic roles such as EMPLOYEE, CONTRACTOR, CONTACT, SHAREHOLDER and SUPPLIER.

Accounts

Financial services enterprises such as banks, credit card companies and securities organizations set up accounts for their customers in order to track and group financial transactions according to the customers’ wishes.

Figure 2 shows a portion of a bank

account data model with some of the specific roles associated with bank accounts. A BANK ACCOUNT must be of an ACCOUNT TYPE such as a checking, savings or money market account. A BANK ACCOUNT may be overseen by one or more BANK ACCOUNT TRUSTEES, be set up for the benefit of one or more BANK ACCOUNT BENEFICIARIES, have one or more BANK ACCOUNT SIGNERS, have one or more BANK ACCOUNT HOLDERS and, depending on the business rules of the enterprise, have one and only one PRIMARY CUSTOMER. Of course, the last relationship mentioned will vary based on the enterprise; however, many financial institutions require that one party be the primary customer and need the social security number for that party for income tax reporting purposes.

Another relationship to BANK ACCOUNT is the ORGANIZATION that initiated the account. One may alternately form a relationship from BANK ACCOUNT to the BRANCH that initiated the transaction (along with the routing number for that branch), which could be shown as either a subtype of ORGANIZATION or a subtype of the PARTY ROLE. Is BRANCH a PARTY ROLE subtype or is it a subtype of ORGANIZATION? Is it possible that the organization that represents a branch could later play additional roles such as a division, affiliate or department of the enterprise? Again, depending on the business rules and needs of the enterprise, one may want to consider alternate options.

Figure 3 shows a similar model to Figure 2 only with the appropriate roles involved in loan accounts: namely, COSIGNERS (people authorized to sign transactions on the account), AUTHORIZED BORROWERS (people authorized to borrow against the account), GUARANTORS (parties that guarantee payment is case of default), CO-BORROWERS (the various parties that are responsible for repayment) and the PRIMARY CUSTOMER (the primary party responsible). The roles involved and their relationships to ACCOUNT (as well as

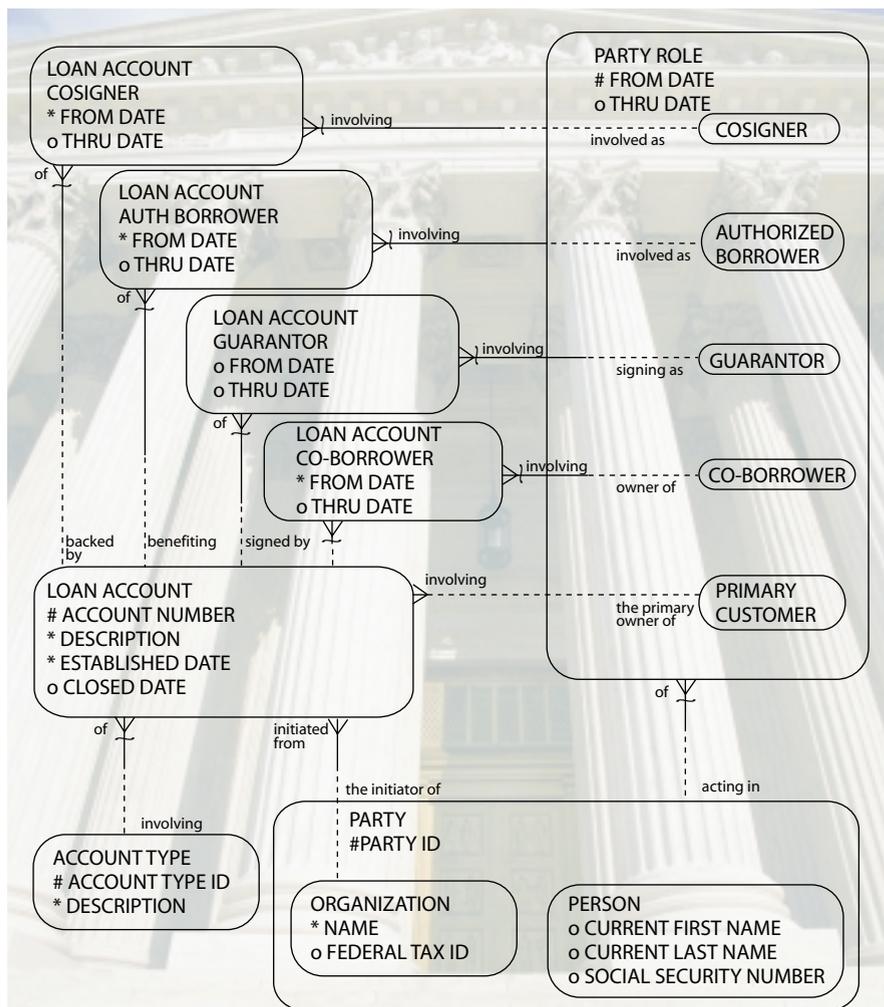


Figure 3: Specific Loan Roles Data Model

business terminology) may vary depending on the business rules and needs of the enterprise.

Figure 4 again shows a similar model only with the roles associated with a brokerage account. While these roles are very similar to BANK ACCOUNT roles, most enterprises maintain separate accounts for securities transactions. However there may be a current or future opportunity to manage both bank account and securities transactions together.

Specific Modeling or Generic Modeling?

Figure 5 provides a more generic account model that meets the needs of a wide variety of financial service enterprises including savings and loan accounts, brokerage accounts, line-of-credit accounts and many other types of accounts.

Whether a data model should be designed more specifically or more generically depends on a number of factors such as the modeling style used, the requirements of the enterprise and the intent of the data model. For instance, is the purpose of a data model to document the data-oriented business rules of an organization or to provide a flexible generic structure to maintain information about the enterprise? If the data model's purpose is to document the business rules and requirements of the enterprise, Figures 2, 3 and 4 accomplish this. They also provide some flexibility by allowing a single place to record the

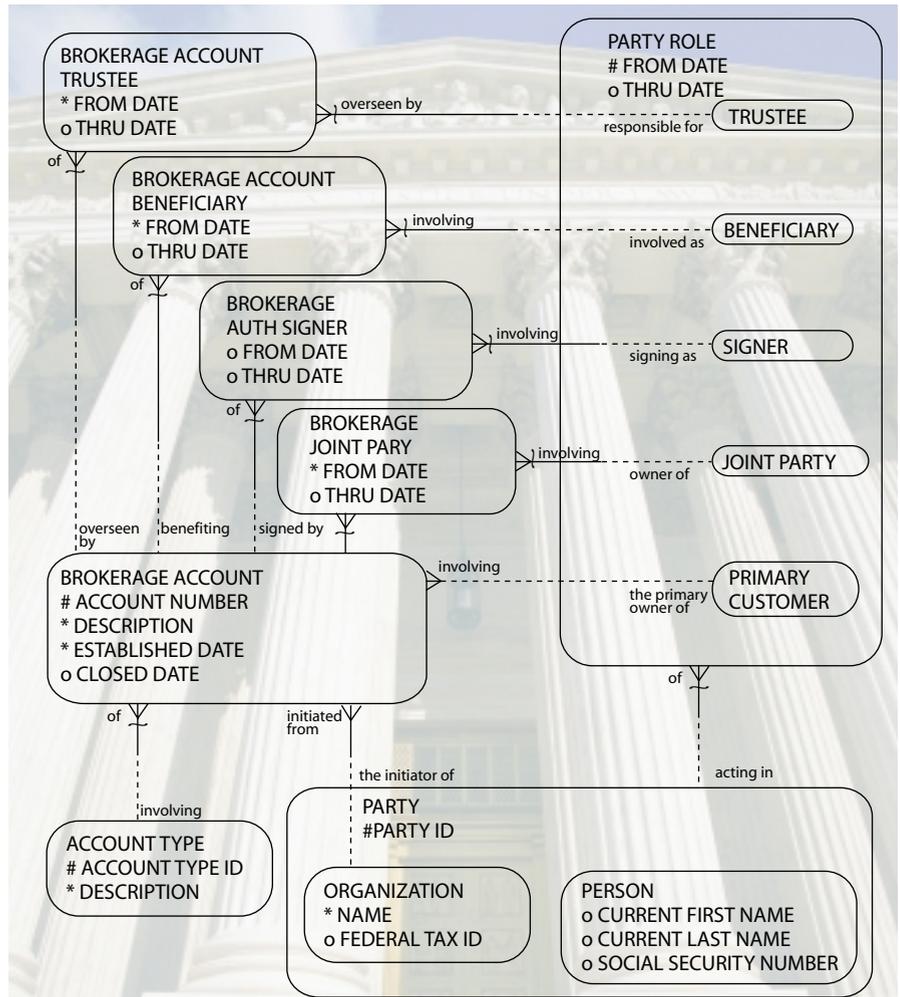


Figure 4: Specific Brokerage Roles Data Model

information about a person or organization, independent of roles.

If the purpose of the data model is to provide a flexible structure that will accommodate changes to the enterprise over time, Figure 5's model

works well. For instance, the previous models show that there is one and only one PRIMARY CUSTOMER for a bank account. Many financial institutions require that there be one primary customer for tax purposes, and they

Notes about this data modeling notation:

- A crow's foot (three prongs at the end of the relationship line) indicates that there are many occurrences of the entity near the crow's foot for each entity that is not near the crow's foot. For example, each PARTY may be acting in one or more PARTY ROLES (entity names are shown in caps in this article).
- The dotted line indicates optionality (as opposed to mandatory) for each side of the relationship. Each PARTY ROLE TYPE may be (because this is a dotted part of the line) the description for one or more PARTY ROLES. Reading the other way, each PARTY ROLE must be described by one and only one ROLE TYPE.
- A '#' in front of an attribute indicates that this attribute is a key. An '*' indicates that the attribute is a mandatory attribute. An 'o' before an attribute indicates that the attribute is optional.
- Boxes within boxes indicate subtypes or subtentities.
- The tilde (~) on the relationship line represents foreign key inheritance. This means that the primary key of the entity closest to the tilde includes, as part of its key, the primary key of the entity without the crow's foot.

need to record the social security number for the primary customer and hold that party primarily responsible. What if the business rules and policies of the financial services organization change over time and there could be equally responsible JOINT PARTIES on an account? In the more specific data models, the underlying data model (and any corresponding data structures) would need to change.

Thus, in the model represented in Figure 5, an ACCOUNT may have any number of ACCOUNT ROLES, which are each for a PARTY and described by an ACCOUNT ROLE TYPE. In this model, the ACCOUNT ROLE TYPES are stored in instances of the ROLE TYPE entity and could be “signer,” “guarantor,” “joint party” and so on. This structure allows additional roles to be easily added over time just by adding a new ACCOUNT ROLE TYPE instance. For instance, a new business practice may involve another role of “party for notices” which designates to whom notices should be delivered. Additionally, ACCOUNT TRANSACTIONS may involve many roles by parties such as the person(s) or organization(s) depositing funds, the person accepting these funds, the person that records the transaction in the system, the person responsible for verifying that the transaction is correct and so on. Again, the roles and the business rules could change over time, so the model easily accommodates change.

However, this latter model does not enforce many business rules. Any number of parties can play any number of roles, and new role types can even be added over time. Where, then, are the business rules documented and recorded? If our purpose is to provide a flexible data structure, another business rules layer could be defined over the data model structure in order to accomplish flexibility as well as precision of defining the requirements. How, then, does one document the business rules? There are many answers such as documenting business rules and relating them to the attributes affected; and as this is the topic of ongoing debate in the business rules movement, alternatives are available

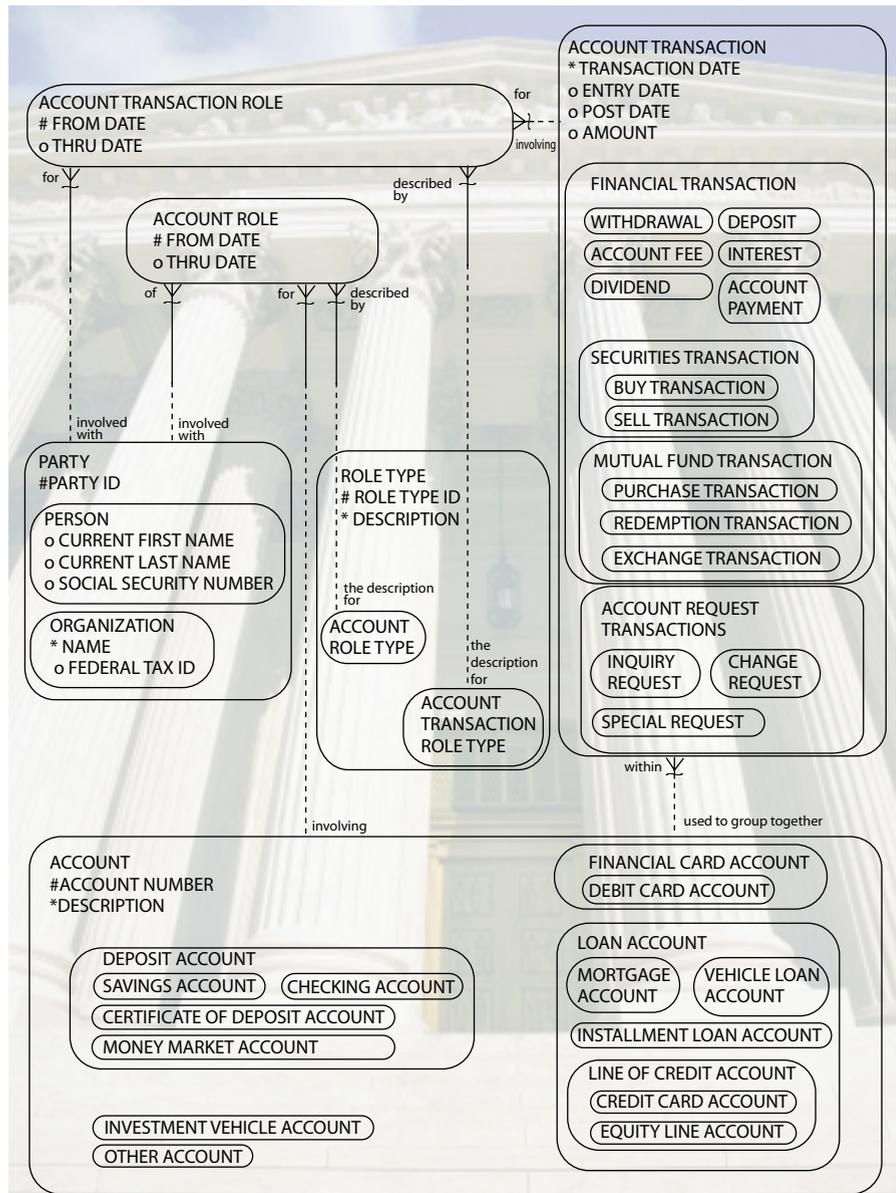


Figure 5: Account Roles Data Model

through avenues such as the Business Rules Forum.

Agreements and Products

Although there are many more details available within the financial services model, there are two additional major data objects within financial services organizations: products and agreements.

Figure 6 illustrates some of the key relationships from ACCOUNTS to AGREEMENTS and from each of these to FINANCIAL PRODUCTS. Each AGREEMENT may be comprised of several AGREEMENT ITEMS which – although not shown in Figure 6 – ultimately relate to

either the terms or conditions of the agreement and could ultimately include relationships to various assets held by the parties involved in the agreement (these are examples of some of the details stored in the more full version of the model). Each ACCOUNT may be governed by one or more agreements; for instance, there could be numerous contracts that are associated with a loan account. An AGREEMENT may also govern many ACCOUNTS; for instance, a banking customer may sign an agreement with the bank governing his/her various savings accounts at the bank.

The financial services products of an enterprise may include both prede-

financed products such as a “checking plus” offering or a customized product that is defined and established for a particular customer. Each financial service offering may include a number of features (FEATURE) such as the interest rate (the value attribute stores the amount of interest), the annual fee (the value attribute stores the fee), and a standard feature such as “checks are provided free with the account.” Customized products may be formed for specific customers (for instance, for high-wealth customers). A customized product may be a higher interest investment account with special FEATURES that are established for use by a customer. Hence, there may be agreement items that specifically entitle certain features that are negotiated for by the agreement. A high-worth individual may negotiate a higher interest rate; hence, the AGREEMENT ITEM is applicable to a FEATURE (through the FEATURE APPLICABILITY entity) that was negotiated such as a higher

interest rate or a waiver of any minimum balance.

Are financial products tied to accounts or to agreements? The model shows the relationship from the FINANCIAL PRODUCT to both the ACCOUNT and the AGREEMENT, depending on the circumstance. A LOAN AGREEMENT may be in place for a specific type of product, and an account may not be needed (for instance, if the customer agrees to borrow a lump sum and pay it back a year later). Other scenarios would suggest that one or more agreements are associated with an ACCOUNT, which is then associated with one or more FINANCIAL PRODUCTS. For instance, there may be a brokerage agreement with terms and conditions for the brokerage account as well as an agreement that provides power of attorney privileges to a close family member.

How can there be more than one financial product for an account or for an agreement? Over time, the product

for an account or an agreement may change. The “from date” and “thru date” on AGREEMENT PRODUCT and ACCOUNT PRODUCT allow the enterprise to record a change in products on the account (for example, changing from a “checking plus” product to a “standard checking” product on an account).

Are there Right and Wrong Data Models?

The models in this article are intended to provide ideas, concepts and alternatives and illustrate possible pitfalls within modeling for financial services (although many of these concepts apply to many other industries as well), allowing the data modeler to be more effective and develop higher quality models in less time. Are the models presented in this article the “right” answer to modeling financial services applications? Or are they “wrong” compared to other ways of modeling the information? I believe that these models can be used as productivity tools and perhaps offer perspectives that the modeler may not have seen, especially because many of these universal data models offer insights from actually implementing these types of designs. Furthermore, each of these models should be used and customized as needed for the needs of the enterprise. However, they are not a substitute for effective requirements analysis. Thus, I believe these universal data models are not right or wrong, but useful in effective modeling. If modelers do not have to redevelop the wheel and we can share ideas and models, then we are also practicing the art of data sharing and integration. ^{DM}

Special thanks to Graeme Simson, Patrick Wilson and Jeffrey Katz for sharing their expertise as I developed this article.

*Len Silverston is a data management consultant with more than 20 years of experience in helping enterprises integrate data. He is the author of the best-selling **The Data Model Resource Book** series (Wiley, 2001), which describes more than 230 integrated, reusable generic and industry-specific data models. Silverston has developed extensive software versions of these data models some of which are now licensed worldwide by Microsoft and some that are available for licensing directly. Silverston's company, Universal Data Models, provides consulting, training and software to jump-start data modeling and data warehouse design efforts while increasing design quality and facilitating data integration. Silverston can be reached at lsilverston@univdata.com.*

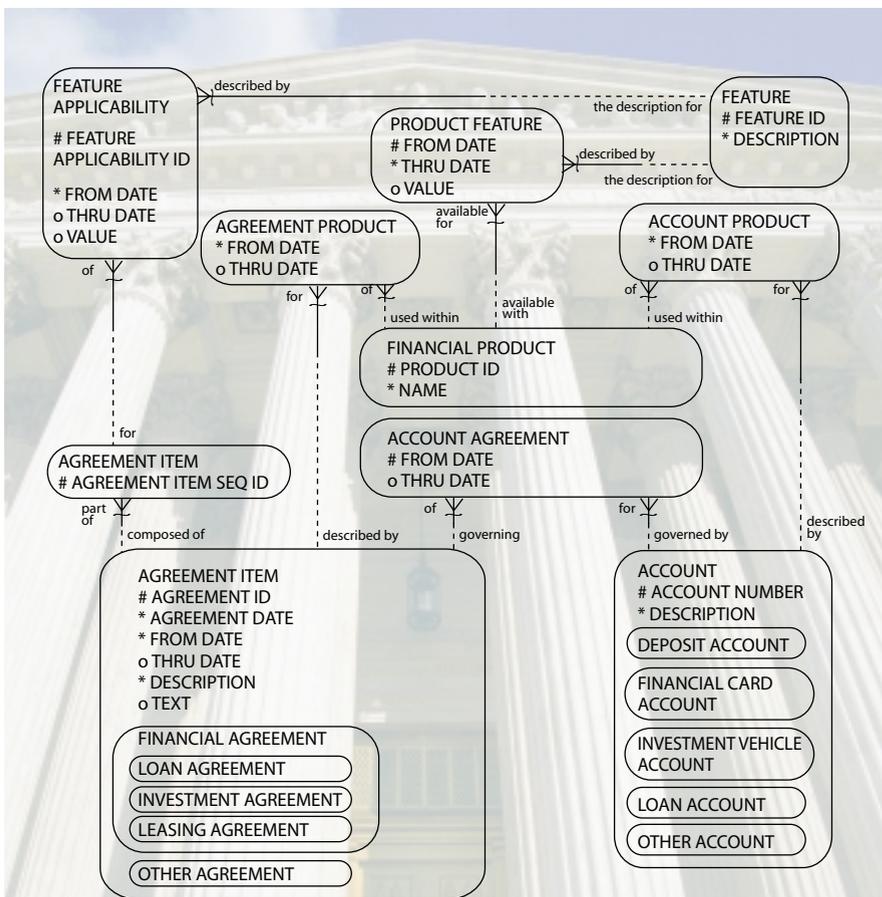


Figure 6: Financial Segments and Products