Incorporating a "Universal" Data Model at a Large Insurance Company

The Insurance Application Architecture (IAA) from IBM is a universal model. What this means is that it is highly generalized and abstract. Thus, instead of modeling entities named CUSTOMER, EMPLOYEE, DEPARTMENT and so forth, the IAA model includes a single entity, PARTY, with two sub-types, PERSON and ORGANIZATION.

Built over many years with the help of many insurance organizations around the world, the IAA model is intended to support the constructs needed by any and all insurance companies. The idea is that insurance organizations using IAA can save time and money by leveraging the use of common database structures, rather than having to reinvent the wheel each time they develop a new system.

The Project

Our client had purchased the IAA model from IBM about one year prior to our being called in. Some work had been done, but basically the model had not been touched for over six months. The client wanted to give it another try to see if they could leverage the investment already made in IAA and to position themselves for the future. One of our tasks would be to discover why the model had been so difficult for them to incorporate.

We agreed on a six-week pilot project, where we would provide process guidance, IBM would provide an IAA expert, and the client would provide a team from their enterprise data architecture group. The selected project focused on member enrollment for group health insurance. The six weeks would be used to analyze requirements; we did not intend to get to implementation issues.

Tools and Process

Because IAA as it was then offered came as a set of books, we agreed that the client needed a modeling tool to capture their work. System Architect™ from Popkin Software & Systems was selected for several reasons, not the least of which was that both the client and ourselves were familiar with the tool. We wouldn't need to spend time trying to learn a tool at the same time we were trying to learn a new approach to modeling. But a more important reason for selecting System Architect was that it is easy to customize (we would find we needed to build several new diagram types and add new definitions to capture the information we wanted) and that it is a repository-based tool. Pictures would not be enough; we needed to build and capture definition information.

We also agreed that if the client were to have any hope of successfully incorporating IAA into their requirements and analysis practices over the longer term and not just for this project, they would need to have a process to follow. Our specific task was to provide guidance in the use of the tool, to develop and document a process that could be used in the future, and to contribute our experience in capturing requirements.

The IAA Models

IAA provides a business architecture from which models can be built that describe applications for insurance companies. The architecture in Edition 3, which our client was using, includes three models:

- The IAA data model consists of a highly generalized set of entities and attributes that support the insurance industry. It describes the 'what to know' that the insurance industry deals with, for example, persons, agreements, and so forth.

- The IAA function model contains descriptions of business functions that occur. It describes the 'what to do' that the insurance industry deals with and is organized into a hierarchy, with elementary functions at the lowest level.

- The IAA function flow model describes the 'when and in what order do I do what I need to do' that a particular business deals with. Unlike the other two models, the function flow model will be unique to each company. It is here that business rules, decision points, and sequence are spelled out.

The IAA data model is highly abstract. Instead of modeling entities named CUSTOMER, EMPLOYEE, DEPARTMENT and so forth, the IAA model includes a single entity, PARTY, with two sub-types, PERSON and ORGANIZATION. It is in the relationship of a PARTY entity with another PARTY entity that a role is taken on. This role information is captured in the 'nature' of the relationship. Each organization specifies the natures, and therefore the roles, which apply to its business policy.

For example, a PERSON entity can be related to another PERSON with a role of DEPENDENT. An ORGANIZATION can be related to another ORGANIZATION with a role of DEPARTMENT. A PERSON can be related to an ORGANIZATION with a role of EMPLOYEE and, perhaps, also a role of CUSTOMER. Figure 1 illustrates the basic PARTY structure in IAA.

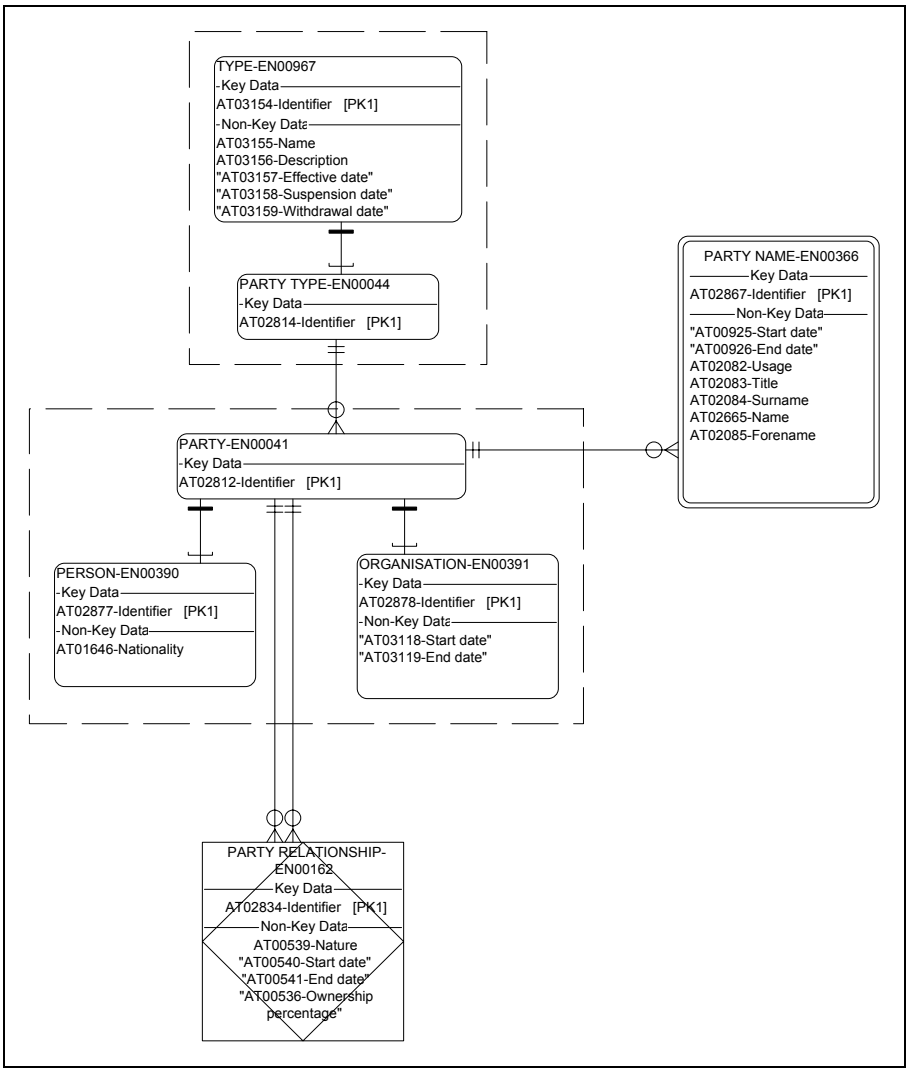

Figure 1: PARTY View

The Problem

Our client agreed that such a generalized model held many advantages. It would be flexible and easy to extend. New roles need only be added as natures to allow new relationship types between PARTY entities. So why hadn't the model been used?

The problem, as we eventually discovered, was that the model seemed to have no relation to the client's business. Interestingly, IBM applies the term 'business models' to IAA. But our client did not see their business in IAA models. Once we understood that this was the roadblock, we set about defining a solution.

Three Levels of Models

We discovered that we needed to develop a process that would allow the client to build models at several levels. The IAA models, while very complete and detailed, did not address the capture of business requirements. Without those requirements, our client felt at a loss to know which parts of the IAA to use, or whether its basic constructs needed to be modified to capture their business-specific rules. So we designed a process that recommended three levels of models:

- ***Business-level models*** would allow us to gather knowledge about the business using familiar terms such as *client*, *member*, or *group plan*. This level of modeling would also allow us to capture information about business processes and the organizations within the business that participated in those processes.

- ***Generic-level models*** would translate our business information into IAA components. Thus our business terms of client and member would become *party*; our group plan would become *agreement*. At this level we would also discover the IAA elementary business functions that would apply to our business processes, and organize those functions into a function flow.

- ***Implementation-level models*** would move the generic models into a physical description. Since we were using the System Architect tool, we could generate SQL and skeleton code from these models. We planned for implementation in our process; however the pilot project did not go this far due to time limitations.

Again, System Architect proved to be an ideal choice, since it allowed us to incorporate many different model types in one repository. We could use industry-standard swim-lane diagrams for business process modeling and entity-relation models for business data modeling, but then move to special diagram types for building IAA data models and IAA function flow models. (Note that the newest release of IAA, Edition 4, focuses on object models. Object-oriented diagram types are supported in System Architect as well.)

How Does the Process Work?

As the project progressed, we not only accomplished modeling work but built a step-by-step process for gathering requirements and for transitioning to IAA. As we developed the process, we found it was extremely important to emphasize the requirements phase. Here is where our clients felt comfortable and could grasp what entities and relationships would be needed and what the business process flow was intended to be.

In our pilot project, we spent approximately one week out of the six conducting JAD sessions with subject matter experts to understand their needs, and to model their requirements. We did not spend longer because the need was to develop skills working with IAA. I would expect that in a live project this work would consume a somewhat larger percentage of the project's lifespan.

Once we had the basic requirements, we could work with the IBM IAA expert to translate our requirements into IAA-compliant constructs. After the remaining five weeks, we felt we had developed a fairly complete IAA data model and function flow model. The process we ultimately settled on looked like this:

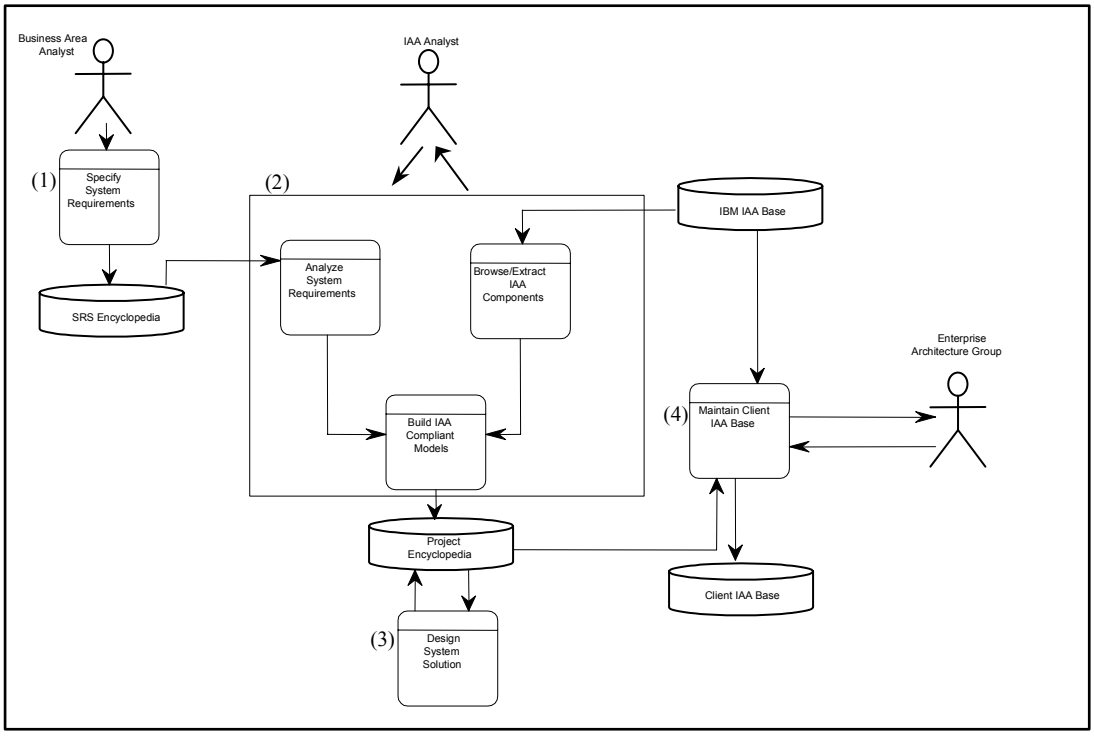

Figure 2: Process for Building Business Requirements and Generic Models

The different symbol types in the diagram represent different facets of the process:

- The stick figures stand for the people, or actors, who participate.
- The rounded rectangles represent sub-processes within the larger process.
- The 'disks' represent stores of information, in this case, the System Architect encyclopedias that held both diagrams and textual definitions.

There were four major process phases: (1) specifying business requirements, (2) translating the requirements into IAA models, (3) designing the solution, and (4) maintaining the client base information. The second process phase, translating the requirements into IAA models, holds three other sub-processes: analyze system requirements, browse/extract IAA components, and build IAA compliant models.

1. Specifying Business Requirements

This was the area of focus that was missing from the IAA approach. We recommended that several models be developed during this phase, including a context diagram, a conceptual entity relation diagram, a list of business events, and a set of event response diagrams. We built these models without worrying about how IAA would

represent the information – we needed to capture the business owners' understanding and we needed to capture it in their terms.

We used a separate System Architect encyclopedia to capture what we learned (the SRS Encyclopedia shown in Figure 2). This encyclopedia was not customized with any IAA definitions or model types. We simply needed a repository to store our business-level information. Once we completed the business knowledge capture, we wrote a Visual Basic script to extract the models and definitions and present them in a Word document. This allowed our business owners to verify what we had done.

2. Translating Business Requirements into IAA

This was the crux of the matter. We now felt we understood the business requirements. How did those requirements get expressed in IAA's universal models?

We started by pre-loading all the information from IAA into an encyclopedia named IAA BASE. We could browse that encyclopedia and extract information from it for use in our project.

Working with IAA Data

IAA entities from the IAA Base encyclopedia are highly generalized and are, therefore, likely to have different names than the entities identified with the business users during requirements analysis. For example, our client had identified a Member entity, and listed "Name" as an attribute within that entity. In finding the corresponding IAA entity, they needed Person, Party, and Party Name to fully represent their business concept of "Member".

Once these candidate entities were identified, we merged them into the project encyclopedia using System Architect's "merge" capability.

The next step was to build the IAA-compliant view of the data using an entity-relation model that had been customized to accommodate IAA definitions. We built this model in two versions: a standard view and an "exploded" view, which we used to reveal the individual natures each relationship could participate in. This model was useful because it was easier for the end users to read, understand, and verify.

Working with IAA Functions

Much as they matched entities identified during requirements analysis with corresponding IAA entities, our client examined their functional requirements and matched them to their IAA equivalents or near equivalents. Each process flow was mapped to the IAA Function Hierarchy, so that appropriate IAA functions could be determined. For example, if our client used "Underwrite Member" as a process flow, the function hierarchy would show that IAA's equivalent is "4.4.3.2 Underwrite Insurance Agreement."

These candidate IAA Functions and Flow Control Functions were then merged to the project encyclopedia. During this activity, AFC modeled their process flow and then built IAA-compliant function flow models. We found that typically, one high-level IAA flow control diagram could be built for each process flow diagram. Then we could build child diagrams for lower-level flow control functions. These diagrams were used to show how the system would respond to external events, such as the receipt of information.

Completing Deliverables

Once we felt that the data and function models represented accurate information, we cross-checked the entities and attributes against the business functions, making sure that all data required is accounted for by some function. We then walked through the models with users, confirming everyone's understanding of the application needs and identifying any unresolved issues. Finally, we produced a specification document and archived the project deliverables.

3. Designing the System Solution

The next phase of the process would be to take the IAA-compliant models and translate them into (1) physical data models optimized for the selected database and (2) physical process models. Again, System Architect is a good choice because it generates schema for many different databases. It can also be scripted to produce code.

4. Maintaining the Company Base Information

As the client worked with IAA, they would need to develop an encyclopedia for their own Company Base definitions. In other words, they needed to build a client-specific universal model. If what IAA calls "Forename" is always called "First Name" at the client, then they should change the name of that attribute permanently. We recommended that our customer store the IAA Base information unchanged in one encyclopedia, and maintain a set of Company Base information in another.

Conclusion

Through the customized use of a repository-based modeling tool, it is possible to model system requirements based on IAA constructs. The effort requires careful analysis of each business's individual characteristics and processes, but it ultimately rewards the company with IAA-compliant documentation. Perhaps the most important lessons we learned are these:

- Users of universal models need not only those models, but must also have business-oriented models so that they can gather knowledge from their business users using their own terms.
- Users of universal models need a process to help them gather business requirements and then transition successfully to the universal view.

About Doreen Evans Associates

Doreen Evans Associates (DEA) is a professional services firm that focuses on business process improvement. We can help you change a process, build an enterprise architecture, or define requirements for your systems and technologies. Founded in 1992 as a woman-owned, privately-held small business, our mission is to ensure that business need drives solutions.