NISTIR 90-4292

# Data Administration: Standards and Techniques

# Proceedings of the Second Annual DAMA Symposium

**Judith J. Newton**
**Frankie E. Spielman, Editors**

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899

May 3 1989

Issued April 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

## FOREWORD

This document represents the proceedings of a one-day symposium held at the National Institute of Standards and Technology on May 3, 1989. It was the second in an annual series of symposia on the subject of data administration. As more and more organizations recognize the need to treat data as a corporate resource, data administration is gaining acceptance as an important area of specialization for computer professionals.

This symposium was jointly sponsored by the National Capital Region of the Data Administration Management Association (NCR DAMA), the Federal Data Management Users Group (FEDMUG), and the Association for Federal Information Resources Management (AFFIRM). We wish to thank the following individuals for their commitment to and assistance with the symposium:

Carole Anderson, DAMA
John Coyle, AFFIRM
Alice Cohen, DAMA
Rene Fecteau, DAMA
Tammar Paynter, DAMA
Ronald Shelby, DAMA

We also wish to express our gratitude to the speakers, session moderators, and participants who made this symposium possible. We are especially grateful to Joyce Myrick for all the administrative support she put forth in making these proceedings a reality.

With one exception, the papers in this proceedings represent manuscripts submitted to the editors for publication. Mr. Zachman's talk has been summarized by the editors from audio recordings and is marked "transcribed." An attempt to retain a feeling of the dynamic structure of this talk has been reflected in the colloquial nature of the transcription.

Because the speakers in the symposium drew on their personal experience and knowledge, they may express views which do not necessarily reflect those of the National Institute of Standards and Technology, DAMA, or AFFIRM. Additionally, they sometimes cite specific vendors and commercial products. The inclusion or omission of a particular company or product does not imply either endorsement or criticism by NIST, DAMA, or AFFIRM.

Judith J. Newton
Symposium Chair

Frankie E. Spielman
Program Chair

CONTENTS

# DATA ARCHITECTURE: THE TRANSITION FROM BUSINESS MODEL TO DATA MODEL

John A. Zachman
International Business Machines Corporation

Transcribed by
Frankie E. Spielman

I am delighted to be here today to talk to you about an area that I have been working on for a long time, "Data Architecture, the Transition from Business Model to a Data Model." I will first discuss the framework for Information Systems Architecture and will then make some observations about the framework. Lastly, I will draw some conclusions that have to deal with the transition from the business model to the data model.

I have been working in the area of a framework for information systems architecture for about 20 years. The broadest term for this area might be called **Enterprise Analysis.** The whole concept of **Enterprise Analysis** is to understand the enterprise as a system in its own right before you start to overlay against that enterprise the information infrastructure required to support it. This concept has been around for a long time. I will be focusing on a subset of the enterprise analysis, the information systems architecture, for a few minutes. Why is information systems architecture so significant to us? For many years, the rest of the data processing community and the business community have thought of us as "the people where the rubber meets the sky"; it's a little esoteric. However, several things are happening that are forcing these issues out on the front burner, not only in the technology management environment but also in the business environment. In the technology management environment, the fundamental driver is the price performance of the technology. The price performance stated by the hardware people has been $10^6$ over a 20 year period. This level of improvement is expected to continue over the next 20 years.

When you see this improvement in performance, it has two impacts on you from a technology management perspective. First it brings enterprise-wide integration into the realm of cost feasibility. A few short years ago, we were just trying to get the payroll program to work. Now, we are trying to integrate the implementation from the scope of the entire enterprise, which is a different kind of a problem. It is so large and complex that one brain cannot comprehend the whole thing at one time. We are discovering that we must have explicit ways to depict the things that we are trying to accomplish so that many people can create the same baseline without losing control of the integration. The second impact on the technology management perspective is that it allows you to package the technology differently; basically with smaller and smaller machines at lower and lower prices. It gives you a great facility for decentralizing the information systems capability out through the rest of the enterprise. We are discovering that decentralizing the information systems without structure, or an architecture, is chaos. It's anarchy! You can **disintegrate** the business. If you keep driving the price performance of technology out of sight, sooner or later you will end up having to deal with these kinds of issues.

From a business or general management standpoint, a similar type of thing is happening. Only there, the fundamental value has to deal with the rate of change. Alvin Toffler wrote the book, <u>Future Shock</u>, where the hypothesis was that not only is change increasing exponentially, but the rate of change is increasing exponentially. That places a very high premium from a management standpoint on the flexibility for infrastructure change; the fundamental structural aspects of the enterprise like the organization structure, product or service structure, distribution channel, geographic structure, or control structure. Changing the infrastructure is the only device that management has at its disposal to maintain the viability of the business or enterprise in a dynamic environment. Management must restructure as the environment changes around itself or end up with a dinosaur on its hands.

As soon as I say infrastructure change, from a technology management perspective, that is the worst possible news that somebody could hear because the implication is that the change is not cosmetic, but it is a fundamental or structural change. You have one of those "Good night, let's throw this stuff away and start over again" kinds of problems. However, from a general management perspective, that turns out to be the name of the game in a dynamic environment. The implication of the infrastructure change is that if you have no baseline against which you are going to attempt to manage change, forget about it - you are not going to manage change. You can change things alright but, in effect, you become the changee. What tends to happen is that the dynamic environment begins to change around the enterprise and management begins to discern that it is no longer relevant in that environment or the market place and decides that it must do something about this situation. **So, let's reorganize!** Then we will change the product structure! Then we'll redefine the distribution channel, change the geographic boundary, then decentralize, then wait for 20 years to see what happens. By that time, the environment changes around you and you end up with a dinosaur. The point is that if you have no baseline, the explicit specification of the infrastructure of the business to serve as a baseline, against which you are going to attempt to measure change, forget about it, you are not going to manage change. That is suggesting a need for information systems architecture, explicit specifications of the infrastructure of the business to serve as a baseline for managing change. So it doesn't really make too much difference whether you argue the point from a general business management or technology management perspective, all these paths start to cross and are based upon information systems architecture.

The reason I developed this material that I am going to share with you today is that our company, an enterprise, was being affected by these kinds of issues. They created a task force to focus on the subject of information systems architecture. They were going to make us talk to each other and draw up some grand and magnificent conclusions on information systems architecture. But all of these people had their own views of what information systems architecture was, they used the same words but meant different things. So, therefore I thought that the only way to have a meaningful dialogue was to find something totally independent from what the group was working on to talk about, something outside the scope of information systems, some neutral ground. If we could get an agreement there, then we could come back into the information systems community, hold the agreement, and establish a basis for having a meaningful dialogue. The goal was to describe information systems architecture without talking about information systems architecture. I will share with you today this approach of discussing something outside the information systems community, return to information systems and draw the parallels between the two, and then develop a framework for information systems architecture. The purpose of stepping you through the framework is to establish a context from which I can draw some conclusions and make some observations of the modeling issues that we are trying to deal with.

For independent, objective thoughts about architecture, I thought why not try an architect? They have been in the game for a thousand years. So I went to one of my friends and said: "Talk to me about architecture". When someone wants to build a building, they go to the architect. A sample of such an initial conversation goes like this:

"I'd like to build a building."

"What kind of building do you have in mind? Do you plan to sleep in it? Eat in it? Work in it?"

"Well, I'd like to sleep in it."

"Oh, you want to build a house?"

"Yes, I'd like a house."

"How large a house do you have in mind?"

"Well, my lot size is 100 feet by 300 feet."

"Then you want a house about 50 feet by 100 feet?"

"Yes, that's about right."

"How many bedrooms do you need?"

"Well, I have two children, so I'd like three bedrooms."

The first thing the architect does as a result of this conversation is to create a "bubble chart" (see figure 2). The bubble chart depicts, in gross terms, the basic intent of the final structure, and the size, shape and spatial relationships. The architect prepares this bubble chart for two reasons. First, the prospective owner must express what is in his mind that will serve as a basis for the architect's actual design work. Second, the architect must convince the owner that the owner's desires are understood well enough so that the owner will pay the bill for the creative work to follow. In effect, the purpose of the bubble chart is to initiate the project.

If the project is in fact initiated, then the next step for the architect is to produce an architectural drawing. The architectural drawing is significant because it is a depiction of the final product as seen by the owner. The drawings include three views: horizontal sections (floor plans), vertical sections (cutaways), and pictures depicting the artistic motif of the final structure. The purpose of these drawings is to enable the owner to say: "You got it, that's exactly what I had in my mind!" or "Make the following modifications." Once the owner agrees that the architect has captured what he has in mind, then everyone signs the contract to continue to the next set of architectural deliverables, the architect's plans.

The architect's plans are different from the drawings. The architect's plans represent the final product as seen by the designer. The architect is thinking of what the owner has in mind and he has translated that into a product. The architect is developing plans composed of 16 categories of detailed representations putting an explicit specification around the material composition of the final product, including wood structure, joints, fasteners, and so on.

The whole reason for the architect to produce the architect's plan is to serve as a basis for negotiating with the general contractor. This is the last product that the architect produces. He may stay involved with the process but this is his last product. He gives it to the owner who takes the plans to a general contractor and says, "Build me one of these". If the contractor builds according to the plans, the owner knows that he should be getting the desired product as depicted in the architect's drawings.

At this point, the contractor redraws the architect's plans to produce the contractor's plans representing the builder's perspective. The builder is constrained by the laws of nature and by the fact that complex engineering products are not normally built in a day. Some phased approach is required which comprises: digging a hole, pouring cement for the foundation, then constructing the first floor, then the second floor, and so on, until the building is completed. If you happen to get that out of sequence, forget it, you've lost it! Furthermore, the contractor may have technology constraints. Either the tool technology or the process technology may constrain his ability to produce precisely what the architect has designed. The contractor will have to design a reasonable facsimile which can be produced and yet satisfies the requirements. These technology constraints, plus the natural constraints requiring phased construction, are reflected in the contractor's plans which serve to direct the actual construction activity.

Now, the general contractor hands the contractor's plans to the subcontractor. The subcontractor produces another representation called the shop plans. The shop plans are the detailed descriptions of the parts or pieces of the final components or parts of the total structure. The subcontractor is never interested in the total structure itself, only in specific parts or pieces. These shop plans might even serve as patterns for a quantity of identical parts to be fabricated for the project. For example, a specific subcontractor might only produce the fasteners.

Finally, we end up with a complete building. It is interesting that in the process of building a complex engineering product like a building, there's not one architectural representation produced but a set of them. As a matter of fact, there appear to be three fundamental architectural representations being produced because there are three fundamental people involved, the owner, the designer, and the builder. Each one of them has a different perspective, different motivation, different constraints, different diagrammatic constructs, different semantics, etc. Now, they precede that with a **ball park** representation within which all of the ensuing architectural activities will take place. They succeed that with the out-of-context representations required for actual implementation purposes. But basically, there appears to be the three fundamental architectural representations representing the viewpoints of the key players who are playing in the game.

In seeing all of this, I thought apparently there is some logical construct that is driving the architectural construction folks through a series of transformations as they take an idea from its conception to its implementation. These may be merely the manifestations of those transformations that are taking place. And if that is the case, then the high probability is that anybody who builds complex engineering products is likely to be driven through the same logical transformations as they take any idea from conception to implementation.

If we examine the process of building airplanes, we find that they happen to produce the same set of architectural representations. The primary difference is that the names of the different viewpoints change. They start out with a **concepts** package representing the **ball park** describing the specifications. For example, concepts for the final product indicate its size, shape and whether it will fly high or fast. They then produce the **work breakdown structure.** The owner, government in this case, requires that the aerospace manufacturer produce a representation of the final product against which the government controls the costs and schedules. In this way, the government controls the manufacturer to ensure that they produce the product the way the government wants. Then engineering translates the work breakdown structure into the **engineering design**, producing drawings and bill-of-materials which begin to specify the nature of the product that the owner has in mind. Then manufacturing engineering produces the **manufacturing engineering design** which basically constrains the engineering design based upon the laws of nature and available technology. It describes how to build the product (i.e., inside-out, bottom-up) and ensures that the product is actually producible. Then you have the assembly and fabrication drawings, the out-of-context representations used on the shop floor for actual fabrication and assembly.

Then manufacturing inserts another level of representation not ordinarily found in architectural construction. This allows the manufacturer to use computer-controlled equipment to produce multiple copies of the same product. They code up the out-of-context representation into machine language representation. This is just one more representation of the product short of the actual, physical product itself.

In comparing the manufacturing industry and construction industry, there is a basic underlying set of logical transforms that are driving anybody who builds large complex engineering products as they take ideas from conception to implementation. And if you believe that information systems are complex engineering products, then we should be able to find the analogous architectural implementations being produced in the information systems as are being produced in other disciplines. And the fact of the matter is, we can find the analogous representations (see figure 3).

In the building industry, they start out with the bubble charts. We in information systems start out with a scope and objectives statement. This describes the ball park that we all are playing in. They produce the architect's drawings, the building as seen by the owner. We produce a model of the business, a description of the business, a system as seen by the user or the owner. They produce the architect's plans, the building as seen by the designer. We produce a model of the information system, the system as seen by the designer who translates this into a design product. They produce the contractor's plan, architect's plans as constrained by nature and the available technology. We produce a technology model, an information system as constrained by the available technology. We also insert that other level of architectural representation that manufacturing inserts which is called the detailed representation. This is the description of the pieces or the object code. We finally end up with the functioning system. In any case, we can find the analogous representations being used in the information systems as they are being used in other disciplines.

Beyond all of this, there are different ways to describe the same thing or object (see figure 4). Three such descriptions are material, function and location (spatial). If you are going to describe a product to be built, you can describe it from a functional perspective. If you are going to describe the function, the description is based on the transformation that is going to take place; that is, the input-process-output process. You can also describe the product from a material perspective which addresses the structure of the product. It's like a bill-of-materials. These are two different independent, and not interchangeable ways to describe the same thing. You cannot substitute one of these for the other. If you are describing function, you cannot use thing-relationship-thing as a basis to describe function. This means that you can work with a bill-of-materials for as long as you like but you will never describe the functional specifications of that product. Or vice versa, if you are trying to describe material, you cannot use input-process-output to describe that material. The third description is spatial in nature which describes the location or the flow of the work or product. In short, each of the different descriptions has been prepared for a different reason, each stands alone, and each is different from the others, even though all the descriptions may pertain to the same object and therefore are related to one another.

We can find analogous representations in information systems. In information systems, we produce functional representations with the focus on the transformation of **input-process-output**. This is the functional model. Then there is the "stuff the thing is made of" which, for information systems, is the data. In information systems, the analog for the material description would be a data model. In the data vernacular, **thing-relationship-thing** would become **entity-relationship-entity**. The data model fundamentally is the same thing as the bill-of-materials for the information systems product. We also have the spatial representation or the geometry which is the focus on flows or connections between the various components. In the information systems network vernacular, **site-link-site** would become **node-line-node**. Once again, the implication is that we can define in information systems the analogous representations being produced as they are being produced in other systems.

Now, two ideas have been discussed today. They are:

> Over the process of building a complex engineering product, there is not an architectural representation being produced but there is a **set** of architectural representations. They tend to represent the different viewpoints of the different players playing in the game-the owner, the designer, and the builder.
>
> There are different ways to describe the same thing - the data model, functional model, and network model representations.

If we put these two ideas together, it would suggest to you that there is a relationship between these two ideas which could be depicted in a classical relationship representation, e.g., a matrix. This suggests that for every one of these different ways to describe the same thing (models), there are the different viewpoints; owner's, designer's, and builder's representations. Figure 5 illustrates the total set of different perspectives for each type of description. It depicts a framework for information systems architecture, not the more generic manufacturing or construction names.

Now, the one single factor that makes this of any significance to you is that you can explicitly differentiate the elements on either axis of the matrix. That basically says any one element on either axis of the matrix is explicitly different from all the other elements on the axis of the matrix. In effect, from data processing terminology, it is not a decomposition that is taking place along the axis, it is a series of transformations. That is saying that the contractor's plans are different from the architect's plans, they are not just more detailed. They might be more detailed but they are different in nature, in context, in structure, diagrammatic constructs, semantics, constraints, motivation, perspective, and so on. The architect's plans are different from the architect's drawings, not just more detailed. **They're all just different!** Because each of the elements on either axis is explicitly different from the others, it is possible to define **precisely** what belongs in each cell with some rigor.

To illustrate how each cell differs from all the others, examine the data description (analogue of bill-of-materials) column. The first row is the objectives/scope or **ball park** row, the architect's bubble row. You would expect to find in the cell at this intersection the list of things that are important to the business. Things in the data vernacular would probably be called an entity. At this level, an entity is a high level aggregation of that entity. We are not talking about a lot of detail because we are not doing any design. We're trying to say, "What's the ball park that we are all playing in?" In effect, what you are trying to do with this architectural implementation is to make a strategy decision; this is the information systems strategy decision as it pertains to data. We have been struggling for years on how to map the business strategy to the information systems strategy. Here is where it is taking place. Basically, a decision is being made at this point. Out of the total set of things that the business is interested in and therefore manages, what is the subset of the total set that you are going to invest your money in? It's like any other investment decision, its always nice to know what the comprehensive set of alternatives are before you start picking the subset. If you pick the subset before you know what the complete comprehensive set is and invest your money in it, you get down the line four or five years and someone says, "Oh, I forgot to tell you one!" That changes the whole structure of your business decision. In any case, there is a list of real-life things that the business is interested in; products, parts, supplies, equipment, employees, customers, and whatever. The question to the CEO is, "Which one of these do you want to invest in?" The number selected will depend upon the amount of money available to invest in them.

Now if we look at the next level down, this is the owner's view or model of the business. The description model will be an **entity-relationship-entity** diagram. The entity to the owner is a business entity. The owner, for example, says employee. The owner is thinking about a real person as an employee. In contrast, the owner does not think about a record on a machine. That is an entirely different kind of concept of an entity than would be found in detailed representations in another row. When the owner thinks about relationships, he is thinking about the business rules or business strategies that are an association between the business entities. An example might be, "In this business, we ship this product out of **this** warehouse." That is the rule or strategy. A different strategy might be, "In our business, we pick this product out of **every** warehouse that we have". It's an entirely different strategy. These are business rules and not data relationships such as would be expected in the model of the information system (designer's view).

Finding good **real-life** examples which illustrate each of the architectural representations is difficult. For the last several years, we have been trying to produce **an** information systems architecture, one picture! That says that we have all the independent variables varying dependently in the same picture. No wonder we are having trouble in design decisions! I was the first one that I know of who has tried to sort these independent variables out into categories and draw clean boundaries around them. It isn't always easy to find a clear picture (see figure 6). Once there is a nice clear picture, it is not always easy to decide which cell it maps into; for example, is this figure the owner's view or the designer's view? It is clear that his view maps into the data column because it concerns data about a department. In this picture, we see **many-to-many** relationships. We know that in real life there are **many-to-many** relationships but as soon as you start reducing real life down into two dimensions of the machine, you can't have **many-to-many** relationships. You have to resolve them, for example, the way to resolve this relationship is to create **artificial** entities. Before this could become a legitimate model of an information system, a "data bigot" would have to normalize it. In any case, this picture is a model of the business and not of the information system. For the information system representation, the data would be normalized.

If you look at the next level down (figure 5), this is the designer's view or model of the information system. The meaning of entity changes to that of a record on a machine. The designer thinks of an employee as a record on a machine. For relationships, the designer thinks in terms of a data relationship or the linkages between the files that allow you to have access from one file to another. Now if we look at the previous example model from a designer's view, it has gone through a transformation (see figure 7). It is a different model but it is a structural derivation from the business model. You can see the intersection of the entities involving the **many-to-many** relationship. Clearly, it is a model of an information system and not a model of the business because of the existence of **artificial** entities, specifically the **DEPTPROJ** entity. It results from the concatenation of department and project and is not a real-life entity but something that is required to make the information system run on a machine.

Looking at another level down (figure 5), we would expect to see the builder's view or technology model. The laws of nature and technology constraints are being applied. The builder is going to say something like this, "We are going to use an IMS lathe to build this baby, or a DB2 press!" In using IMS, entity means "segment" and relationship means "pointer." In DB2, entity means "row" and relationship means "key." Now conveniently, it is exactly the same statistical data model that we looked at previously but now it has gone through another transformation.

In the next level down (figure 5), you would expect to find the out-of-context representation or the database description language. The entities are now specifications of the "fields" and relationships are the specifications of the "addresses." This description is compiled to produce the machine language representation.

So in this data description column, you can find the information systems real-life examples that map very cleanly into the hypothetical constructs that come out of the architectural conception, manufacturing engineering, and so on. We have only put the information systems names around exactly the same logical constructs.

Now, we could work down the other two columns (function and network) in exactly the same manner as the data column. The model for describing the process is **input-process-output**. Each of the representations in the different cells in this column have different meanings associated with **input**, process, and **output**. At the scope description cell (ball park view), we have the functions or processes. At this level, the processes are high level aggregation or process classes. It is not a lot of detail because you are not ready for design yet. Once again it's a strategy decision to select some subset of the appropriate business processes in which to invest money or information systems resources for automation purposes. It is a different investment decision than before but it is still a decision based upon a different set of alternatives. This is the owner's view in which you would expect to find a functional flow diagram in which process is a business process. The inputs and outputs are people, cash, material, products, etc.

In the designer's view for the function column, you would expect to find a data flow diagram. Here, process is an information system or application process, not a business function. The inputs/outputs are **user views** of some data that flow into and out of the application processes.

In the technology model, you would expect to find a structure chart which is a builder's view. In applying the physical constraints of the technology, the processes become computer functions such as storage devices, terminals, and compilers to be used.

Last of all, there is the out-of-context representation. You would expect to find a program in which process is a language statement and the inputs and outputs are control blocks. The program is compiled to produce object code, the machine language representation.

Finally, we could do the same thing in the network column. Again, each of the cells will be different because the cells are based on location. The **nodes** mean something different to each of the users at the different levels and you would end up with a set of network architectures.

In any case, I have developed for you a framework for information systems architecture. All this says is that there is not **an** information systems architecture, there is a set of them! In fact, the information systems architecture is relative to who you are. If you are a programmer, for example, you probably think a structure chart is the information systems architecture. If you are the database administrator, you think data design is the architecture. If you are the data administrator, you think the data model is the architecture. And so on. So it depends upon who you are as to what you are thinking about when discussing information systems architecture. It is little wonder that we are having difficulty communicating with one another about the subject. As a matter of fact, you can see exactly why we can all be using the same words meaning something totally different and having arguments over it! So one observation is that there is <u>not</u> **an** architecture, there is a **set** of them.

Let's take a look at a couple of other observations. Suppose you are trying to produce an architecture, for example a program structure chart (function column, technology model row). If another architectural representation happens to exist above the model of the information system that you are working on, but has not been explicitly described by anybody before you began work on yours, then you are going to have to make assumptions about the next higher level architectural implementation. Those assumptions might be correct, or they might be wrong. If they are wrong, you will find out as soon as you start a systems test. When you start combining all the programs together, forget about it, they are not going to go together! So at that time you are going to have to set out to define the architectural representation which you will do explicitly, or implicitly by continually beating on the program. One way or the other you're going to have to define the higher level architectural representation (model of the information system), then rewrite the program, regenerate the object code and finally get the system.

Now, taking it one step further, if you are working on the model of the information system representation and the business model above it has not yet been defined, then you are going to have to make assumptions about the model of the business while working on the model of the information system. Again, the assumptions might be correct, or they might be wrong! And again you will find out when you implement the system. If anyone changes any one piece of the technology while you are developing the program, forget it, it is not going to go in! If someone says, "Let's now use PCs instead of the 3270s," it won't work. It is a technology based design. If you change the technology, you have just changed the design. So, you are going to have to change the higher level architectural representations which you will do explicitly or implicitly. If you do it implicitly, you probably are not going to implement a system that the user had in mind. It is not going to support the business. With this implicit approach, you are going to continually beat up on the user and redefine your system. Unless you can redefine the user! This places more emphasis on explicitly defining the architectural representations at each level. There is a message in this. The reason you are building higher and higher levels of architectural representations is to minimize the erroneous assumptions. You are dealing with the product quality issue, in manufacturing they call it the scrape and rework problem! The question is, "How much do you want to spend on scrape and rework?" If you don't want to spend the money on scrape and rework, then you are going to have to spend resources at the beginning to define the set of architectures to ensure that you have a high quality product when you implement it.

Now, is there a secret message in defining these architectural representations? Is there a secret message that you must start at the top level representation (objectives/scope) first, then do the next one down, then the next one, and so on? The framework doesn't say that. I could supply you with the logic as to why you would want to do it that way. I can also provide you with the logic to start at the bottom; that is, to go directly to writing the code first. If we go back to the building analogy, we can define the logic for starting with the coding. For example, you are going to build a log cabin. What do you need an architect for? Or a general contractor? Go get yourself an ax, find a forest, cut down trees, and build your log cabin. On the other hand, if you are going to build a 100 story building, forget about the ax and the forest. You will get an architect and a general contractor and work methodically down through all the different architectural levels. You will bring out every possible erroneous assumption that you can so that when you get the building constructed and standing there, it will stand there for longer than 15-20 minutes! There are times when you want to work top-down and times when you want to work bottom-up. It is a risk management issue. The question is, "How much risk are you willing to assume?" If you are willing to accept enormous risk, go for it and start at the bottom. If you don't want to take the risk, start at the top. How much money are you willing to spend up front to minimize the risk and maximize the quality of the product? Again, it is a product quality issue.

One additional thought while discussing the framework chart: it is easy to trace the evolution of programming tools and methods in the context of the framework over the 45 year history of data processing. We started by writing object code which doesn't even show on the chart. Then along came assembly language, Fortran, COBOL, structured programming, structured analysis, etc. All of these primarily supported the functions column of the information systems architecture. We can see the evolution of these programming tools and methods as the price/performance of technology increased. These changes in technology affect each level of the function architectures in a chain reaction sort of way. It wasn't until the 1960's that we started placing more emphasis on the data. From that period, we have seen the evolution of data definition languages, database management systems, entity-relationship modeling, semantic modeling, etc. Now the tools and methodologies are beginning to pour out. On the other hand, we haven't figured out the last (network) column yet because networks didn't become a factor until the 1980's. Now we have PCs on everybody's desk with connections to the networks. We have not done very much as far as algorithms, conventions, or analytical tools on where to locate or place the nodes and equipment on the network. The state of the art has not yet matured that much for this network column. So, in summary, we have been working on functions for 45 years, data for 20 years, and networks for only about 5 years. More work is required in this latter area. We do have a lot more to learn about information systems architecture. In any case, the heart of working in the different levels dimension is a quality issue. In discussing the quality of a product, we have to address the resources we want to spend at each level of the chart.

On the other hand, adding additional columns of architectural implementation turns out to be a productivity issue. For the last 45 years, we have been building information systems based upon the functional specifications alone to the exclusion of the data. Is it possible to build complex engineering products on the basis of functional specifications alone? What are the implications of doing this? In information systems terms, do you want functionally driven design, or do you want data driven design? That is the name of this issue. We are probably going to have to find some totally neutral, independent, unbiased basis for understanding these issues and then we must go back inside information systems and find analogues. Can we find something neutral? They have done this in the manufacturing world. Products built on functional specifications are called **job shops**, which are made to order business products, or customized products. The customer walks in the door and places an order with the job shop describing the product that he wants based on the functional specifications. Engineering decides how to design the product to satisfy the customer's needs. Manufacturing engineering figures out how to build the product. Manufacturing operations gets the raw material and produces it. It is customized to the functional needs of the user.

Incidentally, the very same thing happens in data processing. The customer walks into the door and says, "Give me an application product!" Then process manufacturing asks for the functional specifications which the user defines. Engineering, or the analyst, designs the product to satisfy those functional specifications. Manufacturing engineering, or the programmer, determines the manufacturing processes and procedures to build the product. Manufacturing operations, or data processing operations, gets the raw material or data to produce the product. The product is a customized product! If you stay in the job shop for a long enough time, the marketplace has a tendency to evolve and mature and the marketplace will begin to drive the manufacturer out of the job shop or customized manufacturing business into a standard production environment manufacturing standard product and ultimately into a factory of the future, an assembly order business.

There are some inherent limitations on customized products. There is a long lead time and high creative product cost. One product is produced by going through the complete manufacturing process. There is no product flexibility; once the product is produced it will perform that one function and nothing else. So, there are high maintenance costs over a period of time because you never make spare parts for a custom product. When parts are needed, you have to go back to the manufacturer to have a new part built. There is no part interchangeability. All of these inherent characteristics of the manufacturing business also apply to the information systems business resulting in long lead times, custom software, no flexibility, and high maintenance costs. In the manufacturing environment, the market place tends to drive the manufacturer into a standard production environment where you don't manufacture to order but manufacture to storage. The customer is provided off-the-shelf products which result in reducing lead time to zero, spreading the cost over several products, reducing the maintenance costs, and so on. Ultimately, the market drives the manufacturer out into the factory of the future or assembly to order. In effect, that forces you to begin to deal with assembling from standard bill-of-materials or forces you to formalize a set of architectural implementations. It forces you to form another column if you are trying to deal with orders of magnitude greater flexibility at orders of magnitude less costs. You cannot do that in a custom environment or job shop. This is fundamentally the same in information systems, manufacturing data to storage and assembling to order against the demands of the product. You assemble from off-the-shelf standard data what looks like a custom product. That is fundamentally the factory of the future concept for information systems. So in a nutshell, it forces you to add another column to improve the productivity. You are trying to add a magnitude of flexibility while at the same time reducing the cost - this is productivity. In summary, it is a productivity issue that forces you to form more columns in the horizontal dimension and it is a quality issue that forces you to form more rows in the vertical dimension.

Now, let me draw a couple of conclusions regarding the business model versus the data model. Remember, the business model is the owner's view and the data model is the designer's view. When people discuss business models, there is a tendency for them to think of a business process model as opposed to a business semantic model, rules model, or logistics network model. Business models are for business design, and we have little experience with regards to business design. The tools and methodologies are basically growing up from the information systems community into the business community. It is clear that if you are going up the data column, we know how to build bill-of-materials. The same fundamental structure should be useful for building the work breakdown structure or the business designer's representation as well. However, we have never done business design as a whole, we tend to build the semantic models for information design purposes. We don't really try to do business design and, consequently, are unsure how the languages have to be extended or enriched in order to do business design. So, we do have a lot more to learn about business modeling. Therefore, there are some constraints with regards to formulas as they are used to do the business modeling. Probably, if you drive the rate of change up very much more the business will become dependent upon the business models to manage the change in the business. So, if we keep driving the rate of change up, the business is going to be more inclined to produce those formalized models because they form the baseline for managing change in the business. In the next decade, we should see a lot more work in the area of business models. For the time being, we have limited experience, so when you are talking about making the transition from the business model to the data model, it is a little academic at this point. We don't really have good business models to work with.

The second point is that the constraints of the higher level model must be carried over to the lower level model. When you look at the framework, it becomes clear if you take the owner's view of the business and transform that into the information systems model you don't carry over the constraints from the next higher level. You might as well not have produced the higher level model. Incidentally, this is a quality issue. When you make the transformations and if what you get at the bottom doesn't map all the way back up to the top, then you have a product quality problem. So, the constraints must be carried from level to level to produce the lower level model. Thus, if you derive the data model from a function cell, you are basically customizing the data column to the function column. That tells you that you are not going to make the data reusable. So, if you derive the data model from an adjoining cell rather than the cell above, then you are building customization into the data and not generalization. If you optimize the data design in the technology cell to the program or functional requirements, in effect you will be losing both the constraint and the reusability because you will customize from function to data at the technology level. You are losing the reusability because the data is derived from the function. So, when you are optimizing the data design based upon the current technology you will lose the results of the higher level models. And last, just because you use relational technology doesn't mean you can ignore design. When you use relational, and forget about the higher level models, then the quality and productivity from producing the higher level models is lost.

In any case, these are some conclusions about the transformation that I would make regarding the transformation of business model to the data model based upon the context of the information systems architecture.

In summary, business models are for <u>business</u> design and we have limited experience in this area. The constraints of higher level models must be carried over into lower level models or you might as well not bother to produce higher level models. If you derive the data model from a function "cell," you customize the result, it makes the data not re-usable. If you optimize the data design to the program (function) requirements, you lose both constraints and reusability. Just because you use relational technology doesn't mean you can ignore design. Relational is great, but it isn't magic!

## Reference

Zachman, J.A., "A Framework for Information Systems Architecture," IBM Systems Journal, Vol. 26, No. 3, 1987.

---

John A. Zachman is a consultant for IBM's Applications Enabling Marketing Center. He joined the IBM Corporation in 1965 and has held various marketing-related positions in Chicago, New York, and Los Angeles. He has been involved with Strategic Information Planning methodologies since 1970 and has concentrated on Information Systems Architecture since 1984. In 1989 he joined the CASE Support organization of the Applications Enabling Marketing Center where he continues his work on Information Systems Architecture.

Mr. Zachman travels nationally and internationally, speaking and consulting in the areas of Information Systems Planning and Architecture and has written a number of articles on these subjects. His current responsibilities include working internally with IBM as well as externally with IBM customers in supporting management with information systems.

Mr. Zachman holds a degree in Chemistry from Northwestern University. Prior to joining IBM, he served for a number of years as a Line Officer in the United States Navy and is a retired Commander in the U.S. Naval Reserve.

# A FRAMEWORK FOR INFORMATION SYSTEMS ARCHITECTURE

*A RESEARCH PROJECT*

*Presented by:*
*JOHN A. ZACHMAN*

Figure 1

## THE CLASSIC ARCHITECT'S PRODUCTS

(AS IN REAL ESTATE AND CONSTRUCTION)

| PRODUCT | NATURE/PURPOSE |
|---|---|
| "BUBBLE CHART" | ■ BASIC CONCEPTS FOR BUILDING<br>■ GROSS SIZING, SHAPE, SPATIAL RELATIONSHIPS<br>■ ARCHITECT/OWNER MUTUAL UNDERSTANDING<br>■ INITIATE PROJECT |
| ARCHITECT'S DRAWINGS | ■ FINAL BUILDING AS SEEN BY THE OWNER<br>■ FLOOR PLANS, CUT-AWAYS, PICTURES<br>■ ARCHITECT/OWNER AGREEMENT ON BUILDING<br>■ ESTABLISH CONTRACT |
| ARCHITECT'S PLANS | ■ FINAL BUILDING AS SEEN BY THE DESIGNER<br>■ TRANSLATION OF OWNER'S VIEW INTO A PRODUCT<br>■ DETAILED DRAWINGS -- 16 CATEGORIES<br>■ BASIS FOR NEGOTIATION W/GEN. CONTRACTOR |
| CONTRACTOR'S PLANS | ■ FINAL BUILDING AS SEEN BY THE BUILDER<br>■ ARCHITECT'S PLANS CONSTRAINED BY LAWS OF NATURE AND AVAILABLE TECHNOLOGY<br>■ "HOW TO BUILD IT" DESCRIPTION<br>■ DIRECTS CONSTRUCTION ACTIVITIES |
| SHOP PLANS | ■ SUB-CONTRACTOR'S DESIGN OF A PART/SECTION<br>■ DETAILED STAND-ALONE MODEL<br>■ SPECIFICATION OF WHAT IS TO BE CONSTRUCTED<br>■ PATTERN |
| BUILDING | ■ PHYSICAL BUILDING |

ZACHFRAME INF06

Figure 2

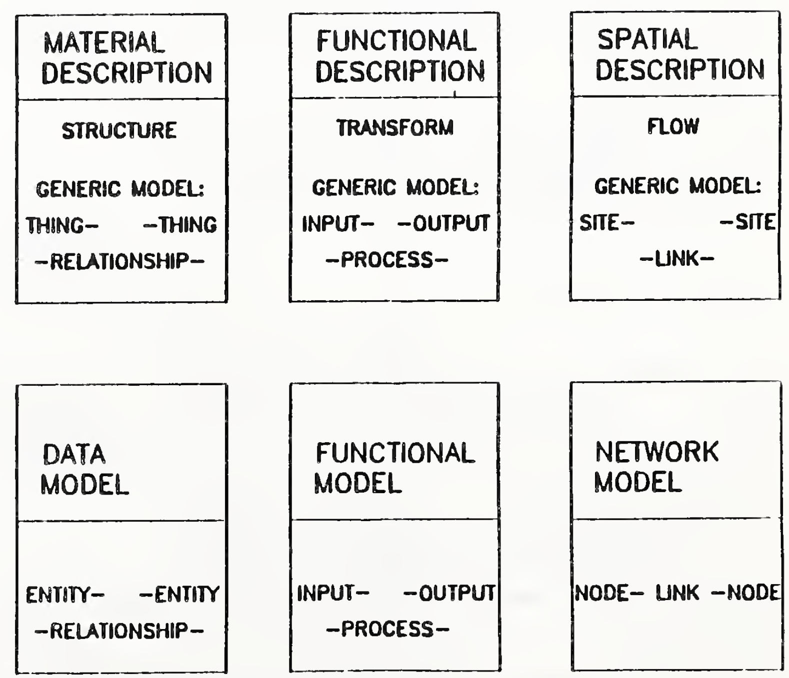
# ANALAGOUS CONCEPTS

| *BUILDINGS* | *SYSTEMS* |
|---|---|
| ARCHITECT'S PRODUCTS | SYSTEMS EQUIVALENTS |
| "BUBBLE CHARTS"<br>GROSS SIZING, SHAPE,<br>SPATIAL RELATIONSHIPS | OBJECTIVES/SCOPE<br>GROSS SIZING, SCOPE |
| ARCHITECT'S DRAWINGS<br>BUILDING AS SEEN BY OWNER | MODEL OF THE BUSINESS<br>"SYSTEM" AS SEEN BY USER |
| ARCHITECT'S PLANS<br>BUILDING AS SEEN BY DESIGNER | MODEL OF THE INFO. SYSTEM<br>SYSTEM AS SEEN BY DESIGNER |
| CONTRACTOR'S PLANS<br>ARCHITECT'S PLANS AS<br>CONSTRAINED BY NATURE &<br>AVAILABLE TECHNOLOGY | TECHNOLOGY MODEL<br>INFORMATION SYSTEM MODEL AS<br>CONSTRAINED BY AVAILABLE<br>TECHNOLOGY |
| SHOP PLANS<br>DESCRIPTIONS OF PARTS/PIECES | DETAILED REPRESENTATIONS<br>DESCRIPTION OF PARTS/PIECES |
| FUNCTIONING BUILDING | FUNCTIONING SYSTEM |

ZACHFRAME INF07

Figure 3

## DIFFERENT WAYS TO DESCRIBE THE SAME THING




ZACHFRAME INF12

Figure 4

# INFORMATION SYSTEMS ARCHITECTURE – A FRAMEWORK

| | DATA | FUNCTION | NETWORK |
|---|---|---|---|
| OBJECTIVES/ SCOPE | List of Things Important to the business<br>ENTITY = Class of Business Thing | List of Processes the Business Performs<br>Process = Class of Business Process | List of Locations in Which the Business Operates-<br>Node = Business Location |
| MODEL OF THE BUSINESS | e.g., "Ent/Rel Diag"<br>Ent = Business Entity<br>Reln = Business Rule | e.g., "Funct Flow Diag"<br>Proc = Bus Process<br>I/O = Bus Resources (Including Info) | e.g., Logistics Network<br>Node = Business Unit<br>Link = Business Relationship<br>(Org, Product, Info) |
| MODEL OF THE INFORMATION SYSTEM | e.g., "Data Model"<br>Ent = Data Entity<br>Reln = Data Reln | e.g., "Data Flow Diag"<br>Proc = Application Function<br>I/O = User Views (Set of Data Elements) | e.g., Distributed Sys. Arch.<br>Node = I/S Function (Processor, Storage, etc)<br>Link = Line Char. |
| TECHNOLOGY MODEL | e.g., Data Design<br>Ent = Segment/Row<br>Reln = Pointer/Key | e.g., "Structure Chart"<br>Proc = Computer Function<br>I/O = Screen/Device Formats | e.g., System Arch<br>Node = Hardware/Sys Software<br>Link = Line Specifications |
| DETAILED REPRESEN-TATIONS | e.g., Data Design Description<br>Ent = Fields<br>Reln = Addresses | e.g., "Program"<br>Proc = Language Stmts<br>I/O = Control Blocks | e.g., Network Architecture<br>Node = Addresses<br>Link = Protocols |
| FUNCTIONING SYSTEM | e.g., DATA | e.g., FUNCTION | e.g., COMMUNICATIONS |

INF29

Figure 5

Sample entity relationship model — Model of the business (owner's perspective) — Data column[3]

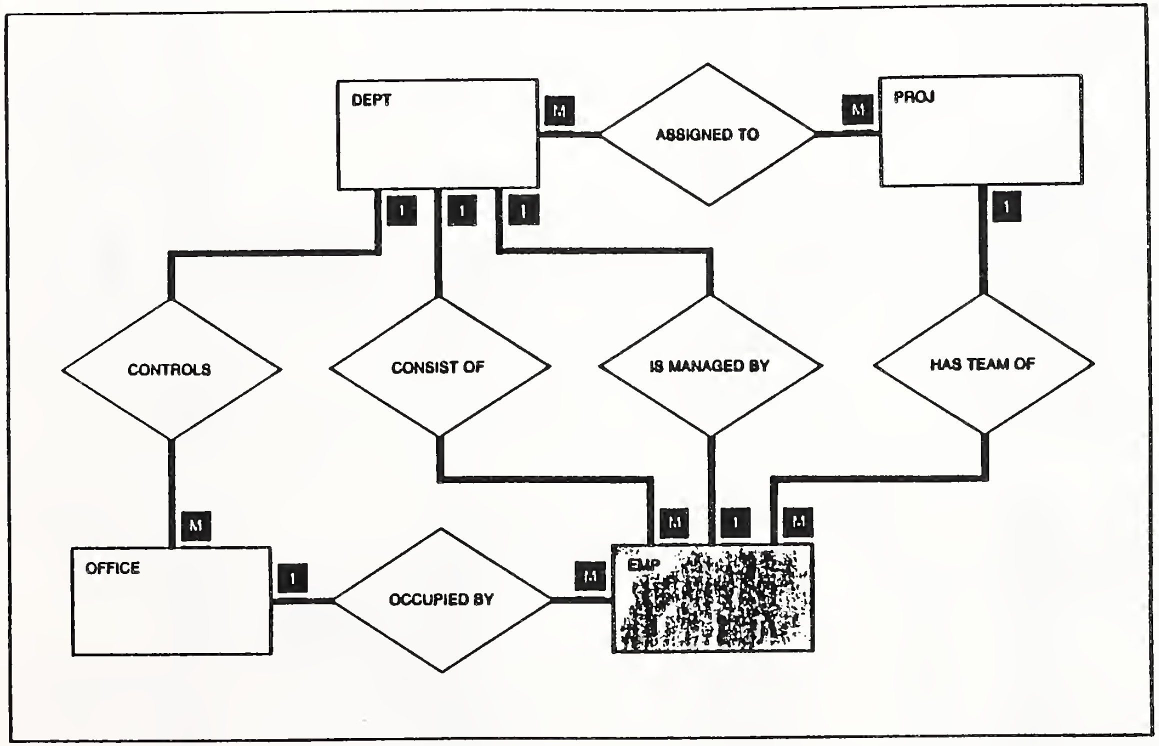



Figure 6

Sample conceptual data model — Model of the information system (designer's perspective) — Data column[3]

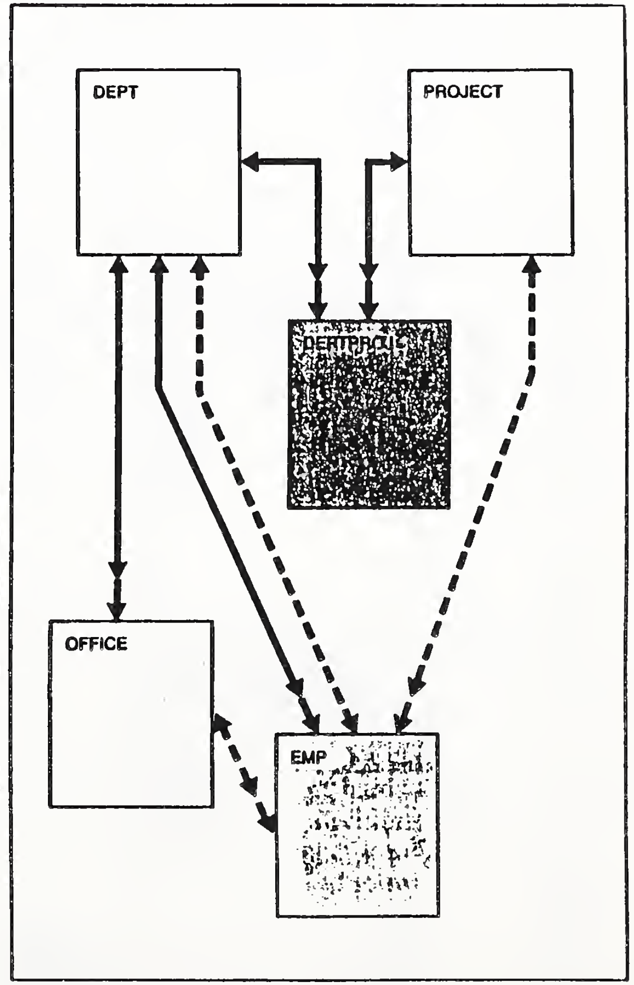
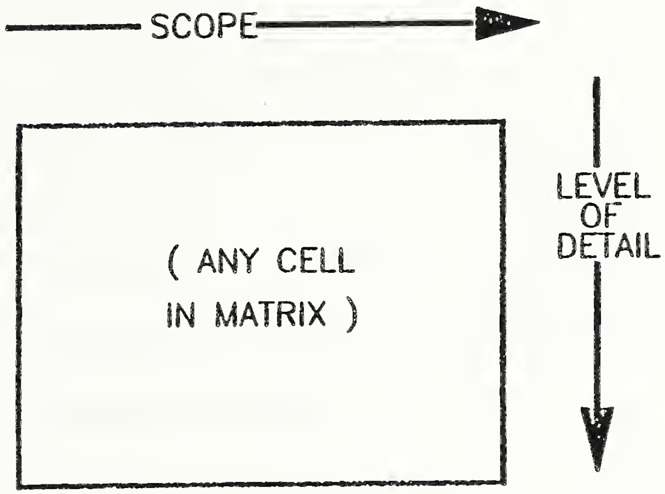



Figure 7

## CELL DIMENSIONS




SCOPE = PORTION OF THE ENTERPRISE ADDRESSED

LEVEL = DEGREE OF DETAIL

ZACHFRAME INF30

Figure 8

## TOOL/METHODOLOGY

### COMPARISON

*POTENTIAL TOOL/METHOD FUNCTIONS*

- DESCRIBE A CELL
- STORE CELL DESCRIPTION
- ANALYZE/REPORT ON DESCRIPTION
- GENERATE CELL DESCRIPTION
- MAP (MAINTAIN DESCRIPTIONS AND RELATIONSHIPS BETWEEN MULTIPLE CELLS)

*STARTER LIST OF TOOLS/METHODS*

E.G.

| | | |
|---|---|---|
| BSP | DD | DSSD |
| SADT | DMS | DDI |
| IDEF | BIAIT | BIPS |
| SSP | YOURDON | FOCUS |
| PDM-80 | DBMS | INFO. ENG. |
| ADF | PSL/PSA | ETC. |

ZACHFRAME INF31

Figure 9

## CONCLUSIONS

*A FRAMEWORK FOR:*

- CLARIFYING PROFESSIONAL COMMUNICATIONS
- IDENTIFYING REASONS AND RISKS
- POSITIONING TOOLS AND/OR METHODOLOGIES
- CONCEPTUALIZING IMPROVED APPROACHES

Figure 10

ZACHFRAME INF32

## MANAGEMENT IMPLICATIONS

- "ARCHITECTURE" IS NOT GOING AWAY
- THE ROLE OF I/S IS LIKELY CHANGING
- NOTHING WILL HAPPEN WITHOUT OVERT DECISIONS

Figure 11

ZACHFRAME INF33

# STANDARDS: ROLE OF DATA STANDARDS

# IN ESTABLISHING A DATA QUALITY PROGRAM

SESSION CHAIR

Thomas McKnight
American Management Systems

PANELISTS

Brad Ellis
John McGuire
Joan Monia
Gail Gorge

# ROLE OF STANDARDS IN ESTABLISHING A DATA QUALITY PROGRAM

Brad Ellis
McDonnell Douglas Corporation

Good Morning. I am Brad Ellis with McDonnell Douglas Corporation (MDC) and I am delighted to be here this morning to discuss the role of standards in Establishing a Data Quality Program. McDonnell Douglas is a large international, multi-divisional company. Within McDonnell Douglas I am responsible for the Corporate Data Management Architecture Program.

My mission is to define a design for managing and integrating MDC information resources. In other words, define a design to achieve Data Quality within the Corporation. This is supportive of a broader Corporate objective of implementing a Total Quality Management System (TQMS) within McDonnell Douglas. This TQMS Program supports implementation of the team concept, reduces levels of management, provides horizontal integration, and avoids vertical silos. The change in process/organization is absolutely essential before an effective Data Quality program can be established.

In order to understand the role of standards in establishing a Data Quality Program one must understand where we have been and where we are going with Data Management. Once this is understood, then the only conclusion is that data standards must be in place for the enterprise to enable a Data Quality Program (fig. 1).

Where have we been with Data Management? Interdependent organizations direct resources and labor to provide automation individually resulting in significant data redundancy, inconsistency, and translation (fig. 2). Data Management is characterized by slow evolution and dominant vendors, with unique proprietary interfaces (definition and manipulation). The portability and distribution of applications and data is limited (host based). Engineering and manufacturing implementations are independent.

In the future (fig. 3), Data Management is characterized by data managed top-down through data planning and business planning integration. Cooperative data administration among business functions utilizes common modeling methods to support data integration. Cooperative dictionaries (IRDS/Protocols) among business functions support the sharing of definitions for company products, tools, processes, and other business resources via a common data model (IDEF1X, EXPRESS, NIAM, 3 Schema). Common data models are also used to understand the business.

In addition, application development and procurement are driven by models that carry semantics, rules, and operations. Cooperative data administration and database administration occur to maintain data integrity and support multiple processing requirements. Requestor-server technology is enabled through remote database access (RDA) standards and a standard application program interface (SQL). As this technology, along with data exchange formats such as PDES, supports physical integration across processing platforms, greater utilization and integration of distributed computing resources are accomplished. Relational and object-oriented DBMS technologies, via a uniform SQL interface, support CAD/CAM/CALS and business process integration. The SQL interface will also allow access to IMS data.

Standards (fig. 4) are the key to enable this future vision of Data Quality. There are two aspects to maintaining the quality of data in the to-be state. The <u>first</u> is managing the definition, structure, and integrity constraints placed on data. These control elements must be managed throughout the information systems life cycle. Integrity constraints will place limits on the changes that may be applied to data. Within these constraints, users and user applications will have the flexibility to manage data content. A <u>second</u> aspect of data quality is data's usefulness in responding to a requirement. Data structure and data semantics must be complete and must support the requirements of the user. Standards (both Enterprise unique and Industry) are the only way to assure Data Quality in this new environment.

The scope of standards (fig. 5) must include: physical environment (Storage, Data Base Machines etc.), tools (Data Modeling, Data Base Design etc.), Methodologies (SDLC to support Data Quality), administrative functions (DA, DBA, etc.), and ANSI/ISO (SQL, PDES, Three Schema, etc.).

Standardization is the only way to enable Data Quality within the Enterprise. However, as stated earlier, processes and cultural changes need to be made before this new environment can be achieved.

------------------------

Brad Ellis is Senior Principal Specialist at McDonnell Douglas. He is currently manager of the Data Management Architecture Corporation. He has had a wide range of experience including Data Base Administration, Dictionary Administration, Data Administration, System Development, and

Data Architectures. He is past Division Manager at GUIDE International (an IBM users group) responsible for GUIDE's activities in Data Administration, Data Dictionary and Architecture. He has over 22 years experience in Information Technology.

## <u>DATA QUALITY ASPECTS</u>

- <u>Managing</u>
  - Definition } Across System
  - Structure } Development
  - Integrity Constraints } Life Cycle
- Data <u>Usefulness</u> in Responding to a Requirement

Figure 1

## DATA MANAGEMENT ENVIRONMENT

*WHERE HAVE WE BEEN:*

- Heterogeneous DBMS platforms inhibit capabilities to integrate data and applications
- Project unique internally developed dictionaries
- Proprietary interfaces for data definition and manipulation
- Application development without common data model
- Interdependent organization direct resources to provide automation individually

Figure 2

## DATA MANAGEMENT ENVIRONMENT

*WHERE ARE WE GOING:*

- Data is managed across business functions
- Common data modeling methods will support integration of data and applications
- Data Administration functions established
  - cooperation between business functions
  - provides direction for database administration
- Cooperative dictionaries based on ANSI IRDS (Information Resource Dictionary Standard) established
- Application development/procurement driven by data models
- Integration of applications within homogeneous platforms enabled by requestor/server technology
- Relational DBMS (Database Management Systems) is platform
- SQL (Structured Query Language) is interface
- Common data exchange formats facilitate data translation - Corporate solutions, PDES - still maturing
- Old DBMS platforms are stabilized

Figure 3

Standards are the key to enabling the "to be" environment

Scope

- Physical Environment
- Tools
- Methodologies
- Administrative Functions
- National./International Standards

Figure 4

## EXAMPLES OF DATA QUALITY "SUBJECTS" SUPPORTED BY STANDARDS

- Data Administration
- Data Planning
- Three Schema
- Common Data Model
- Data Dictionary/Directory
- Data Sharing Tool Set
- Legacy Data Interface (Inventory & Semantics)
- Security Administration
- Software Development Integration
- Data Base Administration
- Physical Data Base Design
- Data Distribution
- Knowledge Base Integration
- Relational Database Systems
- DBMS Firmware
- Object DBMS
- MDC SQL Guidelines

Figure 5

# ROLE OF DATA STANDARDS IN ESTABLISHING A DATA QUALITY PROGRAM

John McGuire
Department of Health and Human Services

## BACKGROUND

Health Care Finance Administration (HCFA) is one of the five major operating components of the Department of Health and Human Services (DHHS). The agency is responsible for the management of the medicare and medicaid programs. Medicare is a Federal health insurance program for the aged, disabled, and persons with end-stage renal disease. To summarize, medicare covers hospital, physician, laboratory, skilled nursing, and home health services, and is administered through private fiscal agents called intermediaries and carriers. Medicaid is a Federal/State program which finances health care for certain low income individuals and their families. States administer the medicaid programs within Federal guidelines.

These two programs, enacted by Congress in 1965, provide health insurance coverage for 33 million aged and disabled individuals plus 24 million beneficiaries eligible for Aid to Families with Dependent Children (AFDC) or Supplemental Security Income (SSI). Thus, HCFA touches the lives of more than 50 million Americans (1 in every 5) -- or about 22 percent of the total population of the United States.

With such a large volume of beneficiaries, it is necessary to collect a massive amount of data to administer the medicare and medicaid programs. This data must be standardized to produce accurate and usable information. The Bureau of Data Management and Strategy (BDMS) is the data processing component for HCFA. As part of its functions, BDMS manages the medicare and medicaid data collection systems. The information collected provides the basis of the statistical system designed to measure the effectiveness of the programs. We are currently involved in an effort known as PRISM (Project to Redesign Information Systems Management) to redesign this statistical system.

## HCFA'S DATA FLOW

Before I discuss how HCFA is standardizing its data under PRISM it would be helpful if you understood the types of data that we collect and the flow of that data.

The attached chart (fig. 1) is somewhat simplified; however, it depicts the fundamental sources and uses of data which BDMS converts to information. For manageability, the data systems are divided into four logical applications groups (LAGS): Health Insurance/Supplementary Medical Insurance (HI/SMI) systems, medicare/medicaid decision support systems (MDSS), program management systems and administrative systems.

The principal source of medicare entitlement data is the Social Security Administration (SSA). BDMS maintains entitlement status on approximately 33 million beneficiaries. About 10 percent of the file represents the records of deceased beneficiaries whose billing records remain active.

The primary sources of medicare utilization data are the providers who generate claims and bills. These claims and bill data are passed through the contractors, i.e., fiscal intermediaries and carriers -- the organizations that provide services to medicare beneficiaries and adjudicate the claims on HCFA's behalf. The data then pass to the HI/SMI systems where they are merged with records containing entitlement status.

The data derived from the medicare entitlement and utilization/billing records form the primary source of data for the Medicare Decision Support Systems (MDSS). Data from the MDSS are processed to support actuarial reports, program development and policy analysis, cost projections for legislative proposals, payment rate analysis and development, research studies and demonstrations.

The state agencies are the principal source of medicaid data. Much of this data flows through our regional offices to provide input to HCFA's medicaid statistical systems and program management systems. The medicaid statistical systems receive paid claims and eligibility data from the state agencies which are used to support activities similar to those using medicare data. Data sent to the program management systems relate to accreditation, cost, budget, and workload.

The medicare contractors also provide management data in the form of costs, budget, workload, and cash flow, generally through our regions, to the program management systems. Data in these systems are converted to cash flow reports, budgets, and trust fund reports.

Administrative data including personnel, payroll, and non-program budget and cost reporting is synthesized centrally in HCFA's administrative systems. Most of these systems are supported by our Office of Budget and Administration.

Standards

In general, HCFA is interested in four levels of standards for data collection and storage - industry-wide standards, department-wide standards, agency-wide standards, and HCFA's internal system standards.

Industry-wide standards - To comply with the health care industry's standards, we made a commitment to use the Standard Form UB-82 developed by the American Hospital Association, along with the Health Insurance Association of America, Blue Cross and Blue Shield Association, HCFA and other interested parties. This form incorporates definitions and coding conventions designed to be used by hospital providers of medical care to bill all third-party payers (insurance companies, etc.). ICD-9-CM, the International Classification of Diseases, Version 9, Clinical Modification is used on the form for reporting diagnosis and procedures. ICD is a coded classification system developed by the World Health Organization and followed by many countries. Use of this standard coding system allows for easy exchange of disease classification data.

HCFA is also committed to the use of a standardized physician billing form (which we call the HCFA-1500) that was developed by the American Medical Association (AMA), HCFA, and the insurance industry. The form uses the standard AMA-developed procedure codes known as CPT-4 to which HCFA adds codes for describing medicare-covered medical procedures performed by professionals other than physicians.

Use of these industry-wide forms and codes lessens the cost and burden on the providers/suppliers, reduces errors for the claims processors, and provides a more usable and accurate database for research, marketing, and for managing the involved insurance programs.

Department-wide Standards

The Department of Health and Human Services has developed and continues to modify minimum uniform health data sets. These are sets of core data elements, uniformly defined throughout the department, which are collected through the department's operational or research activities. The original and most successful of these data sets is the Uniform Hospital Discharge Data Set (UHDDS). The UHDDS is collected by HCFA on the UB-82. The data set includes items such as personal identification, date of birth, sex, race, admission and discharge dates, etc. Collection of these data elements by all agencies in the department allows for the necessary sharing and comparison of health service information.

Agency-wide Standards

Built into HCFA's programs are many standards that affect the quality and scope of our data. These include national standards for beneficiary eligibility, health service provider certification, and fiscal intermediary and carrier performance. We have issued unique, standardized identifiers to beneficiaries, providers, intermediaries, carriers, and in the near future to physicians. In addition, we have developed and require the use of standardized computer specifications by our contractors and providers to ensure that beneficiary utilization data are consistent throughout the country. We also perform uniform edits on these data.

HCFA-Internal Systems Standards

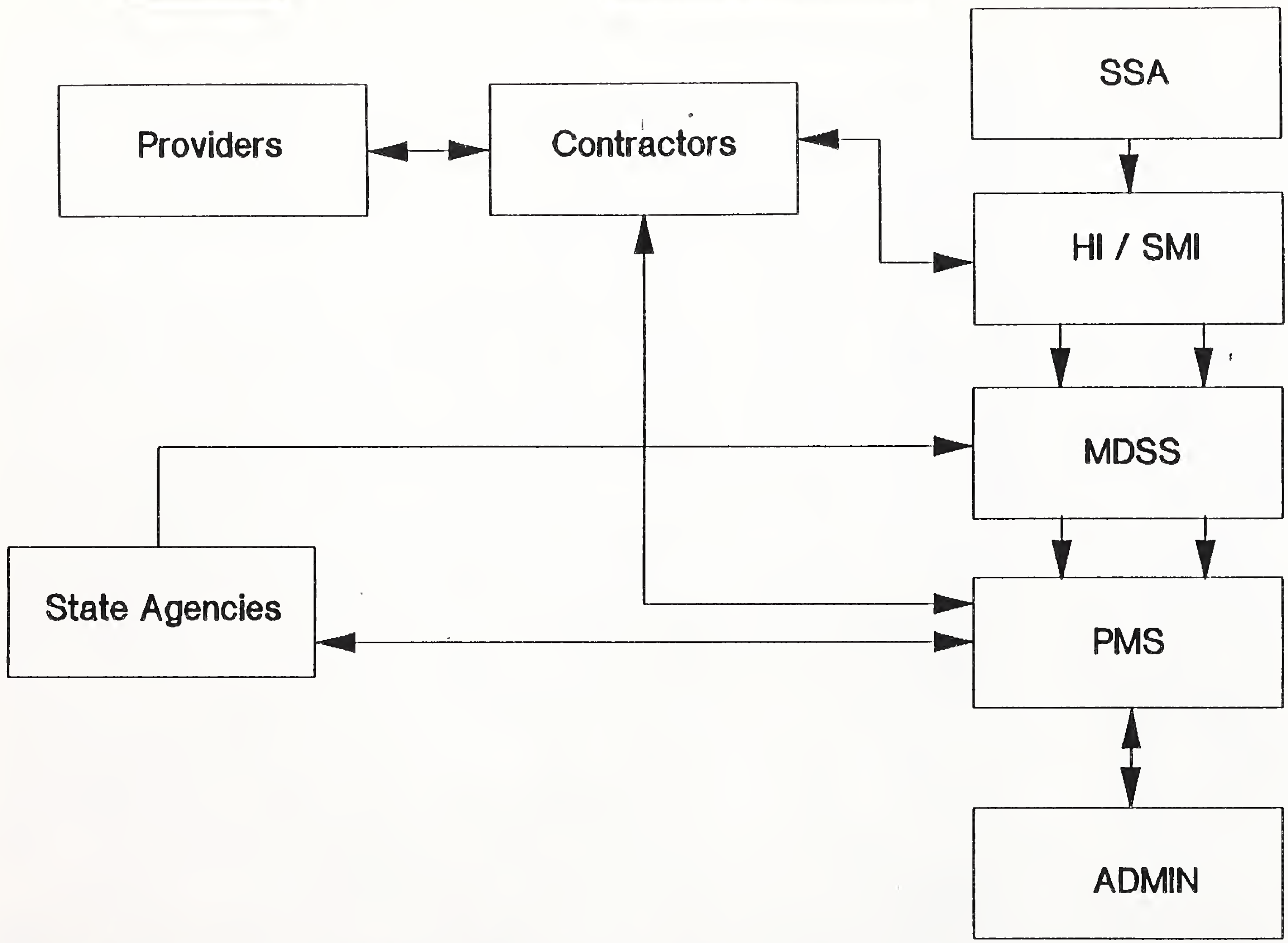
The development of HCFA internal systems standards is a relatively new function for our agency. The data administration branch, which performs this function, was created about one and a half years ago. Under our new redesign effort (PRISM), we will be moving from a bottom-up programming approach programming to a top-down, integrated database management system environment. PRISM is following a structured design methodology.

We are working with our PRISM contractor in creating a data dictionary that will eventually serve as the basis for the corporate dictionary that will be used by the agency for our redesigned systems. The contractor has followed standard naming conventions which we developed for all data elements that were identified as needed in our current and future systems. The data names must contain a standard class word in the final position, at least one key word, and all necessary modifiers. A list of approximately 3,000 approved abbreviations and acronyms was developed by HCFA and is being used in creating the names. When additional abbreviations/acronyms are necessary in the systems modeling process, approval is given by the Data Administration Branch (DAB) and the acronym/abbreviations is added to the official list. DAB has also identified standard attributes and requires that values for these attributes be retained for each element. The elements and associated attributes are reviewed by DAB and the particular LAG representatives associated with the element. After approval is received from the LAG, elements that cross LAG boundaries are reviewed by a joint review committee made up of representatives from all LAGS and DAB. The JRC approval is considered the agency-wide approval on the element. The JRC is also reviewing agency-wide issues. These issues involve topics such as the standard use of specific terms, (e.g., facility vs. provider) the logical representation of dates on an agency-wide basis, standard field lengths for items such as names or addresses, standard values for codes or tables of code, and the ambiguous use of words (e.g., use of "PATIENT" by the HI/SMI LAG and the MDSS LAG).

Since we are just in the process of establishing the standards for data processed in our internal systems, I cannot give any concrete evidence as to the benefits of this effort; however, we have confidence that going through an effort to rename data elements will produce substantial benefits to our program.

SUMMARY

As you have seen, HCFA is very accustomed to using standards. Regardless of the level of their implementation, all standards that are established place us one step closer to maintaining a program with quality data that are easily accessible and usable by us, as well as by other Government agencies, the industry, and the public.




# OVERVIEW OF HCFA DATA FLOW

Session 1: "Role of Data Standards in Establishing a Data Quality Program"
John McGuire, Department of Health and Human Services

# INSTITUTIONALIZING DATA ADMINISTRATION

Joan Monia
GTE Government Systems

Institutionalizing Data Administration involves management of infrastructure change across the enterprise. In the 1990's the only constant will be change. Change in international politics, monetary alignment and international business organizations is accelerating.

Operation of Federal agencies, the Department of Defense, computer systems and even stock markets of different countries are interrelated in such a way that metrics used traditionally to predict future trends are no longer reliable. Changing metrics without an appreciation for impact of the international environment causes enterprise crisis.

Absence of methods and strategic systems which adapt to change impact an organization's ability to respond to the dynamics of global change. The rate of global change exceeds the lead time to implement changes in traditional computer systems.

What is required are systems which can be adjusted quickly for all in the enterprise with features that support management requirements of the 1990s. Rapid response requires rapid decision making by selected management. Communicating through a hierarchical organizational structure impedes timely response. Procedural change lags policy change. Uncoordinated procedures across an organization can cause inadequate response and subsequent failure.

The scope of Data Administration to address the problems in the 90's (figure 3) extends to all units of the organization, and is no longer limited to Information Services. The management of the Information Resource involves infrastructure change across the entire organization. The intelligence of the collective organization must be networked across levels of management (figure 4). Translating that intelligence into information suited to each organizational unit's action must be done under algorithms which sense, translate, and distribute the appropriate information to decision makers.

Methodology and organization (figure 5) to achieve this defines Information Resource Management as an infrastructure of roles throughout an enterprise. Models of the Policy of the enterprise related to its Functions, Objects and Events are built and maintained by those supporting the primary objectives of organization units. These models are then translated into models indicating content and placement of Functions, Data, and System Entry points and relating these models to the supporting computerized devices. Models of optimized computer components are derived from the models of content and placement. The roles of those performing the analysis and design are shown. Intersections between roles show direct relationships in information exchange.

There are several unique features about the methodology which support its use in the management of infrastructure change. Among these are:

A template of generic transforms for Functions;

- A layer containing generic models of Functions, Objects, and Events which reflect the policy of the enterprise;

- A set of methods for translating the generic models of the enterprise into Entity-Relationship models, Data Flow Diagrams, State diagrams and other models related to different implementation technologies;

- A set of methods for quality assurance across model types.

With a template for analysis of organization functions (Figure 10), analysis of an organization can proceed independently across an enterprise and yet be integrated to form a cohesive model of the policy level. The template is used to assess an operation as well as to specify a proposed version of it. The template identifies transforms which are basic to human processing of information and types of data stores which are basic to dynamic processing of information: Profile (or Dictionary), Directory, and Data.

The functions of an organization are called out by the type of transform entailed. The template calls for certain attributes of data needed to perform the individual transforms and does not imply sequence of use. Objects and attributes are identified and related in a separate model, the Object Model. Event sequences are identified and related to attributes of objects in the Event Model.

By separating the three models at the policy or Concept of Operation level, objects about which data is collected, essential transforms basic to a function, and various event sequences may be well understood before combining them into automated procedures.

Another benefit is that these models form a model of the data needed in a dynamic generic strategic planning system. Databases aligned toward strategic planning (Object Databases) can be used to map the individual tactical and operational databases (Subject Databases) into the strategic plan. In fact, the data of such a system is the basis for automated distribution of intelligence about the Information Resource through Information Resource Dictionaries on devices across the enterprise. The models become the policy standards on which the enterprise operates. When this state occurs, Data Administration is no longer relegated to MIS.

The final component of institutionalizing Data Administration is a program (Figure 12) to evolve toward an infrastructure supported by automation to address the issues of the enterprise. Organizations in crisis afford the best opportunity for building infrastructure in the early stages of institutionalizing Data Administration. Facilitation of cross training can be done: by having operational personnel participate with the technical modeling personnel; by workshops given by modeling personnel to operational personnel; by joint participation in policy and procedure development for the organization; by jointly developed prototypes of automated systems; and ultimately by recognition of personnel who find or develop a "better method" which can be incorporated in the Information Resource Management Methodology. When the analysis brings results directly related to enterprise objectives and not just to MIS, Data Administration will be driven by the strategic management of the enterprise as a needed infrastructure for survival.

The final result: organizational commitment for institutionalizing Data Administration.

---

Joan Monia is a senior member of technical staff in GTE Government System Corporation's WIS Division where she has contributed to DoD data administration policy and to an IRDS oriented translator for workstation to mainframe control and administration support. She has been in the forefront of information management technology since her pioneering work on Library Bibliographic Search and Selective Retrieval. Her contributions include specification of the commercially successful dictionary, Data Catalog; an integrated data base design of the first documented enterprise-wide database system; and major contributions to the dictionary centered VAX Information Architecture. These efforts also paralleled her development of data management functions from data administration to data and information resource management while with Digital Equipment Corporation.

# INSTITUTIONALIZING DATA ADMINISTRATION

JOAN MONIA

MAY 3, 1989

GTE GOVERNMENT SYSTEMS

Figure 1

INSTITUTIONALIZING DATA ADMINISTRATION

## OVERVIEW

- PROBLEMS IN THE '90S
- SCOPE
- METHODOLOGY AND ORGANIZATION
- FRAMEWORK
- INFORMATION TEMPLATE
- IMPLEMENTATION
- PROGRAM MANAGEMENT PROCESS

Figure 2

GTE GOVERNMENT SYSTEMS

INSTITUTIONALIZING DATA ADMINISTRATION

## PROBLEMS IN THE 90'S

- RAPIDITY OF GLOBAL CHANGE
- UNRESPONSIVE METRICS
- UNRESPONSIVE DECISIONMAKING
- ABSENCE OF STRATEGIC SYSTEMS
- TECHNOLOGICAL LEAD TIME IN MATURE BUSINESSES
- MANAGEMENT STYLE OF MATURE BUSINESSES

GTE GOVERNMENT SYSTEMS

INSTITUTIONALIZING DATA ADMINISTRATION

Figure 3

## SCOPE

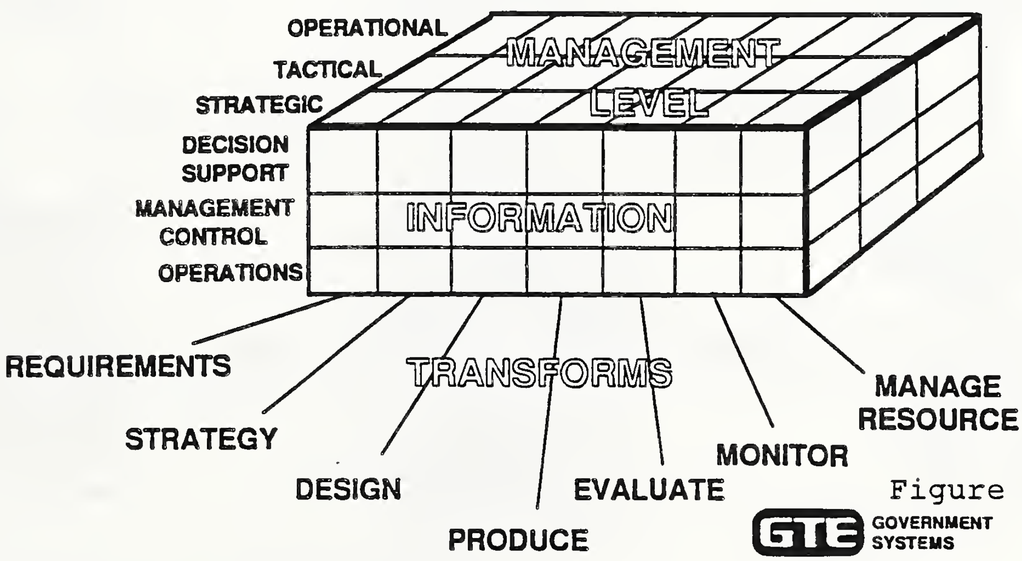



Figure 4

GTE GOVERNMENT SYSTEMS

# METHODOLOGY AND ORGANIZATION

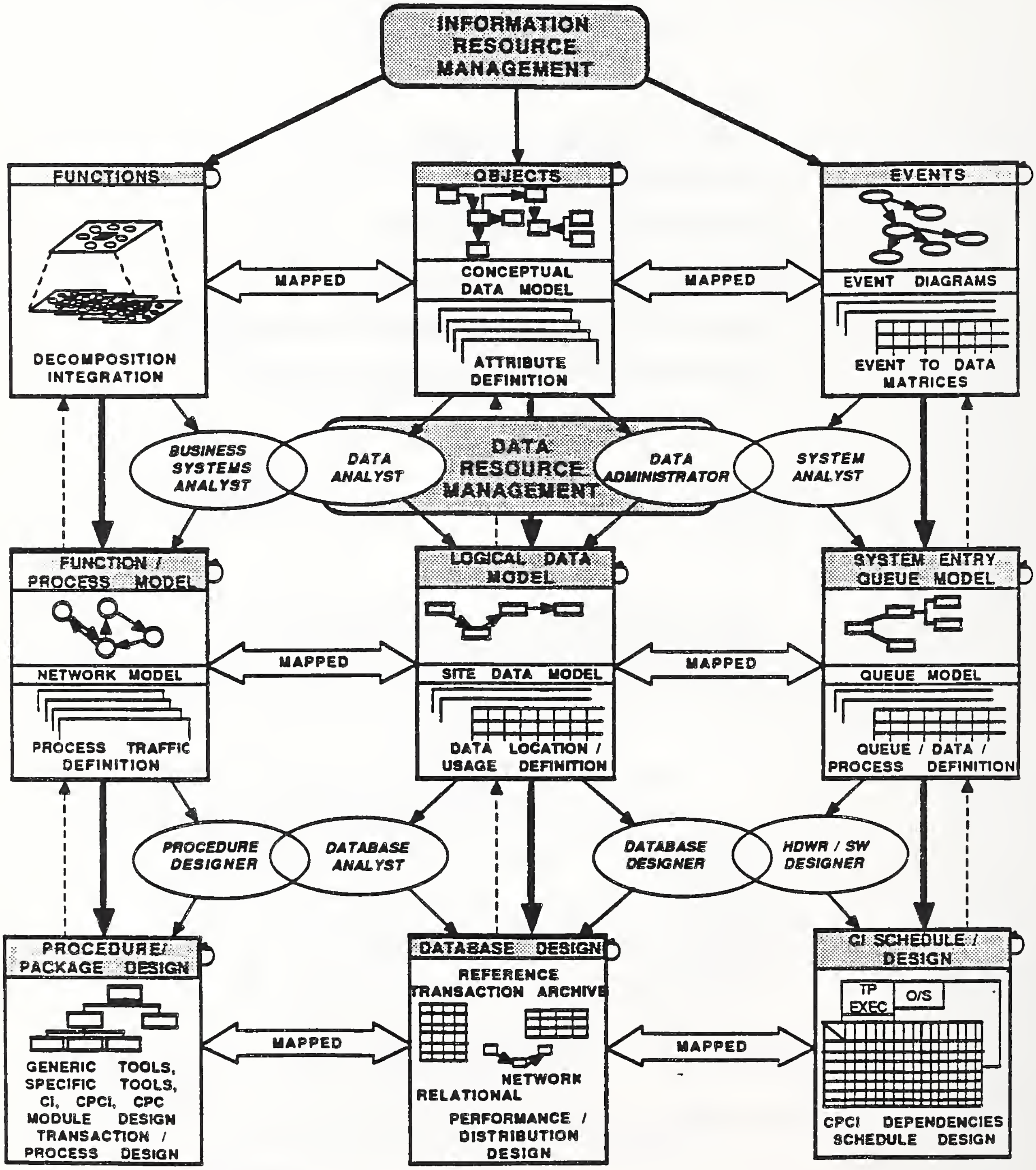


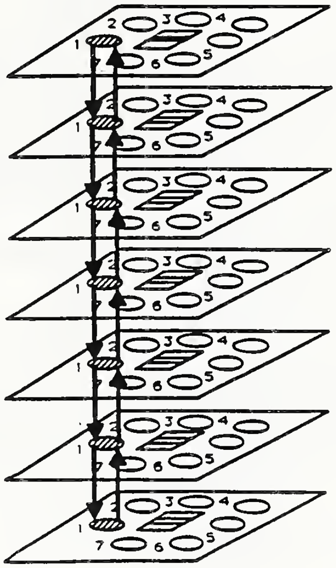

Figure 5

GTE GOVERNMENT SYSTEMS

Joan Monie
March 89

INSTITUTIONALIZING DATA ADMINISTRATION

## OPERATIONAL

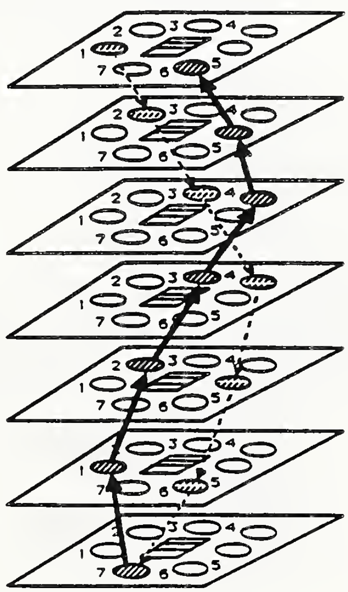



GTE GOVERNMENT SYSTEMS

Figure 6

INSTITUTIONALIZING DATA ADMINISTRATION

## TACTICAL

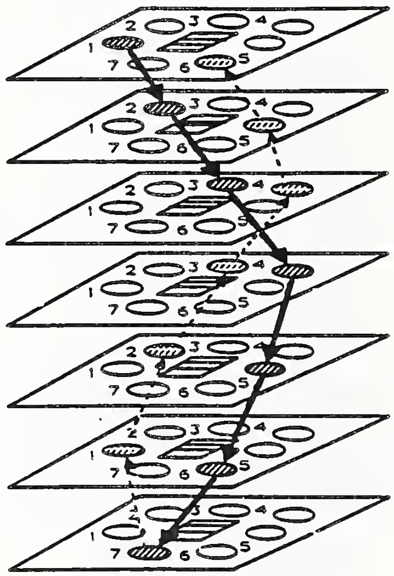



GTE GOVERNMENT SYSTEMS

Figure 7

INSTITUTIONALIZING DATA ADMINISTRATION

## STRATEGIC

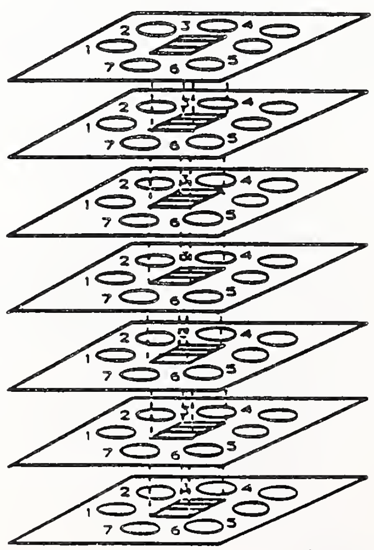


GTE GOVERNMENT SYSTEMS

INSTITUTIONALIZING DATA ADMINISTRATION

Figure 8

## INFORMATION RESOURCE DICTIONARY



GTE GOVERNMENT SYSTEMS

Figure 9

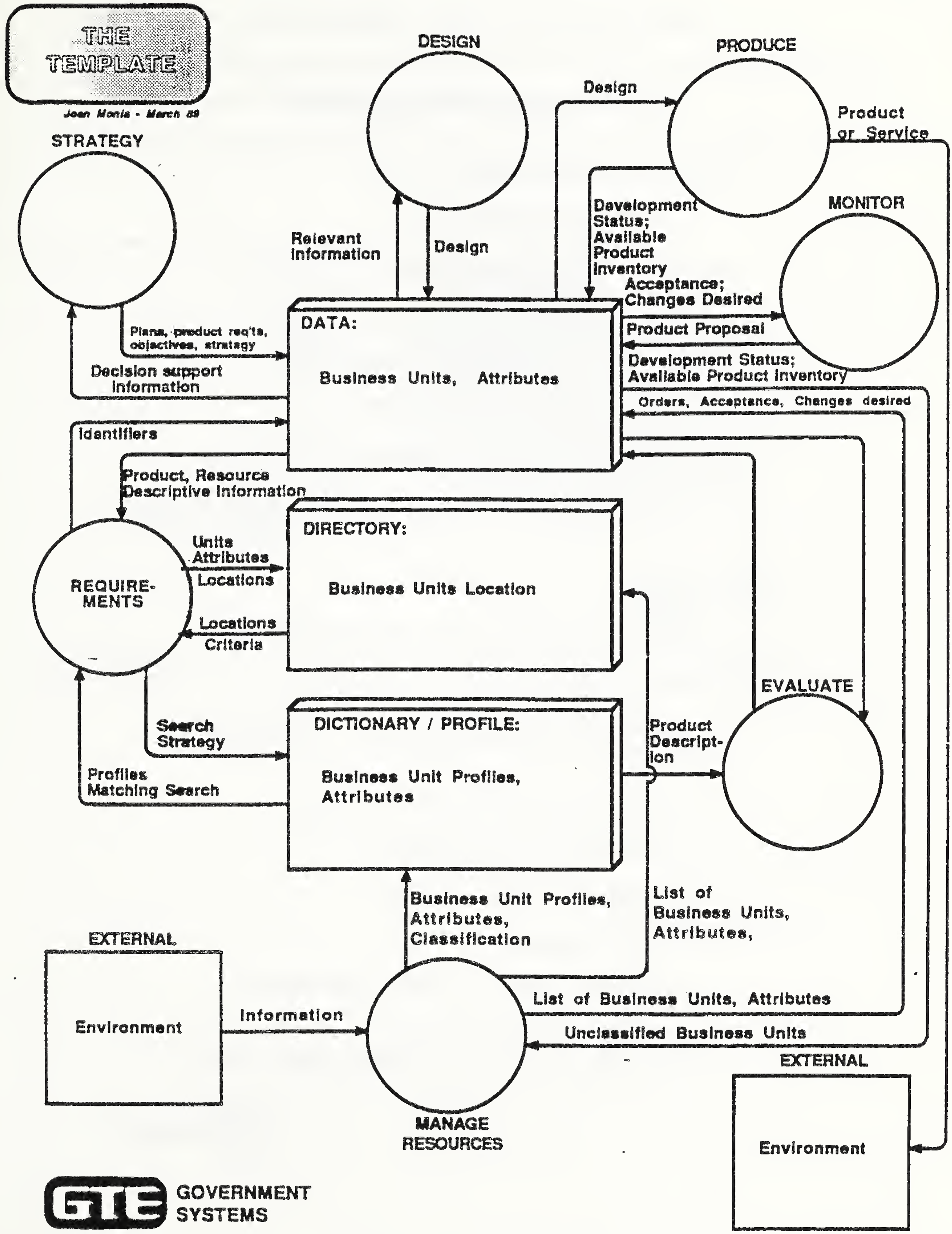

THE TEMPLATE
Joan Monia - March 89
STRATEGY
DESIGN
PRODUCE
MONITOR
Design
Product or Service
Development Status; Available Product Inventory
Relevant Information
Design
Acceptance; Changes Desired
Product Proposal
Development Status; Available Product Inventory
Plans, product req'ts, objectives, strategy
Decision support information
DATA:
Business Units, Attributes
Orders, Acceptance, Changes desired
Identifiers
Product, Resource Descriptive Information
DIRECTORY:
Business Units Location
Units Attributes
Locations
Locations
Criteria
REQUIRE-MENTS
EVALUATE
DICTIONARY / PROFILE:
Business Unit Profiles, Attributes
Search Strategy
Profiles Matching Search
Product Description
Business Unit Profiles, Attributes, Classification
List of Business Units, Attributes,
EXTERNAL
Environment
Information
List of Business Units, Attributes
Unclassified Business Units
EXTERNAL
Environment
MANAGE RESOURCES
GTE GOVERNMENT SYSTEMS

Figure 10

## PHASED IMPLEMENTATION

- OPERATION ASSESSMENT
  - OPERATION STRATEGY
    - ORGANIZATION DESIGN
      - ORGANIZATION DEVELOPMENT
        - MONITORING PROCEDURE
          - EVALUATION
            - RESOURCE MANAGEMENT

GTE GOVERNMENT SYSTEMS

Figure 11

INSTITUTIONALIZING DATA ADMINISTRATION

## PROGRAM MANAGEMENT PROCESS

- JOINT BUSINESS / MIS MANAGEMENT
- FACILITATION PARTNERSHIP
- PROCEDURE DEVELOPMENT
- PHASE PLANNING
- OBJECT / FUNCTION / EVENT INTEGRATION

THE RESULT: ORGANIZATIONAL COMMITMENT

GTE GOVERNMENT SYSTEMS

Figure 12

## SUMMARY

- REQUIREMENTS
- STRATEGY
- DESIGN
- PRODUCE
- MONITOR
- EVALUATE
- MANAGE RESOURCE

***RESULT:***

***INSTITUTIONALIZATION OF DATA ADMINISTRATION***

GTE GOVERNMENT SYSTEMS

Figure 13

# THE DATA QUALITY PROGRAM AT PERPETUAL SAVINGS BANK

Gail Gorge
Perpetual Saving Bank

We are both members of the Mission Impossible team of system development and maintenance.

Picture the project team - interacting with the Data Administration and Quality Assurance. Quality Assurance pushing for definition of functional processes, Data Administration grabbing for data definitions. Each trying to play their part in this scene of political drama.

But this Mission Impossible political situation has only reached this point because management didn't plan carefully and prepare the necessary work leading up to the change. We know this never happens in your environment, right.

Anyway, back to our story. Quality Assurance asks for the project plan or a list of activities. Data Administration asks for data definitions, data flow diagrams, and business rules. The project leader asks for extra strength aspirin and a glass of water. Everyone talks at once saying how unorganized this project is.

Then, a calm comes over the group - cue the Mission Impossible music and a PLAN emerges. But as in mission impossible, no one knows what is going to happen next except the project team. You know and I know that everything always comes out OK on TV, but what about in real life? Commando units couldn't save some of these projects.

I'm here to talk about quality data. The real key to getting quality data is the way in which we get our information about the data, or data management. Data Management to me is more than simply doing data definitions and data structures. Managing data means setting quality parameters for the data. These quality parameters are the acceptable data criteria set by the business needs.

Typically, we depend on the analyst and project team members to define our data attributes as part of the interview process in the requirements phase. This is probably our first problem area because everyone knows that it is very hard to remember and relate everything in interviews. Even after definitions have been published and data flow diagrams have been drawn, things get left out or aren't always totally accurate. The acceptable data criteria often don't get established correctly if at all.

We depend very heavily on the analyst and client users to define business rules and related procedures that give us the acceptable data criteria. Then, in typical projects we ignore these criteria in the reiterations and changes that take place in the next phase - DESIGN. Our Data Dictionaries usually capture changes to element definitions but what about the effects of the changes on the business needs, do they really matter?

The point is that many times we don't go back and review the impact on the original data requirements. If the data definitions comply with our data standards and our naming conventions, what else matters? Data standards are very good and necessary but are designed to ensure consistency in form on a general basis. It is easy to get lost in compliance to these types of standards and miss the acceptance criteria/business issues. (I'm not trying to get the rotten egg award - in the heat of the moment, it is easy to overlook the not obvious but necessary review.)

Now, here is where a part of Quality Assurance can help out. Between reviews, inspections, walk throughs, etc. many of these missing items come out. No news there, we've known that for years - so what?

Well, this depends on how the changes and additions to these items are handled. Too often this is handled lightly and not documented well, if at all. One of my responsibilities in Quality Assurance is to ensure that we manage our changes in projects. People often document changes to requirements and process definition changes but what about data related changes? Who is responsible for managing the data changes?

These changes to data do make a difference. They impact test results, user acceptance test criteria, procedures (manual and automated), etc. How many times does a program fail or system loop because of a business rule definition gone bad? Quality Assurance is recognizing this effect and is trying to find a way of tracking and measuring the occurrence. This is not a simple, straightforward measure. Quality Assurance is attempting to define aspects of data quality.

What this really comes down to is that the cross over between Data Administration and process (functional) definition comes out in the business rules and data flow diagrams. Changes to these are critical and need to be documented and incorporated into all aspects of system change. Quality systems require a harmony of controlling changes to data definitions (data management) and related data information, as well as requirements and functional analysis. Quality data will result from ensuring that data definitions, descriptions, usage, etc., meet the acceptance criteria established by the business need or requirement.

Our Mission, should we decide to accept it, is to find a way to make sure that these changes are recognized, documented, incorporated into the system and tracked/measured for impact.

---

Ms. Gail Gorge has had over 16 years working with developing quality processes in changing environments with 12 years in data processing related Quality Assurance and Data Quality Control. She has implemented Quality Assurance policies and procedures in five different data processing environments which ranged in size from very small to data processing departments of 450 people. She initiated the Data Administration function at the American Automobile Association, bringing in the concept of the corporate data dictionary, as well as corporate information sharing.

Having worked with life cycle methodologies, project management, change management, production control procedures and management systems, Ms. Gorge has been able to incorporate data controls along with process controls for the total effect of Quality Assurance.

Ms. Gorge has a Masters degree in Organizational Development and Change Management.

# TECHNIQUES: BRIDGING THE GAP BETWEEN THE STRATEGIC PLAN AND SYSTEMS DEVELOPMENT

SESSION CHAIR

Jack Durner
NASTEC Corporation

PANELISTS

Ellen Levin
Ron Shelby

# BRIDGING THE GAP BETWEEN THE STRATEGIC PLAN AND SYSTEMS DEVELOPMENT

Jack M. Durner
NASTEC Corporation

When I was preparing to chair the Panel on BRIDGING THE GAP BETWEEN STRATEGIC PLANNING AND DEVELOPMENT, I tried to think of all the things that needed to be considered to make that bridging successful. As the introductor slide indicates, planning is where ideas germinate for the overall job to be done and development is where the job is to take the results of planning and make it work. Both parts are important. Planning is needed to ensure that the "big picture" is adequately addressed and development is needed to make sure the job is implemented well.

The success factor for information management in the 1990's is to have in place a series of structured methodologies and tools (in that order) which lead the information managers from high level business information planning through the generation of applications and code.

It is important to realize that Project Management plays an important part in the process, as well. Some methodologies have attempted to merge the managing of the process with the process itself, resulting in a large, complex set of manuals and procedures. Instead, our approach follows the KISS principle (keep it simple stupid). The result is a set of flexible, easy to use methods that are thought of as tools in a tool box, to be used as necessary for the task at hand.

The first step in the process is planning, sometimes affectionately known as the "P" word. Everybody knows we ought to do it, but the usual response is, "We don't have time to do planning, we have real work to do!" Well, unfortunately, the result is the situation we face when coming to work every day. We have systems that won't (can't) talk to each other, programs that don't produce the results the user really needs, users that are angry at us because we don't understand their business and the most nagging problem of all, fire fighting.

A successful planning process produces an information architecture for the business, or business area under study. This could be the entire organization, a product line, a major functional area or simply one function. The point is that the process is the same, no matter what size study you want to do (that's what a methodology is supposed to be, a structured, generic process adapted to meet the objectives of the task at hand).

The five steps in the planning process are to:

- document the functionality of the study area, developing functional models and information usage models,

- document the information entities, producing an enterprise information model,

- analyze the information gathered, looking at improving the business and also consider how and what to automate for the future,

- produce an information architecture (strategic view) for functions, information and technologies,

- develop a plan to implement the architecture over the next N (usually 3-15) years.

Once these planning steps are accomplished, usually in 3-6 months with a great deal of user involvement, the real test is what happens next. How will the results of the planning process be effectively used by the development teams that must implement the plan(s)? In our experience, most of the teams that do the planning are not the same teams that implement that plan. The big question then becomes, how do you communicate all the information gathered during planning to the development teams without losing valuable pieces? The answer is simple (a la KISS). Begin each development project with the same models and supplementary documentation developed during the planning study.

The above would seem to be "intuitively obvious". Unfortunately, most organizations, without having the benefit of a consistent, continuing process, (from planning through development, design and implementation) require the individual project team to translate the results of the planning study into their development methodology language, redraw diagrams (if they were supplied at all) and worst of all, have to reinterview the same people who gave the information to the planning team in the first place. Bottom line is a lot of wasted time and replicated effort.

The "next step" uses the output of the plan to define the system. Depending on how good a job "they" did in building the planning level models, a large part of what used to be done in development, namely defining the high level requirements, has already been done, thereby saving valuable time for each project.

During system definition, we also look at the functional business (Business Analysis and Requirements Definition) and the detailed data used in that business (Relational Data Modeling). The concepts started in planning are thus carried down, in more detail, at this level. The output of the definition process should be about 80-90% of what is required for a Functional Specification document. Again, the user plays a very important part in the process, sometimes to the point of leading the project. After all, if the reason for developing the system in the first place is to support the user's business, why not let them lead the effort? The nice part of it all is that, because the process is relatively simple, the users can (and want to) be part of the solution development effort.

Once the definition step is complete, then the remaining components provide the capability to design the system and develop the programs and procedures necessary to complete the implementation.

As with any building process, the most critical part is the foundation, or starting point. That's why it is so important to begin with thorough planning and then make effective use of the results of the planning process to successfully implement meaningful systems that meet the users needs. If that critical beginning isn't done right, then we get to ask the age old question, "If we don't have time to do it right, when are we ever going to find time to do it over?"

------------------------

Mr. Durner has been in many aspects of Data Processing for 27 years. He came "up through the ranks" in programming and analysis, project management and systems management. He has considerable experience in many diverse industries, including consulting, hospitals, retailing, engineering design, energy, banks, manufacturing, museums and several military agencies.

For the past 8 years, Mr. Durner has been involved in the development and implementation of Information Management technologies. This includes methodologies, structured techniques and supporting software. His responsibilities include training, consulting, facilitation as well as product development. Prior to his current position as a Principal Consultant with Nastec Corporation, he was a Vice President and co-founder of Technology Information Products (TIP).

# THE ENVIRONMENT FOR IMPLEMENTING A STRATEGIC INFORMATION PLAN

Ellen Levin
Federal Home Loan Mortgage

One of the most effective methods of planning for long-term systems development is to undertake a Strategic Information Systems Plan. The plan specifies the system development sequence for 3-5 years to meet current and projected business priorities, often based on four major components:

1. An information architecture consisting of models of business functions and data, information usage, and conceptual applications ordered in a technically-optimum sequence.

2. A current systems evaluation that inventories existing systems and assesses user and technical satisfaction with existing systems.

3. Technology requirements that specify hardware, software and communications alternatives.

4. Information management policies that indicate the approach the organization is to take toward implementing the systems plan. These policies may set management principles, analyze human resource requirements, indicate organizational roles and responsibilities, address methodology, standards and procedures, indicate plan maintenance activities, and explain the basis for project selection.

While the first three components provide the technical basis for the information system plan, it is the fourth component, information management policies, that link the other components and help ensure successful plan implementation. The management component is the essential ingredient that enables the organization to take the plan and move into a system development project environment. This paper attempts to indicate some of the important issues that need to be addressed as management policy.

Within any organization conflict is inevitable. Implementing a strategic systems plan often requires changes in the way an organization selects and develops systems. To deal with the conflict that results from major cultural changes, the plan should address the likelihood of conflict and develop approaches to resolve it. To succeed, senior executive endorsement of a planned, systematic approach that emphasizes an organization-wide viewpoint over narrow parochial interests is essential. Responsibility for obtaining organization-wide consensus, managing the plan, and coordinating system development projects should be clearly specified. The means of determining system development priorities should be communicated and well-understood. As time passes, new business priorities and assumptions will become clear.

The plan needs to specify how the organization will incorporate these changes into the plan so that the plan remains a living and useful tool. The management policies should clearly assign responsibility for maintaining the plan so that the organizational units know what information they are expected to produce or use. Finally, to help alleviate conflict, a development methodology with supporting standards and procedures should be adopted and taught to all participants.

While the strategic systems plan addresses long-term development needs, it is important to plan for the support and enhancement of existing systems. Failure to address immediate concerns could have serious impact on the ability of the enterprise to respond to critical short-term business requirements. It could lead to discarding the strategic plan altogether. Thus, the organization should initially identify and continue to consider "must-do" enhancements. These immediate priorities should be assessed in comparison to the long-term development sequence so that the impact of undertaking the short-term projects is known. The development sequence may be altered from the technically optimum one. The costs associated with the alternative development sequence, such as bridging, system redesign and conversion, should be addressed by the project planning and coordination function.

Plan maintenance needs to be addressed by the management policies. The information system plan components such as functional models, data models, matrices, project descriptions and schedules need to be maintained by the organizational units assigned to that responsibility. The plan may need to be extended in scope to include business areas not initially included. As projects are undertaken, the increasing level of detail generated must be integrated with the strategic models and the models revised as needed to reflect the increased level of understanding. An automated CASE tool is essential to keep the models up to date. A change control procedure that specifies the means of approving, integrating, and tracking changes should be implemented.

A system development lifecycle methodology should be developed and universally employed to provide a consistent development process. This methodology should also be supported by an automated CASE tool, preferably the same one used for the strategic systems plan. The system development lifecycle method should specify project development phases and milestones. The project initiation phase should be derived from the strategic plan. The methodology should provide a consistent approach, procedures and tools. A standard set of analysis, design and development deliverables should be specified. Milestone reviews should include evaluation of the products for conformance to a corporate-wide perspective.

The chosen system development methodology should enforce top-down design. This is controlled at the project initiation phase by a central project planning and management group whose function is to implement the plan and coordinate multiple projects. Each system development project should start with the strategic plan products such as the conceptual data model and the functional business model. The project teams perform business area analysis to understand system requirements at a detailed level. The teams further decompose business functions identified by the strategic plan, and they validate and extend the data model. Function-data usage is confirmed at the lowest level of detail.

Within the top-down scenario, Data Administration has an opportunity to perform an essential coordinating function. Its traditional role may be expanded to include the wider area of models administration to reflect a concern not only with the data model but also with the functional model and its interaction with the data model. To maintain an organization-wide perspective, Data Administration should conduct data definition workshops to include a wide range of functional areas with an interest in the data under consideration. Project developers, who typically have a more application-specific viewpoint, need to be included in this process which should take place at the start of every project. Data Administration manages the development of a detailed data model, coordinates the concurrent uses of portions of the model, and approves and integrates model changes. In the case of data conflicts, Data Administration facilitates the reconciliation of differences resulting in organizational consensus.

For the information systems plan to succeed, the organization must have in place a set of comprehensive standards for system development. The standards specify the acceptance criteria for data and process model deliverables. These include naming standards, diagramming conventions, abbreviations, and development techniques. The standards help to facilitate communication and consistency and are an essential step for an enterprise-wide information resources dictionary/directory. Responsibility and authority for enforcing compliance with the standards should be assigned.

Since the implementation of an information systems plan represents, for many organizations, a change in the way systems are developed, it may require restructuring of the information systems organization and the creation of new organizational units. It may also lead to changes in the end-user organization based on recognized functionality and data usage. Within the information system organization, some of the essential functions that need to be accommodated include plan maintenance and extension, project coordination and management, data and process modeling, quality control, configuration management, methodology development and CASE tool support. The responsibilities for physical system implementation should also be assigned. In this environment the flow of information among these groups as well as their relation to system development project teams should be clearly defined.

There may be significant cultural changes required for an organization to reorient its thinking toward top-down systems development based on an information systems plan that stresses an enterprise-wide perspective. An active education and training program will help to ensure success. This program should educate both the business user and information systems organizations in the new development approach. It should encompass training in new technical skills such as development methodology, data and process modeling, and new tools. Changes should be made gradually through a series of measured steps toward the goal and with an appreciation of the sensitivities and concerns that the affected individuals may experience. Prototyping the new tools, techniques and development approach on a carefully-selected project will help to build credibility. Specific measurement criteria for evaluating the prototype project and subsequent projects should be established before the start of the efforts and can be used to demonstrate the benefits of the new methods.

By addressing management issues early, mechanisms are established to move the plan from the strategic level to development. If the issues are not addressed until the plan is released, valuable momentum and time may be lost as the organization struggles to address these important concerns. Failure to address these issues may result in the plan becoming just another planning document that sits on a shelf and is interesting for historical purposes only. Management issues are often controversial and there may be a tendency within the strategic systems planning group to focus most of its attention on the relatively straightforward technical and factual aspects of the plan. In order to ensure that the planning effort will be fully successful, as demonstrated by its active use in the system development environment, management issues must be seriously addressed and resolved.

-------------------------

Ellen Levin is currently Manager of Corporate Models at Freddie Mac, the Federal Home Loan Mortgage Corporation. She is responsible for the development and maintenance of conceptual and logical data and functional models to support the organization's information requirements. She recently completed a strategic enterprise model project and has been instrumental in the development of integrated methodologies. She has held previous positions in data administration at INTELSAT and COMSAT.

Freddie Mac

# THE ENVIRONMENT FOR IMPLEMENTING A STRATEGIC INFORMATION PLAN

ELLEN J. LEVIN

FEDERAL HOME LOAN MORTGAGE CORPORATION

MAY 3, 1989

Figure 1

Freddie Mac

# STRATEGIC INFORMATION SYSTEMS PLAN

- DEFINITION:
  - THE SYSTEM DEVELOPMENT SEQUENCE FOR 3-5 YEARS TO MEET CURRENT AND PROJECTED BUSINESS PRIORITIES BASED ON THE FOLLOWING COMPONENTS:
- INFORMATION ARCHITECTURE
  - MODELS OF BUSINESS FUNCTIONS AND INFORMATION USAGE
  - CONCEPTUAL APPLICATIONS ARCHITECTURE AND SEQUENCE
- CURRENT SYSTEMS EVALUATION
  - EXISTING SYSTEMS INVENTORY
  - USER AND TECHNICAL ASSESSMENT
  - COMPARISON OF CURRENT SYSTEMS AND INFORMATION ARCHITECTURE

Figure 2

Freddie Mac

## STRATEGIC INFORMATION SYSTEMS PLAN (Continued)

- TECHNOLOGY REQUIREMENTS
  - HARDWARE
  - SOFTWARE
  - COMMUNICATIONS

- INFORMATION MANAGEMENT POLICIES
  - PRINCIPLES
  - HUMAN RESOURCE REQUIREMENTS
  - ROLES AND RESPONSIBILITIES
  - METHODOLOGY, STANDARDS AND PROCEDURES
  - PLAN MAINTENANCE
  - PROJECT SELECTION PRIORITY

- INFORMATION MANAGEMENT POLICIES LINK THE OTHER COMPONENTS AND HELP TO ENSURE SUCCESS

Figure 3

Freddie Mac

## ADDRESS MANAGEMENT ISSUES TO DEAL WITH CONFLICT

- PROJECT MANAGEMENT RESPONSIBILITY
- SENIOR EXECUTIVE ENDORSEMENT OF PLANNED, SYSTEMATIC APPROACH, ORGANIZATIONAL PERSPECTIVE OVER PAROCHIAL
- GRADUAL ORGANIZATIONAL RESTRUCTURING DRIVEN BY FUNCTIONS AND DATA
- CHANGES IN BUSINESS PRIORITIES
- BUSINESS ASSUMPTIONS
- PLAN MAINTENANCE RESPONSIBILITY
- POLICIES, METHODOLOGY, STANDARDS AND PROCEDURES

Figure 4

Freddie Mac

## ADDRESS DATA OWNERSHIP ISSUES

- DATA OWNERSHIP POLICY
  - SECURITY
  - PRIVACY
  - ACCESS
  - RESPONSIBILITY
  - INTEGRITY
  - DISSEMINATION
- IDENTIFY DATA USERS
- JOINT DATA DEFINITION WORKSHOPS WITH CURRENT AND FUTURE APPLICATION USERS

Figure 5

Freddie Mac

## PLAN TO SUPPORT AND ENHANCE EXISTING SYSTEMS

- SHORT-TERM "MUST-DO'S"
- IMPACT ON STRATEGIC PLAN
- ACKNOWLEDGE REDESIGN AND CONVERSION REQUIREMENTS

Figure 6

Freddie Mac

## MAINTAIN STRATEGIC MODELS

- EXTEND SCOPE
- REVISE WITH INCREASED DETAILED ANALYSIS
- APPROVE AND INTEGRATE MODEL CHANGES
- RECORD EXPANDED MODEL IN AUTOMATED GRAPHIC AND DICTIONARY TOOLS

Figure 7

Freddie Mac

## USE SYSTEM DEVELOPMENT LIFECYCLE

- PROJECT DEVELOPMENT PHASES AND MILESTONES
- PROJECT INITIATION FITS INTO STRATEGIC PLAN
- CONSISTENT APPROACH, PROCEDURES AND TOOLS
- STANDARD ANALYSIS, DESIGN AND DEVELOPMENT DELIVERABLES
- MILESTONE REVIEW FOR CONFORMANCE TO CORPORATE PERSPECTIVE

Figure 8

Freddie Mac

## ENFORCE TOP-DOWN DESIGN

- ENFORCE TOP-DOWN DESIGN
  - PROJECT INITIATION PHASE
  - CENTRAL PROJECT PLANNING
  - COORDINATE MULTIPLE PROJECTS
- PROJECTS START WITH STRATEGIC PLAN PRODUCTS
  - BUSINESS AREA ANALYSIS
  - FUNCTIONAL DECOMPOSITION
  - DETAILED REQUIREMENTS ANALYSIS
  - VALIDATE AND EXTEND DATA MODEL
- DATA ADMINISTRATION (MODELS CONTROL)
  - ATA DEFINITION WORKSHOPS
  - DETAILED DATA MODEL
  - MANAGES CONCURRENT USED OF MODELS
  - INTEGRATES MODEL CHANGES
  - MEDIATES DIFFERENCES

Figure 9

Freddie Mac

## DEVELOP AND ENFORCE STANDARDS

- DOCUMENTATION REQUIREMENTS
  - NAMING STANDARDS
  - FUNCTION NAMES
  - ENTITY NAMES
  - ABBREVIATIONS
  - DIAGRAMMING CONVENTIONS
  - METHODOLOGY STANDARDS
- SUPPORT WITH STANDARD CASE TOOL SET

Figure 10

Freddie Mac

## ASSIGN STAFF ROLES, RESPONSIBILITIES AND AUTHORITY FOR:

- PLAN MAINTENANCE AND EXTENSION
- MODEL CHANGE REVIEW AND APPROVAL
- IMPACT ASSESSMENT
- CHANGING PROJECT PRIORITIES
- CONFLICT RESOLUTION
- DATA MODELING
- PROCESS MODELING
- MODELS INTEGRATION
- PHYSICAL IMPLEMENTATION
- QUALITY CONTROL
- CONFIGURATION MANAGEMENT
- TOOL SUPPORT
- REPORT PRODUCTION

DEFINE INTERFACES BETWEEN GROUPS

Figure 11

Freddie Mac

## ORGANIZE HUMAN RESOURCES

- ALLOCATE SUFFICIENT STAFF
- INFORMATION SYSTEMS ORGANIZATION MAY NEED RESTRUCTURING
- TRAIN DEVELOPMENT STAFF IN NEW SKILLS:
- FUNCTIONAL DECOMPOSITION
- INFORMATION FLOW DIAGRAMMING
- ENTITY RELATIONSHIP/DATA MODELING
- NEW TECHNOLOGY
- EDUCATE END-USERS IN NEW DEVELOPMENT APPROACH
- RECOGNIZE CULTURAL CHANGES REQUIRED

Figure 12

# BUSINESS AREA ANALYSIS:
## The Bridge From Strategy Planning To Systems Development

Ron Shelby
American Management Systems, Inc.

### Introduction

Increasingly today, leading corporations and government agencies are calling upon their information systems professionals to support a new generation of products and services that can't be implemented without managing common data, networks, and information systems successfully. Implementing systems that share customer or product data among them requires top-down planning and bottom-up implementation. How are leaders in the information systems field bridging the gap between top-down strategy planning and systems development (essentially bottom-up)? Analysis of business areas is the approach being used successfully by many organizations to build data-sharing systems based upon the architectures outlined in strategy planning.

Today, I'd like to take a few minutes to discuss business area analysis, the bridge from strategy planning to systems development.

### Information Systems Role in the 1990's

The dramatic increase in the power of information technology in the past ten years is changing the role of information systems. Greater power and lower costs have allowed organizations to automate tasks which are not repetitive, not shared broadly, but which require considerable processing and data. As a result the centralized, mainframe-based systems world of the 1970's has been transformed to the decentralized, multi-layered systems world which is now emerging.

Increased technological power enables business and government to increase the scope of its automation, creating new ways of fulfilling missions. Indeed, entirely new markets, services, and industries have been created as a result. For example, in the financial services industry the widespread use of credit cards, rapid electronic funds transfer, and the many personalized financial investment options available today were not feasible 25 years ago, because the available technology would not support them.

As technology enables new business needs to be met, these business needs demand an ever-increasing level of electronic automation to compete, increase service levels and decrease unit costs. When dealing with a financial institution we expect to be able to withdraw cash from our banks no matter where we are. Credit card companies speed the handling of transaction approvals by using artificial intelligence automation embedded in transaction processing systems, allowing them to cut credit approval costs while increasing the speed at which low risk transactions are approved.

And yet, businesses and government frequently hit a wall as they pursue this ever-increasing spiral of automation enabling new business capabilities. This wall exists because common data about customers, products, and facilities are lacking. As information systems organizations attempt to manage customer data, they find it scattered across hundreds of files and databases which are implemented on different processing platforms. To make matters worse, this data is edited in hundreds of places by routines which vary widely, giving inconsistent results from one database to the next.

The scenario I've described here would be accurate for most corporations or government agencies today. A few leading companies have already remedied the most critical of their problems. Others, seeking competitive advantage within a market (or on a global scale), are in the midst of large software redevelopment projects which are targeted to provide integrated databases and systems for those things that must be widely shared within the company or agency. In the insurance and financial communities, the emphasis is upon customers and customer service. In the petroleum industry and government agencies, the emphasis is upon the products and services themselves, while the telecommunications industry is in the midst of a shift toward becoming part of the world's services market.

What does this all mean for information systems people generally, and data administrators in particular? It means we face the opportunities we have always said we wanted to have. Senior management is sponsoring high-visibility initiatives to build common systems and databases to support their organizations in the future. Increasingly, we are asked to architect, integrate, and manage shared-data environments that will support the core of tomorrow's enterprises.

To succeed we need a practical, rigorous methodology for linking our information systems strategies with the systems development process.

Engineering Data-Sharing Systems

For years data administrators have discussed, promoted, and attempted to win support for data-driven system development techniques. We have tried normalization and canonical synthesis of data from the bottom-up once development began.

We have tried strategic data planning and data modeling from the top down to define the details of all core databases before major development projects begin. Success has been a stranger to initiatives using these approaches.

A third data-driven approach to planning and developing systems and data bases is information engineering based upon the work of James Martin. While there are many different versions of information engineering in existence today, all share a similar approach. This approach combines top-down planning with bottom-up implementation of systems and databases. It is this information engineering approach to data management that offers the best opportunity for success when a data sharing systems environment is an objective.

Strategic Information Systems Planning

The objective of strategic information systems planning hasn't changed much since the early 1970's. Strategic planning strives to provide an enterprise-wide information management plan to support the organization's business strategy. At the same time, an effective strategic planning effort increases management's awareness of technology's potential while alerting information systems management to critical information management priorities.

A strategic planning effort should deliver a model of the business functions and data of the enterprise, including their interactions. This broad view of the enterprise's information requirements is called an information architecture. This architecture should satisfy the information requirements of the enterprise. Products included in the information architecture include a high-level business function (or process) model, a data model, and a matrix which summarizes the interaction between functions and data. While the individual products that make up an information architecture have long been familiar to us, their use in planning a broad analysis of the business is a key feature of information engineering.

The strategic plan should also include a business systems architecture which describes the probable business systems and data stores required to support the enterprise's information architecture. This high-level prediction concerning the future information management environment will be refined as business analysis proceeds. Nevertheless, the business systems architecture is an important early blueprint of the enterprise's target information systems which will be used in the early stages of migration planning.

A complete strategic plan should also contain a technical architecture describing the hardware platforms, software, and networks required to implement the business systems architecture. This architecture defines the key technological infrastructure required for the future. Taken together, the three architectures provide a blueprint which we will call a strategic plan. This strategic plan should allow an enterprise to manage its information effectively in the future by developing information systems that support its business objectives.

Bridging from Strategy Planning to Systems Development

Instead of proceeding to develop a large number of narrowly-scoped systems, the information engineering approach is to perform a detailed analysis of areas of the business that are cohesive and play a key role in creating and maintaining shared data. This analysis of business areas is the key step in tying strategic planning to the development of systems. During the analysis of a business area, several system design projects are clearly identified and scoped. Since this analysis is broader than a traditional "systems analysis", it forms a better basis for stable, integrated systems and databases.

The issue of which business area is analyzed first depends upon the unique priorities of the enterprise. Most service industry companies start with customer identification and development, while manufacturing companies frequently start with product design and development. Each enterprise should address its unique priorities first when analyzing the business.

Business Area Analysis (BAA)

A business area is a cohesive, logical collection of business functions and data which are managed together and which are bundled together to define the scope of an analysis project. This "bundling" should be done as part of strategic planning after the information architecture is defined.

The objectives of analyzing a business area include identifying what detailed business activities must occur to define and use data to meet business objectives. These business activities are commonly called processes, although some versions of information engineering call them functions or activities. Defining the sequence of activities and the interaction of activities and data takes up most of the time and effort required during business area analysis.

Each business area analysis project should take from 4 to 8 months to complete a detailed analysis of business activities and data. Each BAA should scope out the logical processes and data for from two to five system design projects. As you can readily see, integration of data requirements during business area analysis is the key to delivering shared databases later in the systems life cycle. Integrating the definition of detailed data requirements during BAA is a task for the data administrator.

The Data Administrator's Role in BAA

If information systems are going to share data, then the data administrator, or someone fulfilling the data administrator's role, must play an important part in planning and carrying out analysis of business areas. The data administrator (DA) should play a lead role in selecting the CASE (computer-aided software engineering) tools to support analysis. The DA should define data definition standards and integration procedures, and assist project teams in understanding the scope of the business area defined for them during the strategic planning process. In every way it is the data administrator who plays the key role in coordinating between the strategic plan and the initiation of the more detailed business area analysis. The DA should also define the key roles of information analysts, business clients (end-users), CASE encyclopedia manager, team data administrator, and an overall data architect.

During business area analysis, the project team verifies their understanding of the scope of their analysis with management, proceeds to analyze data requirements (entity analysis), functional requirements, systems supporting the business area currently, and then delivers a detailed logical description of functions and data for the next stage of the information engineering life cycle, business system design.

The key deliverable which describes data requirements is a data model composed of an entity relationship diagram and definitions of the entity types and attributes. As the integrator of the data definition process during BAA, the data administrator is the key to defining shared data that can later be implemented as shared, common databases. The DA should assume responsibility for maintaining a master data model that is used to share data definitions across all business areas of the enterprise. Obviously, an automated data dictionary or CASE encyclopedia is essential to succeeding in this role.

The data administrator should also establish guidelines to communicate changes across teams, coordinate data definitions responsibilities among teams, and resolve disagreements on a regular basis. Ultimately, the DA must assume responsibility for ensuring the data definitions provided by the teams are complete and verified by business clients (end-users). The DA's integrator role, linking the analysis within each team with the analysis being done by other teams, is the key to defining an enterprise's common, shareable data.

Conclusion

As I have discussed, the DA's role during analysis of business areas is essential to bridging the gap between strategy planning and systems development. The DA manages the creation of logical data models within each business area which share definitions of important data. This logical data model will form the basis for the physical data base design which is created in the next phase of the life cycle. Without successfully integrating data definitions during BAA, shared data bases are not likely to be produced by the system development projects that follow. The DA, acting as an integrator, change agent, and manager of information about shared data is in the best position to move the enterprise closer to shared data during business area analysis.

-------------------------

Ron Shelby has ten years experience as a data administration practitioner and consultant. He founded the data administration function at a major insurance company in Toronto, and then served as the data administrator for the U.S. Department of the Interior.

While in Toronto, Ron served as President of the Data Base Association of Ontario, Canada's largest data administration professional association. Once he relocated to Washington, D.C., he co-founded the National Capital Region Chapter of DAMA in 1985. Ron continues to serve as the membership Vice President of the National Capital Region Chapter.

As Membership Vice President for DAMA International, he established the DAMA newsletter as a means of communication amongst the chapters.

As a consultant, Ron has advised and trained data administrators in the financial, oil, publishing, and telecommunications industries, as well as in the Federal Government. He has helped clients use information engineering techniques and CASE tools, and taught numerous courses in data modeling and the use of data dictionaries. Ron speaks frequently at professional conferences on data administration topics.

## BUSINESS AREA ANALYSIS

# THE BRIDGE FROM STRATEGY PLANNING TO SYSTEMS DEVELOPMENT

Presented by
Ron Shelby
May 3, 1989

1989 NCR-DAMA Symposium

Figure 1

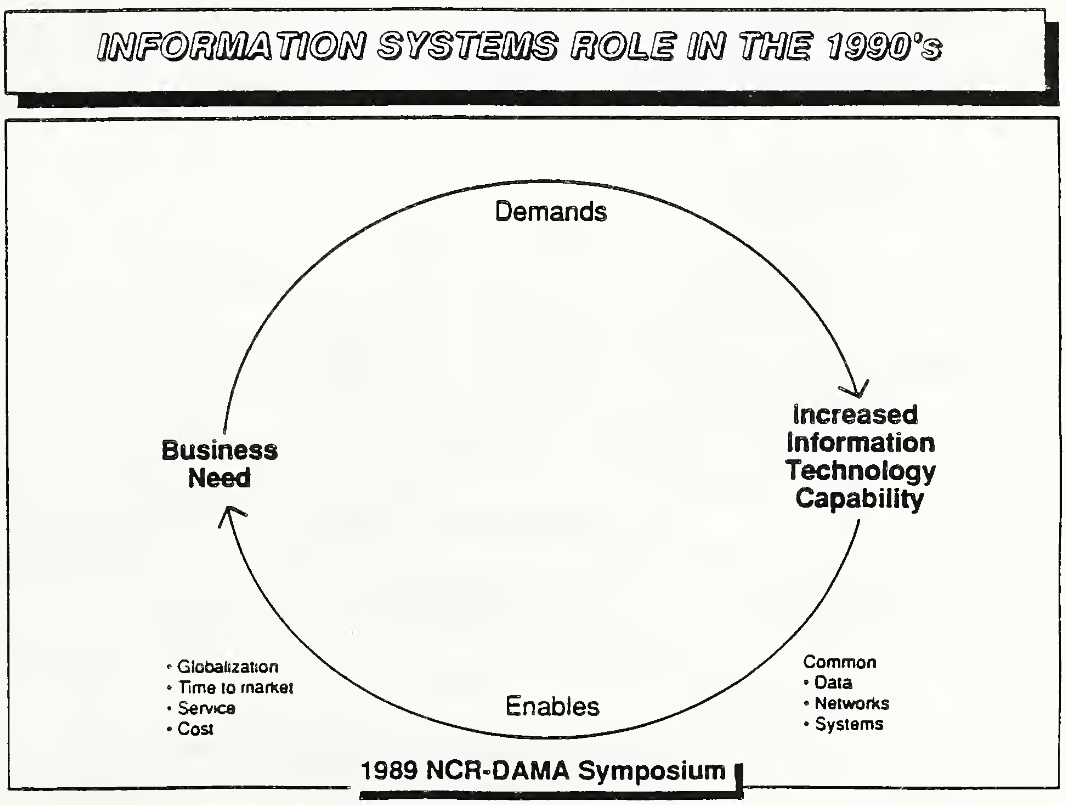



Figure 2

## WHY DO STRATEGIC PLANNING?

- Establish an information strategy based upon business strategy
- Increase management awareness of information technology's potential for the business
- Establish a plan to invest in systems which meet business information needs
- Define an information architecture for future development of data sharing systems
- Plan a technical architecture to optimize the use of information technology

1989 NCR-DAMA Symposium

Figure 3

## STRATEGY PLANNING DELIVERABLES

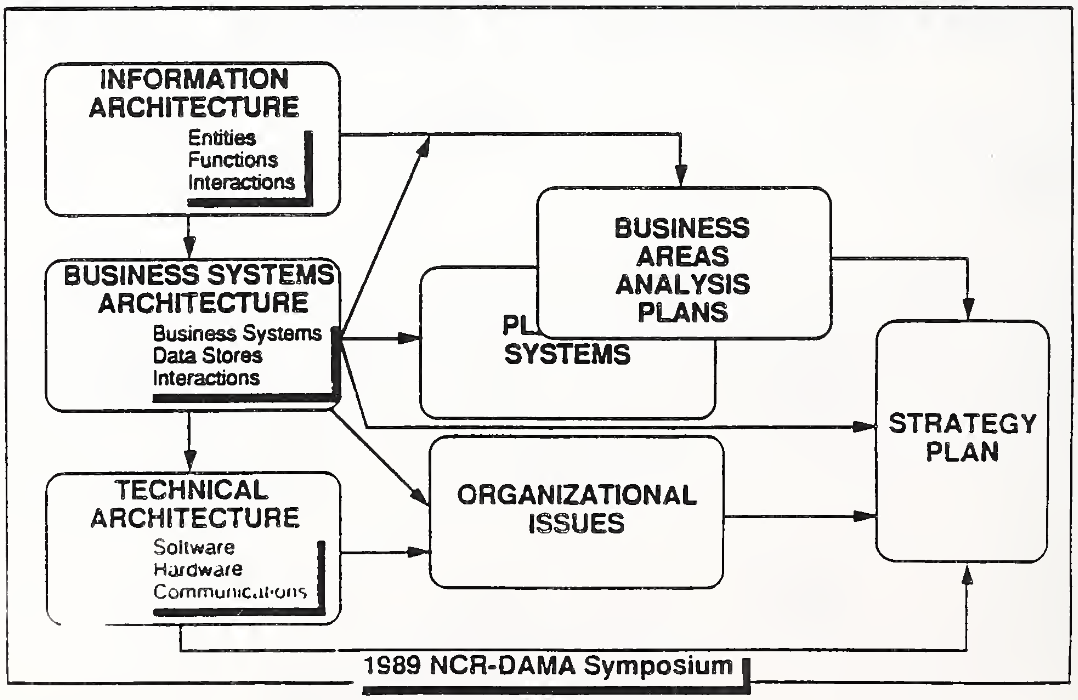
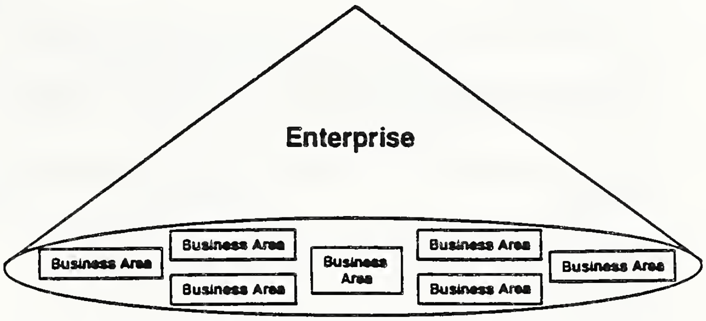



Figure 4

# WHAT IS A BUSINESS AREA?

**A logical collection of business functions and entity types which are carried out together and which are "bundled" to define the scope of an Analysis Project.**




The grouping of business functions and entity types which scope an Analysis Project is done during Strategy Planning.

**1989 NCR-DAMA Symposium**

Figure 5

# BUSINESS AREA ANALYSIS

*"The period in the systems life cycle in which a detailed analysis of business objects is carried out within a defined Business Area in preparation for the design of systems to support that area."*

**1989 NCR-DAMA Symposium**

Figure 6

## OBJECTIVES OF BUSINESS AREA ANALYSIS

- To identify and define the business activities of a major part of a business
- To define the data required for each business activity
- To identify the necessary sequence in which activities should occur
- To define the manner in which the data is affected by business activities
- To scope out discrete design areas for development

1989 NCR-DAMA Symposium

Figure 7

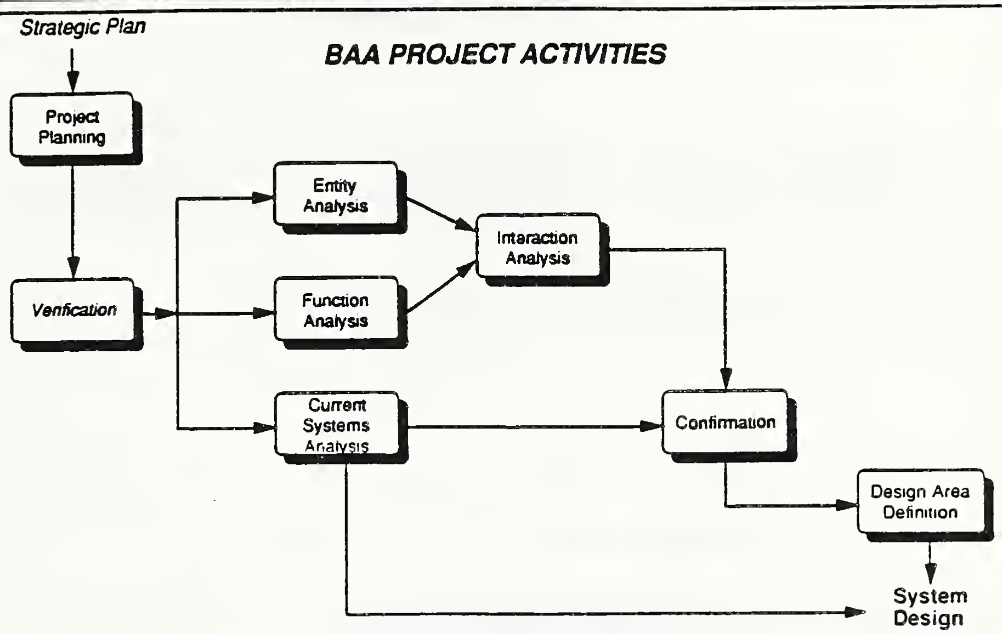
## PLANNING THE ANALYSIS

- Scoping document for each project
- Roles and procedures for data definition management
- Prepare CASE models
- Standards
- Staffing, space, and tool selection
- CASE model management plans

1989 NCR-DAMA Symposium

Figure 8

# ACTIVITIES DURING ANALYSIS




1989 NCR-DAMA Symposium

Figure 9

# BAA DELIVERABLES

- Entity Relationship Diagram
- Entity Hierarchy Diagram
- Process Hierarchy Diagram
- Process Dependency Diagram
- Process Logic Diagram
- Process Action Diagram
- Design Areas For Development

1989 NCR-DAMA Symposium

Figure 10

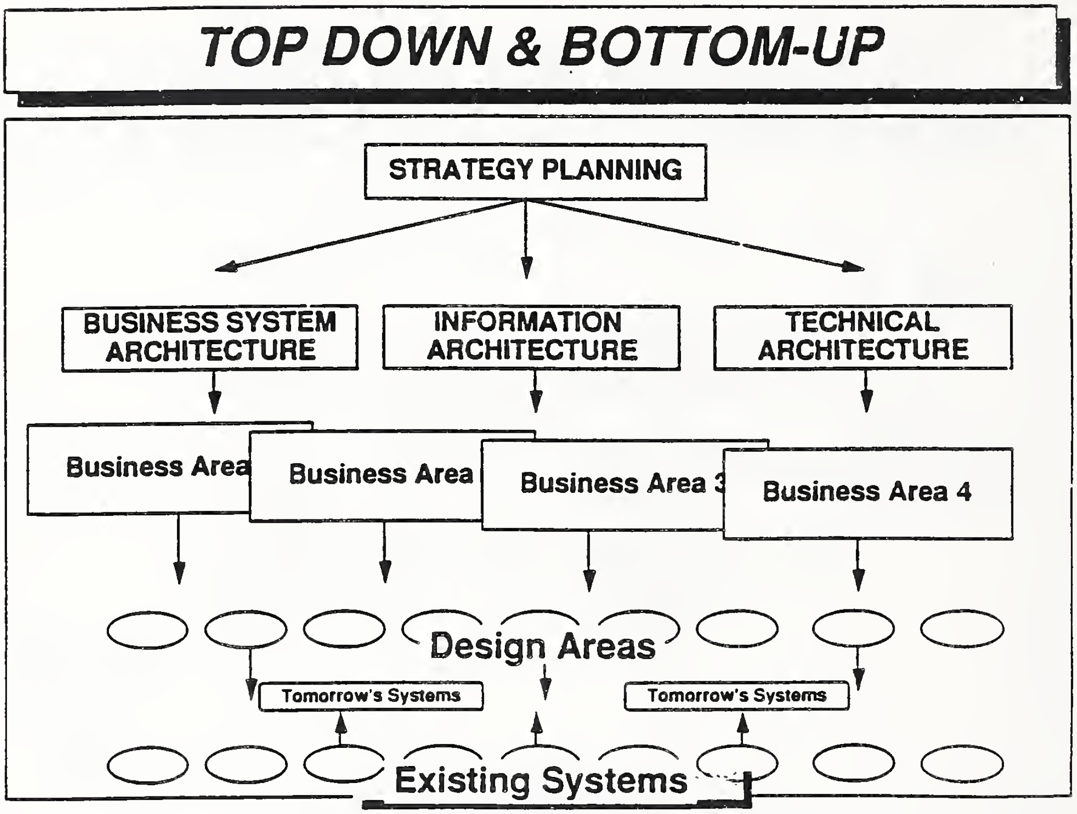



Figure 11

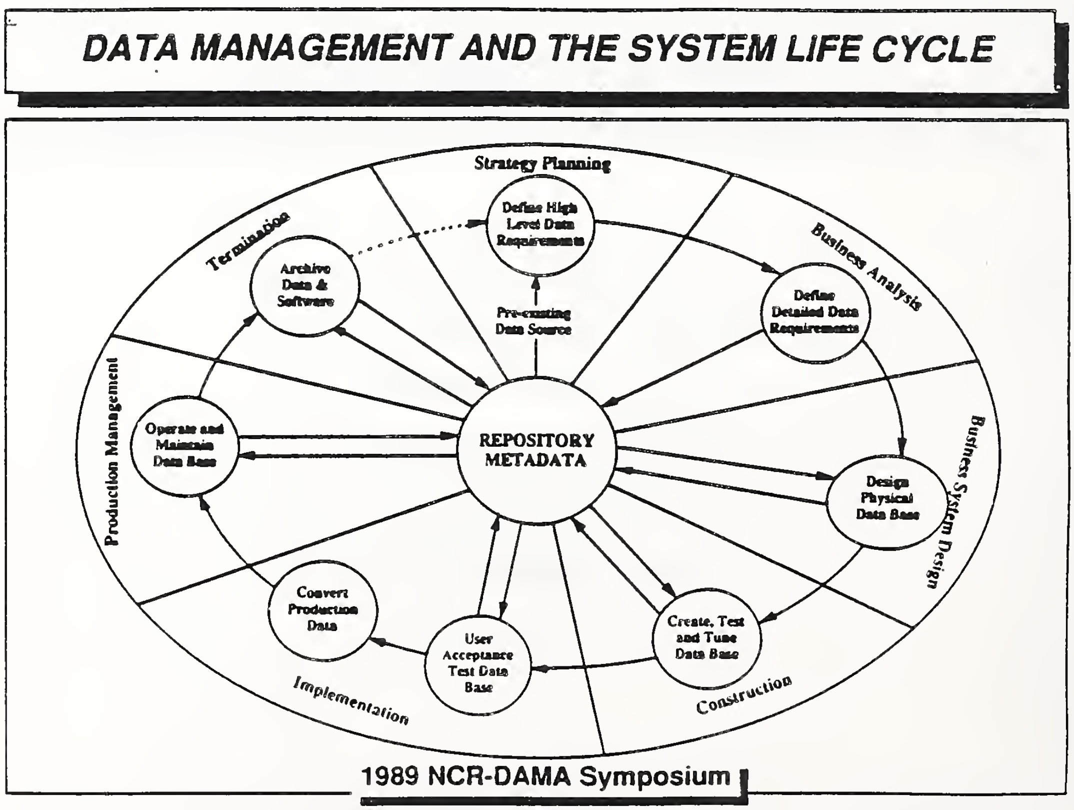



Figure 12

## LEVERAGING THE DATA ASSET

***The Data Administrator's Roles***

1. Architect of a Vital, Shared Asset
2. Change Agent Supporting Innovation
3. Supplier of Information About Shared Data
4. Integrator

1989 NCR-DAMA Symposium

Figure 13

## DATA DEFINITION MANAGEMENT

***OBJECTIVE***

- Consistent Data Definition to Support Data Sharing

***GOALS***

- Ensure Data Integration Across Functions
- Communicate Definition Changes Across Teams
- Provide an Audit Trail for Data Definition Tracking
- Link Intra-Team Procedures with Inter-Team Integration
- Clarify Data Definition Responsibilities

1989 NCR-DAMA Symposium

Figure 14

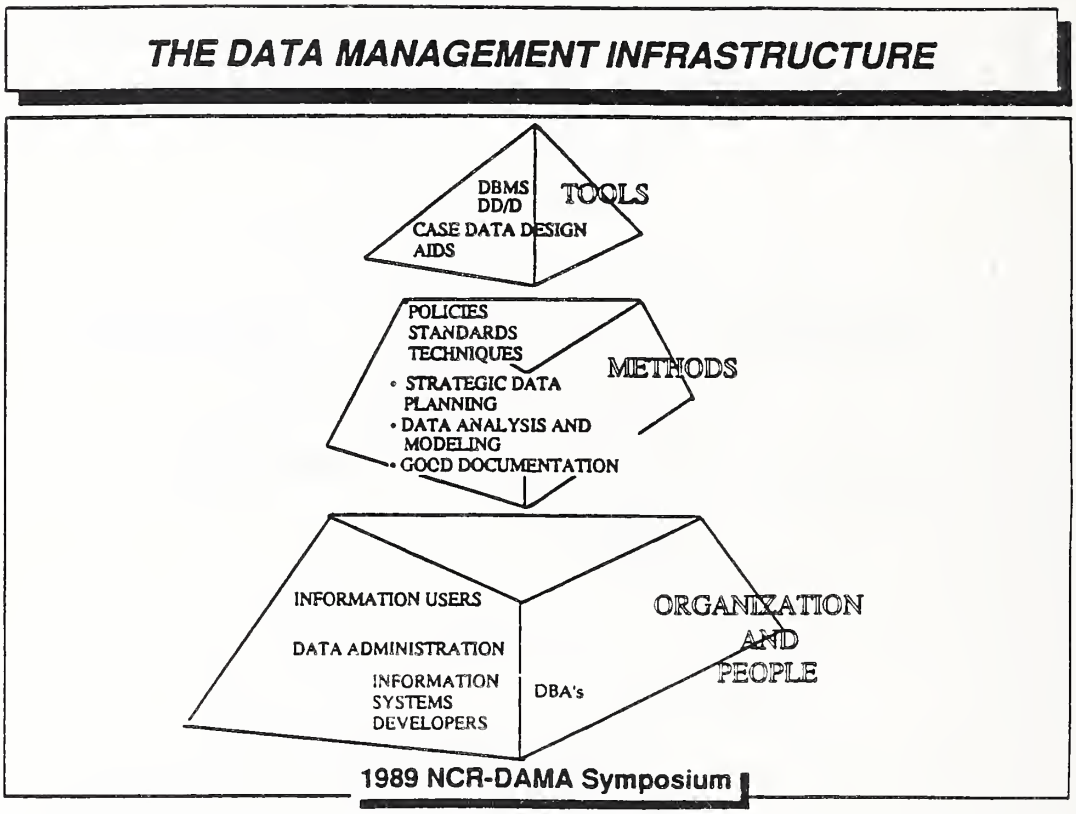



Figure 15

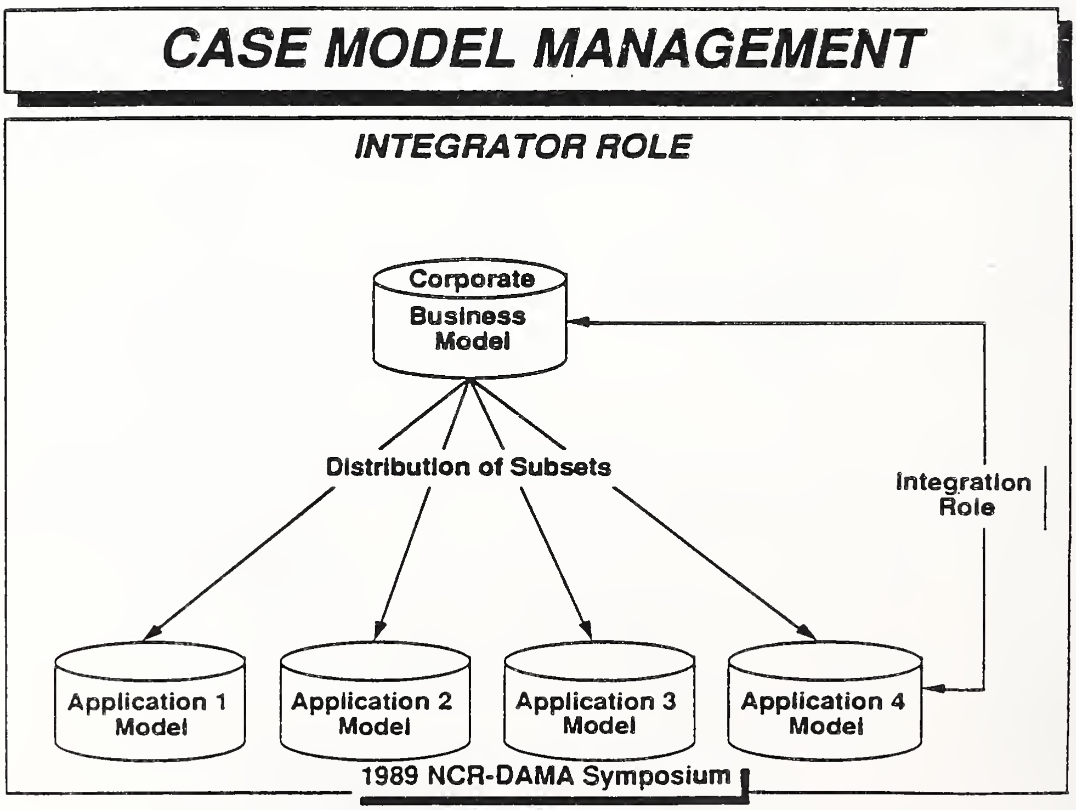



Figure 16

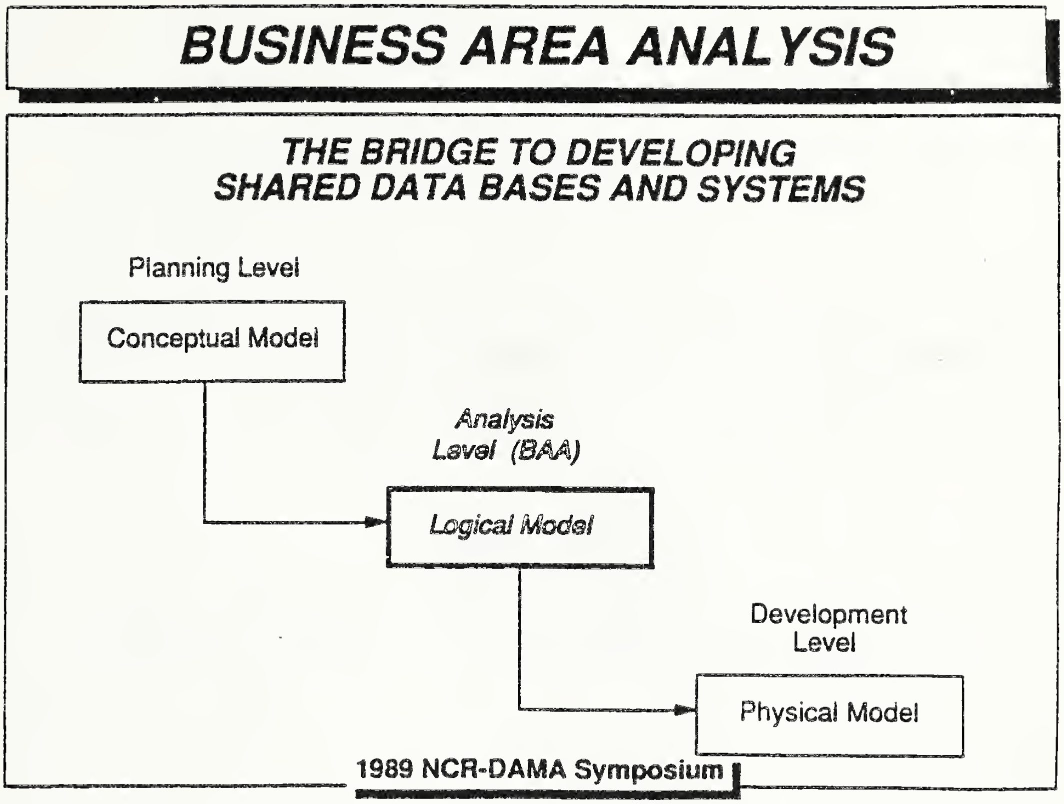

BUSINESS AREA ANALYSIS
THE BRIDGE TO DEVELOPING
SHARED DATA BASES AND SYSTEMS
Planning Level
Conceptual Model
Analysis
Level (BAA)
Logical Model
Development
Level
Physical Model
1989 NCR-DAMA Symposium

Figure 17

## STANDARDS: USING STANDARDS TO SUPPORT DATA SHARING

SESSION CHAIR

Alan Goldfine
National Institute of Standards and Technology

PANELISTS

Judith Newton
Margaret Law
Thomasin Kirkendall

# THE INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS)
# A STATUS REPORT

Alan Goldfine
National Institute of Standards and Technology

## The Status of the IRDS

The IRDS is a computer software system that provides facilities for recording, storing, and processing information about an organization's significant data and data processing resources. It is a generalization and standardization of commercially available data dictionary/directory systems, and is defined by a series of standard specifications.

The initial IRDS specification is a 764 page document. It defines a Command Language and a screen-oriented, menu-driven Panel Interface. It also defines the underlying data model of the IRDS, a variant of the Entity-Relationship approach. The specification also includes the Basic Functional Schema, a "starter set" of IRDS entity-types, relationship-types, and attribute-types.

The IRDS became a voluntary American National Standard in October, 1988. Copies ($65 each) can be ordered from the American National Standards Institute (ANSI) at (212)642-4900. "X3.138-1988, Information Resource Dictionary System" should be specified.

The IRDS has just become a Federal Information Processing Standard (FIPS Publication 156). The announcement appeared in the April 5, 1989 Federal Register, and copies of the FIPS Publication will be available from the National Technical Information Service in a couple of months. The effective date of the FIPS is September 25, 1989.

The IRDS development community has always recognized the need for an interface to the IRDS suitable for use by software external to the IRDS. The IRDS technical committee X3H4 has developed specifications for such an interface, called the Services Interface. The Services Interface specifies generic, low-level, navigational operations for accessing an IRDS. The draft of the Services Interface standard has been completed, and should be out for public review in the Spring of 1989.

## Standards Activity

Several other standards in the IRDS family are being developed or are under active consideration:

- The Export/Import File Format--under development. This project will produce a standard format for the controlled transfer of dictionary data from one IRDS to another. The format, when official, will complete the specification of the IRD-IRD Interface in the IRDS standard.

- Naming Convention Verification--under development. This technical report, which we anticipate will serve as the basis of an IRDS Module, will assist data administrators by storing standard names and their relationships to other, synonymous names, by enforcing the organization's rules for the formation of standard names, and by producing name analysis reports on demand.

- Model Integration--under consideration. This technical report would outline the steps required in synthesizing an integrated data model or conceptual schema from a set of component user views for ultimate placement in an IRDS. It would specify the minimum functionality required for a tool that provided computer-aided support of the model integration process.

- The IRDS in a Distributed Heterogeneous Environment--under consideration. This technical report would provide a framework for the logical placement of the IRDS in a data administration environment. This framework would clarify the role of the IRDS in current distributed multi-platform environments and would illustrate the interfaces to CASE software, network software, and intelligent device controllers.

NIST Activities

The National Institute of Standards and Technology (NIST) is enhancing its IRDS prototype to include a Panel Interface and IRD-IRD Interface capability. The current source code, which is available for outside use and testing, consists of a C program interface to an SQL database, and implements a subset of the IRDS Command Language.

NIST also plans to develop validation tests for IRDS software. We invite the cooperation of interested vendors and users in this effort.

IRDS Documentation from NIST

- A Technical Overview of the Information Resource Dictionary System (Second Edition), NBSIR 88-3700, (Revision of NBSIR 85-3164).

- Guide to Information Resource Dictionary System Applications: General Concepts and Strategic Systems Planning, NBS Special Publication 500-152.

- Guide on Data Entity Naming Conventions, NBS Special Publication 500-149.

---

Alan Goldfine is a senior staff scientist with the National Computer Systems Laboratory of the National Institute of Standards and Technology. He is the leader of the NIST project to develop Federal Information Processing Standards for the Information Resource Dictionary System. He is also the Technical Editor of the IRDS Specifications document for Standards Committee X3H4.

Dr. Goldfine holds a Ph.D. in Computer Science from Pennsylvania State University.

National Capital Region
Data Administration Management Association
Second Annual Symposium
May 3, 1989

# THE INFORMATION RESOURCE DICTIONARY SYSTEM A STATUS REPORT

ALAN GOLDFINE
NATIONAL COMPUTER SYSTEMS LABORATORY
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
BUILDING 225, ROOM A266
GAITHERSBURG, MD 20899
301/975-3252

Figure 1

## THE IRDS

- IS A COMPUTER SOFTWARE SYSTEM
- PROVIDES FACILITIES FOR RECORDING, STORING, AND PROCESSING INFORMATION ABOUT AN ORGANIZATION'S SIGNIFICANT DATA AND DATA PROCESSING RESOURCES
- IS A GENERALIZATION AND STANDARDIZATION OF COMMERCIALLY AVAILABLE DATA DICTIONARY/ DIRECTORY SYSTEMS
- IS DEFINED BY A SERIES OF STANDARD SPECIFICATIONS

Figure 2

## THE IRDS (Initial Specification)

- 764 PAGE DOCUMENT
- DEFINES
  -- COMMAND LANGUAGE
  -- PANEL INTERFACE
  -- UNDERLYING E/R DATA MODEL
  -- BASIC FUNCTIONAL SCHEMA
- BECAME AN ANSI STANDARD IN OCTOBER 1988
  COPIES ($65) CAN BE ORDERED FROM ANSI:
  (212)642-4900
- HAS JUST BECOME A FEDERAL INFORMATION PROCESSING STANDARD (FIPS PUB 156)

Figure 3

## THE IRDS (Services Interface)

- SPECIFIES GENERIC "LOW LEVEL" EXTERNAL SOFTWARE INTERFACE WITH IRDS
- DRAFT STANDARD HAS BEEN COMPLETED BY STANDARDS COMMITTEE X3H4
- SHOULD BE OUT FOR PUBLIC REVIEW IN SPRING 1989

Figure 4

## THE IRDS (Other Standards)

- EXPORT/IMPORT FILE FORMAT -- UNDER DEVELOPMENT
- NAMING CONVENTION VERIFICATION -- UNDER DEVELOPMENT
- MODEL INTEGRATION -- UNDER CONSIDERATION
- DISTRIBUTED HETEROGENEOUS ENVIRONMENT -- UNDER CONSIDERATION

Figure 5

## THE IRDS (NIST Activities)

- IRDS PROTOTYPE
  -- TO BE EXTENDED TO INCLUDE PANEL INTERFACE AND EXPORT/IMPORT FACILITY
  -- CURRENT SOURCE CODE (C INTERFACE TO SQL DBMS, IMPLEMENTING A SUBSET OF THE COMMAND LANGUAGE) IS AVAILABLE FOR OUTSIDE USE AND TESTING
- DEVELOPMENT OF VALIDATION TESTS FOR IRDS IMPLEMENTATIONS
  -- NIST INVITES COOPERATION

Figure 6

# THE IRDS (Documentation Available from NIST)

- A TECHNICAL OVERVIEW OF THE INFORMATION RESOURCE DICTIONARY SYSTEM, Second Edition
  NBSIR 88-3700 (Revision of NBSIR 85-3164)
- GUIDE TO INFORMATION RESOURCE DICTIONARY SYSTEM APPLICATIONS: GENERAL CONCEPTS AND STRATEGIC SYSTEMS PLANNING
  NBS SPECIAL PUBLICATION 500-152
- GUIDE ON DATA ENTITY NAMING CONVENTIONS
  NBS SPECIAL PUBLICATION 500-149

Figure 7

# DATA ENTITY NAMING CONVENTIONS

Judith Newton
National Institute of Standards and Technology
National Computer Systems Laboratory

Naming conventions are guidelines for the format and content of data entity names, and are enforced by the organization's data administrator. They help to establish consistency of data throughout the organization. This results in greater efficiency through reduced data handling as the number of discrete data elements is reduced, and a reduction in confusion among both staff and management, as communication is enhanced. Guidance for developing and applying naming conventions is found in Guide on Data Entity Naming Conventions, NIST Special Publication 500-149, October 1987.

At first glance, data entity names may seem no different from natural language nouns. But they differ from nouns in the same way programming languages differ from natural languages: by the constraints imposed upon them by hardware, software, and human users, and by the possibility for the expression of the organization of the data itself.

Data entity names can reflect the organization of the data both logically, through prime words, and associatively, through class words. Prime words represent the logical groupings of data, such as all information which describes the concept **employee**; class words describe the basic nature of a class of data, such as **name**, **code**, or **date**. Data elements, one type of entity, may need a set of class words to fully describe all elements, while other entities such as file or record may need only one. Modifiers, which establish uniqueness of the data entity name, are the third name component.

While there may be many rules to be established for a set of naming conventions, there are a few guiding principles to follow while writing those rules:

Clarity - names are as clear as possible to a casual user.

Brevity within uniqueness - names are short while still maintaining uniqueness within the database.

Conformance to rules of syntax - each name is in the proper format. If there are too many names which cannot be made to fit the naming conventions, the rules may be too rigorous.

Context-freedom - each name is free of the physical context in which the data entity is implemented.

The IRDS provides a framework for establishing the structure of the names of each entity and the names' relationships to each other, i.e., the metanaming structure. There are three types of names for each entity: access name, descriptive name, and alternate name.

The access and descriptive names are functionally identical, but by providing two names, the IRDS allows them to share the burdens of the guiding principles of clarity and brevity. The access name may be terse, with abbreviations and acronyms but no connectors allowed (for example, EMPLOYEE-NAME), while the descriptive name allows for a longer and more discursive style (NAME OF EMPLOYEE). A user familiar with the database may want to use the access name for retrievals, while a more casual user would prefer the descriptive name. The alternate name may encompass any number of contingencies, such as physical name(s), report header name, and form input name. The majority of this discussion about names is concerned with access name grammar and usage.

The content component of naming grammar has been discussed above; the other component is format. Establishing format rules completes the process by which naming consistency is achieved. For instance, if the prime word is always the first word in the name and the class word last, there is no ambiguity in their identification. Searching by logical group (prime word) or basic nature (class word) is greatly simplified.

Application of naming conventions assists the data administrator in the analysis of data by (for instance) facilitating identification of coupled data elements and their decomposition into atomic data elements; and restructuring data names in which data is mixed in with metadata.

A hierarchy of data elements can be developed based on class words. A "kernel" of class words can be used to form a set of standard or generic elements. These generic elements consist of a class word and modifier combination. Full data elements, called application elements, can then be formed with the addition of a prime word and any extra modifiers as needed. For instance, an application element EMPLOYEE-BIRTH-STATE-NAME is formed of the kernel class word NAME, which is contained in the generic element STATE-NAME; the prime word EMPLOYEE; and the modifier BIRTH.

Descriptive names are derived from access names by casting the access names into natural language grammar and adding connectors as needed. It is important to retain the prime and class words. For instance, EMPLOYEE-BIRTH-STATE-NAME becomes NAME OF BIRTH STATE OF EMPLOYEE.

Like most design activities, the effort expended in advance of the application of data entity naming conventions will pay off over the life of the enterprise.

---

Judith Newton is a computer specialist at the National Computer Systems Laboratory in the National Institute of Standards and Technology. She participates in the American National Standard Committee X3H4 (IRDS) and the ANS Committee X3L8. She is the author of NIST Special Publication 500-149, Guide on Data Entity Naming Conventions.

She is president of the National Capital Region Data Administration Management Association (NCR DAMA) and chair of the NCR DAMA Data Administration Symposium. She leads an International DAMA workshop in Standards and Procedures for Data Administration.

Previously, she worked for the Navy Regional Data Automation Command on development of the RAS data element dictionary, a forerunner of commercial data dictionary systems.

# DATA ENTITY NAMING CONVENTIONS

JUDITH NEWTON

NATIONAL COMPUTER SYSTEMS LABORATORY

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

Figure 1

## WHAT ARE NAMING CONVENTIONS ?

GUIDELINES FOR FORMAT AND CONTENT OF DATA ENTITY NAMES

ENFORCED BY DATA ADMINISTRATOR

## WHAT ARE THEY GOOD FOR ?

CONSISTENCY OF DATA THROUGHOUT ORGANIZATION

MEANS:

- o GREATER EFFICIENCY - REDUCED DATA HANDLING
  SYNONYM RESOLUTION
- o COST SAVINGS - LESS COMPUTING TIME
- o CONFUSION REDUCTION AMONG STAFF AND MANAGEMENT

Figure 2

## GUIDING PRINCIPLES
## FOR RULE DERIVATION

o CLARITY

o BREVITY WITHIN UNIQUENESS

o CONFORMANCE TO RULES OF SYNTAX

o CONTEXT-FREEDOM

Figure 3

## FOUR MAJOR CONCERNS

METHODOLOGY FOR NAME CONSTRUCTION

CONTENT OF NAMES

FORMAT OF NAMES

NAMING CONVENTION ADMINISTRATION

Figure 4

## IRDS NAMES

ACCESS NAME

PRIMARY ID

TERSE

DESCRIPTIVE NAME

LONGER THAN ACCESS NAME

FUNCTIONALLY THE SAME AS ACCESS NAME

ALTERNATE NAME

ATTRIBUTE OF ENTITY

ALIAS OR SYNONYM

Figure 5

## NAMING CONVENTION

## GRAMMAR

- O INFORMATION CONTENT

- O FORMAT

Figure 6

# NAMES USED TO EXPRESS DATA ARCHITECTURE

- o LOGICAL DATA MODEL

- o CLASSIFICATION OF DATA ENTITIES

Figure 7

# LOGICAL GROUPINGS
## in the logical data model

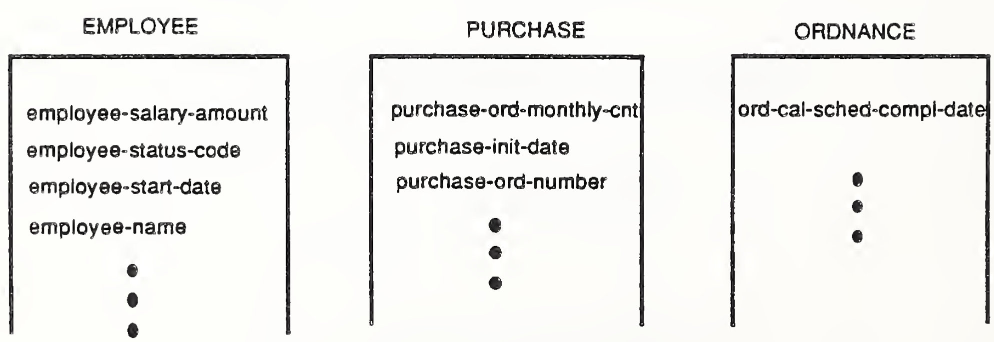



Figure 8

## FORMAT OF NAMES

ALLOWABLE CHARACTERS

LENGTH (MIN/MAX)

ABBREVIATIONS/ACRONYMS

CONNECTORS/MODIFIERS

WORD FORM

CAPITALIZATION

WORD ORDER

Figure 9

## NAMES AND DATA CLASSIFICATION

**DATA ENTITIES MAY BE CLASSIFIED IN CATEGORIES SUCH AS:**

FOR DATA ELEMENTS:

NAME
AMOUNT
DATE
TEXT

FOR OTHER ENTITIES:

FILE

RECORD

DATABASE

...ETC.

Figure 10

# TOOLS

STANDARD ABBREVIATION LIST

STANDARD ACRONYM LIST

ALLOWED WORD LISTS

THESAURUS

GLOSSARY

GUIDANCE FROM DA

DATA DICTIONARY/DIRECTORY

Figure 11

# AND IN CONCLUSION

GOALS: CONSISTENCY AND EFFICIENCY - COST SAVINGS AND CONFUSION REDUCTION

PRINCIPLES:

CLARITY, BREVITY, RULE CONFORMANCE, CONTEXT-FREEDOM

Figure 12

# IRDS Export/Import Facility

Margaret H. Law
National Institute of Standards & Technology

The Export/Import Facility of the Information Resource Dictionary System (IRDS) is under development in the X3H4 Committee responsible for the American National Standard IRDS. X3H4 is part of the X3 Committee that operates under the auspices of the American National Standards Institute (ANSI).

The National Institute of Standards and Technology (NIST) actively participates in X3H4 and has played a key role in the development of the IRDS. The planned IRDS Export/Import Facility is one area in which NIST is actively involved. NIST is participating in developing Export/Import specifications for the IRDS, and is developing a prototype to demonstrate this interchange concept.

While the content of a dictionary or repository is often referred to as **data**, technically it should be called **metadata**, or descriptive "data about data." To reflect this terminology, information exchange between dictionaries or repositories should be called **metadata interchange**, not data interchange. Data interchange among databases running on database management systems (DBMSs) is significantly different from metadata interchange among dictionaries, repositories, and CASE tools. Data interchange is supported by standard query languages such as the Structured Query Language (SQL) and the Network Data Language (NDL). Metadata interchange will soon be supported by the standard repository interchange method, the IRDS Export/Import Facility and File Format.

The planned IRDS Export/Import Facility impacts CASE tools in that it provides a potential mechanism for CASE metadata interchange and integration. For repository-based CASE tools, the IRDS Export/Import Facility will provide a neutral method of metadata interchange that does not depend on a particular, predefined schema.

The functionality of the IRDS Export/Import Facility is based on: (1) the IRDS meta-schema constructs, and (2) the extensible schema capability for Information Resource Dictionary (IRD) applications.

The IRDS is designed with a top level of meta-schema constructs (in the schema description layer) that are used to build schemas for each IRD application. These meta-schema constructs also provide a flexible foundation on which communications protocols can be built. The IRDS Export/Import Facility will use Abstract Syntax Notation One (ASN.1), a protocol language and representation method used to support the presentation and application layers of Open Systems Interconnection (OSI).

The IRDS meta-schema constructs can provide a "foundation" for metadata interchange because they are a finite group of structures that can be coded into protocols; they can provide a "flexible foundation" because they can be used to describe a wide variety of structures in any application schema layer. The flexibility of the IRDS meta-schema layer directly supports the extensible schema capability of every IRD application.

Several aspects of the IRDS Export/Import Facility are discussed:

- o Proposed IRDS Export/Import File Format, which has been specified and defined in ASN.1.

- o Limited IRDS interchange functionality that now exists, to export a schema and metadata to an intermediate file, check schema compatibility between the target and the intermediate file, and to import only the metadata into the empty partition of the target IRD.

- o Additional IRDS interchange functionality that is envisioned to support continuing metadata interchange among multiple IRDS and CASE tools.

The proposed IRDS Import/Export File Format, based on ASN.1, is expected to be approved by X3H4 in 1989, and by ANSI in the early 1990's. This repository interchange file format will provide a mechanism for exchanging both schema and metadata information among tools. The IRDS Export/Import File Format is eagerly awaited by users anxious to interchange metadata. To release the file format as quickly as possible, X3H4 has separated the interchange file format from the definition of additional IRDS export/import functionality, which will require extensive work.

The existing IRDS interchange functionality is discussed in terms of its limitations. For the existing IRDS interchange functionality, it is awkward that the schema exported from the source IRD cannot be imported into the target IRD. It is also awkward that subschemas cannot be defined in an IRD, so they cannot be exported or imported at this time. The empty partition in the target IRD can be empty only once, so that a dictionary administrator must move metadata laboriously from partition to partition to effect dictionary integration.

Plans are described for extending this functionality with an IRDS Export/Import Facility. Schema subsetting functionality is required so that subschemas can be defined, selected, exported, and imported. IRD imports should be able to be received in a non-empty partition of a target IRD, so that IRDS functionality can help support the process of schema integration. The role of the IRDS command for "check schema compatibility," and the role of versioning mechanisms for export/import are discussed.

Finally, the real world problems of repository interchange are addressed, with emphasis on the problem of schema integration. Metadata interchange is only part of the problem. What do you do when the schemas of the source and target dictionaries are not compatible? The valuable efforts of the X3H4 working group on Schema Integration are mentioned.

---

Dr. Law is a member of the Data Administration Group of the Information Systems Engineering Division of the National Institute of Standards and Technology (NIST). She participates in the X3H4 Information Resource Dictionary System (IRDS) Export/Import Facility working group, and is involved in developing the IRDS Export/Import File Format prototype at NIST. She initiated the Federal CASE Conference Series and is a Program Co-Chair for FedCASE'89, addressing "Integrated Data Management for Software Engineering." Margaret has authored a publication that demonstrates the use of the extensible schema capability of the IRDS -- <u>Guide to Information Resource Dictionary System Applications: General Concepts and Strategic Systems Planning</u>, NIST Special Publication 500-152, 1988. She has also co-authored <u>Guide to Data Administration</u>, soon to be released as a NIST Special Publication, and <u>Guide on Logical Database Design</u>, NIST Special Publication 500-122, 1985.

# IRDS
# Export/Import Facility

MARGARET H. LAW

NATIONAL COMPUTER SYSTEMS LABORATORY

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY

Presented to DAMA
Second Annual Symposium
May 3, 1989

Figure 1

# IRDS Export/Import Facility

- Under development in X3H4 (ANSI)
- Impacts CASE and Data Dictionary/Directory tools
  - o Potential mechanism for CASE data interchange and integration
- Several aspects of Export/Import Facility
  - o IRDS Export/Import File Format
  - o Existing IRDS Interchange Functionality
  - o Additional IRDS Interchange Functionality

Figure 2

# Relation of IRDS to CASE

- IRDS provides a standard for the type of data dictionary system software that underlies computer-aided software engineering (CASE) tools

- IRDS features exceed the functionality of many CASE tools

  o Extensible schema capability

  o Extensible lifecycle phase partitioning

  o User-defined views within lifecycle phases

  o Proposed Export/Import File Format for standardization

Figure 3

# IRDS Export/Import File Format

- Uses Abstract Syntax Notation One (ASN.1), a standard communications protocol with encoding rules

  o ASN.1 is consistent with Open Systems Interconnection (OSI) as an Application Layer and a Presentation Layer Protocol

  o ASN.1 is an international standard approved by the International Organization for Standardization (ISO)

    - ISO 8824 and ISO 8825

Figure 4

# IRDS Import/Export File Format

(continued)

- o Based on IRDS standard schema description constructs
- o Supports interchange of both IRD schema and metadata, as defined and selected by user
- o Expected to be completed by subcommittee in 1989, approved by X3H4 in 1990, and approved by ANSI in 1990
- o Additional IRDS Export/Import functionality will take longer

Figure 5

# Information Resource Dictionary System

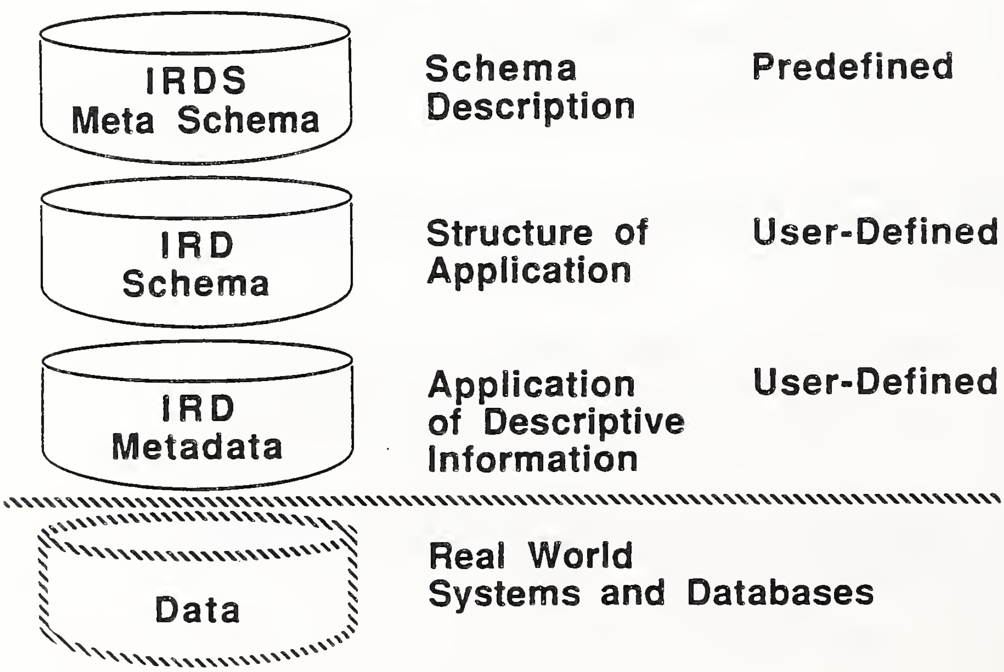



Figure 6

# Existing
# IRDS Interchange Functionality

- o Schema and metadata export from any source Information Resource Dictionary (IRD) application
  - Supported by Export/Import File Format expected to be approved in 1990
- o Source IRD schema is exported from source IRD, but is not imported into target IRD
- o Check IRD Schema Compatibility procedures used to identify schema discrepancies between source and target IRDs

Figure 7

# Existing
# IRDS Interchange Functionality

(continued)

- o Metadata import into an empty (i.e., without metadata) life cycle phase partition of the target IRD
  - Problem: Partition can be empty only once
  - Additional import functionality planned
- o Administrator of target IRD has to manually merge the imported dictionary with contents of other life cycle phase partitions
  - Additional integration functionality planned to provide automated support

Figure 8

# Existing Export/Import Functionality

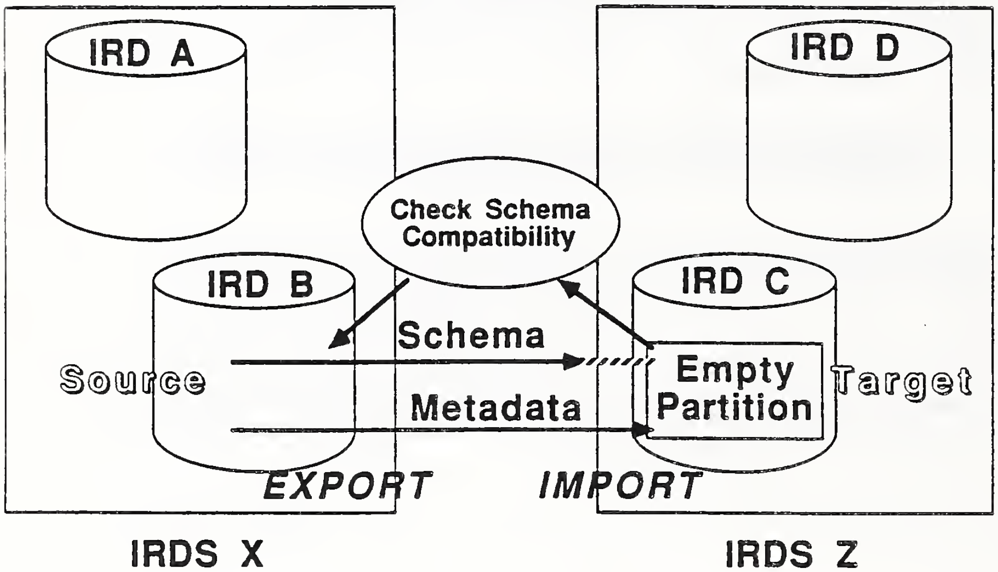



Figure 9

# Proposed IRDS Interchange Functionality

- o Identification and exchange of schemas among IRDs
- o Support for multiple, sequential interchanges among IRDs
- o Identification and exchange of subschemas among IRDs, so that only the relevant part of the source schema must be transferred
- o Registration of subschemas to control multiple subschema interchange

Figure 10

# Proposed IRDS Subschema Subsetting Functionality

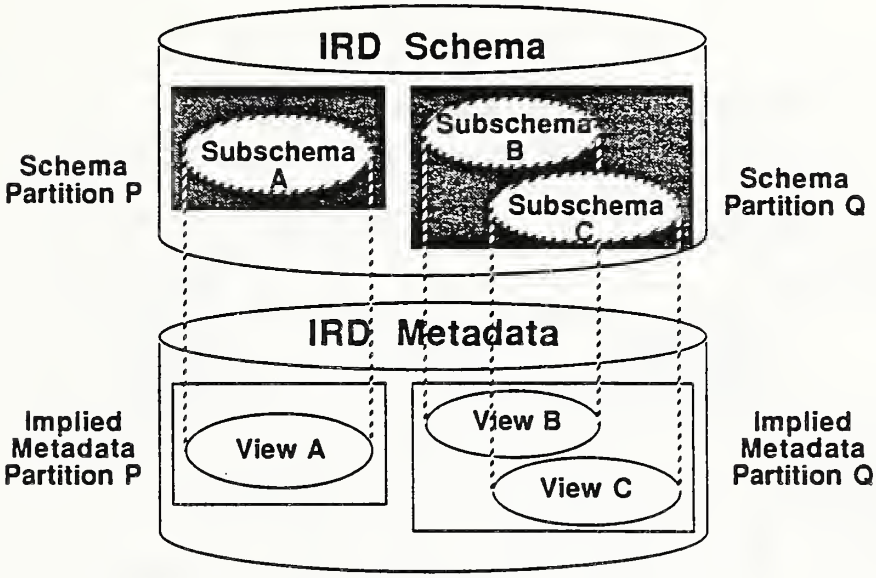



Figure 11

# Proposed IRDS Interchange Functionality

(continued)

- Versioning control for multiple schema, subschema, and metadata interchanges among multiple systems and IRD applications
- Procedures for importing schemas, subschemas, and metadata into non-empty IRD partitions (i.e., with pre-existing metadata)
- Support for logical deletion, as is already provided for addition and modification
- Procedures for interchange with non-IRDS (untrusted) vs. IRDS (trusted) systems

Figure 12

# Proposed Export/Import Functionality

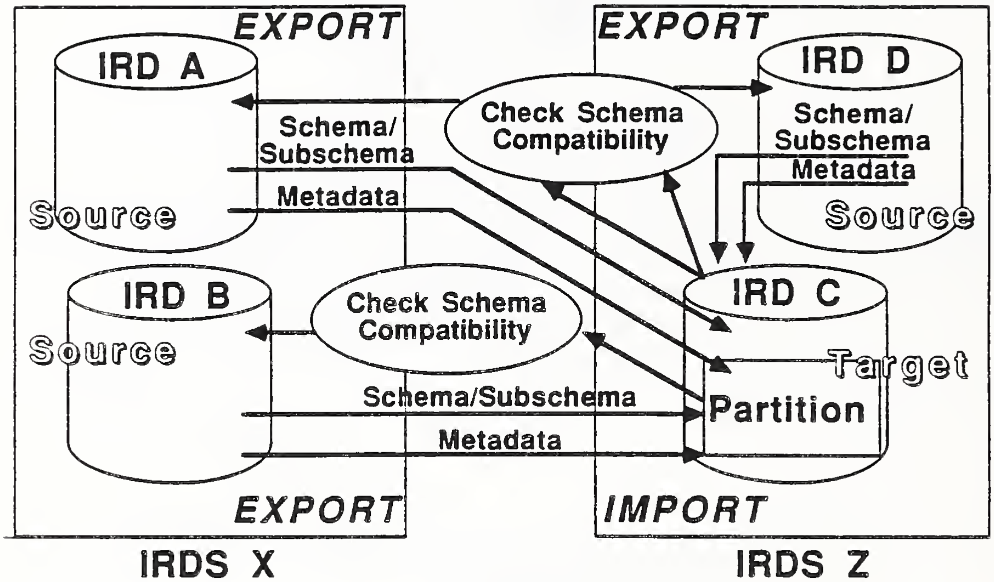



Figure 13

**TECHNIQUES: DATA INTEGRATION ISSUES IN SYSTEMS DEVELOPMENT**

SESSION CHAIR

David R. Skeen
Department of the Navy

PANELISTS

Anthony J. Winkler
Harold Boylan

# DATA INTEGRATION ISSUES IN SYSTEMS DEVELOPMENT

David R. Skeen
Department of the Navy

Within the decade of the 80's, the importance of data as a resource has become one of the challenges which organizations must solve. With the advent of the personal computer and other techniques, the end-user has now become the change agent, not the information systems developer. Managing data and realizing the organizational opportunities available to the end-user are ingredients which must be harnessed as we go into the decade of the 90's. However, before this trend is realized, organizations must solve the inevitable fragmentation of their information in a distributed information systems environment. The two key ingredients which must be managed in the systems development process are the data and its interface, i.e., data communications.

One of the biggest hurdles which an organization must overcome is to realize that its culture must change. Top-level management, end-users and information systems personnel must change their perspective before an information environment can be obtained. Within the Department of the Navy, we are progressing with an approach known as the Data/Technology Strategy. Its primary principle is to place the data before technology. This strategy is the first step in progressing toward the Navy's Corporate Information environment. It requires an understanding of the corporate's business and its information flows, involvement of the functional managers, and managing data as a resource. It suggests a data-driven solution to the company's mission. The technological infrastructure to their strategy requires that the information systems function control data and its data communications resources. Other tenets include: integrating data to understand the company's information systems; understanding that end-user computing is critical; automation of data at the source where its value is recognized; and development of corporate-wide database strategies.

In the Navy, a methodology has been developed which entails four layers of information architectures which begin with the company's mission and functions. These four architectures are: Information Flow Architecture or Business Processes, Data Architectures, Data Base/Applications Architecture, and Technical Architecture.

The specific products of each architecture are listed below:

1. Information Flow Architecture
   - Corporate-wide organizational structure
   - Business Process
   - Detailed Information Flows

2. Data Architectures
   - Functional Data Model
   - Logical Data Flows

3. Data Base/Applications Architecture
   - Corporate Data Bases
   - Applications Information Systems

4. Technology
   - Data Communications
   - Information Systems Facilities
   - Hardware
   - Systems Software, including Database Management Systems

A key management strategy for integrating data is the Data Base/Application Architecture which uses the traditional Management Information Systems (MIS) "triangle" to relate the company's databases and their usefulness to the three layers of management, i.e., Strategic Control, Management Control, and Operational Control. The five types of databases can be categorized into Corporate, Decision Support, Executive, Departmental MIS, and Field Systems/Data Collection. Each database satisfies various levels of management but work together to form the organization's total MIS.

Once an organization is structured to accommodate such a philosophy, several key aspects must be addressed before an organization can realize the value of its information resource. Such aspects can be listed under three categories: Management, Data Management, and Information Systems and Technology. The more important elements are usually those associated with Management, such as, Top-Level Management Support, Corporate Planning, Life-Cycle Management including performing information benefits analysis, and positioning the organization's structure to move into the information era. How management introduces such a data-driven philosophy into the organization is crucial to its success.

Other aspects which must be addressed can be categorized under Data Management. Such elements include: Data Standardization, how the business is decomposed or described and documented, and how data are integrated. The first three architectures described above are developed within this data management task.

Once the Management and Data Management programs have been established, the Information Systems and Technology can be addressed. The Data Base/Applications Architecture is the bridge between the organization's data and its information systems infrastructure. This architecture is the "plan" for integrating an organization's data.

It is critical that an organization realize the value of data, its relationship to the structure of the business and its mission, and how to develop a strategy for its integration. This is the real challenge!

---

David R. Skeen is the Director, Total Force Information Resources and Systems Management Division, Office of the Deputy Chief of Naval Operations (Manpower, Personnel and Training). Mr. Skeen is directly responsible for the Navy's Information Systems which support manpower, personnel, and training functions.

Mr. Skeen is an associate professor at the School of Engineering and Applied Science, George Washington University, where he teaches courses in Management and Information Resources and Data Communications. Mr. Skeen has published several articles, developed and presented training curricula, and has lectured extensively at international and national computer conferences.

He is the past-President of the Federal ADP Users Group (FADPUG) which has over 3,000 Federal ADP managers and senior technicians as members. In 1979, Mr. Skeen participated on the Personnel Task Team of President Carter's Reorganization Project for Data Processing.

## CRITICAL FACTORS OF THE DATA/ TECHNOLOGY STRATEGY

FUNCTIONAL (DATA) VIEW

- FUNCTIONAL MANAGERS' DIRECTION AND INVOLVEMENT
- MANAGING DATA AS A RESOURCE
- DESCRIBING THE BUSINESS AS A "WHITE COLLAR" BUSINESS (THE FIRST STEP OF TQM)
- DECOMPOSING THE BUSINESS USING ARCHITECTURAL TOOLS AND TECHNIQUES
- MANAGE INFORMATION AS A FORCE MULTIPLIER

Figure 1

## STEPS TO A DATA RESOURCE MANAGEMENT PROGRAM FOR LARGE ORGANIZATIONS

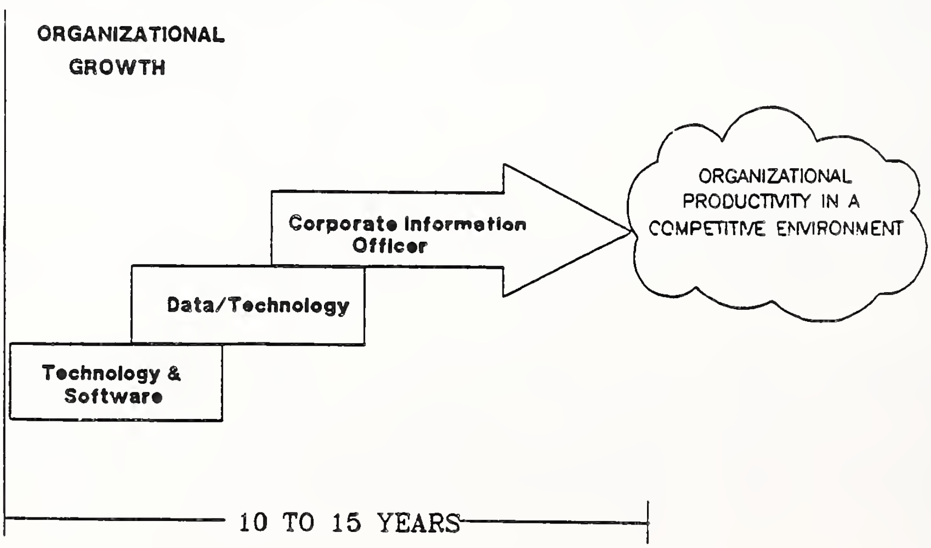



Figure 2

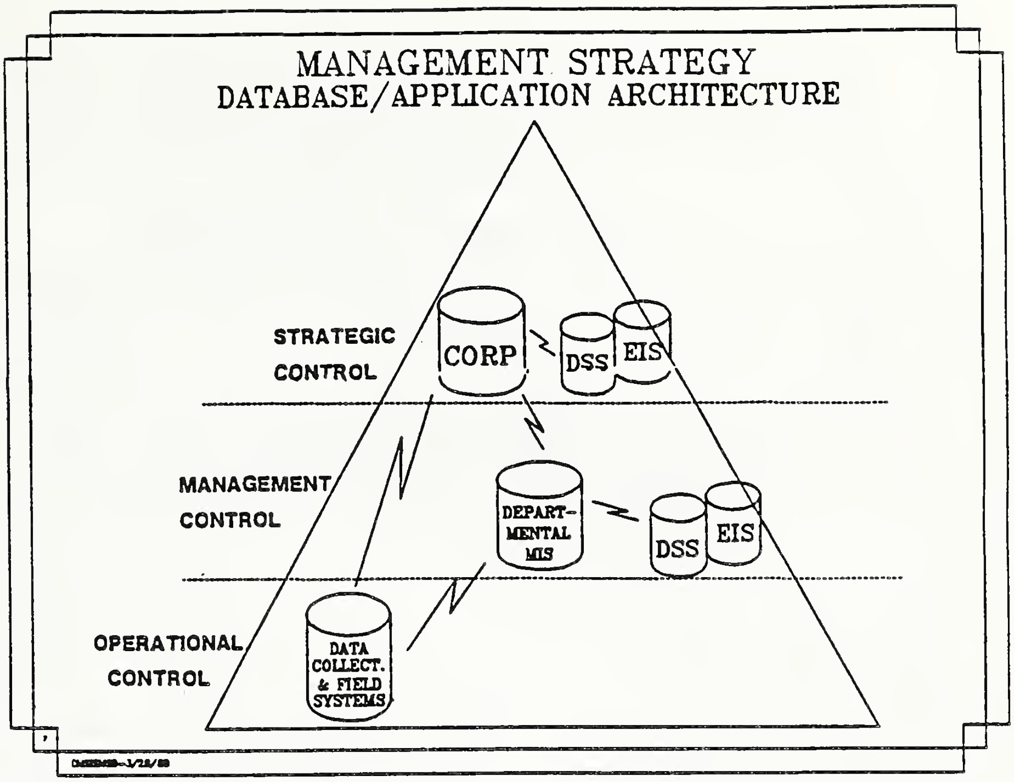



Figure 3

## CRITICAL FACTORS OF THE DATA/ TECHNOLOGY STRATEGY (CON'T)

<u>TECHNOLOGY</u>

- CONTROL DATA AND DATA COMMUNICATIONS
- INTEGRATE DATA AND INFORMATION SYSTEMS WILL FOLLOW
- ENCOURAGE END-USER COMPUTING
- AUTOMATE SOURCE DATA ENTRY
- DEVELOP CORPORATE-WIDE DATA BASE STRATEGIES

Figure 4

# THE REAL CHALLENGE

HOW TO ESTABLISH A PROCESS WHICH RECOGNIZES THE VALUE OF DATA WHILE ENSURING QUALITY SYSTEMS SUPPORTS WITHIN THE ORGANIZATION'S CORPORATE STRATEGY.

Figure 5

# DATA INTEGRATION ISSUES IN SYSTEM DEVELOPMENT

Dr. Jerry Winkler
CTA INCORPORATED

The objective of Data Integration is to assure that all data acts together in such a way as to appear to be a single complete unit. This presentation examines the issues concerned with achieving this objective. Principally, these issues can be categorized as management, technical and legacy. In the management arena, the function normally referred to as Data Administration or more globally as Information Resource Management is concerned with standards and control of the data resource. But what does this imply? In the technical area, the complexity of the data integration issue becomes more relevant. While aspects of this problem are normally under the auspices of the function often referred to as Database Administration, which is normally restricted to the design, integrity and performance of database management system (DBMS) implementations, the problems are much broader in the distributed heterogeneous environments in which many of us find ourselves today. The legacy issues are just that; there are often many existing data sources that exist in a number of different forms which have not been subject to enforced standards and control. Successful data integration must accommodate legacy environments.

## Management as a Data Integration Issue

Why is management of the data resource a data integration issue? Consider that systems development, when it occurs in an environment in which data is not managed, is like attempting to communicate in an environment without a common vocabulary. In such an environment, data integration is often illusory, and system integration is a pipe-dream.

What are the principal characteristics of this managed data environment? First, the most critical "element" of this environment is the data element; data element names are the vocabulary of systems. If one is to manage this environment:

- o All data elements must be identifiable and identified.
- o All data elements must be named according to a naming standard.
- o Synonyms and homonyms of data element names must be recognized as such.

- o The relationships that exist between data elements and other information resources must be known and documented.

It should be apparent that in order to be managed, the object that is to be managed must be identifiable and identified. How the object is identified (i.e., named) must be based on a standard approach, otherwise one may be creating new names for the same object, or using duplicate names for different objects. This is one of the reasons why Data Administration is the headache it is for the legacy. Synonyms and homonyms exist because of the lack of naming standards; of course, synonyms exist in the English language, so it may be impossible to eliminate all synonyms and homonyms, but it is critical that they be identified as such. Knowledge of relationships is necessary, because of higher level issues concerned with, for example, design of files, databases and distributed architectures.

**Technical Aspects of the Data Integration Issue**

Often, one considers that the data integration issue in systems development involves only using the proper data to produce the desired information. This perspective is very narrow and eliminates, out of hand, the myriad of considerations that occur during system development. In the attached set of transparency masters, the one titled "Data to Information Transformation Aspects" (fig. 7) is intended to depict the complexity masked by the simplistic view of the situation. In this figure, from the top:

- o The presentation aspects involve providing the end product of the information production process to the customers. Considerations in this process are:

  - Human factors.

  - The content of the message the customer is expecting.

  - The purpose of receiving the content; i.e., what action is to be taken.

  - Constraints concerning presentation, e.g., the device or the time sensitivity of the information.

- o The processing aspects are concerned with the types of processing required to obtain the data/information necessary to prepare the desired information. These aspects include:

- How does one identify specific data/information within the system, based on non-specific identification by the end-user?

- How does one find the specific data/information once it has been identified?

- How is the data, once located, to be transported to the requestor?

- Is it necessary to transform the data either at its source or its destination in order to be used at the requestor's site? Potential transformations might be summarization, translation or fusion.

- How should the data be presented to the processes that are concerned with presentation to the requestor?

o The data storage aspects are concerned with the knowledge about the data that applications must possess in order to process it. These include:

  - How is the data structured and what is the impact of that structure on the semantics of the data?

  - What is the storage media of the data and how does this influence its accessibility?

  - Where is the data stored, e.g., locally or remotely?

  - Are there access control restrictions regarding accessibility of the data?

  - Are there special access mechanisms, e.g., indexes, to facilitate access?

  - Is the data encoded or compressed?

o Finally, the source data aspects are concerned with the class of data. These are important considerations because of their impact on processing, especially in heterogeneous environments.

## Legacy as a Data Integration Issue

A legacy of applications, procedures, forms, data files, databases, etc., exists within 99.9% of today's information environments. Normally, this legacy is not pure in terms of naming or other standards. This fact makes transition to a managed data environment even more costly than it might have been, and the cost does not reduce over time. It is always more expensive later.

## What to do?

It is important that organizations realize that postponement of moving to a managed data environment is like riding the crest of a wave; eventually, the wave will collapse and come crashing down. Thus it is important to recognize that data, information, systems, etc., are all information assets that should be managed. In order to manage these assets, it is necessary to establish objectives, allocate resources to managing the assets, and then proceed to manage them. This should precede systems development.

It is important that technology be used to support this management. A kernel technology is that represented by the American National Standard for Information Resource Dictionary Systems (IRDS, ANSI/X3.138-1988), which is also a Federal Information Processing Standard (FIPS) Publication 156. Additional technology is necessary to support the management of these assets. Such technology would support naming standards and synonym/homonym resolution. These features do not currently exist within the IRDS standard, but it is expected that they will be a future capability since the need is well-recognized.

---

Dr. Jerry Winkler is Chair of the American National Standards development technical committee responsible for Information Resource Dictionary Systems (IRDSs). He is a Chief Engineer with CTA Incorporated of Rockville, Maryland. He is principally involved with two NASA projects -- one involves integrating IRDS and Open Systems Interconnection (OSI) directory service technologies to provide access to any object of interest in the Space Station Freedom Program; the other involves developing standards for automated interchange of international space data.

# DATA INTEGRATION ISSUES IN SYSTEM DEVELOPMENT

DATA ADMINISTRATION:
STANDARDS AND TECHNIQUES SYMPOSIUM

Presented by
Dr. Jerry Winkler
Chair, ANSC/X3H4

CTA INCORPORATED
McLean, VA 22102

May 3, 1989

Figure 1

PROVIDING INFORMATION—THE SIMPLIFIED VIEW

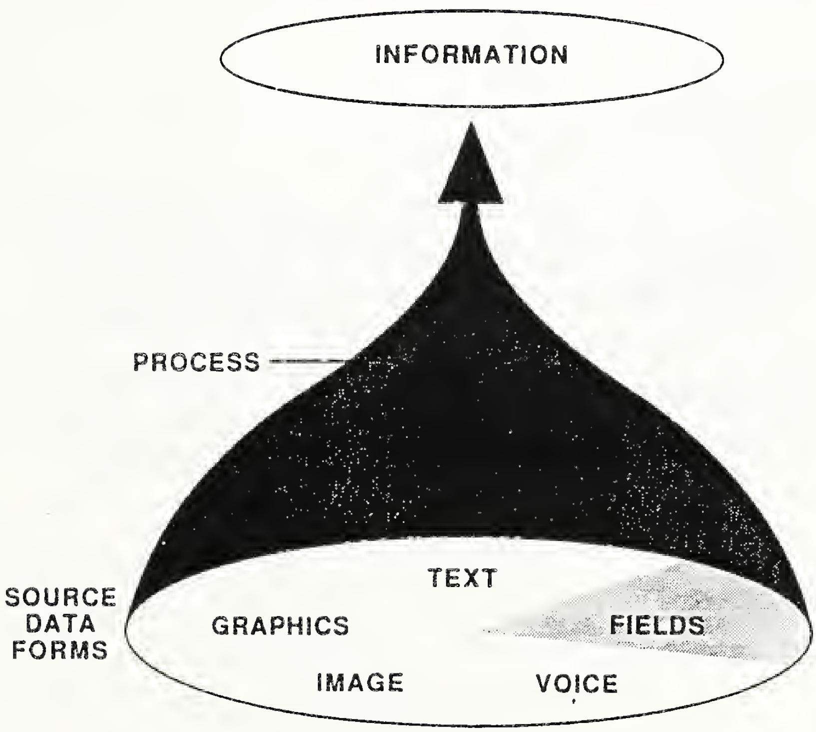



Figure 2

## DATA INTEGRATION

ALL DATA ACTING TOGETHER TO ACHIEVE
A SINGLE COMPLETE UNIT.

Figure 3

## IMPLICATIONS OF DATA INTEGRATION

- STANDARDS
- MANAGEMENT AND CONTROL
- EACH UNIQUE DATA ELEMENT IS UNIQUELY IDENTIFIABLE
- MUST APPEAR AS A WHOLE—NO REPLICATION OF DATA INSTANCES
- PRACTICALITY—MUST ALLOW FOR:
  - LEGACY
  - PURPOSEFUL REPLICATION OF DATA INSTANCES

Figure 4

## OBSERVATIONS

- DATA INTEGRATION OCCURS ONLY IN A MANAGED DATA ENVIRONMENT
- SYSTEMS DEVELOPMENT WITHOUT A MANAGED DATA ENVIRONMENT IS LIKE ATTEMPTING TO COMMUNICATE WITHOUT A COMMON VOCABULARY
- SYSTEMS INTEGRATION CANNOT OCCUR WITHOUT DATA INTEGRATION

Figure 5

## WHAT IS A MANAGED DATA ENVIRONMENT

- ALL DATA ELEMENTS ARE IDENTIFIABLE AND IDENTIFIED
- ALL DATA ELEMENTS ARE NAMED ACCORDING TO A NAMING STANDARD
- SYNONYMS AND HOMONYMS ARE RECOGNIZED AS SUCH
- RELATIONSHIPS BETWEEN DATA ELEMENTS AND OTHER RESOURCES ARE KNOWN AND DOCUMENTED

Figure 6

## DATA TO INFORMATION TRANSFORMATION ASPECTS

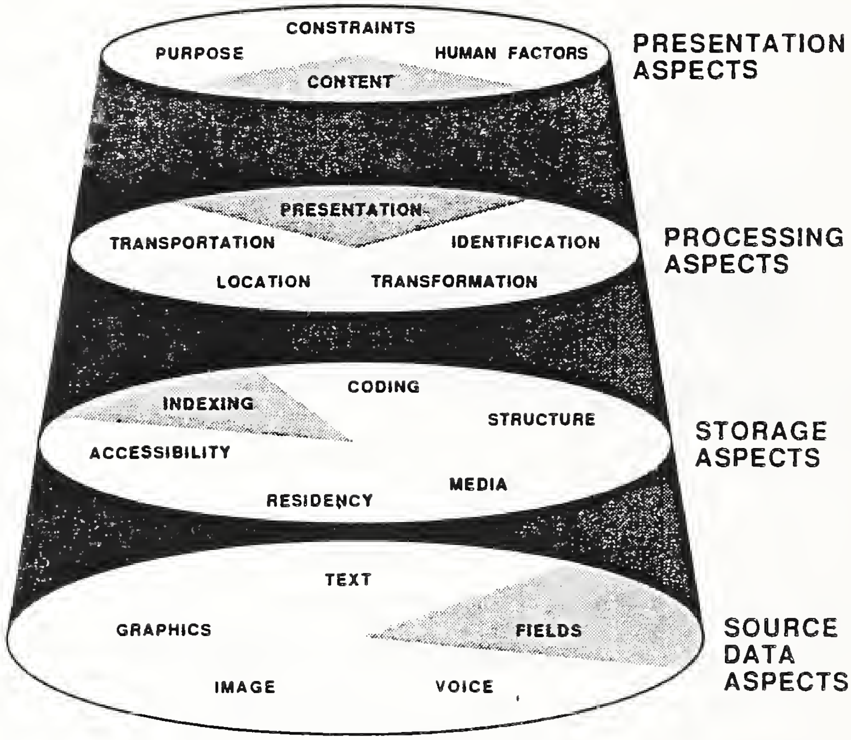



Figure 7

## ANSWERS TO ISSUES

- REALIZE THAT DATA, INFORMATION, SYSTEMS, ETC. ARE ASSETS
- DEFINE OBJECTIVES, PLAN, ALLOCATE RESOURCES, MANAGE
- TECHNOLOGY SUPPORT
- "PAY ME NOW OR PAY ME LATER"

Figure 8

## PURPOSE OF THE IRDS

- PROVIDE A COMMON SOURCE FOR UNDERSTANDING THE INFORMATION ENVIRONMENT OF AN ORGANIZATION
- PROVIDE A TOOL FOR MANAGING THE INFORMATION RESOURCE ASSETS OF THE ORGANIZATION
- PROVIDE AN INVENTORY SYSTEM FOR THE INFORMATION ENVIRONMENT

Figure 9

## ANSI X3.138-1988

- THE SPECIFICATION FOR AN AMERICAN NATIONAL STANDARD (ANS) INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS)
- CORE MODULE DEFINES THE IRDS
  - FUNCTIONS IN TERMS OF A COMMAND LANGUAGE AND PANEL INTERFACE
  - UNDERLYING DATA MODEL:
    - INFORMATION RESOURCE DICTIONARY (IRD)
    - IRD SCHEMA
    - IRD SCHEMA DEFINITION
- NON-CORE MODULES SPECIFY
  - A BASIC FUNCTIONAL SCHEMA
  - THE IRDS SECURITY MODULE
  - THE EXTENSIBLE LIFE CYCLE PHASE FACILITY
  - IRDS PROCEDURES
  - THE APPLICATION PROGRAM INTERFACE
  - ENTITY LISTS
- DOES NOT ASSUME AN IMPLEMENTATION ENVIRONMENT

Figure 10

## IRDS—THE FUTURE

- IRDS REFERENCE MODEL
- NAMING CONVENTION SUPPORT — TECHNICAL REPORT/STANDARD
- DATA MODEL INTEGRATION — TECHNICAL REPORT
- IRDS IN A DISTRIBUTED HETEROGENEOUS ENVIRONMENT — TECHNICAL REPORT
- CASE TOOL DATA MODEL
- N-ARY INTERFACE/DATA MODEL

Figure 11

# DATA INTEGRATION ISSUES IN SYSTEMS DEVELOPMENT

Commander Harold Boylan
Department of the Navy

Of fundamental concern is not so much whether one should or should not integrate data, but rather the need to establish a management process that recognizes the value of data to functional decision makers. It is this information resource management focus on data and its role in directing and controlling the organization that results in strategies and specific actions. The extent that data integration belongs as part of a strategy to improve the quality of data available to decision makers determines its relative importance and justifies the substantial resource commitments required.

A simple economic approach to determining the value of data is to determine what one is willing to pay for it. The converse of this approach is much more difficult to comprehend; that is, determining the cost of not having data or of having poor quality data. As an illustration, the Navy maintains up to eleven separate systems to collect data from and provide some support to its field personnel and pay offices. These systems have evolved over a number of years, and for the most part, do what they were designed to do at reasonable visible cost. Much of the data that are input to these systems are duplicative. In addition to the cost of data entry, multiple communications capabilities, and multiple databases, there are significant configuration management problems imposed when requirements must be orchestrated across multiple organizations. However, the greatest costs of poor data to the Navy, in this example, is not in the information systems budget, but in the systemic inefficiencies of managing a work force of two million people. It is the management information produced by these data systems that has the biggest impact on utilization of manpower resources, including decisions made in recruiting, retention, targeted pay policies, training, promotions, separations and the management of the $17 billion appropriation of military pay.

The first issue in data integration, therefore, concerns the scope and role of integration within the context of IRM policies and strategies to improve the quality of data provided to decision makers. These policies and strategies must be driven by a good understanding of the organization's mission, how the organization consumes resources, who makes the real decisions about those resources, and how the information flows or doesn't flow. Navy IRM strategies focus on centralized management of data and communications and decentralized, to the maximum extent feasible, technology supporting specific functional applications. Centralized management of data includes integration of data within the boundaries of major policy and resource management responsibilities.

If integration is justified in the realization of systemic efficiencies to the organization, the second issue deals with determining what specific data should be integrated. A fairly safe principle (and a reasonable place to start) is that one must first integrate metadata before attempting to integrate actual data. This issue essentially is one of redefining, redesigning, and reorganizing data elements used by multiple organizational areas. Categorizing and standardizing data elements based on "subject areas" of the business independent from specific functional uses or existing information systems provides the basic framework for integration and is the only practical way of minimizing the inherent political problems of data ownership. It is particularly important to analyze separate processes that collect the same data, separate validations of the same data, separate storage of the same data, and separate sources that distribute the same data. Frequently, to determine the "same" data requires looking again at the real world thing or event the data are attempting to represent. This data approach can have major impact on systems design since it forces one to rethink basic central control processes (such as record gains and losses), to review basic business rules and transaction design, and to engineer more generic functionality within the system. Because of the magnitude of change implied by this approach, ultimate constraints on what can be done and when it can be done may be driven by the transition strategy necessary to move from the existing systems environment to the integrated environment. Navy experiences in large scale data modeling have shown significant reductions in the number of data elements, simplified validation and control processes, more flexible response to new requirements, and data quality improvements are achievable, but the transition in information systems must occur in a modular and evolutionary fashion.

The final set of issues are administrative in nature and represent organizational barriers which must be overcome to achieve systemic efficiencies through data integration. These issues involve direction, commitment, division of labor, and a tolerance of change. Direction and commitment imply a shared vision of the future and a realistic expectation of progress. Traditional systems design must give way to shared roles by application specialists, data administrators, and data base administrators. Each party must be willing and able to lead different phases of projects and act as change facilitators with functional counterparts.

New tools and methodologies must be assimilated in the way business is done. Performance criteria and reward structures must begin to reflect desired outcomes and encourage cooperative and creative approaches. Finally, basic information systems decision processes such as Life Cycle Management must be expanded to build upon the data and communication infrastructure, incorporate information benefits analyses or other methods to assess the quality of data and its impact on decisions, and to more closely tie together the business needs, data requirements and information systems design.

---

Commander Boylan is currently assigned to the staff of the Director, Department of the Navy Information Resources Management Office. His primary responsibilities include strategic planning, architectures and data administration. With over twenty years in the Navy, he has gained extensive experience in the development, operation, and management of advanced technology systems. In his previous assignment to the Deputy Chief of Naval Operations, Manpower, Personnel, and Training, he built the Navy's first large scale, centralized Data Resource Management Program to support the management of two million active duty, reserve, retired, and civilian personnel. He has served as Project Manager in the design and development of centralized and distributed information systems.

CDR Boylan is also a Navy pilot with substantial operational experience in the command and control of airborne weapon systems to provide direct fleet support and collect intelligence data.

CDR Boylan graduated from the U.S. Naval Academy in 1968 with a BS in systems engineering. He has a MS degree in computer systems management from the Naval Post Graduate School and has completed two years of graduate work in financial management at George Washington University. He teaches part time at George Washington's School of Engineering and Applied Science.

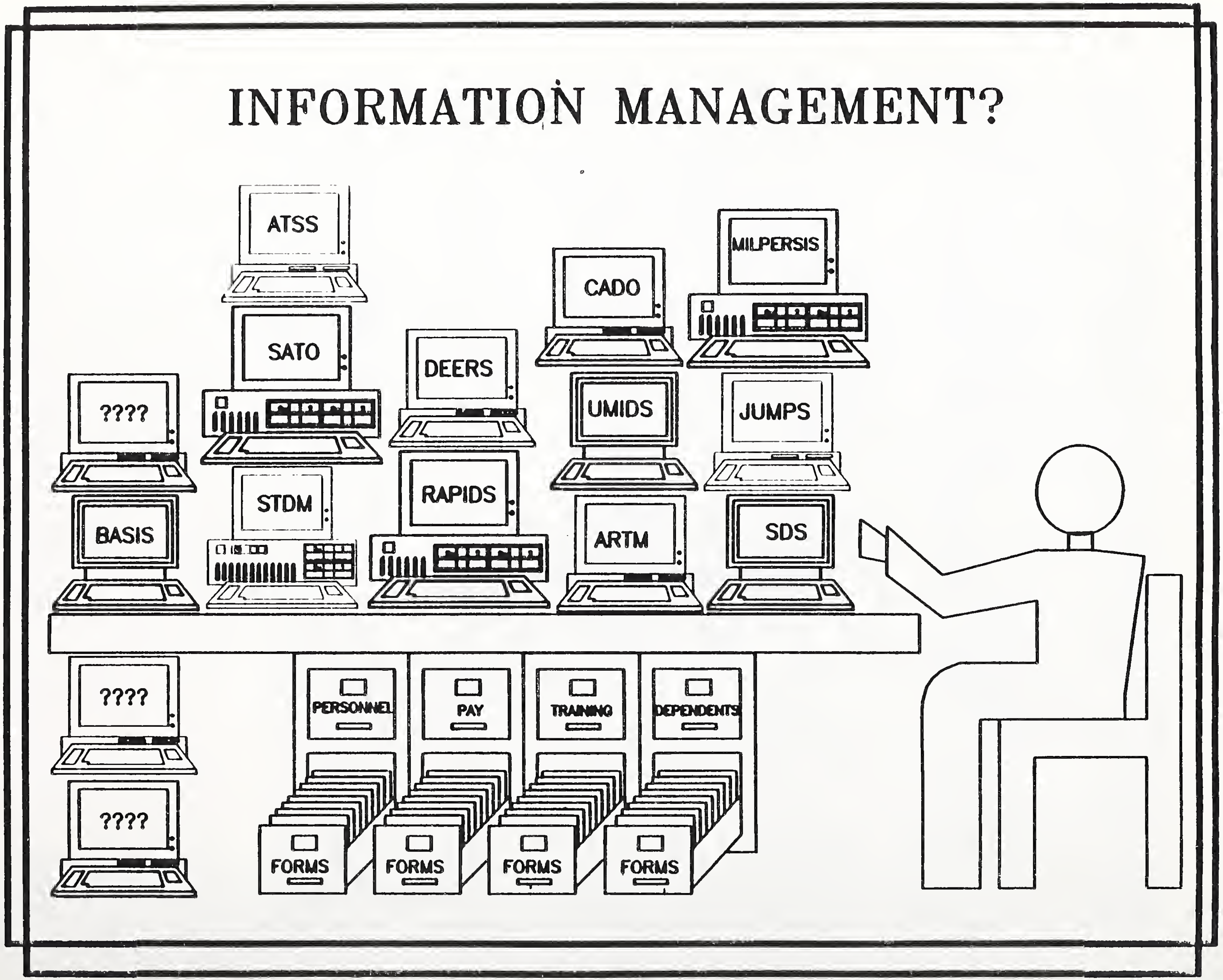

INFORMATION MANAGEMENT?
ATSS
SATO
STDM
DEERS
RAPIDS
CADO
UMIDS
ARTM
MILPERSIS
JUMPS
SDS
????
BASIS
????
????
PERSONNEL
PAY
TRAINING
DEPENDENTS
FORMS
FORMS
FORMS
FORMS

Figure 1

# MPN End Strength Accounting & Reporting

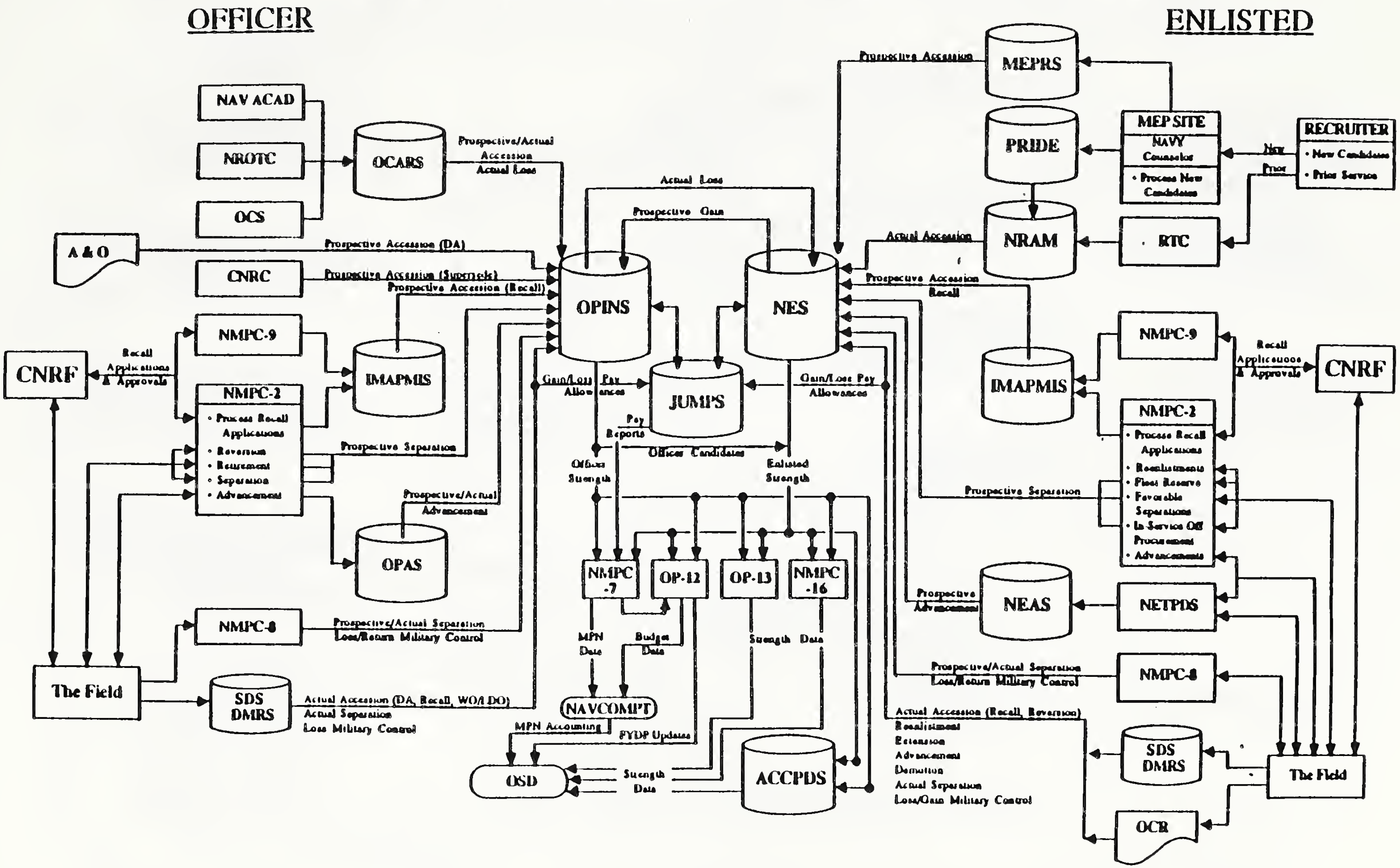
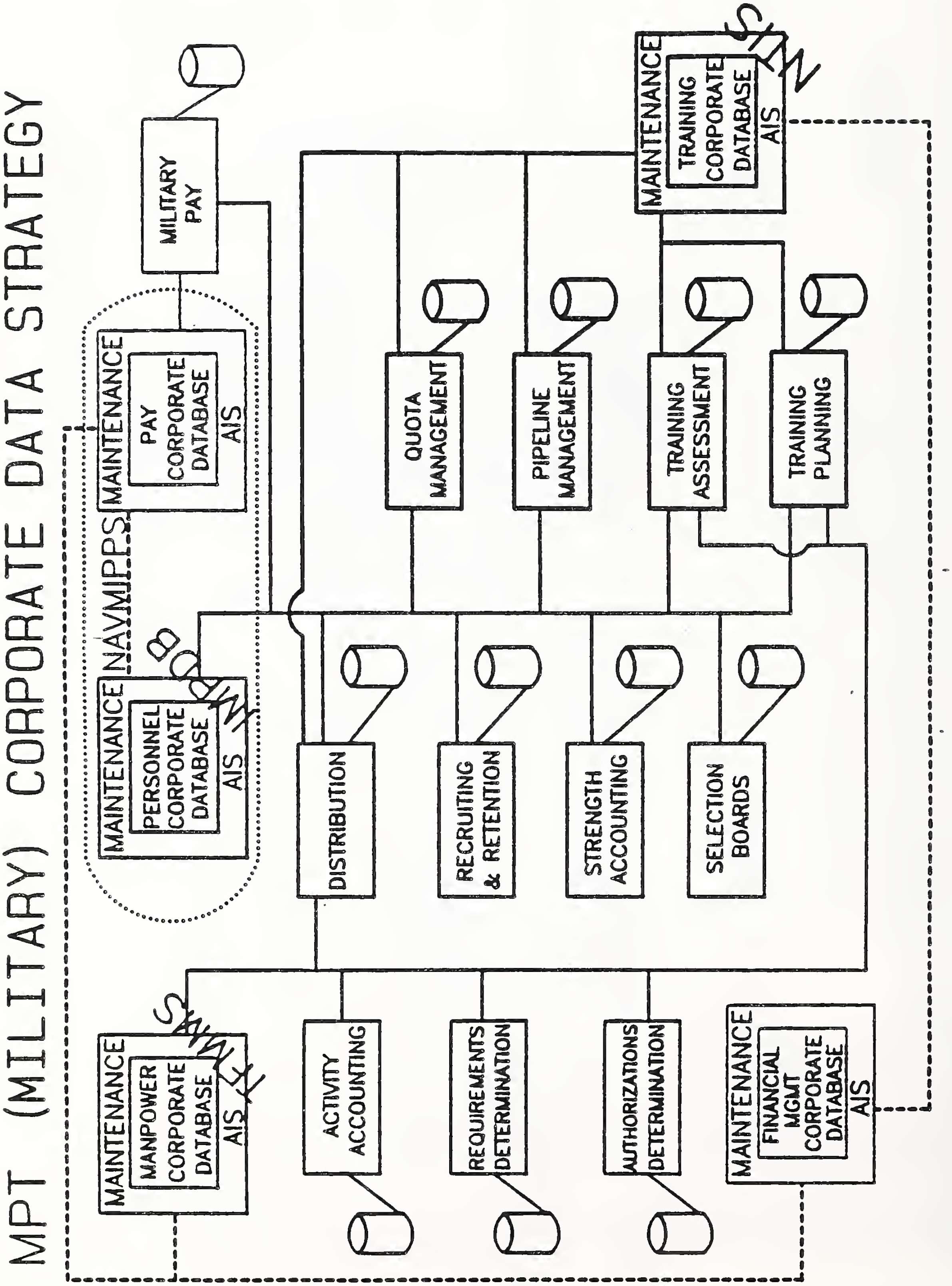



Figure 2



MPT (MILITARY) CORPORATE DATA STRATEGY
MAINTENANCE
MANPOWER CORPORATE DATABASE
AIS
MAINTENANCE
PERSONNEL CORPORATE DATABASE
AIS
NAVMIPPS
MAINTENANCE
PAY CORPORATE DATABASE
AIS
MILITARY PAY
ACTIVITY ACCOUNTING
REQUIREMENTS DETERMINATION
AUTHORIZATIONS DETERMINATION
MAINTENANCE
FINANCIAL MGMT CORPORATE DATABASE
AIS
DISTRIBUTION
RECRUITING & RETENTION
STRENGTH ACCOUNTING
SELECTION BOARDS
QUOTA MANAGEMENT
PIPELINE MANAGEMENT
TRAINING ASSESSMENT
TRAINING PLANNING
MAINTENANCE
TRAINING CORPORATE DATABASE
AIS


Figure 3

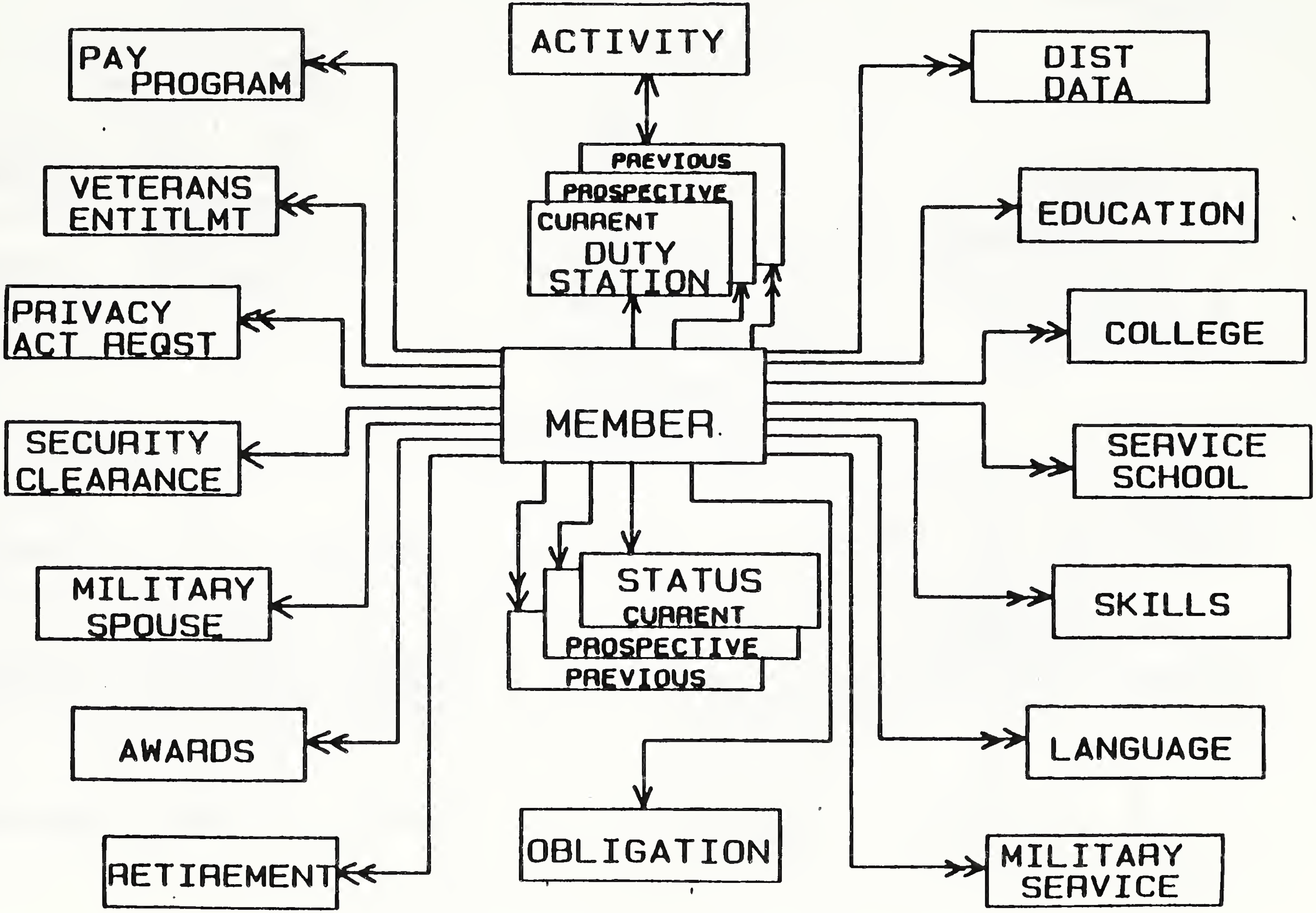

PAY PROGRAM
VETERANS ENTITLMT
PRIVACY ACT REQST
SECURITY CLEARANCE
MILITARY SPOUSE
AWARDS
RETIREMENT
ACTIVITY
PREVIOUS
PROSPECTIVE
CURRENT
DUTY STATION
MEMBER
STATUS
CURRENT
PROSPECTIVE
PREVIOUS
OBLIGATION
DIST DATA
EDUCATION
COLLEGE
SERVICE SCHOOL
SKILLS
LANGUAGE
MILITARY SERVICE

Figure 4

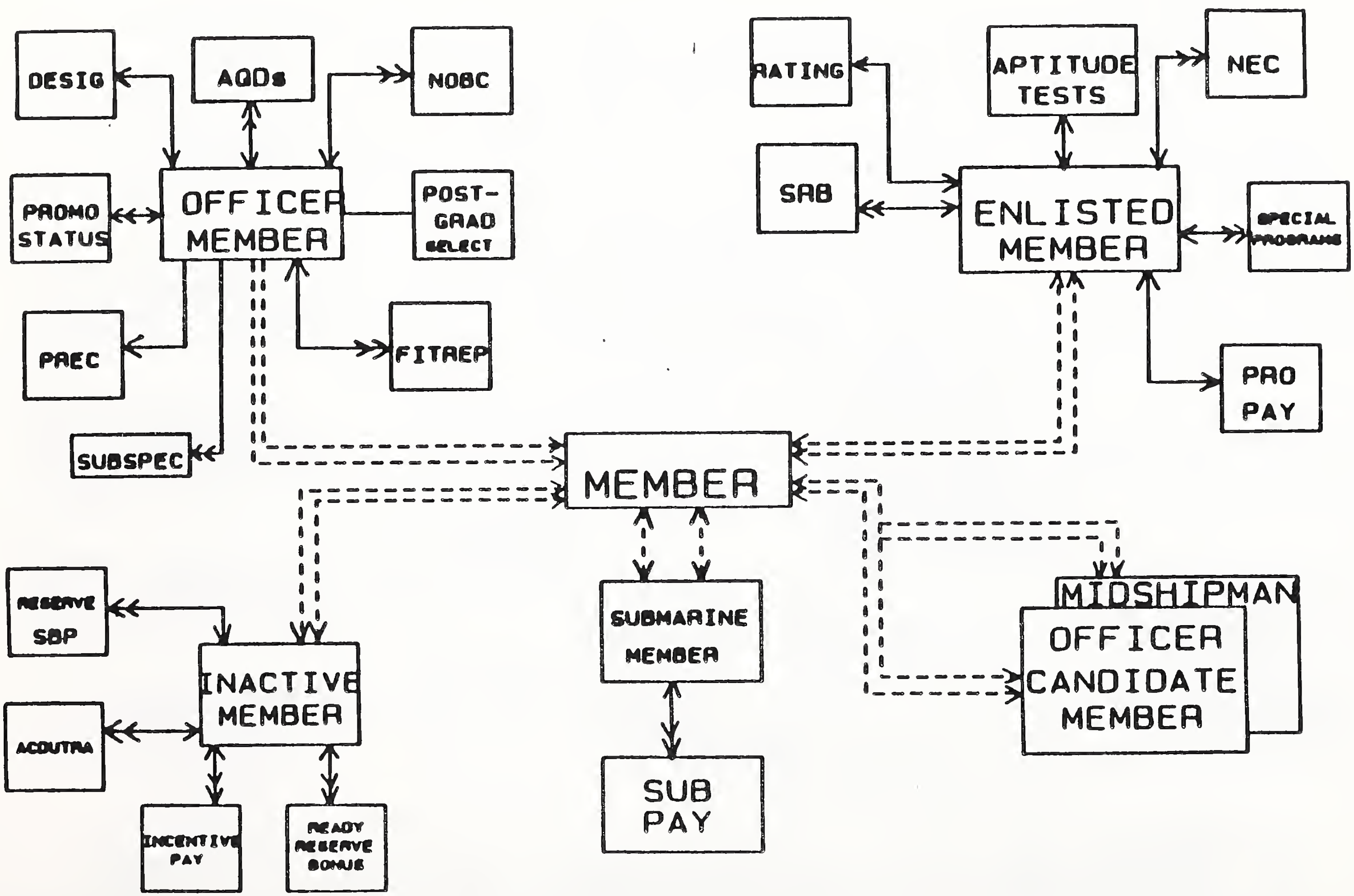

DESIG
AQDs
NOBC
PROMO STATUS
OFFICER MEMBER
POST-GRAD SELECT
PREC
FITREP
SUBSPEC
RATING
APTITUDE TESTS
NEC
SAB
ENLISTED MEMBER
SPECIAL PROGRAMS
PRO PAY
MEMBER
RESERVE SBP
INACTIVE MEMBER
ACDUTRA
INCENTIVE PAY
READY RESERVE BONUS
SUBMARINE MEMBER
SUB PAY
MIDSHIPMAN
OFFICER CANDIDATE MEMBER

Figure 5

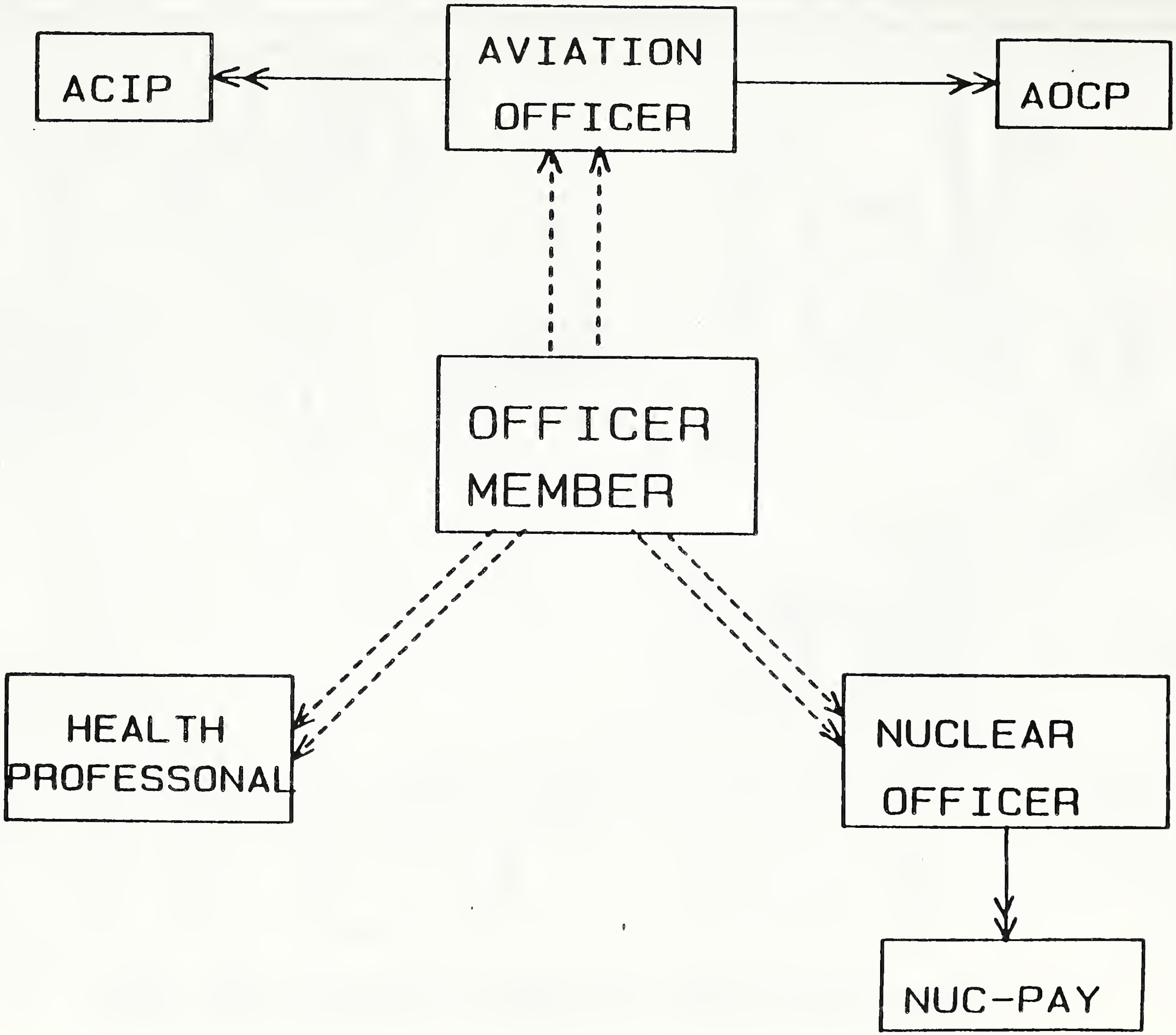

AVIATION OFFICER
ACIP
AOCP
OFFICER MEMBER
HEALTH PROFESSONAL
NUCLEAR OFFICER
NUC-PAY

Figure 6

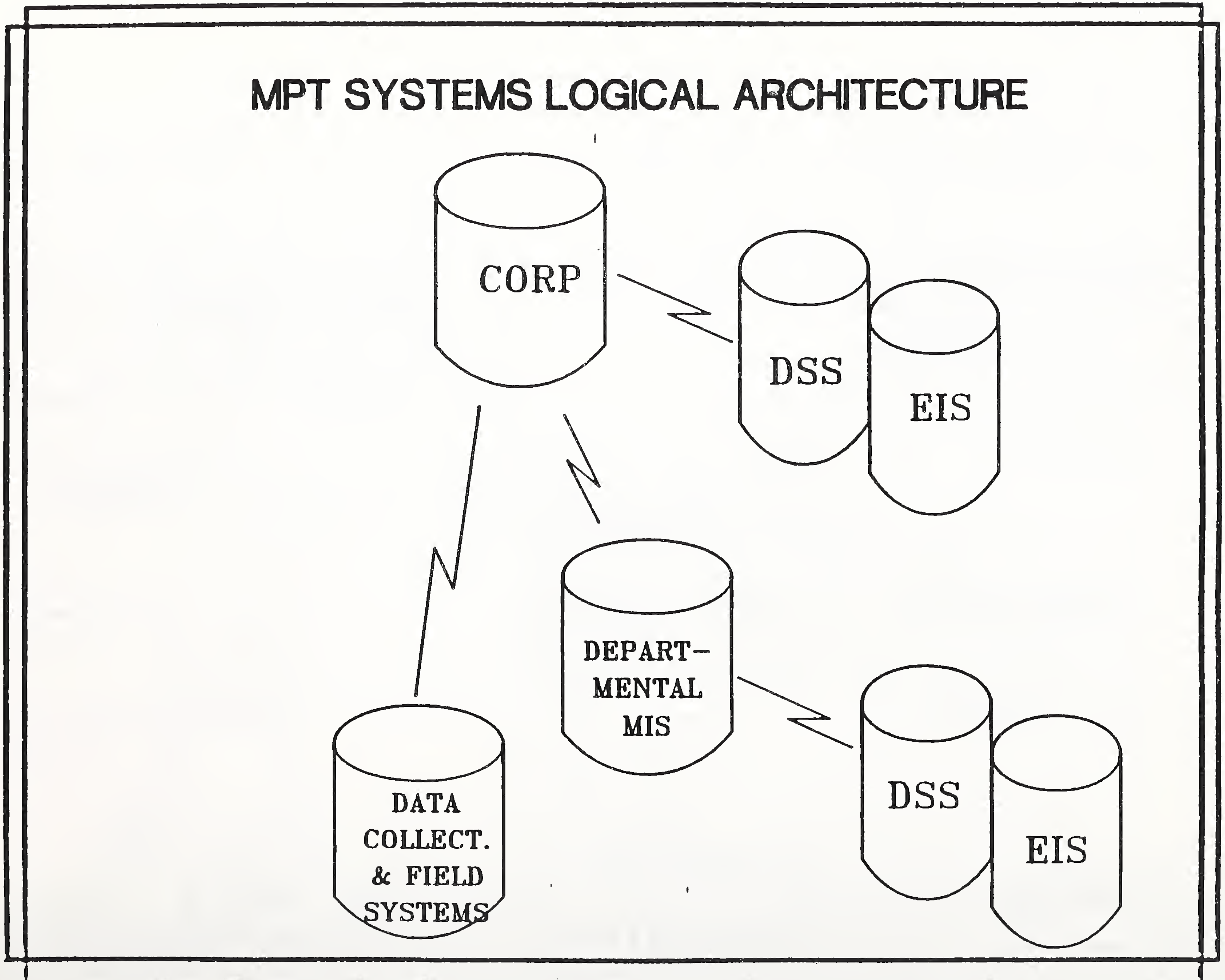
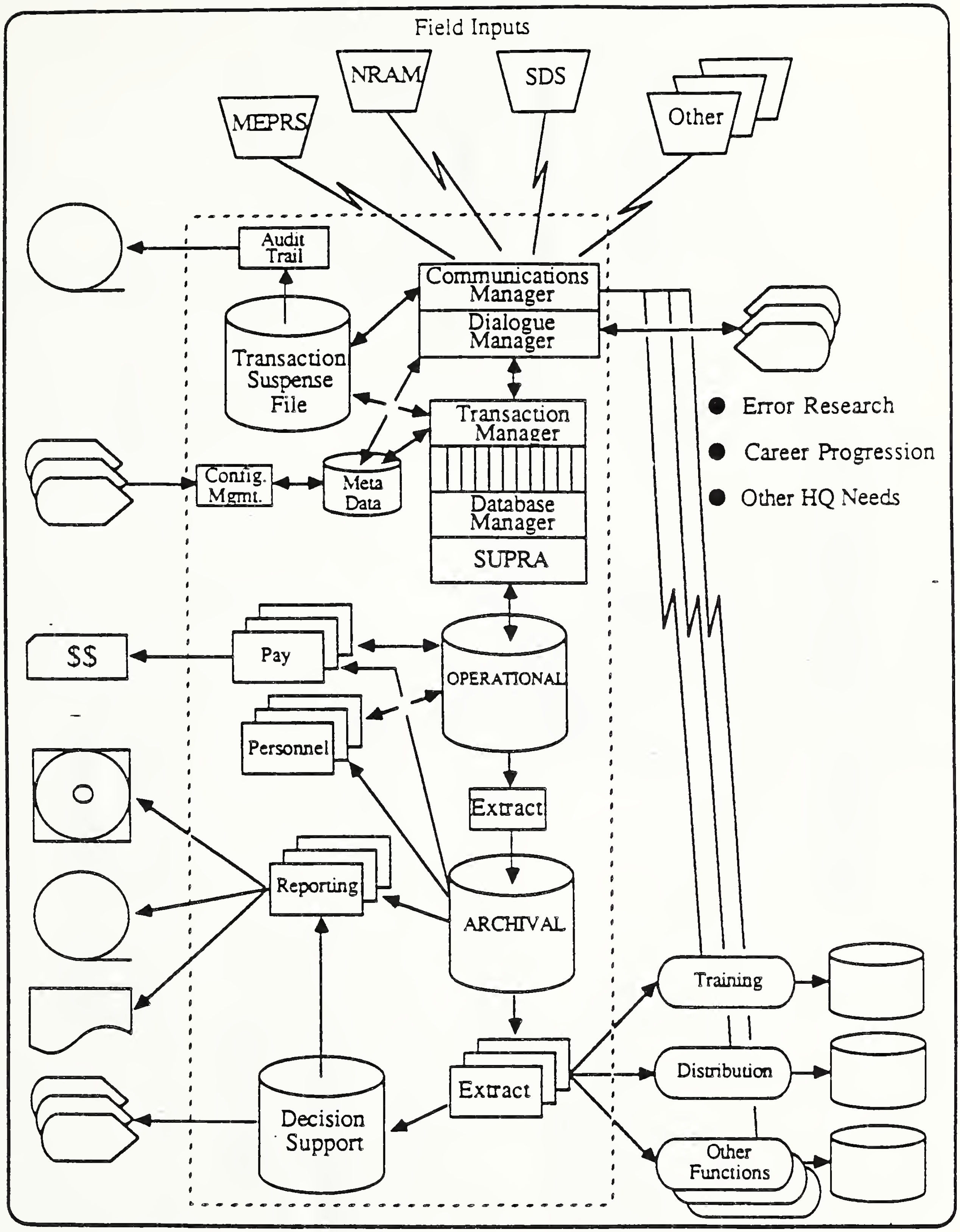

MPT SYSTEMS LOGICAL ARCHITECTURE
CORP
DSS
EIS
DEPART-
MENTAL
MIS
DSS
EIS
DATA
COLLECT.
& FIELD
SYSTEMS

Figure 7




Target Architecture for Integrated Pay and Personnel Systems

Figure 8

## The MPT IRM Direction

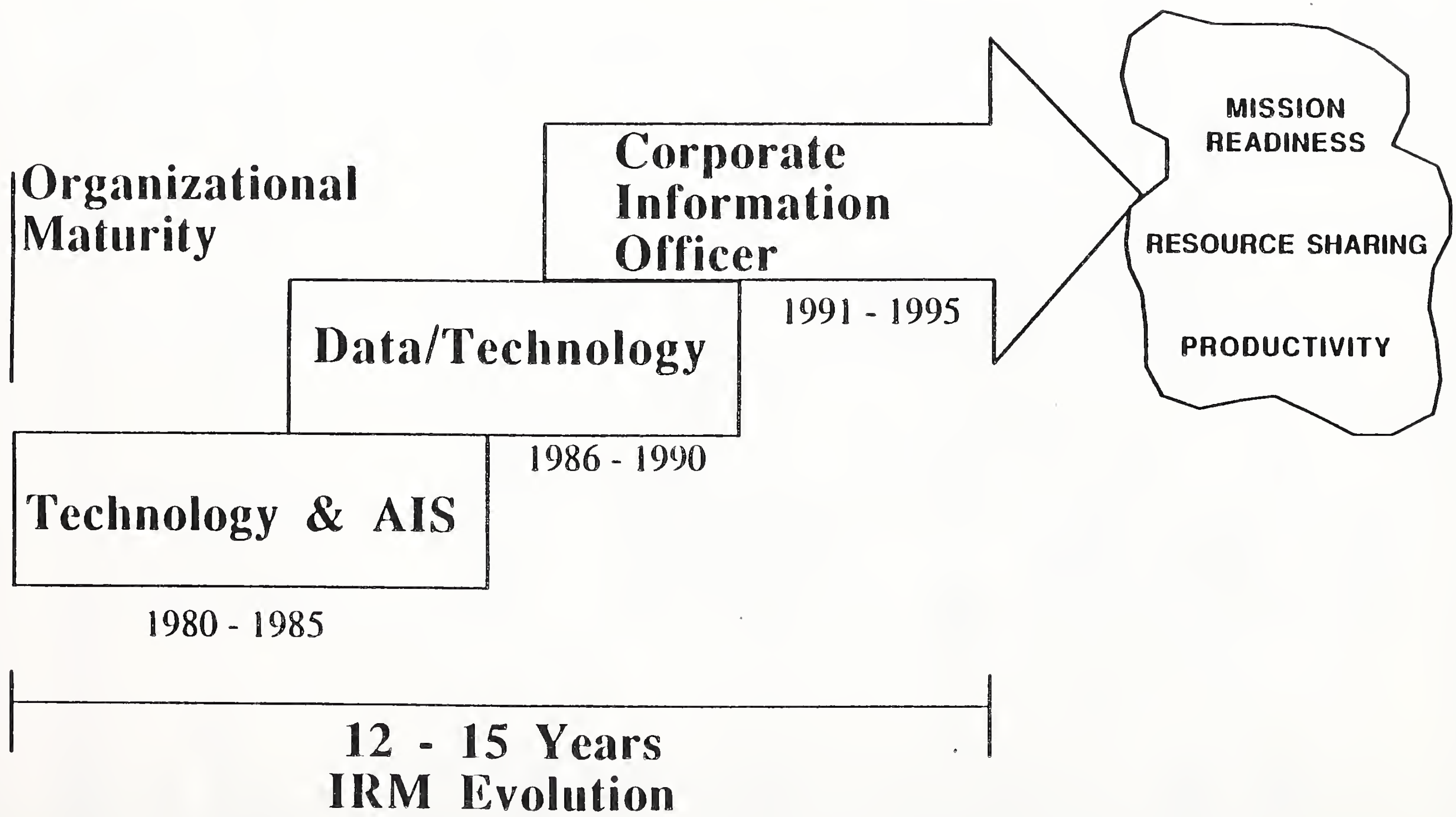



Figure 9

GENERAL SESSION

# THE DATA ADMINISTRATOR: ACHIEVING EXCELLENCE

Robert M. Curtice
Arthur D. Little, Inc.

# THE DATA ADMINISTRATOR: ACHIEVING EXCELLENCE

Robert M. Curtice
Arthur D. Little, Inc.

My remarks this afternoon have to do with achieving excellence in data administration. I propose three broad criteria for judging the degree of excellence that a data administration function achieves.

The first criterion has to do with how relevant what you do as a data administrator is to your organization's business. After all, this is the bottom line: if you are not doing something meaningful to the business, then you can hardly be counted as achieving excellence, even if you are producing something of high quality.

The second criterion has to do with explicit support for the strategic direction that your business is pursuing. I know a lot of people here are from agencies and government departments whose purpose is not profit orientation as it is in the commercial world. Nonetheless, there are strategies your organization has for carrying out its mission and for achieving its goals and objectives. You should be able to relate your activities as data administrator very directly to these goals and strategies.

Lastly, we will consider what you might think of as more traditional elements of quality or measures of excellence, namely producing high quality products. Are the outputs and deliverables that you are involved with and that you produce of consistently high quality? What factors might be taken into account in judging the quality of data administration products?

## Relevance to the Business

Let's explore this question of relevance to the business. First of all it seems to me that there is a dichotomy in data administration organizations between those that have a technical orientation and those that have a business orientation. Those of you who were around at the time when the distinction began to be made between database administration and data administration will recall that many of the technical aspects of data management would be embraced by the position of database administration whereas a business orientation was the purview of the data administrator. And there was a lot of talk at that time about the data administrator not even reporting within the IS organization.

Some people even thought that data administrators would report to the Chief Executive Officer. I don't know that any do. Does anybody in attendance today report to the Chief Executive Officer of your organization? No, I didn't think so.

Interestingly, the most successful and excellent data administration people that I am familiar with have come from a business orientation. In other words, they have their roots in the business world rather than a technical world.

However, despite the fact that we have made the distinction between database administration and data administration, many of today's data administrators come from a technical background. That doesn't mean that they can't be business oriented and obviously we have to have a mix of technical understanding and business understanding. In fact, one of the important roles of the data administrator is to bridge that gap between the business and technical environments. Nonetheless, it seems to me that in many organizations the data administrator not only comes from a technical background, but his or her interests and orientation are very technical.

Lets take a poll of the audience present here today:

> If your job was eliminated in your organization, would your inclination be to take another job not related to data administration or even information systems in your current organization, or would your intention be to take a similar data administration or information systems oriented job in another organization?

Does everybody understand the question? Think about it for a minute. Let's have a show of hands of people whose interest would generally be to take another job (not a similar kind of job in data administration or systems) in your current organization. Consultants by the way can't vote on this question. [About 25% of the audience raises their hands]. OK, that shows where your allegiances are!

This doesn't necessarily mean you aren't doing an excellent job; its just an interesting observation. I think we had at least three-quarters of the people whose orientation would be to stay in their profession rather than remain in their particular business or agency.

Now, as I mentioned earlier, those data administration people that I have come across and consider to be achieving excellence would definitely consider themselves more loyal to their organizations than to their professions.

A second aspect of relevance to the business concerns the use of business terminology: in order to be relevant to the business you have to use the terminology of the business--not technical terms. This is a sin that, for some reason, we keep committing. We turn off the users and we turn off the management of the organization by forcing them to learn our language and that is just not going to work. Whether we are talking about data models or standardizing data or DBMSs or whatever, we have got to find a way to couch what we have to say in terms that are meaningful to the business and not technical jargon.

Another measure of our relevance to the business deals with the involvement of business personnel. These are your constituents, business personnel who are the ultimate users of the services of the data administration function. Who do you consider to be your customers? In those excellent data administration organizations that I have seen, they definitely consider the customers to be the end-users. Not other people within the IS organization.

At the opposite extreme, I have known organizations in which the data administration function was not allowed to talk to end-users! Let's have a show of hands:

> How many people here in their normal course of doing data management or data administration kinds of work have day-to-day, regular interaction with users as opposed to systems people?

Think about it now and let's be honest. Nobody is keeping score on individuals. This is a blind survey. Let's have a show of hands for the people who regularly interact with business people who are not in the IS function. [About two-thirds of the audience raises their hands].

That's pretty good. We had about two-thirds of the audience who in fact have regular interaction with business people and that is very enlightening: I believe in some way that shows a measure of excellence.

The final topic I will discuss under the category of relevance to the business has to do with educating management. By educating management I mean not only business management but in many cases, IS management as well. I have to tell you that many IS managers are not attuned to what data administration is about and what the benefits of data integration and data architecture are. It's up to you to educate both within the IS function and in the user community. Part of that education means explaining the advantages of certain data policies, and the most important of those I think are policies having to do with sharing data: "thou shalt share thy data" if you want to put it in theological tones.

To repeat, the most important data management policy from the business perspective has to do with the need to share data. Explaining the concept of data sharing and selling it is the responsibility of data administration; there is nobody else in the organization who is going to push that idea. What are the benefits of sharing data? What are the costs of not doing it? That is the way we want to express it. If you can do that in a manner convincing to the business, then you are doing an excellent job.

A second aspect of educating management has to do with the benefits and use of the data model. We need to introduce the data model, educate management on why we need data models, how business people can use data models to help think about changes in the business and achieve business objectives.

Finally, a third area of education I want to mention has to do with standards and standardizing data. There are important aspects of standardization that fall within the IS function (things like data element naming conventions, standards for using the data dictionary, etc.). But equally important (and perhaps more so), data standards can impact the business itself. For example, things like we will all use a common customer number. Those are not internal IS kinds of standards. Those are standards about how the business operates, about data sharing, about common use of codes and meaning of codes. If you can educate your management on why those things are important, then that is a way to judge yourself on the degree of excellence of your data administration function.

Support for Business Strategy/Mission

The second major topic has to do with direct support that we are providing in the data administration function for the business strategy or mission. As I mentioned before, perhaps your organization does not think in terms of a business strategy or a competitive strategy. Nonetheless, even in not-for-profit organizations, government agencies, etc., there are strategies, there are mission statements, there are goals and objectives that need to be achieved and each organization has them.

The first measure of the quality of excellence I would like to put forward in this category has to do with the degree to which that business strategy has input to the data administration function. Do you know what the business strategies, objectives, critical success factors, what ever you want to call them, of your organization are? Are they explicitly factored into your plans and activities? So let's have another little quiz here:

> If I asked you to get up and tell me say the 4 or 5 key business objectives of your enterprise, could you do it? Not the objectives of the IS organization -- this doesn't have anything to do with information systems per se. But what are the 4 or 5 key objectives or strategies or goals that your company or organization wants to achieve or needs to achieve in the next 2-5 years? The question is do you know what they are? I am not asking you if you think you know what they are, I am asking do you know what they are? Are they explicitly published? Have you talked with a business manager about what they are?

OK, let's have a show of hands of people who know what their organization's strategies and objectives are; who feel confident you know what they are. OK, and those who are not so clear on what they are. I would say about fifty-fifty on that one.

First of all, it is amazing how many organizations don't have clear objectives and that goes for the business world as well as the not for profit world. It's more amazing to me the number of organizations that do have objectives but don't communicate them. Only the senior management knows what they are. It is hard to help the organization meet those objectives if you are not sure what they are.

So you are at a little disadvantage if you are in the category that your organization doesn't issue and promulgate its objectives and strategies. A lot of people think they are secret; for example, that we are going to acquire some other company tomorrow. That is not what I am talking about. I am talking more about strategic things. Acquiring a specific company is a tactical level kind of activity; an objective to grow by acquisition is a strategic statement. What's going to be most important to our organization over the next 2-5 years? That's what we are really talking about.

The second criterion for judging the support of data administration for business strategy would have to do with what's in the data model. There are a number of ways in which the data model specifically can support the business strategy.

First there are specific things in the data model that allow us to do certain things that relate to our objectives or our goals. I call this the Ragu effect. All of you have seen the Ragu spaghetti sauce commercial - I think it's Ragu: the guy comes to smell the spaghetti sauce and it is out of a jar and says "Yeah, but does this contain all those herbs and spices my mother used to make?" And the other guy says "it's in there." Well, the same is true for the data model, will it enable us to achieve a business objective? We ought to be able to point to the data model and say "it's in there." And we must say it in a way that is understandable to the business personnel. For example, you ought to be able to say things like "our data model can easily allow us to compute customer profitability," or "easily allow us to compute product profitability." Now I use these two examples, obviously from the commercial world, because inherent in these two examples is data sharing. You are not able to compute either customer or product profitability by just looking at a narrow set of data. You have got to combine data about sales and marketing and costs and purchasing and labor and cost accounting and financial accounting and all sorts of things in order to get a bottom line of: are we making any money on this product or are we making money on this customer? This implies you know what a customer is. That may not be so easy.

Another example has to do with being able to compute last year's budget or revenues as if we were organized as we are now. Those of you who are data mavens will recognize this as a problem in maintaining historical data and having your data model to be able to go back and reconstitute the situation as it was at 3 p.m. on July 5th, 1986: what did the data look like at that point in time? That is a real challenge for the data model. We need to go back and look at what the revenue and budgets were then, and to recast that information according to the organization structure in place now. Unless your data model is set up to do that properly, you are not going to be able to do that very easily, if at all. Restating historical results is an extremely meaningful and frequent kind of question management wants to ask: trends, historical data, the "what if" kind of question.

Another specific example might be the ability to accept purchase orders in the EDI (electronic data interchange) standard format. Or, as was mentioned in another session, the ability for a bank to use the ATM standard for exchanging account data, balance data etc. Obviously, I being able to communicate with other companies using ED has impact on the way we define and store and format data. This is another example of being very directly relevant to a business capability. Our data model can support the ability to exchange data with others using the standard, or it can't.

I was interested in finding some data model examples in the not for profit area, and I asked John Harpold of the United States Postal Service who is here today if he had an example in which specific data model capabilities added to the organization's objectives and strategies. He described an interesting example dealing with mail forwarding, i.e., when you change your address and you want your mail forwarded. Well suppose you live in New York and you moved to California. You fill out one of those little cards and give it to your local Post Office and they will forward your mail. The problem is somebody from Chicago is going to mail you a letter and it's going to go to your old address in New York and then they are going to forward it to California.

By having a common data model for both the database that supports the local address forwarding system and what's now called the National Address Data Base (the Post Office actually has one big database that contains supposedly every address that they deliver mail to), they can intercept the letter and send it directly to California.

By having the same data model for the address in those two systems, they are able to catch your letter on-line as it is being mailed from Chicago and not use up the transportation costs of sending it to New York. So there is a cost saving objective that is met. Second of all, obviously, it is going to get to you in California where you moved quicker than being sent to New York first. So there is customer service objective being met. Here then is a nice example where specific data management capabilities enabled the postal service to achieve two very strategic business objectives--cost control and customer service. The data administrator ought to be able to stand up in front of management and say "look what we've been able to do because our data model was used in both of those systems. We have the ability to share that data and thus the ability to achieve these business objectives."

Another kind of support that the data model might have for the business strategy would be to enable simplification of a business process. For example, we may want to relate our engineering bill-of-material to the manufacturing bill-of-material. In most manufacturing companies engineering creates their engineering bill-of-materials and manufacturing creates their manufacturing bill-of-materials. John Zachman talked about that this morning. Engineering does their thing, then throws the result over the wall to manufacturing. They use a lot of effort trying to translate what was done in the engineering world into the manufacturing world. By integrating that data, we can greatly streamline the organizational interface and the work procedures between engineering and manufacturing and achieve some important business objectives in addition to simplifying the work flow. For example, we can make sure that the products that engineering designs can be manufactured within cost.

Another simplifying example might concern the ability to consolidate shipments across orders. Suppose you receive two orders from one company to deliver goods; you have to ship them in two shipments. Why? -- because every order relates to one shipment and every shipment relates to one order. That's the way we do business now; that's the way our company works. It would be a tremendous mistake if, in constructing the data model, we never challenged this assumption that we took it as a given. This example of having two orders with one shipment is of course trivial; but believe me there are many much more subtle opportunities to simplify things in every enterprise.

My experience on this is that if you go back and challenge the business as to why we don't do something in a certain way, you will find something very interesting. The reason, in a lot of cases, of why the business practice is done the way it is has to do with the way our information systems were originally built. Thus the reason we don't have two orders on one shipment is that the computer system that was first built to support these functions didn't allow this condition.

Now this practice has become institutionalized, and as we are developing our new database to support the new future systems, if we are not very careful (and I mean very careful), we are going to build that constraint right back into the system.

What began as a constraint in the first implementation of a system now becomes a requirement! This particular example is perhaps simple. You may say: "Oh well, we would never do that." But I submit to you there are almost certainly cases in your data model right now where you are doing just that and you haven't found them yet. You need to go and find them and challenge the business to simplify its practices by managing data more flexibly. The objective is not to make the data model mimic exactly the way we do business today, even though we think we are doing the right job.

Another example might concern the elimination of duplicate data entry and cross-checking, frequently the result of not sharing data. If we don't share data we have to input it several times at different source locations; not only does that result in an extra cost of collecting the data, but we will wind up somewhere along the line with one of those data sources checking what the other data source did. In validating what the other data source did and looking for incompatibilities between those data sources, we add extra costs, time, and complexity to the whole process. If you are able to stand up and describe in your data model how it can simplify some of these business practices, then you are doing what I consider to be an excellent job.

Third, you need to be sure that future flexibility is built into the data model. The data model should be able to adapt to future directions in the business. In order to do that, you have to be privy to what direction the business is likely to go in. Each of you probably can sit there and think of a thousand reasons why you can't do anything about understanding where the future is going or what likely outcomes might happen; a good excuse is, "We're at the whim of Congress." Well, you know, there are people who make a living anticipating what Congress is going to do. Congress doesn't do things that are totally by surprise, certainly not in the time period that we are talking about. This doesn't say we have to understand exactly what the future is going to be; this is saying we have to understand what the possible scenarios would be and to look at our data model and for example, determine that if this legislation were enacted in the future, what would the impact on our data model be? Or to consider that while we only operate a single warehouse today, it is not at all unlikely that we are going to have two warehouses in our business in the future. If we grow and we expand to the West Coast, we are going to need two warehouses, etc.

If you are business oriented, this sort of scenario thinking would occur to you, especially if you are privy to the directions and strategy of the business. You would then be in a position to challenge your data model and ask what would happen to the data model if such and such legislation were enacted? What would happen if the business policy changed in certain areas? What would happen if we had two warehouses? Would the whole data model fall apart? Or would we be able to accommodate that situation either with no change or a very slight change to the data model? This question seems to me to be of very significant interest to the business.

Are we investing in something here that is going to be robust and be able to accommodate the evolution of the business over the next couple of years?

Moreover, it's just as important to be able to describe the things the data model cannot do. If you do not say what the data model cannot do, there is always the assumption on the part of the management that it can do it. There may well be specific things that you are able to tell just from looking at the data model. If we go in this direction, if this legislation comes in, if this capability is needed, then the data model as it exists now is not going to be able to handle these changes easily. This is a very significant piece of information for senior managers: to know what they are investing in and what limits of what they are buying in their systems and databases. I believe it is just as important to say what the data model cannot support as what it can support.

Producing Quality Products

The last area having to do with excellence in data administration deals with producing quality products. We can do things that are meaningful to the business, but if we don't do them in a quality way, we are not going to have lasting value and our efforts are not going to be considered excellent. There are two main areas I would like to talk about in this respect.

First and foremost is the quality of the data model. The data model is a major product for the data administration organization. If we are using entity-relationship kinds of diagrams, then I think to do an excellent job and to make them understandable by the business, they have to be layered. I have seen many data models in the form of entity-relationship charts that I would call spaghetti diagrams. They take up the size of a wall, they have lines going all over the place, and it looks like somebody took a bowl of spaghetti and threw it. That kind of a diagram is not very understandable to business people. In fact, it's barely understandable to technical people and I don't know what good that kind of diagram is to be honest with you. So I suggest you consider, if you are not doing it already, layering your data model. Sure it has to be integrated, it has to be enterprise-wide, but it is not something we should be proud to show users and confuse them and say look what a wonderful, complicated, totally incomprehensible data model we have here. All they can do is say well I guess it's right.

I would recommend layering the data model by function and not by organization. The organizational structure is going to change over time. So you want to base this layering on something that is a little more stable. I think that if you look at the functions of the business and produce a data model view that contains the entities and relationships of interest to each function, it will be much more understandable to the business users. Of course some data will appear on more than one functional view. That's what it's all about after all. But if you layer it by function it will become much more understandable.

My second suggestion is this: don't show intersections that are not needed. This is the one thing that I have had users come to me and say we really don't understand that. You will recall that John Zachman this morning had in his example of a technical data model an entity he called DEPT-PROJ, i.e., the intersection of department and project. Well, to the users there are two entities--department and project, and only two entities. The users don't understand a DEPT-PROJ. If you are using a data model methodology that requires intersection entities, fine. I happen to use one that does not require intersection entities, but if you do, then don't show them to the user. Doing so does not add any value, it adds confusion. So think about tailoring your products so that they are understandable and meaningful to the user.

The third suggestion I have is to include in your documentation of the data model real examples. It seems to me that this is a very powerful technique that we don't capitalize on enough as a communication and documentation vehicle. Take your data model and make a real life business example of actual data and show how that situation would look if our data were organized according to the data model. I would suggest that you pick sort of the worst case example. There are probably complex cases you know about, for example: three years ago we had an employee, Harry Smith, who left the company and then he came back but he came back part-time because he also had established another business as a consultant and we gave him a consulting contract so he was kind of working for us as a consultant and as a part-time employee at the same time; but he was also a previous employee. We needed to calculate his length of service and his benefits, etc. Our current Human Resource systems couldn't handle that at all.

The idea is to show that we can handle such a case in the new data model: here are the entity records that would exist for employee, here's the relationships we have nine of these and three of these, and here's how they would be related, etc. Two things then happen. One, it becomes very real for the users who are not used to dealing in the abstract terms of data types of person, employee, etc. They are used to dealing in instances, Harry Smith, and so on. Secondly, if you can demonstrate that the data model can handle that tricky situation, then obviously, you have given them a lot of confidence that you can handle the average situation.

The second topic dealing with quality product has to do with data definition and there are three aspects of quality data definitions I want to talk about. First of all, we must use meaningful business data names. If we name our data using COBOL data names, then the user is not going to be able to understand the data or relate to them. We immediately turn off the user by having data named with a sequence of three character abbreviations. One little trick I use in constructing data names is that if you think about it, data is always about something. If I just told you "July 3rd, 1989" then you wouldn't know anything more than before I told you. On the other hand, if I told you July 3rd, 1989 is Harry Smith's birthday, now you know something. So in order to be meaningful, I have to make that piece of data apply to something. Of course the things to which the data elements apply are the entities. So if you look at the name of a data element, it ought to be clear not only that it is a date, but what it is applying to because the date of somebody's birthday is a very different meaning than the date a purchase order was issued. So, in order to get some meaning out of this, you've got to be clear in the name what is the thing that this piece of data applies to.

Second, we must create better data definitions -- ones that are easily understandable to business personnel. It seems to me that a lot of business managers who get the output of automated systems don't use that automated output fully, despite spending millions of dollars on it. By the way, I came across a figure the other day that might interest you. Did you know that in 1985, 40 percent of all durable goods, capital equipment expenditures in the United States were spent on information technology equipment? That includes all machine tools, all transportation, railroad cars, every piece of capital durable equipment, which only excludes buildings; 40 percent of all that investment was information technology equipment. Kind of interesting. A lot of money being spent there. That surprised me.

Back to the point, in spite of all that money that we are spending, many business managers don't utilize the information that they get and they don't trust it because they don't understand it. They get a piece of data and it says XYZ but they don't really know what assumptions went behind XYZ. Under what conditions was this data collected? What time period does it apply to? What geographic area does it apply to? Well, unless I know that, I can't use this information. I can't trust it. I submit to you that in a large percentage of the reports most business people get, they don't have a good feeling about answers to those questions.

I will give you just one little example. In talking to a president of an automotive supply equipment manufacturing company the other day, he said that his company got some reports out of the computer systems which showed that the cost of repairing and servicing his product in Europe was twice as much as it was in the United States. Very interesting result. He wanted to know why is that? Why does it cost twice as much to service this product in Europe as in the United States? He launched a big task force; they went to Europe; they looked at the suppliers to see whether the raw materials used to make his product was some how of inferior quality in Europe or whether the design wasn't applicable to Europe or whether they were using this product in Europe differently that made it fail more and therefore the costs were greater, and on and on.

Months and months of effort. Well you know what happened? It wasn't true. The cost in Europe wasn't different than the cost in the United States. What was happening was the definition of that piece of data was different in Europe than in the United States. They were including additional costs in the service account in Europe than they did in the United States. This is a good example of business managers getting some data and not understanding it. Of course, if it comes out of the computer, then the average person will say it's got to be right. If the computer says it's more in Europe than it is in the United States, it's got to be right. They spent a lot of money trying to track this down.

Here is another suggestion that has to do with data definition quality: make use of standardized domains.

Let's do another audience survey:

> Will everybody raise their hands who knows what a domain is? Be honest now. About half the people raised their hands. I am using the word domain here in the relational sense, that is, the possible values of a column of a relation. Some people make the distinction between a data element and a domain, and it's very important. I know in the postal service for example, they make the distinction between what they call roles and roots. You may be calling it by a different name but it is fairly commonplace to use the term domain.

Now if you make a distinction between domains and data elements, you are well on the road to improving the quality of your data definitions. Date for example, is a domain. It is not a data element.

By itself, it doesn't have any meaning, since it doesn't apply to anything. Instead it standardizes the values of certain columns. One column might be the date of the employee's birthday. Another column might be the date that he was hired. Another column might be the date he was eligible for certain insurance. Certainly none of us would want to define those three pieces of data the same. Anybody present who would want to define those to be the same thing? No.

But there is something similar about them. What is similar about them? They have the same domain! So the idea is to define the domains separately from the data elements. Domains stand alone; data elements are always in context. The meaning of a data element is dependent on that context. For example, birth date in the context of an employee; issue date in the context of the purchase order. They mean something different. You've got to capture that context in order to get the true meaning and definition of the data element.

If we have already defined the domain date, then when we are creating the definition of date of employee's hire, we don't have to repeat what the definition of a date is. More importantly, the definition of this data element is not really concerned with what a date is. Rather, it is concerned with what does it mean to be hired on a certain date. We are content to know what a date is as the definition of the domain. As for the data element, we must instead focus on the notion of hire date. Is it the actual date you are hired? Is it equivalent date of service going back so many months? Is it the first date you report to work? What is it? That's what the definition of the data element should spell out. Not what a date is.

So you see, making the distinction between domains and data elements actually helps to improve the quality of data definitions. So let's have a show of hands:

> How many people make a distinction between domains and data elements in their data dictionaries? How many people do not make the distinction? That is interesting. We've got, I would say about 60 percent who do and 40 percent who don't.

Well, we have about run out of time. Let me end up by summarizing.

Excellence in data administration means two things. First of all, obviously, doing the right things, being relevant to the business, supporting the business strategy, and being understandable to the business. Second, it means doing things right. By that we mean producing quality products, where quality is defined in business terms, not necessarily in technical terms.

Thank you.

---

Mr. Curtice has been addressing problems in information processing, systems architecture, and database management for clients since he joined Arthur D. Little, Inc. in 1966. His consulting assignments have centered on the design methods, software, and other technical and managerial issues arising in the planning and development of database oriented computer systems.

Recently, Mr. Curtice has been concerned with the application system development methodology as it is affected by the introduction of both a database approach and the use of data management software.

Mr. Curtice has evaluated plans for CAD/CAM systems for several large manufacturing concerns, with emphasis on overall system architecture and database integration issues.

He was the major contributor to Arthur D. Little, Inc.'s Strategic Value Analysis methodology for systems and data planning.

Mr. Curtice is a member of the Association for Computing Machinery and its Special Interest Group on the Management of Data. He received a B.A. in Mathematics and an M.S. in Information Science, both from Lehigh University.

Recent publications include: Strategic Value Analysis: A Modern Approach to Systems and Data Planning, and Logical Data Base Design (with P.E. Jones).

# THE DATA ADMINISTRATOR:
## *Achieving Excellence*

DAMA
Washington, D.C.

Robert M. Curtice
**Arthur D. Little, Inc.**
**Cambridge, Ma.**

**Arthur D Little**

Figure 1

# EXCELLENCE MEANS:

- Relevance to Business
- Support for Business Strategy or Mission
- **Producing Quality Products**

**Arthur D Little**

Figure 2

# RELEVANCE TO THE BUSINESS

- Business Vs. Technical Orientation
- Use of Business Terminology
- Involvement of Business Personnel
- **Education of Management:**
  - Data Policies
  - Data Model
  - **Data Standards**

Arthur D Little

Figure 3

# SUPPORT FOR BUSINESS STRATEGY / MISSION

- Business Strategy Input to DA
- **Data Model Support for Business Strategy**
  **Provide Specific Facility**

Arthur D Little

Figure 4

## Specific Facility Examples

### Our Data Model Can:

Easily compute customer profitability

Easily compute product profitability

Recompute last years budgets/revenues
as if we were organized as we now are

Accept purchase orders in the EDI
standard format

Arthur D Little

Figure 5

## SUPPORT FOR BUSINESS STRATEGY / MISSION

- Business Strategy Input to DA
- Data Model Support for Business Strategy
  - Provide Specific Facility
  - Simplify Business Practice

Arthur D Little

Figure 6

## Simplifying Examples

### Our Data Model Can:

Relate Engineering Bill-of-Material to
Manufacturing Bill-of-Material

Consolidate shipments across orders

Support multiple ship dates per
Purchase Order line item

Eliminate duplicate data entry and
cross checking

Arthur D Little

Figure 7

## SUPPORT FOR BUSINESS STRATEGY / MISSION

- Business Strategy Input to DA
- Data Model Support for Business Strategy
  - Provide Specific Facility
  - Simplify Business Practice
  - Add Future Flexibility
- **Migrate Toward a Vision while Yielding Direct Benefits**

Arthur D Little

Figure 8

## PRODUCING QUALITY PRODUCT

- Data Model Quality
  - Layer E-R diagrams
    - [By Function, not Organization]
  - Don't show unneeded intersections
  - Document real examples
- **Data Definition Quality**
  - Meaningful Business Data Names
  - Quality Data Definitions
  - **Use Standardized Domains**

Arthur D Little

Figure 9

## IN SUMMARY...

Excellence Means

Doing The Right Things
Doing Things Right

In a way that's meaningful to the business.

Arthur D Little

Figure 10

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

# BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 90-4292

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
APRIL 1990

4. TITLE AND SUBTITLE

The Second Annual DAMA Symposium

5. AUTHOR(S)

Judith J. Newton and Frankie E. Spielman

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP) National Capital Region of the Data Administration Management Association (NCR DAMA); Federal Data Management Users Group (FEDMUG); Federal Information Resources Management (AFFIRM).

10. SUPPLEMENTARY NOTES

☐ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This publication constitutes the proceedings of a one-day symposium at the National Institute of Standards and Technology on May 3, 1989. It was jointly sponsored by the National Capital Region of the Data Administration Management Association (NCR DAMA), the Federal Data Management Users Group (FEDMUG), and the Association for Federal Information Resources Management (AFFIRM).

The symposium provided attendees with an opportunity to share the insights of leaders in the Data Administration field. Special emphasis was given to the factors which contribute to successful implementation of Data Administration standards and techniques.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

administration; data; data administration; data architecture; information; information asset management; life cycle; management

13. AVAILABILITY

XX UNLIMITED

☐ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

☐ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

XX ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
163

15. PRICE
A08

ELECTRONIC FORM