

# The Information FrameWork

---

by R. Evernden

*John Zachman's framework for information systems architecture has been widely discussed since its publication in the IBM Systems Journal in 1987. This paper shows how the development of the Information FrameWork (IFW) has built upon the ideas presented by Zachman. However, the objectives and scope of IFW are broader than that of the original Zachman framework. IFW is described and compared with the original Zachman structure, showing the evolution, changes, and the rationale behind the changes based on experiences from within the financial services industry. This paper also shows how the structure of IFW has been populated by industry-wide models and supported by a distinctive methodology. A detailed discussion of each of the six dimensions of the IFW architecture is presented.*

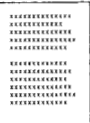
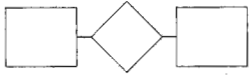
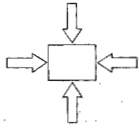
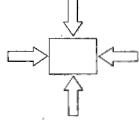
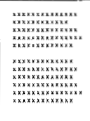
In his 1987 paper, "A Framework for Information Systems Architecture," John Zachman defined information systems architecture, first "by creating a descriptive framework from disciplines quite independent of information systems," then specifying "information systems architecture based upon the neutral, objective framework."<sup>1</sup> This framework has become known as the Zachman framework after its author. The initial Zachman framework consisted of three columns (for data, process, and network descriptions, respectively) and five rows, making a 15-cell grid structure. The 1987 paper proposed a further three possible descriptions covering people, time, and purpose,<sup>2</sup> which were described in more detail in a paper for the *IBM Systems Journal* in 1992.<sup>3</sup> Figure 1 shows the complete six-column Zachman framework.

Although the Information FrameWork (IFW) was initially based on the Zachman framework, during development the structure was changed, taking into account recent research and ideas, and incorporating the experience of developing models and a methodology to support the framework. Zachman uses an analogy from classical building architecture and military aircraft manufacturing to help define information systems architecture (ISA). This approach is useful when the final outcome is a "system"<sup>4</sup> because ISA helps in understanding the components that make up each individual application or system. IFW uses the alternative analogy of a "city plan"<sup>5</sup> rather than a building plan. It provides an effective way to gradually develop a complete "city" of information.<sup>6</sup> This compilation includes information about individual applications and systems, as well as information output from other types of projects such as strategic planning or business process reengineering.

**Objectives of the Information FrameWork.** The need for a clear strategy for managing information has been a major issue in the 1990s.<sup>7,8</sup> The focus of IFW is on information, and its primary objective is therefore to provide a strategy for managing information as a valuable asset; hence the name "Information FrameWork." It was not named after infor-

©Copyright 1996 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of this paper must be obtained from the Editor.

Figure 1 The six-column information systems architecture (Zachman framework)

|                                  | <b>DATA</b><br><input type="checkbox"/> ENTITY<br><input type="checkbox"/> RELATIONSHIP                                                                                                                                                           | <b>FUNCTION</b><br><input type="checkbox"/> FUNCTION<br><input type="checkbox"/> ARGUMENT                                                                                                                                               | <b>NETWORK</b><br><input type="checkbox"/> NODE<br><input type="checkbox"/> LINK                                                                                                                                  | <b>PEOPLE</b><br><input type="checkbox"/> AGENT<br><input type="checkbox"/> WORK                                                                                                                                        |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SCOPE PLANNER</b>             | <b>LIST OF THINGS IMPORTANT TO THE BUSINESS</b><br><br><input type="checkbox"/> ENTITY = CLASS OF BUSINESS THING                                                 | <b>LIST OF PROCESSES THE BUSINESS PERFORMS</b><br><br><input type="checkbox"/> FUNCTION = CLASS OF BUSINESS PROCESS                                    | <b>LIST OF LOCATIONS IN WHICH THE BUSINESS OPERATES</b><br><br><input type="checkbox"/> NODE = MAJOR BUSINESS LOCATION          | <b>LIST OF ORGANIZATIONS/AGENTS IMPORTANT TO THE BUSINESS</b><br><br><input type="checkbox"/> AGENT = MAJOR ORGANIZATION UNIT        |
| <b>ENTERPRISE MODEL OWNER</b>    | <b>E.G., ENTITY/RELATIONSHIP DIAGRAM</b><br><br><input type="checkbox"/> ENTITY = BUSINESS ENTITY<br><input type="checkbox"/> RELATIONSHIP = BUSINESS CONSTRAINT | <b>E.G., PROCESS FLOW DIAGRAM</b><br><br><input type="checkbox"/> FUNCTION = BUSINESS PROCESS<br><input type="checkbox"/> ARGUMENT = BUSINESS RESOURCE | <b>E.G., LOGISTICS NETWORK</b><br><input type="checkbox"/> NODE = BUSINESS LOCATION<br><input type="checkbox"/> LINK = BUSINESS LINKAGE                                                                           | <b>E.G., ORGANIZATION CHART</b><br><input type="checkbox"/> AGENT = ORGANIZATION UNIT<br><input type="checkbox"/> WORK = WORK PRODUCT                                                                                   |
| <b>SYSTEM MODEL DESIGNER</b>     | <b>E.G., DATA MODEL</b><br><input type="checkbox"/> ENTITY = DATA ENTITY<br><input type="checkbox"/> RELATIONSHIP = DATA RELATIONSHIP                                                                                                             | <b>E.G., DATA FLOW DIAGRAM</b><br><br><input type="checkbox"/> FUNCTION = APPLICATION FUNCTION<br><input type="checkbox"/> ARGUMENT = USER VIEW       | <b>E.G., DISTRIBUTED SYSTEM ARCHITECTURE</b><br><input type="checkbox"/> NODE = INFORMATION SYSTEM FUNCTION (PROCESSOR, STORAGE, ETC.)<br><input type="checkbox"/> LINK = LINE CHARACTERISTICS                    | <b>E.G., HUMAN INTERFACE ARCHITECTURE</b><br><input type="checkbox"/> AGENT = ROLE<br><input type="checkbox"/> WORK = DELIVERABLE                                                                                       |
| <b>TECHNOLOGY MODEL BUILDER</b>  | <b>E.G., DATA DESIGN</b><br><input type="checkbox"/> ENTITY = SEGMENT/ROW<br><input type="checkbox"/> RELATIONSHIP = POINTER/KEY                                                                                                                  | <b>E.G., STRUCTURE CHART</b><br><input type="checkbox"/> FUNCTION = COMPUTER FUNCTION<br><input type="checkbox"/> ARGUMENT = SCREEN/DEVICE FORMAT                                                                                       | <b>E.G., SYSTEM ARCHITECTURE</b><br><input type="checkbox"/> NODE = HARDWARE/SYSTEM SOFTWARE<br><input type="checkbox"/> LINK = LINE SPECIFICATIONS                                                               | <b>E.G., HUMAN/TECHNOLOGY INTERFACE</b><br><input type="checkbox"/> AGENT = ROLE<br><input type="checkbox"/> WORK = DELIVERABLE                                                                                         |
| <b>COMPONENTS SUB-CONTRACTOR</b> | <b>E.G., DATA DEFINITION DESCRIPTION</b><br><br><input type="checkbox"/> ENTITY = FIELD<br><input type="checkbox"/> RELATIONSHIP = ADDRESS                     | <b>E.G., PROGRAM</b><br><br><input type="checkbox"/> FUNCTION = LANGUAGE STATEMENT<br><input type="checkbox"/> ARGUMENT = CONTROL BLOCK              | <b>E.G., NETWORK ARCHITECTURE</b><br><br><input type="checkbox"/> NODE = ADDRESS<br><input type="checkbox"/> LINK = PROTOCOL | <b>E.G., SECURITY ARCHITECTURE</b><br><br><input type="checkbox"/> AGENT = IDENTITY<br><input type="checkbox"/> WORK = TRANSACTION |
| <b>FUNCTIONING SYSTEM</b>        | <b>E.G., DATA</b>                                                                                                                                                                                                                                 | <b>E.G., FUNCTION</b>                                                                                                                                                                                                                   | <b>E.G., NETWORK</b>                                                                                                                                                                                              | <b>E.G., ORGANIZATION</b>                                                                                                                                                                                               |

Adapted from Figure 6, J. F. Sowa and J. A. Zachman in *IBM Systems Journal*, Vol. 31, No. 3, 1992.

mation systems, application development, or any of a number of alternative uses of the word "information," because IFW can be used in a wide range of information-related projects. Although IFW was initially developed for the financial services industry, IBM did not limit the name by calling it "Financial Services FrameWork," because the principles and ideas behind IFW can be applied to information in

any industry.<sup>9</sup> Whereas the Zachman framework is focused on an information *systems* architecture, the Information FrameWork is aimed at managing *information*. ISA "provides a systematic taxonomy of concepts for relating things in the world to the representations in the computer." It has been said that since the introduction of the Zachman framework it "has been widely adopted by systems analysts and

| TIME<br><input type="checkbox"/> TIME<br><input type="checkbox"/> CYCLE                                                         |                                                                                                                        | MOTIVATION<br><input type="checkbox"/> ENDS<br><input type="checkbox"/> MEANS |                                  |
|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|----------------------------------|
| LIST OF EVENTS SIGNIFICANT TO THE BUSINESS<br> | LIST OF BUSINESS GOALS / STRATEGY<br> |                                                                               | SCOPE<br>PLANNER                 |
| <input type="checkbox"/> TIME = MAJOR BUSINESS EVENT                                                                            | <input type="checkbox"/> ENDS / MEANS = MAJOR BUSINESS GOAL / CRITICAL SUCCESS FACTOR                                  |                                                                               |                                  |
| E.G., MASTER SCHEDULE                                                                                                           | E.G., BUSINESS PLAN                                                                                                    |                                                                               | ENTERPRISE<br>MODEL<br>OWNER     |
| <input type="checkbox"/> TIME = BUSINESS EVENT<br><input type="checkbox"/> CYCLE = BUSINESS CYCLE                               | <input type="checkbox"/> ENDS = BUSINESS OBJECTIVE<br><input type="checkbox"/> MEANS = BUSINESS STRATEGY               |                                                                               |                                  |
| E.G., PROCESSING STRUCTURE                                                                                                      | E.G., KNOWLEDGE ARCHITECTURE                                                                                           |                                                                               | SYSTEM<br>MODEL<br>DESIGNER      |
| <input type="checkbox"/> TIME = SYSTEM EVENT<br><input type="checkbox"/> CYCLE = PROCESSING CYCLE                               | <input type="checkbox"/> ENDS = CRITERION<br><input type="checkbox"/> MEANS = OPTION                                   |                                                                               |                                  |
| E.G., CONTROL STRUCTURE                                                                                                         | E.G., KNOWLEDGE DESIGN                                                                                                 |                                                                               | TECHNOLOGY<br>MODEL<br>BUILDER   |
| <input type="checkbox"/> TIME = EXECUTE<br><input type="checkbox"/> CYCLE = COMPONENT CYCLE                                     | <input type="checkbox"/> ENDS = CONDITION<br><input type="checkbox"/> MEANS = ACTION                                   |                                                                               |                                  |
| E.G., TIMING<br>DEFINITION<br>               | E.G., KNOWLEDGE<br>DEFINITION<br>   |                                                                               | COMPONENTS<br>SUB-<br>CONTRACTOR |
| <input type="checkbox"/> TIME = INTERRUPT<br><input type="checkbox"/> CYCLE = MACHINE CYCLE                                     | <input type="checkbox"/> ENDS = SUBCONDITION<br><input type="checkbox"/> MEANS = STEP                                  |                                                                               |                                  |
| E.G., SCHEDULE                                                                                                                  | E.G., STRATEGY                                                                                                         |                                                                               | FUNCTIONING<br>SYSTEM            |

database designers,"<sup>10</sup> whereas IFW is suitable for use in any situation where "information" is created or used. This distinction, which has also been made by other framework developers,<sup>11</sup> is of some importance, because the underlying focus of a framework helps to determine the format of the framework itself. The framework construct is created in and through a particular world view such as the one for

IFW in Figure 2. Recognition of this situation is one of the strengths of IFW; hence, it denies any claims for spurious neutrality or objectivity and strives to be reflexive in its methodology.

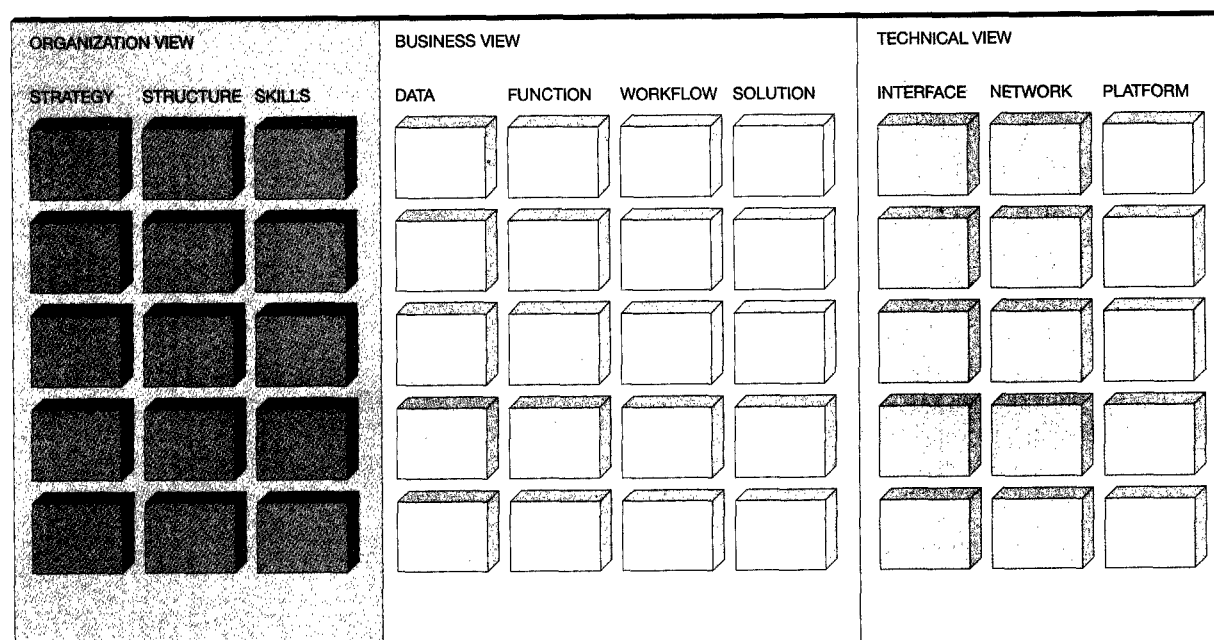
The second objective of IFW is to provide the means to realize the potential value of the information asset in the most effective manner. Much of the development work in IFW has been exploratory. The Zachman framework was tested in a number of projects, and, where necessary, enhancements were made. In philosophy a framework is often used as an outline or hypothesis in the production of new knowledge. IFW has been used in this sense, with additions and changes to the basic hypothesis, evolving as IBM gained more experience through practical customer projects. IFW has drawn upon many sources for its intellectual foundation,<sup>12</sup> including the notion of a "framework" from object-oriented theory. A recent paper by Cockburn<sup>13</sup> states that a framework is "a template for a group of objects that manage a responsibility jointly, using a predefined protocol among themselves," and goes on to say that the framework itself "consists of the statement of how the responsibility is divided and the definition of the protocol." In this sense the components of IFW have responsibility for managing information. The cells in IFW show the division of responsibility, with each cell localizing one aspect of the problem of managing information. The protocol is a methodology that explains exactly how component objects are grouped together in IFW to define solutions.

Inevitably IFW draws comparisons with the Zachman framework. Table 1 is designed to show the major differences. Each of these differences is covered in more detail in this paper. Some cells in IFW were not included in the Zachman framework. Some cells in the Zachman framework are also split into two or more cells in IFW. Appendices A and B give a more detailed comparison between the six-column ISA framework and the ten-column IFW at the cell or component level.

IFW<sup>14</sup> was developed by IBM's Banking Solution Centre in Dublin in conjunction with input and feedback from many of the world's leading financial institutions. Although most of the development work outlined in this paper was derived from experience in the financial services industry, the results can be applied to manage complex information structures in any industry.

In 1988 IBM initiated the Financial Application Solutions for the 1990s (FAS90) project. FAS90 conducted

**Figure 2 The Information FrameWork**



**Table 1 A comparison of IFW and the Zachman framework**

| Criteria                   | IFW                                                                                                                                                                                                                                                                                                                                     | Zachman                                                                                                                                                     |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Focus and nature           | <ul style="list-style-type: none"> <li>• Information</li> <li>• Main deliverables are domain models and reusable information components</li> <li>• Analogy with city planning and urbanism</li> </ul>                                                                                                                                   | <ul style="list-style-type: none"> <li>• Systems</li> <li>• Main deliverable is stand-alone system</li> <li>• Analogy with building architecture</li> </ul> |
| Main processes supported   | <ul style="list-style-type: none"> <li>• Information management</li> <li>• Comparing and integrating multiple methodologies, domain models, architectures, work practices</li> <li>• Processes that create or use information (since most processes have information inputs and outputs, IFW can be used in many situations)</li> </ul> | <ul style="list-style-type: none"> <li>• Systems development</li> </ul>                                                                                     |
| Structure and architecture | <ul style="list-style-type: none"> <li>• Three views, ten columns</li> <li>• Three levels, five rows</li> <li>• Fifty cells</li> <li>• Six dimensions to the architecture</li> </ul>                                                                                                                                                    | <ul style="list-style-type: none"> <li>• Six columns</li> <li>• Five rows</li> <li>• Thirty cells</li> <li>• Two dimensions</li> </ul>                      |
| Rules of the framework     | <ul style="list-style-type: none"> <li>• Deliberate order in the views and columns</li> </ul>                                                                                                                                                                                                                                           | <ul style="list-style-type: none"> <li>• No order intended in the columns</li> </ul>                                                                        |

surveys of financial institution needs through customer advisory boards and other forums that identified a number of projects through which IBM could support the financial services industry. Two of the initial projects were to develop the Financial Appli-

cation Architecture (FAA)<sup>15</sup> and the Financial Services Data Model (FSDM).

In 1991 some IBM customers in Europe wanted to develop an industry-standard function model to com-

**Figure 3** Structural components in the Information FrameWork

| LEVELS OF CONSTRAINT |                           | TYPES OF INFORMATION |                  |               |               |                 |                 |                 |                  |                |
|----------------------|---------------------------|----------------------|------------------|---------------|---------------|-----------------|-----------------|-----------------|------------------|----------------|
|                      |                           | ORGANIZATION VIEW    |                  |               | BUSINESS VIEW |                 |                 |                 | TECHNICAL VIEW   |                |
|                      |                           | STRATEGY COLUMN      | STRUCTURE COLUMN | SKILLS COLUMN | DATA COLUMN   | FUNCTION COLUMN | WORKFLOW COLUMN | SOLUTION COLUMN | INTERFACE COLUMN | NETWORK COLUMN |
| DECONSTRUCTION LEVEL | DOMAIN CONCEPT ROW        |                      |                  |               |               |                 |                 |                 |                  |                |
|                      | DOMAIN CLASSIFICATION ROW |                      |                  |               |               |                 |                 |                 |                  |                |
| COMPOSITION LEVEL    | GENERIC TEMPLATE ROW      |                      |                  |               |               |                 |                 |                 |                  |                |
|                      | DESIGN CONTEXT ROW        |                      |                  |               |               |                 |                 |                 |                  |                |
| IMPLEMENTATION LEVEL | OPERATIONAL BOUND ROW     |                      |                  |               |               |                 |                 |                 |                  |                |

plement the FSDM. They also wanted a framework that would show how the data and function models were used to analyze business information and develop applications. The IFW project started in 1991, growing out of FAA and FAS90. The Financial Services Function Model was first released in 1993, followed by the Financial Services Workflow Model in 1994. Other materials have since been developed and are described later in this paper.

IBM has sponsored other "industry" architectures, most notably the Retail Application Architecture<sup>16</sup> (RAA) and the Insurance Application Architecture (IAA). Each of these industry architectures was based on the Zachman framework with some modifications, and each has focused on developing "business architectures" composed of data, function, process, and workflow models, together with supporting methodologies. The IFW project has incorporated much of the experience gained from these other industry architectures.

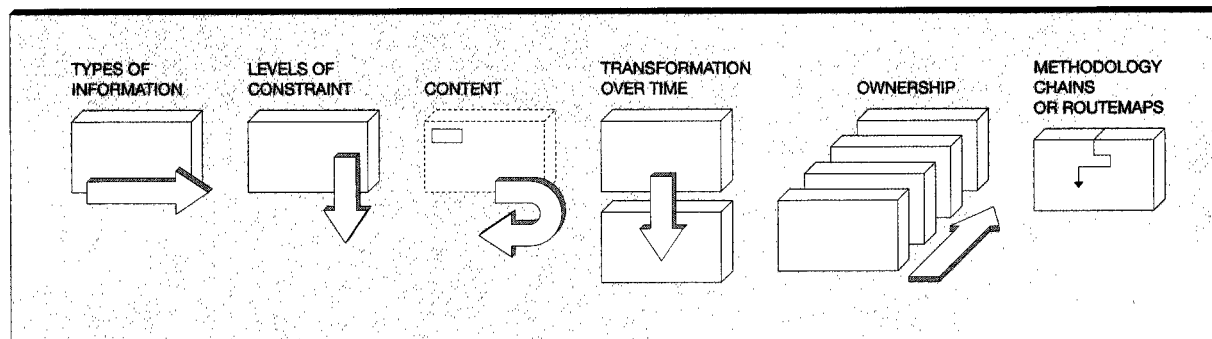
In the rest of this paper, the structure of IFW is first described through its components, the six dimensions of the IFW architecture, and characteristics. The paper concludes by suggesting how IFW might evolve in the future.

## The structure of IFW

**Components of the Information FrameWork.** IFW is a grid structure, made up of a number of components. The fundamental components are shown in Figure 3. Many of these components were used in the Zachman framework, although two constructs—"view" and "dimension" (dimension is later shown in Figure 4)—were defined during the development of IFW and are additions to those used in the Zachman framework. I will now discuss the structural components with reference to the IFW.

A *framework* is an outline structure that defines a set of cells and their relationships. The cells are grouped by columns and rows within the framework. The framework is used to explore different aspects of each cell. The framework is a systematic taxonomy that can represent the subject matter of the framework from a number of different perspectives.<sup>17</sup> For example, IFW is made up of 50 cells that provide a taxonomy for managing information, whereas Zachman's framework has 30 cells, produced by the matrix between six columns and five rows, that provide a way of viewing a system from many different perspectives and showing how they are related.

Figure 4 The six dimensions of the Information Framework



A *view* is a way of looking at a subset of columns in a framework, representing a set of interests. Views are at a higher level of abstraction than the columns in information systems architecture. The views of a framework identify the different perspectives of the broad groups that will use the framework. For example, in IFW the views represent the perspectives of organization strategists and managers, business analysts and designers, or technical architects and builders. IFW uses the notion of "view" to distinguish three broad entry points for understanding the information used across an institution (Figure 2).

The *columns* of the framework are used to distinguish or separate the subject matter of the framework into a number of broad categories or groups. Each column in a framework isolates one abstraction of the subject from other abstractions. This means that one column of the framework can be analyzed in isolation from other columns; for example, data can be analyzed separately from process, which is a useful technique in domain analysis. In IFW the columns represent various ways to represent different types of information. An understanding of different information types was determined from an analysis of the available methodologies for business and information systems modeling.

The *rows* of IFW show information at different levels of constraint on what is possible, ranging from representational forms that simply classify and define information elements, to representational forms that include design decisions, technical constraints, and organizational constraints. In IFW there are three broad levels, each of which can be further subdivided as necessary. Each row in the framework isolates a different representation or set of constraints from other representations to satisfy different purposes

and objectives. For example, one row often uses a modeling technique different from subsequent rows. A representation is one of the many ways of depicting the subject matter within a framework and can be textual, graphic, or pictorial. Within IFW, representations include classification hierarchies, entity relationship diagrams, organigrams, business strategy models, state transition models, workflow models, and process dependency diagrams.

A *cell* is an intersection between a column and a row in a framework. In IFW each cell therefore represents a particular type of information (columns) defined within a specified level of constraint (level or rows), for example, information about data represented as a classification hierarchy. Each cell is a key unit in the framework architecture, and defining the purpose of a cell is important for understanding the value of that cell within the overall structure of the framework. For example, the interaction between the data column and the domain classification row might be used to achieve a rapid and consistent definition of data requirements across multiple projects and across different organization units through the use of a data classification model.

The *dimensions* of a framework represent architectural perspectives. They define a number of criteria that are used to separate different parts of the architecture. For example, columns and rows are two of the six dimensions of IFW (Figure 4). It is often difficult to visualize more than two dimensions at any one time. Research into information frameworks at the Cranfield School of Management suggests that there are multiple dimensions and that analysis is most constructive when any two of these dimensions are viewed together in a matrix.<sup>18</sup> The IFW and ISA diagrams are two-dimensional drawings.

### **The six dimensions of the Information Framework.**

It became clear during the development of IFW that there were a number of dimensions to the architecture. The grid structure diagram of IFW shows the first two dimensions—the columns and rows, representing the types of information and levels of constraint.

The third dimension is the intersection between a column and a row, representing a specific cell in the grid. This dimension records knowledge about aspects or content of a cell. These first three dimensions are all concerned with different components so that information can be managed more effectively. The goal is to replace the complexity of the information resource within a large corporation and break that complexity into more manageable chunks—the cells of IFW. The criteria for defining a cell come from dimensions one and two. At the lowest granularity, a cell is the intersection of a column and a row. At a higher granularity, a cell can be the intersection of a view and a level.

Having defined the components of an information architecture, the next three dimensions have to do with using this resource in the most effective manner. Inevitably there will be changes to the resource over time, and these changes should also be managed and controlled. The fourth dimension is designed to make this task easier. This dimension covers the transition or transformation from one version of IFW to another over a period of time. The time scale can be architectural in duration, covering any period from one to twenty or more years, or project-related, covering any period from one week to five years. This dimension also handles aspects of information management such as versioning.

The fifth dimension acknowledges that content within any cell in the framework could have several owners. For example, much of the data defined in a data model, such as data about an individual or person, would be true not just within the financial services industry but across the insurance, retail, petroleum, or travel industries. The fifth dimension therefore includes information “ownership” at a global, industry, cross-enterprise, enterprise, local, or individual level.

Finally, the material in the other dimensions of IFW are used in many different project situations. The sixth dimension includes the project, process, or routemap views through IFW. Very often the material in individual cells of IFW is used in different ways,

depending on a preference for one methodology or another. This aspect of information management is covered in this dimension. For example, an object-oriented development would use information in the data, function, and workflow columns, but it would typically combine them as objects with related methods and messages—in other words, two or more of the architectural cells of IFW (separated in the first three dimensions) have been combined together because of methodological need (in the sixth dimension).

**Characteristics of IFW.** The ordering of the columns in IFW, together with the basic structure of the framework, is designed to allow for *reuse of information* in any of the cells. The words “data” and “information” have often been used in a similar sense, especially when related to computing.<sup>19</sup> In IFW, “information” is taken to mean a set of components that together comprise the knowledge or experience about a given domain, such as the financial services industry. These components can be both elemental and aggregate. Typically what we regard as information is a complex grouping of components that could be viewed from many different viewpoints. One of these viewpoints is the “data” perspective. Data as an entity is one of the basic building blocks used to create information. Information is often stored as a combination of data pieces. IFW is used to analyze a piece of information, break the information into its constituent components, and position the components in the appropriate cells of the framework. Data information includes such items as an individual’s name, sex, and marital status. Workflow information includes the activities and tasks that are carried out.

Breaking information into these components makes it easier to define each information component once. Each component belongs in only one of the cells of the framework. As with any filing system, it is important that the placing of components within the cells follows some guidelines or rules. For example, internal mail for an office may be sorted and distributed using an alphabetic pigeonhole filing system. In contrast to this system, where each item must be positioned in one pigeonhole, IFW uses a *faceted classification system* where each compound piece of information is composed of items selected from a number of perspectives or facets.<sup>20</sup> (For example, a book might be found in a library by author, title, subject matter, country, and book format.) Information in this context includes not only data, but process and other types of business information, organizational

information, and technical information. Guidelines documented for IFW help to identify the different types of reuse (cell by cell) and show how individual reusable components are created and managed.

As well as providing a comprehensive structure to manage information created from many diverse processes, IFW was *designed to accommodate a variety of approaches to information management*. Research by Michael Earl suggests<sup>21</sup> that there are five different approaches for strategic information systems planning (SISP), each with different characteristics, strengths, and weaknesses. Although the research is based on information systems, the five approaches can be used in a more general sense to understand the different styles of information management. The five different approaches, with a discussion of how they are related to IFW, are:

1. Business-led—where the emphasis is on the business leading information systems through business plans and needs and not vice versa. IFW covers this approach via the business view of the framework (Figure 2).
2. Method-driven—where the emphasis is on the selection of the “best” formal SISP method to enhance information systems strategies. IFW provides a number of routemaps (discussed in a later section of this paper) that provide alternative methodological routes for using the framework.
3. Administrative—where the emphasis is on identification and allocation of information systems resources to meet agreed-to needs within the rules, management planning, and control procedures of a firm. The IFW structure in total supports this approach, using the framework to administer the information resource.
4. Technological—where the emphasis is on the production of models and blueprints and SISP is regarded as an exercise in business and information modeling. IFW covers this approach via the technical view of the framework (Figure 2).
5. Organizational—where the emphasis is on organizational learning about business issues and the information technology contribution and SISP is seen as a continuous decision-making activity shared by the business and information systems. IFW covers this approach via the organization view of the framework (Figure 2).

This taxonomy of approaches can be used “as a diagnostic tool to position a firm’s current SISP efforts” and also “to design a situation-specific (customized) approach on a ‘mix-and-match’ basis.”<sup>21</sup> Earl’s re-

search suggests that a hybrid approach that combines elements from each of the five different strategic information systems planning approaches is potentially the most effective. The structure of IFW can be used in such a way to analyze strategies and approaches cell by cell, combining the cells into an optimum mix-and-match approach. IFW in this sense is a superset of all of the different approaches, methodologies, and types of information that need to be related, integrated, compared, or otherwise coordinated.

### Dimension 1: Types of information

**Three views in IFW.** Projects using IFW have shown that it is useful to have views that separate organization, business, and technical information. Having these views is partly because the users and the nature of change are different for each view. The three separate views also make it easier to develop stable models.

IFW is intended for use by:

- Those involved in planning strategies, in designing the structure of the organization, and in developing skills—hence, in the analysis of change and the impact of change within a particular organization
- Those involved in analyzing the business, in designing customer solutions, and managing, operating, or supporting the business, for example, application developers
- Those responsible for providing the technical infrastructure that supports the computer-assisted part of any solution. This infrastructure includes the analysis of application interfaces and structures, network distribution, and system platforms. It also involves combining the business requirements for a solution with a technical platform that will be used to implement the solution.

A specific *organization* is involved in running one or more types of *business* and may require the support of a *technical* computer-based infrastructure. The three views of information form a progression from the view of the organization itself where the focus is on strategic planning, business transformation, or organizational impact, through a view of the business or the industry that the organization operates (providing financial services or insurance), to the view of the technical platforms and structures used to provide computer support for that business—where the focus is more on architectural principles, isolating the business needs from implementation and interfaces between technologies.

Another reason why the three views are separated is that the pace of change in each view is different. When change is introduced in any one of the views, a financial institution has a number of choices for the remaining views. One option is to contain the impact by trying to minimize the consequent change in the other views. A second option is to evaluate opportunities for simultaneous change in the other views that might enhance the potential benefits. Separating the views provides a means to analyze each perspective individually or in combination.

Finally, a major reason for this separation is that it is much easier to define generic industry-wide models when the views are separated. For example, it was possible to define a stable Financial Services Function Model partly because the organizational information, such as the organization structure, was separated into the organization view.

**The columns.** One of the primary criteria for the six columns in the Zachman framework is the set of questions purportedly used by journalists—what? how? where? who? when? and why?<sup>22</sup> However, the same questions apply within each column as well as across the columns. As an example, the data column contains information about *what* data items a financial institution requires, *how* different data elements are grouped together, and *why* certain data items are associated with others. IFW chose different criteria to determine the nature of its columns. One of the most important criteria was to ask “What were the major categories or types of information required for each view?” Note that this line of thinking is extended into the first row of the IFW, as the “concepts” for each column provide a further breakdown or classification of the information type. For example, the information type “data” is further classified into the nine data concepts discussed later in this paper.

Zachman and Sowa have stated categorically, in the “rules of the framework,”<sup>23</sup> that there is no order in the columns of the ISA framework. In contrast, the views and columns of IFW are grouped into a logical order (see Figure 2). This grouping was partly determined by the stability of information in each column ranging from relatively stable to relatively dynamic. The order of the columns represents a stability factor. For example, in IFW data are regarded as more stable than the functions, which are more stable than the workflows. The most dynamic aspect of the business is likely to lie in the solutions offered to its customers. The notion of mass customization requires a stable base of data, function, and workflow com-

ponents that can be grouped in such a way that the solution is customized for every customer.<sup>24</sup>

There are two aspects to column order in IFW. The columns have been placed in a specific order within the views following the architectural principles described in this paper, but the order in which the columns and cells are used in any given project could vary to allow for individual work preferences. Studying the work practices of people who use information, such as application developers and business analysts, and the methodological sequencing of information analysis also helped to identify what the logical grouping of the columns should be. Research into and experience with reusable components<sup>25</sup> in analysis, design, and development also provided useful insights that were incorporated into IFW. These studies, some describing successful projects and others describing failures, have shown that the development of information assets designed for reuse, techniques for classifying information assets, and viewing systems development from an integrated perspective are among the critical success factors for the governance of information on a large scale.

**The organization view.** The organization view represents many of the information elements that will differentiate one financial institution from another. This differentiating information is categorized into the strategy, structure, and skills of that organization. Information in this view is necessary to complete the requirements for business solutions. For example, it is necessary to know the locations in the organization structure where a solution will be used, as well as knowing who will use the solution. The skills within the organization will determine the core competencies of the financial institution, as well as determining whether a solution is aimed at those with a great deal of experience or those who have little banking knowledge. Many of the components in these columns are defined in the Zachman framework from a systems rather than an organizational perspective.

Organization information is increasingly important as financial institutions strive to build information systems that support business goals, reengineer business processes that cross organizational structures, and recognize that skills within an organization often distinguish one bank from another. Many methodologies have included constructs and techniques that take into account organization structure charts, goals, and critical success factors—for example, in information engineering. The current interest in in-

tegrating diverse methodology chains is a positive step toward enhancing the integration of this information. IFW has been designed to take advantage of these trends by providing the umbrella structure and methodology to compare and coordinate appropriate techniques from diverse methodologies.

**The business view.** The first column of the Zachman framework, as illustrated in Figure 1, describes data, whereas the second column describes process or function. It is a traditional systems view, as defined in information engineering<sup>26</sup> with analysis based on two basic types of models—data and process. Information engineering does not necessarily cover all of the alternative approaches such as object-oriented, prototyping, or workflow analysis. To manage the information output from all sources within a company requires a structure that is sufficiently comprehensive to contain different information types. IBM reviewed alternative ways to categorize information or methodologies. One attempt to classify the different information systems development methodologies proposed a simple and useful classification<sup>27</sup> based on three different aspects of information systems development—function analysis (a process-driven perspective from the late 1960s), object analysis (a data-driven perspective from the mid-1970s), and event analysis (a behavior-driven perspective from the early 1980s). This classification suggested that three columns were necessary to represent business information.

These three aspects, related to IFW, are:

- The data column—focuses on (1) the objects, entities, or things about which we store information, including relationships or associations between two or more of these objects, (2) definitions, identifiers, attributes, properties, or descriptors of these objects and their relationships, and (3) constraints that apply directly to the objects, their relationships, and attributes (such as cardinality and value domains).
- The function column—focuses on (1) the static aspects of function or process, including the business functions or functional areas that are managed by a financial institution, (2) the states and allowable transitions that apply to data defined in the previous column, and (3) the data access actions that apply to data defined in the previous column.
- The workflow column—focuses on (1) the dynamic or behavioral aspects of process or workflow, including the activities, procedures, or processes

within the business, (2) the events or triggers that initiate and connect these activities, and (3) the conditions and behavior logic that control the flow between the activities.

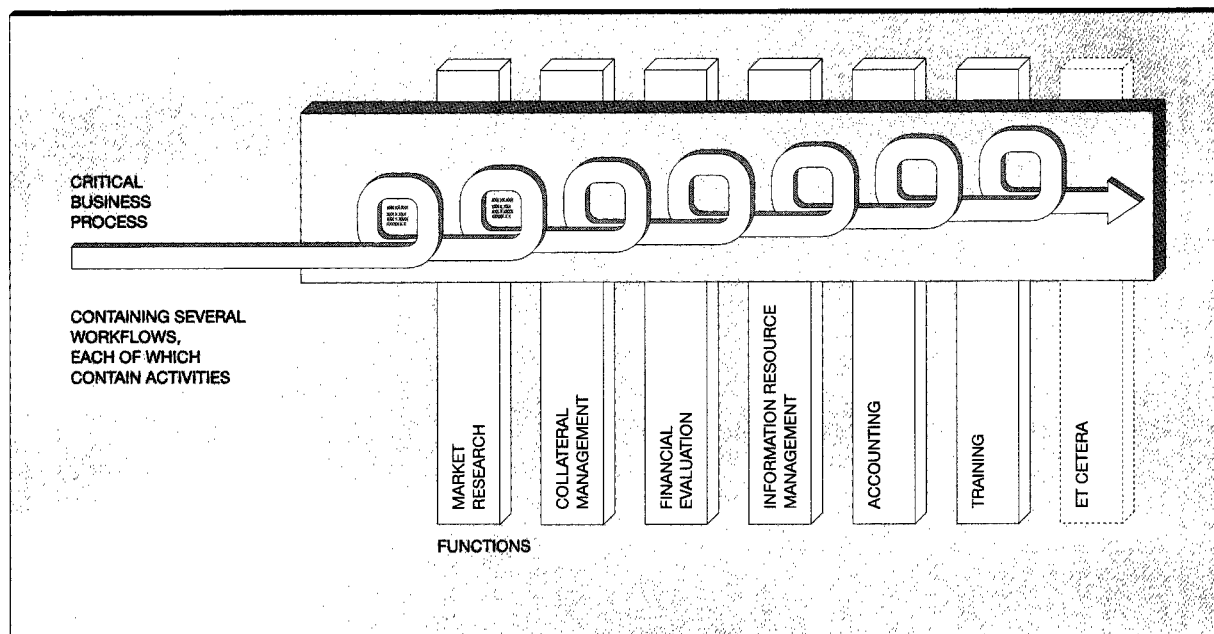
In developing the Financial Services Function Model it became apparent that there was a great deal of confusion over the terms “function,” “process,” “business process,” “activity,” and “workflow.” There is no general consensus on all of the constructs that form such models. The process/function column in the Zachman framework does not differentiate between these terms.<sup>28</sup> From various methodologies and papers, Curtis, Kellner, and Over<sup>29</sup> abstracted a description of the most frequently used elements to give a more precise definition of the term “process model.” These elements include what is going to be done, who is going to do it, when, where, how, and why it will be done, and who is dependent on it being done.

They used these elements as the basis for defining four different perspectives of the actual process—*organizational, informational, functional, and behavioral*. Of these four perspectives, the organizational perspective is covered in the organization view (strategy, structure, and skills columns) of IFW. The informational perspective is covered by the data column. The distinction between the functional and behavioral perspectives is shown by IFW in the next two columns—function and workflow. The function column covers the more static aspects of what is going to be done, whereas the workflow column covers the behavioral aspects of when it will be done and how it will be done. The distinction between function and workflow has proved useful in helping to separate the different types of information and thereby creating reusable components.

Recent texts on business process reengineering<sup>30</sup> make a similar distinction between function and workflow. A function can be found throughout the largely hierarchical or vertical structure of the organization (see Figure 5). Davenport contrasts the vertical and horizontal views by saying that “whereas an organization’s hierarchical structure is typically a slice-in-time view of responsibilities and reporting relationships, its process structure is a dynamic view of how the organization delivers value.”<sup>31</sup>

To summarize the key differences between the function column and the workflow column: a function is relatively static, is found vertically throughout the structure of the organization, is not time-dependent,

Figure 5 Vertical functions versus horizontal business processes and workflows



and is often related to the organization structure and strategies; a workflow is dynamic, is found horizontally across the structure of the organization, has a start, a middle, and an end, and is related to people and their roles within the organization.

The solution column holds information about the combination of generic components from the data, function, and workflow columns that are used to create business solutions. Every time that a financial institution generates an information-based solution, the requirements are based on a subset from the models in these three columns. Note that there is a distinction between the definition of a "business" solution and its possible support by computer software. Some of the solutions, but not necessarily all of them, will be supported or implemented by computer systems.

The solution column includes solutions that are provided by the business to its customers as well as solutions that are required internally within the organization. It therefore includes *products* such as loans and deposits, *services* such as financial investment advice, and *support* such as administrative and accounting systems, management information, and decision support.

For example, two types of solutions might be loan or deposit products. The requirements for each product are defined in terms of the data required to support the product, the functionality provided by the product, and the processes or activities that use the data to provide the functionality of the product. Information in the solution column is used by business professionals to focus energy on the solutions that they are providing to their customers, making changes where there is a rapid impact on their business. Once solutions have been structured from generic reusable components into groups of conceptually similar solutions, it is easier to support mass customization and reduce delivery time of new products into the market.

The Financial Services Data Model was the first model to be defined for the finance industry. This model provided a stable and generic set of data definitions for a particular business and became a foundation to define the function and workflow models. It is not a coincidence that the financial service models were developed in the order of data first, followed by function, followed by workflow. Each model was built using information defined in the previous model(s). The function and workflow models do not repeat data definitions that are in the data model, but

simply cross reference the existing data definition. Similarly, the workflow model does not repeat function definitions, because these were already provided in the Financial Services Function Model.

The different types of information are contained within different cells of IFW. What the data look like, the data definition, is in the data column. This data definition might be reused in many different functional contexts. What we do to the data, such as the basic create, read, update, delete, and add routines, is in the function column. This functional definition might be reused in many different process or workflow contexts. Finally, how we use those data once we have accessed them is in the workflow column.

**The technical view.** Just as the business view is used to understand the nature of the particular business or businesses that an organization is operating, so the technical view is used to understand information about the technical structures that support the business. The subdivision of the technical view into three columns is based on many existing technical architecture models that layer the technical platform into an interface layer, a network architecture, and a systems platform.<sup>32</sup>

In the Zachman framework, the structure of a program is assumed in the design in the function column. In IFW, information about the structure of a program—including the types of components that make up an application such as subroutines or modules, the interfaces that link these components such as application programming interfaces or language constructs, and the overall technical structure of a program—is included in the interface column. This information is viewed separately from the business content defined in the business view of IFW. For example, the definition of the business flow logic supported within an application component is contained in the workflow column, whereas the business definition of the data is contained in the data column.

In a similar way, the network and platform columns are used to record information about the structure of the network that a financial institution operates and about the underlying system platforms. The network column includes information about protocols, devices, and the network topology. The platform column includes information about the operating system, object, document, or data storage systems, and resource managers such as workflow, transaction, or electronic image managers.

Information described in the Zachman framework combines organization, business, or technical representations which in IFW are separated into the distinct views. For example, in the Zachman network column there is a mixture of business location information that is placed in the organization structure column of IFW, the network architecture addresses and protocols that are placed in the technical network column of IFW, and system architecture components that are placed in the technical network and platform columns of IFW.

## Dimension 2: Levels of constraint

### Redefining the rows of the Information Framework.

In examining the rationale for leveling the rows in IFW it became clear that existing definitions were based on more than one distinctive set of criteria. For example, in the Zachman framework the rows are distinguished partly by whether they represent a business or a technical perspective, partly by the role of the person who uses the model at that level, and partly by the type of model (see Table 2). In IFW different criteria are separated into distinct dimensions. For example, the distinction between technical and business is based on the different types of information that are required in each view as discussed in Dimension 1, the information user could be a different person or role depending on which routemap is being used to navigate IFW as discussed in Dimension 6, and the type of model used to represent the information is part of the cell content discussed in Dimension 3. What is left in Dimension 2 are the types of constraint that apply to information.

Redefinition of the Zachman levels by IFW is subtle rather than radical. IFW takes into account object-oriented modeling constructs that are achieving wide acceptance in the 1990s, the need to manage reusable assets, and the existence of industry-wide models. Some of the criteria for defining the rows in the Zachman framework have been moved to separate dimensions in IFW. Combined with the changes to the columns of the framework, IFW represents a significant departure from Zachman's work.

Over the last 30 years, successive development methodologies have tried to remove some of the constraints and provide more flexible information representations. Providing a number of layers of abstraction is one way of addressing this flexibility. The Financial Services Data Model (FSDM) was developed as a layered model, following the practice of information

**Table 2 Criteria for separating rows in the Zachman framework**

| Row Label                                               | Perspective                         | User                     | Type of Model or Representation                                                                                                                                             |
|---------------------------------------------------------|-------------------------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope<br>Enterprise model                               | Business<br>Business                | Planner<br>Owner         | List<br>Some are business oriented, e.g., business plan or organization chart<br>Some are more technically oriented, e.g., entity-relationship diagram or logistics network |
| System model                                            | Business                            | Designer                 | Mixture of detailed information engineering models (data model and data flow model) and architectures (distributed system architecture, human interface architecture)       |
| Technology model<br>Components<br>Functioning<br>system | Technical<br>Technical<br>Technical | Builder<br>Subcontractor | Technical<br>Technical<br>Technical                                                                                                                                         |

engineering that creates conceptual, logical, and physical models, and the refinement of this approach following the experience of banks involved in the IFW project. Among other inputs to this project, IBM bought a retail bank model and enterprise modeling methodology from Westpac Banking Corporation that used classifications of business entities as well as entity-relationship diagrams. The structure of the Westpac model became the foundation for the FSDM. In examining the layers of the FSDM, it became apparent that there was some comparison to a historical pattern in the evolution of application development methodologies.

The pattern is revealed by looking at broad trends across four decades. In the 1960s many banks started to use computers for the first time. Many of the early applications were fairly simple—partly because of the limited capacity of computers at that time and partly because computing was applied to the relatively straightforward accounting transactions of the bank. When these early systems were built, the requirements were not very complex from a business perspective, and code was often constructed without a detailed or formal logical design. In data terms, examples include the data definition language or program declare statements.

During the 1960s banks started to use computers more and more, and by the 1970s applications were becoming more complex. It was around this time that a number of logical modeling techniques were advanced to diagrammatically represent the structure of the data or the logic of the program. Because most applications were developed as stand-alone systems, it often meant that for every program and system there might be a separate set of diagrams. Although analysts knew that some data and functional spec-

ifications were repeated in each new development, there was no simple way to prevent this rework, and the pressure to complete applications left little time to coordinate effort from one project to another. In data terms, examples would include entity-relationship diagrams, data flow diagrams, program structure charts, and function or process decomposition diagrams<sup>33</sup> for a specific application.

In the 1980s several of the larger banks, recognizing that there were redundant data specifications in their systems and overlapping functionality in their application code, tried to develop enterprise-wide models. This development was a sensible progression from the independent models for each system, but it proved to be a complex and time-consuming task. For a start, no one had defined an adequate methodology for integrating a number of models. Typically, a bank might try to integrate 30 or 40 current systems models into one enterprise-wide model, and this task might require rationalizing 15 000 implemented data items with a standardized business definition. Not surprisingly, some of these projects were not completed, and others failed to deliver the promise and benefits of a single corporate model.<sup>34</sup> In data terms, examples would include enterprise-wide entity-relationship diagrams with business definitions of each entity.

In the 1990s, techniques from object-oriented analysis and knowledge-based systems became more widely accepted, resulting in a greater use of, for example, class hierarchies, inheritance, encapsulation, and polymorphism. In some respects, object-oriented models are an alternative way to represent the business information shown in information engineering models. In IFW, object-oriented techniques influenced the development of industry-wide models and

**Table 3 Historical evolution of models**

| Decade | Type of Model                                                         | Row in IFW<br>(Letter) | Row in IFW<br>(Name)                    | Corresponding Row in<br>Zachman           |
|--------|-----------------------------------------------------------------------|------------------------|-----------------------------------------|-------------------------------------------|
| 1960s  | Applications developed with little use of models                      | D                      | Operationally bound                     | Detailed description and technology model |
| 1970s  | Application-specific ERDs and data flow diagrams                      | C'<br>(C Prime)        | Design context                          | Model of the information system           |
| 1980s  | Initial attempts at enterprise-wide models                            | C                      | Generic template                        | Model of the business                     |
| 1990s  | Classification structure for "industry" models, based on key concepts | B<br>A                 | Domain classification<br>Domain concept | Scope description                         |

classification. In data terms this includes lists of entities important to the business, as in Zachman, but these are structured as classification hierarchies, and grouped within concepts.

Table 3 summarizes this historical pattern in relation to the levels of IFW, showing how data models have had a different focus over time. This historical view of the FSDM was applied to the rows across each of the columns of IFW.

In IFW, each row was originally identified by a reference letter. These terms are shown in Figure 3 and described here. The A Level represents the domain concepts used for classifying the information within a given column. The B Level represents domain classifications that provide the classification of information within a given column. Classifications are used for analysis and communication and are based on the domain concepts. The classification structures of IFW enable business analysts to quickly define their needs while at the same time allowing those choices to be traced to predefined logical models. The C Level represents the generic templates that group the information items classified in the domain classifications into high-level logical models. The C' (so-called "C prime") represents the design context that also groups information items into a logical model, and the same modeling techniques can often be used for both this and the previous row; however, the design context row contains detail that is needed for a specific project or application context. The design generally contains information that is constrained by the design techniques that have been used, as well as design decisions that are determined by the organizational, business, and technical contexts for the design. The D Level represents information about a particular design that happens to be implemented. In many cases this perspective is fixed and cannot be changed without going through a managed pro-

cess to ensure that all changes are authorized and correct. It is for this reason that the row is described as "operationally bound"—meaning that it is fixed, constrained, and restricted within the operations of the financial institution. Because the reference letters have no meaning in themselves and were therefore open to interpretation, this identification has been replaced by a name that explicitly describes the use of each level. For historical continuity both reference letters and descriptive terms have been used in this paper. Table 4 shows the techniques that are used for each row in IFW with some consequent benefits and observations.

**Three levels in IFW.** As the columns in IFW have been grouped into three broad views, so the rows in IFW can be seen as three broad levels, the deconstruction level, composition level, and implementation level. The label applied to each level describes the basic process that underlies the analysis and definition of information at that level.

The domain concept and domain classification rows are both included in the deconstruction level. Deconstruction denotes that all contextual constraints have been removed as far as is possible. The process involves taking constructed information models, understanding aspects of the model that are present due to the necessary constraints that applied at the time the model was defined, identifying terms or elements as distinct and definite, then organizing the elements in a classification hierarchy.<sup>35</sup> At the deconstruction level of IFW each view is quite distinct. For example, the domain concepts and classifications of the business view are defined independently of the technology that might be used to implement a solution and of the particular organization that operates the business. The technical view at the same level provides the concepts and classifications to help structure and manage interfaces, networks, and plat-

Table 4 Rows in IFW

| Row Letter | Row Name                  | Techniques                                                                                                                                                         | Benefits and Observations                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A          | Domain concept row        | <ul style="list-style-type: none"> <li>• Conceptual analysis</li> <li>• Domain analysis</li> </ul>                                                                 | <ul style="list-style-type: none"> <li>• Provides a clear structure within the column</li> <li>• Provides concepts that are difficult to identify from bottom-up analysis</li> <li>• Helps to focus on reuse</li> </ul>                                                                                                                                                                                                                   |
| B          | Domain classification row | <ul style="list-style-type: none"> <li>• Classification theory</li> <li>• Domain analysis and modeling</li> </ul>                                                  | <ul style="list-style-type: none"> <li>• Based on concepts from the previous row</li> <li>• Easy to understand and learn</li> <li>• Good communication medium for business and technical people</li> <li>• Supports rapid analysis approach</li> <li>• Valuable foundation for industry- or enterprise-wide models</li> <li>• Does not predetermine or constrain the structure in the next rows</li> </ul>                                |
| C          | Generic template row      | <ul style="list-style-type: none"> <li>• Logical modeling techniques appropriate to the type of information within the column</li> <li>• Generalization</li> </ul> | <ul style="list-style-type: none"> <li>• Uses items classified in the previous row</li> <li>• Provides templates that can be used across multiple contexts to provide consistency and simplification in detailed models at the next row</li> <li>• Provides flexible structures, generalized to a point where the template is useful rather than so abstract that it is useless</li> <li>• Facilitates rapid model development</li> </ul> |
| C'         | Design context row        | <ul style="list-style-type: none"> <li>• Logical modeling techniques appropriate to the type of information within the column</li> <li>• Specialization</li> </ul> | <ul style="list-style-type: none"> <li>• Based on the generic template of the previous row</li> <li>• Allows comparison of details across multiple contexts (e.g., different applications or different organization units)</li> <li>• Used in conjunction with generalization techniques so that there is a good balance between information defined at this row and at the generic template row</li> </ul>                               |
| D          | Operationally bound row   | <ul style="list-style-type: none"> <li>• Implementation modeling techniques appropriate to the information in the column</li> </ul>                                | <ul style="list-style-type: none"> <li>• With a clear trace to the previous rows maintenance and management become much easier</li> </ul>                                                                                                                                                                                                                                                                                                 |

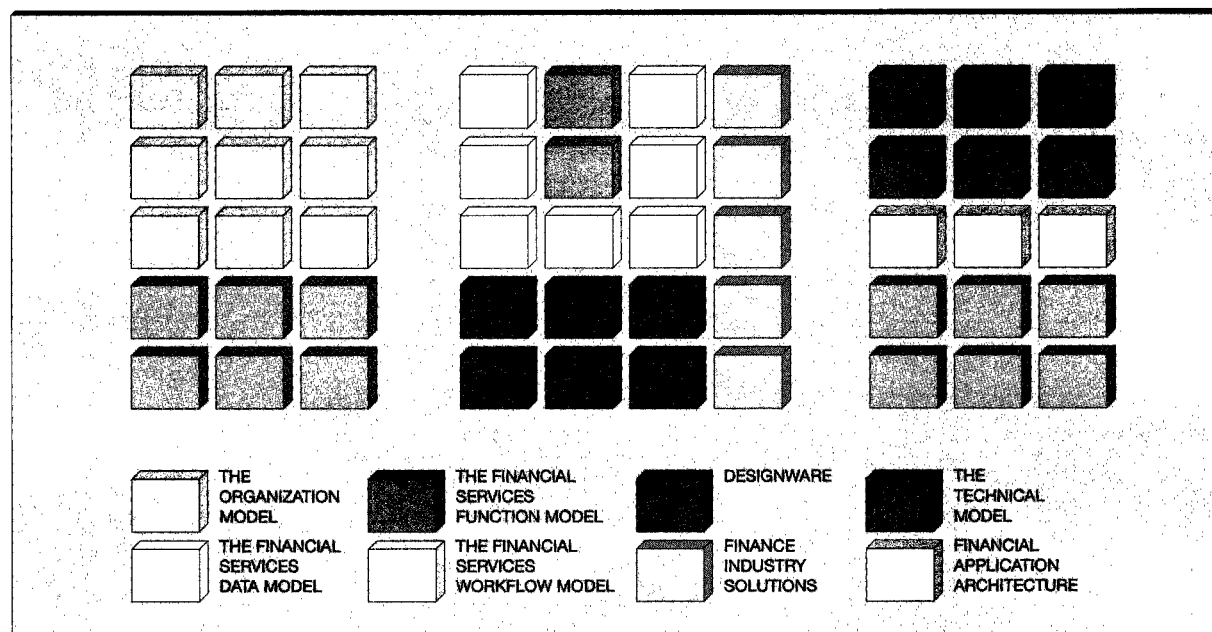
forms. It is only at the composition level, in particular in the detailed design context, that each view needs to take full account of the other views and the constraints that they impose on the design.

The discussion of Dimension 1 gave reasons for each view being distinct. At the deconstruction level this distinction is maintained not only for each view but also across each of the columns since it allows definition of domain models that are generic and can be applied in many different situations. For example, many function models using information engineering reflect the structure of the organization at the time the model was created. When the organization structure changes, the function model is often not updated, so the model rapidly becomes out of date. Defining a function model that specifically excludes organizational aspects can provide a bus-

iness model that is more stable. This stability is a necessary prerequisite for constructing reusable building blocks.

In the generic template and design context rows, information elements are grouped into logical models. This process can be seen as one of composition, where elements are selected or "scoped" from domain classifications for appropriate columns, to be combined as a logical model for a given context. Composition, as in music or literature, uses predefined ingredients according to theory, standards, guidelines, style, and creativity. Finally, in the operationally bound row, the underlying process is implementation of the design that was chosen from the possible options identified during composition. The constraints in this row are generally imposed through historic circumstance, such as inherited data struc-

Figure 6 Content models in the Information FrameWork



tures that reflect the needs of sequential access batch processing rather than on-line relational database access. Operationally we are constrained by history rather than the bounds of our imagination!

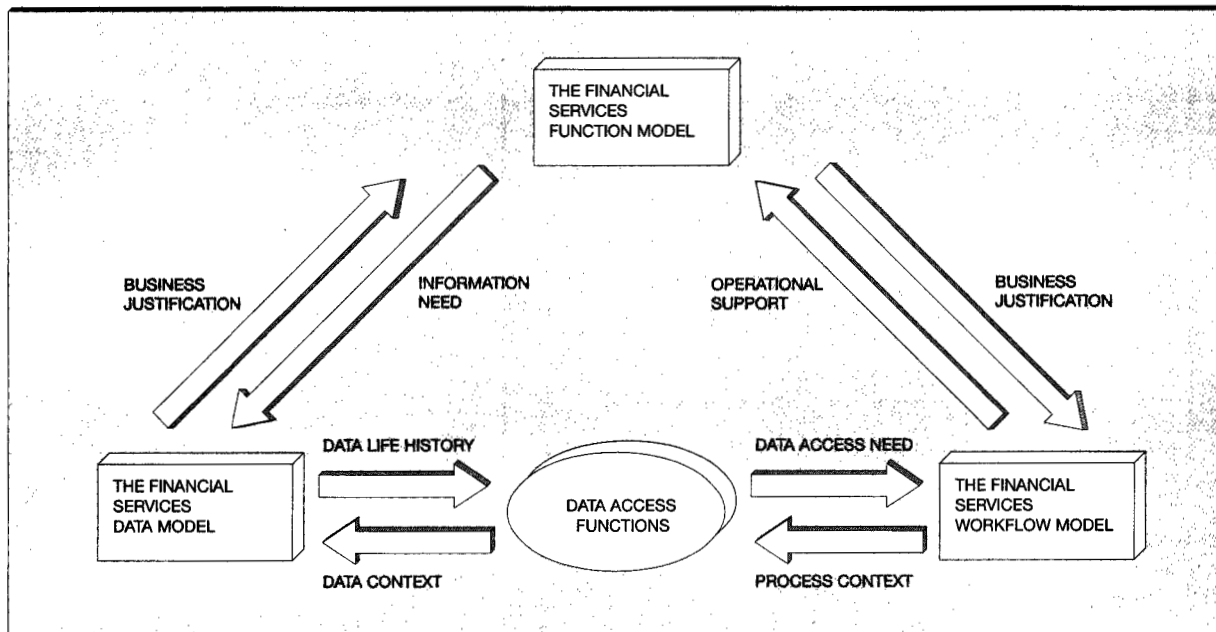
### Dimension 3: (Cell) content

**Domain models.** Perhaps the most significant aspect of IFW is that we have “populated” many of the cells, not only with a detailed architecture, but also with business content using industry-wide models that predefine much of the business knowledge that is common to all financial institutions (see Figure 6). A domain is a field or scope of knowledge, and a domain model is therefore a representation of a particular field or area of knowledge. The body of information recorded in a domain model represents knowledge that can be used when analyzing and understanding the problems that exist in the domain. This information also supports the definition and development of solutions to those problems. IFW is used to manage a complex problem domain by breaking the necessary information into a number of closely related domain models. A body of information can be considered a problem domain if “*deep or comprehensive relationships* among the items of information are known or are suspected with respect to some class of problems, there is a *community* that has a

stake in solving the problems, the community seeks *software-intensive solutions* to these problems, and the community has *access to knowledge* that can be applied to solving the problems.”<sup>36</sup> In the introduction to a tutorial on domain analysis published by the IEEE Computer Society Press it has been stated that “domain analysis has been identified as a major factor in the success of reusability in software construction.”<sup>37</sup> IFW has taken this notion a stage further by suggesting that domain models can be created by a given community, and that these predefined domain models can be used in any project that requires access to the knowledge contained in the domain model, rather than limiting their use to software construction.

The cell content in IFW consists of a number of different elements, each representing different types of information about the domain of the financial service industry. There are three business models—the Financial Services Data Model (FSDM), the Financial Services Function Model (FSFM), and the Financial Services Workflow Model (FSWM). These are generic models for the finance industry and are discussed in more detail later in this paper. Based on these three business models are solution designs called *DesignWare* that represent template or outline designs for common solutions within the financial

Figure 7 The modeling triangle



services industry, such as loans, deposits, electronic banking, general ledger, and corporate management information. Some of these designs have been taken a step further, with generated application code for a number of alternative technical platforms.

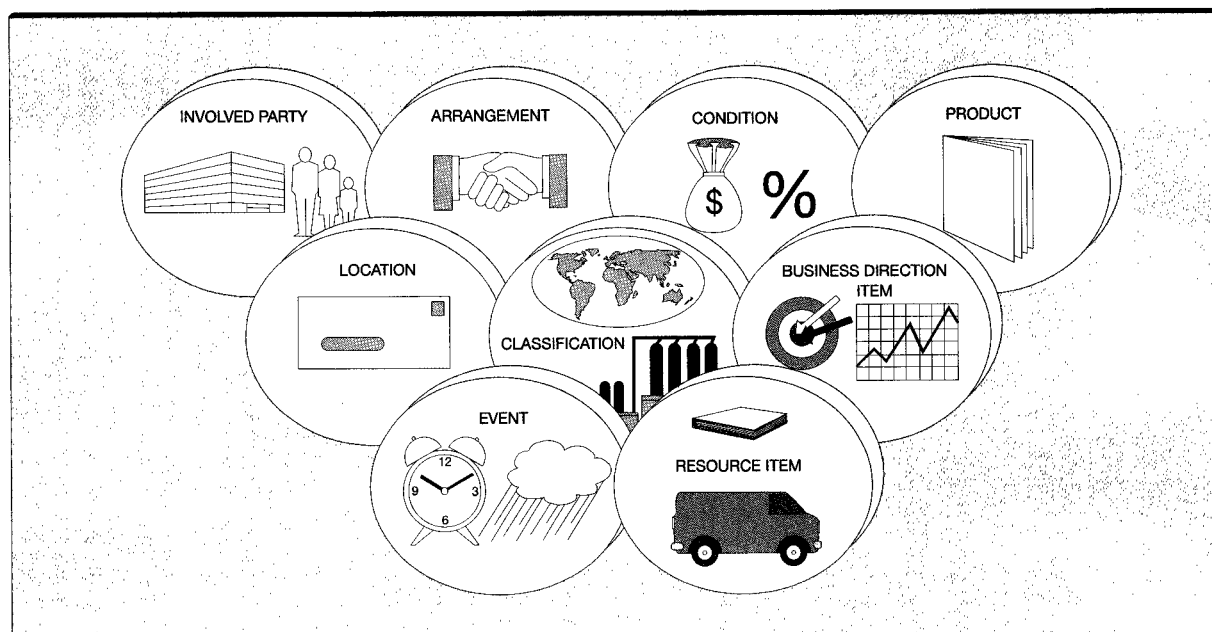
At the time of writing there is an engineering edition of the organization model and work in progress on the technical model that classify organizational and technical information. In the technical view is also the Financial Application Architecture,<sup>15</sup> a conceptual architecture for the financial services industry.

Domain concepts in the organization model include target, constraints, critical success factors, and key indicators in the strategy column; culture, role, and socio-infrastructure in the structure column; and ability, experience, training, and characteristics in the skills column. For each of these concepts, detailed domain classifications define over 2000 items representing information about an organization. The material in this model provides valuable checklists that can be used when defining corporate strategy, understanding the culture of an organization, or planning transformation and change.

Within the technical model, domain concepts include logic, language, component structure, and interface in the interface column. Each of these concepts is explored in more detail by expanding the concepts into a classification hierarchy. For example, logic is classified into types of logic, including control logic, presentation logic, business logic, data access logic, communication logic, probability logic, and rule logic. A COBOL program may contain each of these different types of logic, sometimes clearly separated but more often without these distinctions. Simply separating data access logic from other types of logic in procedural language programs could significantly reduce maintenance costs and simplify the structure of current legacy systems. The information elements in the technical model are used to define the application architecture, interface specifications, distribution architecture, and system platform architecture at the composition level.

**The business modeling triangle.** The modeling triangle shows the relationships between the three generic business models that predefine data, functions, and workflows for the financial services industry (Figure 7). Each of the models in the modeling triangle can be verified using the remaining two models. For

Figure 8 The nine data concepts of the Financial Services Data Model



example, if data items defined in the data column are selected for a given context, a corresponding function from the function column should be selected that needs those data, and the data should be used in an activity or process defined in the workflow column. Similarly, an activity should show how a function is supported and should only use data defined in the data model. In this way the three models work together to ensure a degree of completeness and quality. Each model in the triangle has a dependency on the other two models, creating a balanced structure among the three models that helps to secure accuracy and integration among the models, and also helps to promote reuse.

*The Financial Services Data Model.* The data model was the first industry-wide model built for the financial services industry. The development was initiated to address the problems that individual banks were facing in trying to build enterprise-wide data models. The estimated effort to develop a finance industry model was more than 100 person years. It included the purchase of a number of existing "enterprise-wide," or banking-specific, data models and a project combining experience, involvement, and feedback from leading banks around the world. It became apparent that it was more effective to define an industry-wide model through cooperative ef-

fort and combined resources than it was to develop individual enterprise-wide models.<sup>38</sup> An industry model can predefine at least 80 percent of individual bank requirements and therefore represents considerable savings in time, resources, and effort for any given bank. It can also help to provide a model that was consistent, relatively complete, and of high quality.

The FSDM defines nine data concepts such as involved party, arrangement, location, and product (see Figure 8). Each of the concepts is represented by classification hierarchies. During the development of the FSDM it was apparent that the definition of clear, simple hierarchies required a combination of both bottom-up and top-down analysis techniques. From the bottom it is necessary to have a wide and detailed base, representing enough information to give a full enterprise-wide perspective. From the top it is necessary to clearly define concepts that will distinguish the different information components.<sup>39</sup> To date, the FSDM has provided a model representing 80 percent of the data in each of the banks using it.<sup>40</sup>

*The Financial Services Function Model.* The FSFM is a generic model of 480 functions that must be managed by a financial institution. To define such a model it was necessary to separate the functions from or-

ganization structure and strategy. This is not to say that there are no relationships between the functions, strategies, and structure, but for the functions to be defined in a stable model that will not change significantly over time, it must be separated from the organization structure, which is subject to frequent change. The structure and strategies of an organization, in contrast, are more dynamic, changing with new market opportunities and threats. Any of the functions defined in the FSFM, such as market research, information resource management, or training, will be found throughout the organization structure. For example, training will be found at all levels in the organization, from senior management to branch teller, and from the information systems department to the foreign exchange department. FSFM defines 90 to 95 percent of functional areas managed by a financial institution.

*The Financial Services Workflow Model.* IFW uses the term "workflow" in a broad and narrow sense. In the broad sense, a large-scale workflow that is of major importance to an organization is the equivalent of a business process. Within any industry there are usually between 6 and 15 critical business processes. In the finance industry, these processes might include "negotiate loan arrangement" or "develop product." In the narrow sense, workflow refers to a particular type of implementation using software that supports workflow management, imaging, and document or folder management. In both senses, the term "workflow" reflects a horizontal view across a number of organization functions. The FSWM contains predefined elements that are used to construct workflow or process models. Domain concepts for this column in IFW include activity, trigger, verb, workflow, and critical business process. The activity classification defines more than 3000 activities that are combined to create the workflows of a financial institution. Because the activities are defined out of context, their definitions do not include information about specific linkages between activities. This means that the activities are not layered or decomposed in any way. It also means that the specific triggers that initiate an activity are not included in the classifications for the activity concept. The linkage information is provided by the combination of activities with triggers. A trigger is anything that initiates or starts an activity. The trigger classification includes over 120 triggers, grouped within six categories of generic trigger for the financial services industry.

A significant contribution to reusable activity definitions has come through the use of a naming stan-

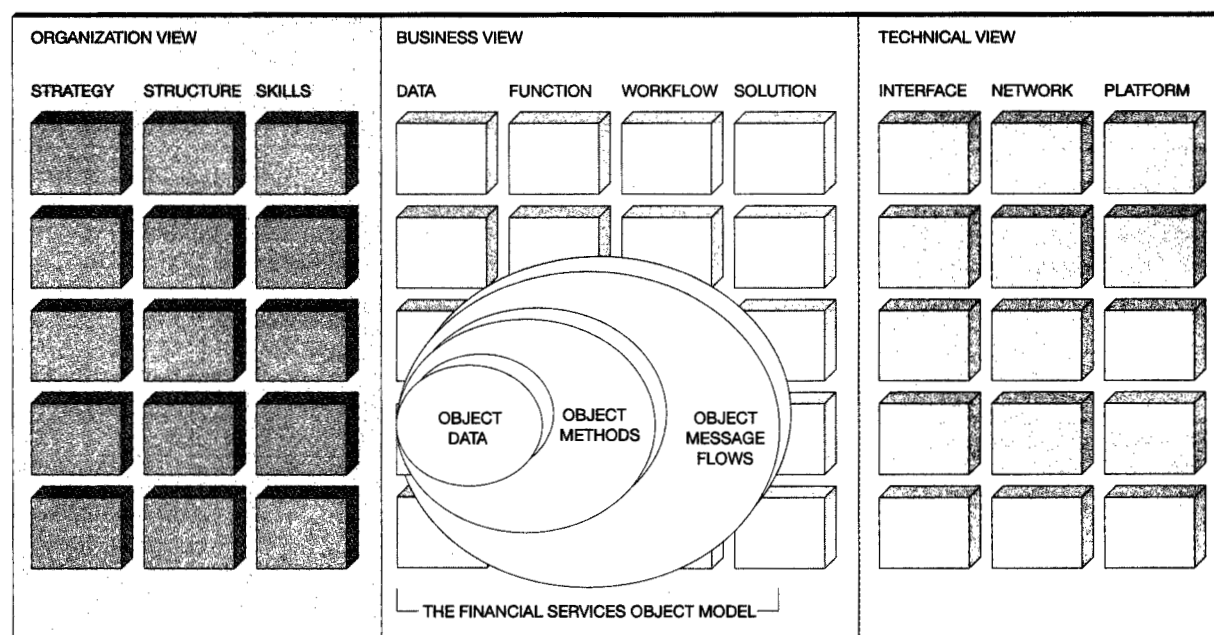
dard that is based partly on the FSDM and partly on a predefined set of industry verbs. Each activity is named using a verb and noun combination. The nouns are all derivatives of the FSDM, and the verbs are all defined in the verb classification, which defines more than 120 verbs from a financial services business perspective.

Activities, triggers, and verbs are combined to form critical business processes and workflows. Critical business processes are processes fundamental to the operation of a financial services organization. Currently IBM has identified more than 20 of these processes. Using the workflow components defined in the deconstruction levels, IBM has defined generic templates for several of these processes.

**Financial Services Object Model.** The Financial Services Object Model (FSOM) is an alternative way to represent the information required by a financial institution. Whereas the FSDM and FSWM models provide an information-engineering-oriented design at the composition level, the FSOM provides a business view of objects that reduces the large effort needed in the analysis and design of objects. This business view of objects is one of the critical factors in delivering successful object-oriented solutions. Although the final implementation of objects can be quite different from, say, that with a procedural implementation language, the underlying business requirements and information are actually very similar. Figure 9 shows how the information content of the FSOM is positioned across the cells of IFW. This object-oriented perspective is a good example of how Dimension 3 is used to compare different types of content model using a consistent information framework.

**DesignWare and business solutions.** Once the basic generic building blocks were provided in the FSDM, FSFM, and FSWM, it was possible to start designing business solutions using these components. IBM is using the term *DesignWare* to cover the generic templates and design context models that are being developed by using IFW. As DesignWare evolves it will extend from the traditional areas of retail and wholesale banking to the components required to construct the new products and services being offered by financial institutions today. Underlying much of this content development is the notion of mass customization.<sup>41</sup> IFW provides techniques that complement other research into mass customization. These techniques draw heavily upon the intellectual foundation of IFW for their theory and the content models to provide the information resource.

Figure 9 Information covered by the Financial Services Object Model



Each solution is defined based on a subset from each of the data, function, and workflow columns. Thus, a personal loan supports a number of functions, requires certain data, and uses those data in workflows that fulfill the objectives of each function. But each solution is also a part of a "set" of similar solutions. The personal loan is part of a family of "loan" solutions that include the personal loan, as well as car loans, housing loans, student loans, agricultural loans, and investment loans. If this family of loans is structured into a hierarchy of loans, some of which are more generalized and some of which are more specialized, it is possible to support the rapid creation of new solutions or variations on solutions within that solution family. Many of the fundamental features of these individual loans are actually very similar, so by grouping the loans into a hierarchy with the more generalized types of loans at the top of the hierarchy and the more specialized loans in the lower parts of the hierarchy, it is possible to structure the solutions for maximum reuse. Grouping into families or classes of solutions also helps to increase reuse of the generic components.

The hierarchical structure of loans in a solution family represents a classification based on the concept of "structured product." At the next level the criteria for classifying the loan family in a particular way

are verified. In this case, the criteria for structuring products are based on two matrices between product and condition, and between product and activity. Conditions for a loan might include interest rate, loan amount, or the term of a loan.

From a business perspective, the product and condition matrix is used to structure products so that the definition of a new product can be as simple as changing the value of some conditions. For example, to create a student loan product, a variation on the already-defined personal loan, might be as simple as changing the product name to "student," restricting the maximum amount of the loan to \$3000, fixing the interest rate at 4 percent, and restricting the availability of the loan to individuals above the age of 16.

From an implementation perspective, the product and activity matrix is used to simplify the number of activities required to support a product, making it easier to generalize activities at a higher level in the loan family hierarchy and improving the way in which the bank can respond to changes in the market. For example, creating the student loan product might be as simple as inheriting a set of activities that were already defined for the personal loan and adding three new activities. Even the three new activities

can be quickly constructed using predefined components in the data, function, and workflow columns!

Over time all of the business solutions can be defined based on the generic industry-wide models of the data, function, and workflow columns. This has a number of significant benefits. Using the predefined industry models (FSDM, FSM, and FSWM) as a starting point when defining a new solution instead of starting from nothing provides a productivity boost. A typical productivity improvement of 40 percent has been measured in a number of IFW sites.<sup>42</sup> A second productivity boost in the area of requirements definition, analysis, and design comes from "template" solutions, classified into a small number of conceptually distinct types of solutions, and based on the generic models. Template solutions also move the focus and control of development to a further level of generalization. This movement can be seen as the focus of analysis shifts across the columns of IFW, with less analysis or design required in the data, function, and workflow columns because many of the elements are predefined, with more analysis and design taking place in the solution column. Furthermore, traditional methodologies often require mediation by technically trained analysts, whereas the definition of solutions from well-defined components structured within an integrating framework can be achieved directly by business users.

Although this type of approach to developing solutions is still in its infancy, a number of banks have already started to use it to control and manage the products and services that they offer. One bank simplified 650 disparate product lines into a more manageable set of 13 product families by using these techniques. A number of banks have developed applications to manage their product catalogs,<sup>43</sup> allowing business analysts to directly view and manipulate the conditions that apply to each product within a family group.

The values for conditions that differentiate the products within a family group can be stored in tables held externally from business logic for maximum flexibility. Eventually, the rules governing the different condition values might be stored in sophisticated knowledge bases that allow a bank to provide mass customization<sup>44</sup> of its products and services, individually tailored for each customer arrangement.

#### **Dimension 4: Structured transformation**

IFW was designed to address the paradigm shift underlying the information age or the knowledge econ-

omy. These terms are both used to suggest the realization that information is the key asset that organizations need to manage effectively in the future.<sup>45</sup> In any major change it is useful to have an overall framework that can guide the transformation process. This type of framework has been called a *transformation framework*.<sup>46</sup> To be effective, a transformation framework has to contain the perspective of the current or dominant paradigm, be self-consciously aware of all its dimensions and implications (location as historically specific), and anticipate future paradigms. IFW is one example of a transformation framework focused on the paradigm shift in information governance. This means that IFW must be sufficiently detailed and comprehensive to analyze both existing and anticipated paradigms, and flexible enough to adapt to any new understanding of the paradigm change. This aspect of IFW has been discussed in detail with reference to the structure of IFW and the dimensions of IFW.

Dimension 4 deals with aspects of time when using the IFW. It supports change or transformation on a small or large scale. This dimension typically addresses the phases of such a transformation, broadly covering from the current or "as is" situation to the future or "to be" situation. A number of models have been used to understand this dimension. One is the phases-of-transformation model<sup>47</sup> that defines three phases as optimization, enhancement, and redefinition. On the basis of this model, IFW can be used to manage change in any of these phases. If the focus of change was to introduce better information management within an organization, a typical approach would look first at optimizing the existing information management practice, then at enhancing that practice by adding new techniques or providing better communication and coordination between information users, and finally by redefining the use of information, perhaps by providing radically new types of financial services or products that took advantage of the information asset held by the financial institution.

Another model that can be adapted for use in this dimension is the software capability model,<sup>48</sup> originally developed as a standard to assess the maturity of the application development process. The five levels of this model—initial, repeatable, defined, managed, and optimized—are easily applied to the maturity of the information governance process. As such, it provides a useful checklist to determine the current level of maturity within an organization or to identify how information management can be improved within an organization.

**Table 5 Ownership levels**

| Ownership Level | Nature of Information                                                                                                                                                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Global          | Commonly found in all or most industry groups                                                                                                                                                                                                                                                                 |
| Industry        | Information that is typical of a particular industry community<br>Defined and benchmarked through joint development projects involving a number of industry representatives                                                                                                                                   |
| Enterprise      | Information that is unique to a particular company<br>Defined internally by members of an individual company                                                                                                                                                                                                  |
| Local           | Often a subset of the enterprise information<br>Defined within a project, organization unit, or business application<br>May consist of variations and versions of the enterprise definitions<br>May include additional definitions that are not recognized, required, or standardized at the enterprise level |
| Individual      | Often a personalized view of information at higher levels                                                                                                                                                                                                                                                     |

A critical aspect of this dimension is the need to manage transformation across the whole framework. This need has a parallel in systems thinking, which emphasizes the whole pattern of change rather than focusing on isolated snapshots of the whole. Peter Senge has said that "Systems thinking is a conceptual framework, a body of knowledge and tools that has been developed over the past fifty years, to make the full patterns clearer, and to help us see how to change them effectively."<sup>49</sup> Without this comprehensive framework, change tends to be managed project by project, with no overriding strategic direction. IFW can be used to ensure that each step in the transformation process adds value to the information asset.

### Dimension 5: Ownership

During the 1980s many financial institutions recognized the value of developing enterprise-wide models, but introducing such a model in a company brought a number of challenges. One was the challenge of creating the enterprise-wide models in the first place. Enterprise models are usually synthesized from a number of project- or organization-unit-specific

sources. Apart from the methodological difficulties in integrating diverse source material, this synthesis can also raise questions about ownership of intellectual assets. Once the enterprise model exists there have to be universally accepted procedures for accessing, using, and updating the information, as well as resolving disputes, allocating funds, and managing redundant development efforts in projects that take advantage of the information asset.

The issues of ownership that arise between the enterprise-wide level and the organization unit or project level are similar to the issues that arise between the ownership of an industry-wide model by IBM and the use of that model under license by individual enterprises.<sup>50</sup> Paul Strassmann, writing about "The Politics of Information Management,"<sup>51</sup> has distinguished between information management and information technology management. Information management is concerned with governance, whereas information technology management provides technical solutions for specific business needs. One of the key aspects of information governance is that long-term interests need to be funded and controlled at a higher level than short-term needs. Within an individual company it would mean that the decision to develop an enterprise-wide model should be funded and managed at the enterprise level. The industry-wide models that populate IFW are owned and managed by IBM.

In the ownership dimension of IFW six levels have been identified. These are the *global* level, which includes materials that are common across all of the lower levels; the *industry* level, which covers the interests of a group such as financial service, manufacturing, or retail institutions; the *enterprise* level, which covers assets that are common across all of the organization units of a particular company; the *local* level, which includes the specific needs of a project, organization unit, or business application; and the *individual* level, which covers assets owned by a person within the organization.<sup>52</sup>

An interesting characteristic that has been verified between the industry and enterprise levels is that as much as 80 percent of the information of an enterprise can be defined at the industry level. This characteristic has been borne out by projects in which financial services models have been customized and adapted for specific enterprises. Table 5 summarizes the key characteristics at each of the levels of ownership.

## Dimension 6: Routemaps

Many organizations use more than one methodology and, therefore, potentially have a conflict between two or more method-driven approaches. Today the information systems department of many banks uses both information engineering and object-oriented methodologies for application development. Other departments of the bank may be using business analysis methodologies such as critical success factor analysis or Porter's value chain analysis. Typically, each institution selects a combination of methodological approaches that matches its needs. It has been suggested that a "super" methodology, giving support to a complete cycle of organizational, business, and technical analysis, could be created using a "methodology chain," which can "coordinate and connect some suitable methodologies into a holistic approach."<sup>53</sup>

Although the IFW project has focused much of its development effort on enhancing and adding to contemporary systems development methodologies, the structure of IFW is designed to coordinate methodologies from many different approaches into coherent, well-defined methodology chains. In IFW these methodology chains are called *routemaps*. For example, IFW has a routemap for business process reengineering which, following the directive that a business process is more than just the tasks,<sup>54</sup> includes references to business process reengineering techniques, workflow and data modeling techniques, and organization impact analysis. The IFW routemaps also suggest the most effective techniques for using the predefined data, function, and workflow models that populate IFW.

Some of the IFW routemaps focus on what Anders Nilsson calls "horizontal integration" creating an alliance between two methodologies from the same group, for example, combining techniques from information engineering and object-oriented technology. Other routemaps provide "vertical integration" between methodologies from distinct groups, for example, in using business modeling and workflow modeling techniques to provide the business process reengineering routemap. The objective of the IFW routemaps is not to repeat tasks that are well-defined in one of the separate methodologies, but to make the "chain" explicit and add value by defining new tasks that are unique to IFW. Very few attempts have been made to show how techniques from different methodologies can be chained together. The IFW routemaps are designed to achieve this chaining by

using the IFW cells to show the "fit" between outputs from a variety of methodologies like information engineering and object-oriented technology. In addition, new techniques and tasks provided by IFW in its routemaps are intended to provide quality or time-related benefits. For example, with use of the FSDM it is possible to reduce the time taken to define data requirements from months to days.

Each routemap represents a typical project undertaken by a financial institution. The following list covers both routemaps that have already been written for IFW and methodology chains that could be created in IFW if required in client projects:

- Strategic alignment between organizational, business, and technical areas, strategic planning and implementation, information strategic planning (ISP), project planning, and information resource management (IRM)
- Customization of industry models, migration of customers' current models to IFW standards, development of a business architecture including establishment of a corporate data model, data mining, and development of a business data warehouse<sup>55</sup>
- Rapid requirements definition and logical design, application development (using a variety of approaches, such as information engineering, object-oriented technology, prototyping, or event-driven methodologies), database design, redevelopment of databases or applications, and legacy system reengineering
- Business process analysis and reengineering (both analyzing and implementing), solution restructuring, for example, of products and services or reports and management information, market segmentation, and customer analysis
- Cultural and organizational understanding, organizational change, change and transformation management, definition of an information management strategy, methodology, or tool comparison, and review

## Further work

A number of opportunities exist for further work and research using IFW. The main areas for further work seem to be in assessing how IFW could be used in other industries, developing content material and understanding in the remaining cells of IFW (especially in the organization and technical views), and developing materials in the solution column based on the other columns in the business view.

It is highly likely that the principles and ideas that have been developed for the finance services industry will apply in other industries. In fact, IFW has been designed in such a way that it could be populated by models from a number of alternative industries. The industry models developed by IBM for the insurance industry and the retail industry,<sup>56</sup> for example, could be positioned within the structure of IFW. Further materials are being developed within the banking, finance, and securities business solution units for IFW. IFW has been the strategic platform for future development efforts worldwide in this industry within IBM since 1994.

Probably the most exciting opportunity for the future is the notion that solutions can be grouped into solution families and these families can be defined based on the generic models of the problem domain. This notion provides an effective means to provide the mass customization and dynamic stability mentioned earlier.

### Concluding remarks

The development of IFW has not been without some difficulties. Initially there was much discussion, debate, and persuasion within IBM and participating financial institutions to ensure that the overall structure was indeed the correct one. Acceptance of IFW has involved both evangelizing and politics. IFW attempts to create and respond to a paradigm shift, and new paradigms are notoriously difficult to understand from the perspective of the old paradigm!<sup>57</sup> IFW has increasingly been accepted as a means of achieving this paradigm shift. It now addresses many of the ingrained problems that face information systems management, such as maintenance backlogs, degenerating legacy applications, inaccurate data, and lack of necessary management information.<sup>58</sup>

A major area of contention in the early stages of development was over terminology. Some of the terms that caused the greatest confusion were function, process, workflow, rule, application architecture, business architecture, and system architecture. It is possible to define or clarify these terms by using IFW. Another question was, how many cells are necessary to document information needs? The Zachman framework defined 30 cells. Some approaches (such as that used by the Retail Application Architecture<sup>16</sup>) suggest that even 30 cells is too large a number and that a simplified Zachman approach should be used. What seems important is not the number of cells required on a project-by-project basis, but that there

are sufficient cells to cover all aspects of information management.

A final issue is whether there is any computer-based tool support for IFW. The requirements for tool support need to be well understood before such a tool can be constructed. Much of the development of IFW has been concerned with understanding the nature of information and the type of framework required to manage it. In this respect IFW has been invaluable even without software support—the first step to managing any complex subject area is to adequately understand the subject matter and to define simple techniques that can be implemented. Once these are understood it is possible to progress to automated supports. Most of the development work in IFW has used existing computer-aided systems engineering (CASE) tools such as KnowledgeWare's Application Development Workbench\*\* (ADW) or Seer Technology's High Productivity System\*\* (HPS) toolset. More recently a tool called *m1* from modelware has been developed specifically to support the classification hierarchies of IFW.

IFW has had a considerable impact on the thinking of information systems departments in many major banks. The work has: created a framework that addresses contemporary needs for the complete information picture; shown how to align business and information systems analysis and modeling techniques in support of application development and business process reengineering; proved the benefits of having industry-wide models that predefine 80 percent of the components used in analysis, design, and development; and defined a set of routemaps that combine the best elements from diverse methodologies into project-based methodology chains.

Any framework for fully understanding information systems needs to be broader than the Zachman framework. The success of Zachman's information systems architecture confirms that a framework is a useful construct for understanding complex systems. IFW has shown that its three views, providing a broad categorization of the types of information available and representing three different approaches to the analysis and understanding of this information, are a valuable addition to the columns of the Zachman framework.

There is much further useful research to be done in this area. In particular, there should be more work in providing methodology chains; these will undoubtedly bring greater advantages to an institution than

stand-alone methodologies, since they can be adapted to the unique situation of that institution. It has already been suggested<sup>53</sup> that this will have an impact on the role of CASE tools in the future, as most existing tools are designed to support a specific methodology for information systems development rather than supporting the integration of diverse methodologies. The dimensions of IFW provide a stimulus to understand information governance from several alternative perspectives. Comparing one dimension against another is a powerful analysis technique.

Considerable savings have been made by defining industry-wide models that fit within the business, organization, or technical views of IFW. Further research should review how appropriate these ideas are to other industries.

It has been said that "discussion of frameworks is complicated by the fact that no really good documentation techniques have been found for them, something that also makes it difficult to build commerce in frameworks and to learn a new framework."<sup>13</sup> IFW shows how a framework has been developed for the financial services industry and shows how the ideas of John Zachman have been helpful in documenting this structure. In practice, IFW and

its underlying principles have proved to be straightforward and easy to learn.

### Acknowledgments

The conclusions and experience described in this paper are based on the collective efforts and insights of many individuals, and of organizations within the finance services industry and elsewhere. The IFW project, although initiated by IBM's Banking Solution Centre in Dublin, has been dependent on the involvement of more than 50 financial institutions around the world. I would like to thank everyone involved in this project.

In particular, I would like to thank John Dermody, Colman O'Sullivan, and Patrick Dillon for their considerable support, Valerie Ivers who kept the show on the road, and all of my many colleagues in the Banking Solution Centre. For reviewing the text and providing valuable feedback, I am grateful to Kieran Timmons, Patricia Burke, Paul Kilcullen, Roger James, Tommy Gannon, and Leanne Hassenbach. Thank you to all of my colleagues for lively and rewarding discussions over the years. Peter Horbatiuk, Robert Peake, and Dan Tasker deserve a special mention for many formative discussions.

## Appendix A: ISA represented within IFW

Structure is based on Sowa and Zachman, *IBM Systems Journal*, Vol. 31, No. 3, 1992.

**Table 6 Organization view**

|                       | Strategy                                                                                    | Structure                                                                                                                                                                       | Skills                   |
|-----------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Domain concepts       | (Not explicitly defined)                                                                    | (Not explicitly defined)                                                                                                                                                        | (Not explicitly defined) |
| Domain classification | List of business goals/strategy<br>Ends/means = major business goal/critical success factor | List of locations in which the business operates<br>Node = major business location<br>List of organizations/agents important to the business<br>Agent = major organization unit | (Not explicitly defined) |
| Generic template      | E.g., business plan<br>Ends = business objective<br>Means = business strategy               | E.g., organization chart<br>Agent = organization unit<br>Work = work product                                                                                                    | (Not explicitly defined) |
| Design context        | E.g., knowledge architecture<br>Ends = criterion<br>Means = option                          | E.g., human interface architecture<br>Agent = role<br>Work = deliverable                                                                                                        | (Not explicitly defined) |
| Operationally bound   | (Not explicitly defined)                                                                    | (Not explicitly defined)                                                                                                                                                        | (Not explicitly defined) |

**Table 7 Business view**

|                       | Data                                                                                                                                                     | Function                                                                             | Workflow                                                                                                                                                                                                                                                                                                                                                                        | Solution                 |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Domain concepts       | (Not explicitly defined)                                                                                                                                 | (Not explicitly defined)                                                             | (Not explicitly defined)                                                                                                                                                                                                                                                                                                                                                        | (Not explicitly defined) |
| Domain classification | List of things important to the business<br>Entity = class of business thing                                                                             | List of processes the business performs<br>Function = class of business process      | List of events significant to the business<br>Time = major business event                                                                                                                                                                                                                                                                                                       | (Not explicitly defined) |
| Generic template      | E.g., "entity-relationship diagram"<br>Entity = business entity<br>Relationship = business constraint                                                    | (Not explicitly defined)                                                             | E.g., master schedule<br>Time = Business event<br>Cycle = Business cycle                                                                                                                                                                                                                                                                                                        | (Not explicitly defined) |
| Design context        | E.g., "data model"<br>Entity = data entity<br>Relationship = data relationship                                                                           | E.g., "data flow diagram"<br>Function = application function<br>Argument = user view | E.g., "process flow diagram"<br>Function = business process<br>Argument = business resources<br>E.g., human/technology interface<br>Agent = user<br>Work = job<br>E.g., security architecture<br>Agent = identity<br>Work = transaction<br>E.g., control structure<br>Time = execute<br>Cycle = component cycle<br>E.g., knowledge design<br>Ends = condition<br>Means = action | (Not explicitly defined) |
| Operationally bound   | E.g., data design<br>Entity = segment/row<br>Relationship = pointer/key<br>E.g., data definition description<br>Entity = field<br>Relationship = address | (Not explicitly defined)                                                             | E.g., "program"<br>Function = language statement<br>Argument = control block<br>E.g., processing structure<br>Time = system event<br>Cycle = processing cycle<br>E.g., knowledge definition<br>Ends = subcondition<br>Means = step                                                                                                                                              | (Not explicitly defined) |

**Table 8 Technical view**

|                       | Interface                                        | Network                                                                                                                                                                                   | Platform                                                                                   |
|-----------------------|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Domain concepts       | (Not explicitly defined)                         | (Not explicitly defined)                                                                                                                                                                  | (Not explicitly defined)                                                                   |
| Domain classification | List of locations in which the business operates | (Not explicitly defined)                                                                                                                                                                  | (Not explicitly defined)                                                                   |
| Generic template      | (Not explicitly defined)                         | E.g., network architecture<br>Node = address<br>Link = protocol<br>E.g., distributed system architecture<br>Node = I/S function (processor, storage, etc.)<br>Link = line characteristics | E.g., system architecture<br>Node = hardware/system software<br>Link = line specifications |

**Table 8 Technical view, continued**

|                     | Interface                                                                                  | Network                                                                        | Platform                                                             |
|---------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Design context      | E.g., "structure chart"<br>Function = computer function<br>Argument = screen/device format | E.g., logistics network<br>Node = business location<br>Link = business linkage | (Not explicitly defined)                                             |
| Operationally bound | (Not explicitly defined)                                                                   | (Not explicitly defined)                                                       | E.g., timing definition<br>Time = interrupt<br>Cycle = machine cycle |

## Appendix B: IFW at a component level

**Table 9 Organization view**

|                       | Strategy                                                                                                                                                                                                                                                                                                                                     | Structure                                                                                                                                                                                                                                                                                                                                                               | Skills                                                                                                                                                                                                                                                                                                                                       |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Domain concepts       | Concepts important for understanding organization strategies<br>E.g., target, constraint, key indicators, critical success factor                                                                                                                                                                                                            | Concepts important for understanding organization structures<br>E.g., socio-infrastructure, role, and culture                                                                                                                                                                                                                                                           | Concepts important for understanding organization skills<br>E.g., experience, ability, characteristics, training                                                                                                                                                                                                                             |
| Domain classification | Classification of strategy knowledge based on domain concepts<br>E.g., classification of constraint types, key indicator types, critical success factor types<br>"Strategy" information, listed within the classification hierarchy, e.g., major business goals and critical success factors, constraints (weaknesses, threats, regulations) | Classification of structure knowledge based on domain concepts<br>E.g., classification of location types, role types, administrative infrastructure types<br>"Structure" information, listed within the classification hierarchy, e.g., major business locations in which the business operates; organizations, organization units and agents important to the business | Classification of skills knowledge based on domain concepts<br>E.g., classification of types of experience, training, ability<br>"Skills" information, listed within the classification hierarchy, e.g., techniques or methods used by the business, training qualifications recognized within the organization, abilities required by staff |
| Generic template      | Cross-enterprise, generic model relating organization strategy components (classified in the domain classifications)<br>E.g., business plan, five-year plan, mission statements, business strategy model, information strategic plan                                                                                                         | Cross-enterprise, generic model relating organization structure components (classified in the domain classifications)<br>E.g., organization structure chart, report distribution structure                                                                                                                                                                              | Cross-enterprise, generic model relating organization skills components (classified in the domain classifications)<br>E.g., human resource development plan, staff development plan, information systems development methodology                                                                                                             |
| Design context        | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., objective/measurement method matrix, objective/schedule matrix, project plan                                                                                                                                                              | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., organization unit structure chart, role/deliverable matrix, role/organization unit matrix                                                                                                                                                                            | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., skill set/project matrix, skill set/role matrix, methodology manual                                                                                                                                                                       |
| Operationally bound   | Implementation of the organization strategy concepts<br>E.g., actual project schedule, product rollout, customer satisfaction                                                                                                                                                                                                                | Implementation of the organization structure concepts<br>E.g., ABC Bank, Loans Department, General Manager, Lois Lane                                                                                                                                                                                                                                                   | Implementation of the organization skills concepts<br>E.g., core competencies, bank teller experience, published methodology                                                                                                                                                                                                                 |

**Table 10 Business view**

|                       | <b>Data</b>                                                                                                                                                                                                                                                                                         | <b>Function</b>                                                                                                                                                                                                                                                                            | <b>Workflow</b>                                                                                                                                                                                                                                                                                                                        | <b>Solution</b>                                                                                                                                                                                                                                                                                                                    |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Domain concepts       | Concepts important for understanding business data<br>E.g., involved party, arrangement, condition, classification, product                                                                                                                                                                         | Concepts important for understanding business functions<br>E.g., direction management, business operations, market management, and resource management                                                                                                                                     | Concepts important for understanding business workflows<br>E.g., activity verbs, activity, trigger, workflow, critical business process                                                                                                                                                                                                | Concepts important for understanding business solutions<br>E.g., product structure, report structure, management information                                                                                                                                                                                                       |
| Domain classification | Classification of data knowledge based on domain concepts<br>E.g., classification of data concept schemes and values, data concept descriptors, and data concept relationships<br>"Data" information, listed within the classification hierarchy, e.g., classes of things important to the business | Classification of function knowledge based on domain concepts<br>E.g., classification of function types, types of state, data access types<br>"Function" information, listed within the classification hierarchy, e.g., functions managed within the business, classes of business process | Classification of workflow knowledge based on domain concepts<br>E.g., classification of verb types, trigger types, activity types<br>"Workflow" information, listed within the classification hierarchy, e.g., events significant to the business, triggers that initiate business activities, verbs used to name business activities | Classification of solution knowledge based on domain concepts<br>E.g., classification of product structure types, report types, management information system types<br>"Solution" information, listed within the classification hierarchy, e.g., business conditions that differentiate products, parameters used to drive reports |
| Generic template      | Cross-enterprise, generic model relating organization data components (classified in the domain classifications)<br>E.g., entity-relationship diagram, object data model, business entity model, subtype-supertype hierarchy, object class diagram                                                  | Cross-enterprise, generic model relating organization function components (classified in the domain classifications)<br>E.g., function context diagram, data access diagrams, state transition table                                                                                       | Cross-enterprise, generic model relating organization workflow components (classified in the domain classifications)<br>E.g., critical business processes, business procedures manual, process context diagram, scenario diagrams                                                                                                      | Cross-enterprise, generic model relating organization solution components (classified in the domain classifications)<br>E.g., product/condition dependency matrix, product/activity support matrix                                                                                                                                 |
| Design context        | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., application data model, database logical design                                                                                                                                                  | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., data access diagram, CRUD matrix, functional dependency diagram, data flow diagram                                                                                                      | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., workflow model, process flow diagram, control flow diagram, message flow diagram                                                                                                                                                    | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., product/condition value matrix, activity/parameter matrix                                                                                                                                                                       |
| Operationally bound   | Implementation of the business data concepts<br>E.g., customer information database, account transaction data definition                                                                                                                                                                            | Implementation of the business function concepts<br>E.g., customer enquiry module, arrangement data access module, customer relationship system                                                                                                                                            | Implementation of the business workflow concepts<br>E.g., program logic to print statement, evaluate customer workflow                                                                                                                                                                                                                 | Implementation of the business solution concepts<br>E.g., condition value table for personal loan, parameters to drive management information search                                                                                                                                                                               |

**Table 11 Technical view**

|                       | Interface                                                                                                                                                                                                                                                                                                                                                                                                             | Network                                                                                                                                                                                                                                                                                                                                                                                            | Platform                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Domain concepts       | Concepts important for understanding technical applications<br>E.g., logic, component structure, language, interface                                                                                                                                                                                                                                                                                                  | Concepts important for understanding technical networks<br>E.g., device, topology, communication medium, protocol                                                                                                                                                                                                                                                                                  | Concepts important for understanding technical systems<br>E.g., storage, resource manager, operating system, platform                                                                                                                                                                                                                                                                          |
| Domain classification | Classification of application knowledge based on domain concepts<br>E.g., classification of types of logic, component types, language types, programming interface types<br>"Application" information, listed within the classification hierarchy, e.g., languages used to code programs within the institution, components used to construct or generate applications, interfaces used to connect program components | Classification of network knowledge based on domain concepts<br>E.g., classification of device types, protocol types, types of topology or network structure<br>"Network" information, listed within the classification hierarchy, e.g., protocols used in the institution's networks, devices connected to the network, communication medium used for distributed systems, communication software | Classification of system knowledge based on domain concepts<br>E.g., classification of types of storage, classifications of resource manager types, operating system types<br>"System" information, listed within the classification hierarchy, e.g., operating systems used by the institution, resource managers available to support applications and networks, database management systems |
| Generic template      | Cross-enterprise, generic model relating organization application components (classified in the domain classifications)<br>E.g., application structure architecture, application programming interfaces                                                                                                                                                                                                               | Cross-enterprise, generic model relating organization network components (classified in the domain classifications)<br>E.g., network architecture, distributed system architecture, communication interfaces                                                                                                                                                                                       | Cross-enterprise, generic model relating organization system components (classified in the domain classifications)<br>E.g., system architecture, hardware configuration map, resource manager interfaces, database management system interfaces                                                                                                                                                |
| Design context        | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., program structure chart, application program interface design                                                                                                                                                                                                                                                      | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., local area network diagram, distribution network diagram                                                                                                                                                                                                                                        | Detailed or project-specific logical model (based on and extending the generic templates)<br>E.g., resource manager interface design, database interface design                                                                                                                                                                                                                                |
| Operationally bound   | Implementation of the technical application concepts<br>E.g., application programming interfaces, dynamic linking, compiler or interpreter logic                                                                                                                                                                                                                                                                      | Implementation of the technical network concepts<br>E.g., communication protocols, connecting cable, node links, network programming interfaces                                                                                                                                                                                                                                                    | Implementation of the technical system concepts<br>E.g., system response codes, system programming interfaces, system software, hardware                                                                                                                                                                                                                                                       |

\*\*Trademark or registered trademark of KnowledgeWare, Inc., or Seer Technologies, Inc.

### Cited references and notes

1. J. A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal* 26, No. 3, 276-292 (1987), from the Abstract.
2. *Ibid.*, listed in Appendix A on p. 292.
3. J. F. Sowa and J. A. Zachman, "Extending and Formalizing

the Framework for Information Systems Architecture," *IBM Systems Journal* 31, No. 3, 590-616 (1992). In this 1992 paper, the information systems architecture (ISA) framework is described as a "taxonomy with 30 boxes or cells organized into six columns (labeled A through E) and five rows (numbered 1 through 5)," page 591. Some references to the ISA framework add functioning system as a sixth row.

4. Zachman, "Framework," Table 2, page 282, where the product from an information systems architecture is the "information system."

5. R. L. Nolan and D. W. Mulryan, "Undertaking an Architecture Program," *Stage by Stage* 7, No. 2 (March–April 1987), where the authors state, "There is a parallel between architecture design and city planning. City planners must design in the face of many unknowns, such as future transportation technologies, changing work, living, and commuting patterns, and so on. To deal with the complexities and unknowns, city planners set guidelines on, for instance, building height, setbacks, and zoning. They can ill afford to delve into such detail as prescribing building materials. As a result of this level of planning, our major cities are able to accommodate new technologies for transportation and communication which remain viable for hundreds of years, and which make a major contribution to each city's brand of urban culture."
6. J. Podolsky, in an opinion column for *Datamation*, says that what our information systems need is landscape architecture not building architecture, to provide "bottom-up ecological and evolutionary models of system growth." He concludes, "We may wish our systems portfolio with its hodgepodge legacy of applications and processes to be a planned community of buildings and roads, but it ain't so. And the kinds of plans we need will come not from the IT version of Frank Lloyd Wright, but from the inspiration of the parks of Frederick Law Olmsted" in *ThinkWrap*, "Parks, not Buildings," *Datamation* 40, No. 19, 90 (October 15, 1994).
7. F. Niederman, J. C. Brancheau, and J. C. Wetherbe, "Information Systems Management Issues for the 1990s," *MIS Quarterly* 15, No. 4, 475–500 (December 1991).
8. T. Moynihan, "What Chief Executives and Senior Managers Want From Their IT Departments," *MIS Quarterly* 14, No. 1, 15–26 (March 1990).
9. At the time of writing, IFW has been used in projects in the financial services industry, the airline industry, and the government sector.
10. Sowa and Zachman, "Extending and Formalizing the Framework," Abstract.
11. For example, K. Seer and M. Wise, "A Framework for Managing Model Objects," in *Database Programming and Design* 7, No. 8, 50–55 (August 1994). The authors developed a framework to classify objects. They state, "Just as the Zachman ISA framework supports a systems-development methodology, a repository framework supports a repository-management methodology."
12. Some of these include information engineering and object-oriented development methodologies, software engineering, domain analysis, business anthropology, artificial intelligence and knowledge-based systems, workflow analysis, business process reengineering, management theory and practice, change management, and organizational behavior.
13. A. A. R. Cockburn, "The Impact of Object-Orientation on Application Development," *IBM Systems Journal* 32, No. 3, 433 (1993).
14. The term *information framework* was first used on a consulting engagement by Westpac Banking Corporation's CS90 Consulting Group in 1991. CS90 was Westpac's "Core Systems for the 1990s" project. The idea of the framework started by taking experience at Westpac in enterprise modeling and reuse application architectures and generalizing the concepts so that they could be applied in any financial institution. The first version of the information framework was a combination of the Zachman framework, the Westpac CS90 project, and contemporary research. See R. Evernden, "Object Management in a Mainframe Repository-Based Environment," *Repository AD/Cycle International Users Group Conference Proceedings*, Chicago (October 27–30, 1991), pp. 188–197, for a detailed discussion of the Westpac project.
15. *Financial Application Architecture—Introduction*, GC31-3932-0, IBM Corporation (1992), and *Financial Application Architecture—Concepts of Application and System Architectures*, LY38-4402-0, IBM Corporation (1992); available through IBM branch offices.
16. P. Stecher, "Building Business and Application Systems with the Retail Application Architecture," *IBM Systems Journal* 32, No. 2, 278–306 (1993).
17. C. Loosley, "Separation and Integration in the Zachman Framework," *Database Newsletter* 20, No. 1, 3–9, Database Research Group, Boston (1992).
18. J. Ward and A. Bytheway at a symposium on *Transforming the Business with Information*, Cranfield University School of Management, UK (July 1995).
19. For example, in one popular dictionary, the definition for data is "1. A series of observations, measurements, or facts; information. 2. Also called: **information**. Computers the information operated on by a computer program," *The New Collins Dictionary and Thesaurus*, Collins, London and Glasgow (1987), p. 247.
20. Two major types of classification schemes are found in library science: enumerative and faceted. For a good discussion of faceted classifications and reusability, see R. Prieto-Diaz and G. A. Jones, "Breathing New Life into Old Software," reprinted in *Tutorial: Software Reuse—Emerging Technology*, W. Tracz, The Computer Society of the IEEE, Los Alamitos, CA (1990), pp. 152–160.
21. M. J. Earl, "Experiences in Strategic Information Systems Planning," *MIS Quarterly*, No. 1, 1–24 (March 1993).
22. See D. Tasker, *The Problem Space* (1993), available only on diskette from Dan Tasker, 37/9 Hampden Avenue, Cremorne, NSW 2090, Australia, for an object-oriented discussion of these questions in the context of the Zachman framework. Tasker labels the five "Ws" as Objects, Events, Rules, Participants, and Locations.
23. Sowa and Zachman, "Extending and Formalizing the Framework," p. 599. Rule 1 states "The columns have no order. Order implies priorities. It creates a bias toward one aspect at the expense of others. Traditional programmers, for example, tend to have a bias toward function. They usually prefer to see the function column first in the framework."
24. B. J. Pine II, *Mass Customization*, Harvard Business School Press, Boston (1992).
25. U. Apte, C. S. Sankar, M. Thaku, and J. E. Turner, "Reusability-Based Strategy for Development of Information Systems: Implementation Experience of a Bank," *MIS Quarterly* 14, No. 4, 421–433 (December 1990). Also J. Karimi, "An Asset-Based Systems Development Approach to Software Reusability," *MIS Quarterly* 14, No. 2, 179–198 (June 1990).
26. C. Finkelstein, *An Introduction to Information Engineering (From Strategic Planning to Information Systems)*, Addison-Wesley Publishing Co., Reading, MA (1989).
27. A. G. Nilsson, "Information Systems Development: A Frame of Reference and Classifications," paper presented at *Polish-Scandinavian Seminar*, Paraszyno, Poland (June 27–30, 1988).
28. In the original Zachman article (Zachman, 1987), this column is labeled "Process Description," whereas in the later article (Sowa and Zachman, 1992) it is labeled "Function." It is interesting that our discussions in IFW had difficulty with choosing appropriate labels for this column. This is partly because, in the many different methodologies available to information systems analysis, the two words "function" and "process" are often interchangeable!

29. B. Curtis, M. I. Kellner, and J. Over, "Process Modelling," *Communications of the ACM* 35, No. 9, 75-90 (September 1992). This paper was part of a special issue on analysis and modeling in software development.
30. T. H. Davenport, *Process Innovation—Reengineering Work Through Information Technology*, Harvard Business School Press, Boston (1993), see pages 5 through 9 for a discussion of "What Is a Process?"
31. *Ibid.*, p. 6.
32. This includes architectures from IBM such as the Systems Application Architecture™, Systems Network Architecture, and Financial Application Architecture, as well as the work of many other proprietary or open architectures.
33. Some of the most important writings for this period include: E. F. Codd, "A Relational Model for Large Shared Data Banks," *Communications of the ACM* 13, No. 6, 377-387 (1970); P. Chen, "The Entity Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems* 1, No. 1, 9-36 (March 1976); C. Finkelstein, "Information Engineering," six In Depth articles in *Computerworld* (May-June 1981); J. Martin and C. Finkelstein, *Information Engineering*, Technical Report on Information Engineering, Savant Institute, United Kingdom (1981).
34. V. Grover and J. T. C. Teng, "How Effective Is Data Resource Management? Reassessing Strategic Objectives," *Journal of Information Systems Management* (Summer 1991).
35. For an interesting discussion of schemata of separation and identity, and decomposition and recomposition, see C. Castoriadis, *The Imaginary Institution of Society*, Polity Press (1987), in particular, pp. 176-183 and pp. 221-227.
36. G. Arango and R. Prieto-Diaz, "Part 1: Introduction and Overview, Domain Analysis Concepts and Research Directions," in *Domain Analysis and Software Systems Modeling*, G. Arango and R. Prieto-Diaz, Editors. The editors are referencing another article within the same compilation: G. Arango, "Domain Analysis—From Art Form to Engineering Discipline," *Proceedings of the 5th International Workshop on Software Specification and Design*, Computer Society Press, Los Alamitos, CA (1989), pp. 152-159.
37. *Domain Analysis and Software Systems Modeling*, G. Arango and R. Prieto-Diaz, Editors, p. 1.
38. R. Evernden, "New Financial Service Models of the 1990s," *Euromoney Financial Technology Review*, 6-10 (Spring 1994).
39. There are several examples of the need for well-defined data concepts. The June 1990 issue of *American Programmer*, in an article on enterprise modeling, listed ten basic entities (organization, individual/person, activity, document, requirement, item, property, money, location, journal/date). IBM's Insurance Application Architecture uses 14 entity families in its high-level data model—see the Insurance Application Architecture Data Model Reference (IBM Licensed Publication).
40. As of August 1995 there were 42 bank customers with a license for the FSDM, 15 with customer licenses for the FSFM, and 12 with customer licenses for the FSWM.
41. B. J. Pine II, *Mass Customization: The New Frontier in Business Competition*, Harvard School Press, Boston (1993).
42. The experiences of many IFW users have been recorded at the annual IFW conferences. At the most recent conference, held in Monte Carlo in May 1995, DataCentralen reported an estimated function point reduction by 40 percent when mapping a traditional data model for an enterprise tax system against the Financial Services Data Model. Bancomer reported the creation of a corporate data model in only six months by customizing IFW models. The resultant model contained 2382 data definitions transferred into Help for Windows™ for general access across the bank. Banque Nationale de Paris, Den Danske Bank, Banca Commerciale Italiana, and Nedcor are other banks that have documented similar benefits at these conferences.
43. For example, Standard Bank of South Africa, as reported at the IFW/FSDM Conference 1994, Rome (February 23-25, 1994) in a presentation by I. Shutte.
44. A. C. Boynton, B. Victor, and B. J. Pine II, "New Competitive Strategies: Challenges to Organizations and Information Technology," *IBM Systems Journal* 32, No. 1, 40-64 (1993), describes the notion of mass customization, bringing the notion of "custom" back into customer, with examples from the banking industry.
45. R. Evernden, "From Stumbling Blocks to Building Blocks—Preparing for the Knowledge Economy Using Information Framework," *IFW Conference 1995—From Business Transformation to Solution Implementation*, Monte Carlo (May 17-19, 1995), Binder Section 2. Describes this paradigm shift in detail, outlining the anticipated changes that will impact organizations and their use of information.
46. Unpublished research by R. Evernden and P. Burke, "Transformation Frameworks in Paradigm Analysis." This research was discussed in Reference 45.
47. W. H. Davidson, "Beyond Re-Engineering: The Three Phases of Business Transformation," *IBM Systems Journal* 32, No. 1, 65-79 (1993).
48. M. Paulk et al., *Capability Maturity Model for Software, Version 1.1*, Technical Report CMU/SEI-93-TR-24, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA (February 1993).
49. P. M. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*, Century Business, Random House, London (1992), p. 7; also Doubleday Currency, New York (1990). For further useful discussion, see G. H. Watson, *Business Systems Engineering: Managing Breakthrough Changes for Productivity and Profit*, John Wiley & Sons, New York (1994).
50. Enterprises can purchase a license for the IFW models. This allows the enterprise to use material while the intellectual ownership of the industry model is retained by IBM. At the time of writing, there were nearly 50 licenses for IFW materials.
51. P. A. Strassmann, *The Politics of Information Management*, The Information Economics Press, New Canaan, CT (1995).
52. These levels can be compared to the seven layers of a federated organization described in Reference 51.
53. A. G. Nilsson, "Business Modelling as a Base for Information Systems Development," *The Third International Conference on Information Systems Developers Workbench—Methodologies, Techniques, Tools, and Procedures*, Gdansk, Poland (September 22-24, 1992); also appears as Report V-4211, Institute V, Department of Information Management, Stockholm School of Economics, Box 6501, S-113, 83 Stockholm, Sweden.
54. M. Hammer and J. Champy, *Reengineering the Corporation—A Manifesto for Business Revolution*, Harper Business, New York (1993).
55. B. A. Devlin and P. T. Murphy, "An Architecture for a Business and Information System," *IBM Systems Journal* 27, No. 1, 60-80 (1988).
56. The Insurance Application Architecture contains a data model, function model, and function flow model. The Retail Application Architecture contains a business entity model/data model, functional model, and organization model.

57. R. Evernden, "Spaghetti Code to Ravioli Code," paper presented at Università Cattolica del Sacro Cuore, Milan (September 23, 1994), International Conference *I modelli in banca: organizzazione, processi, evoluzione dei sistemi informativi*, which provides a detailed discussion of the paradigm shift necessary to address many of the latent problems in the current application maintenance process. Five radical changes are described, which the author suggests are necessary to shift into the new paradigm.
58. The 1994 annual IFW/FSDM Conference, held in Rome (February 23–25, 1994), was attended by 60 financial institutions. Among others, presentations were made by Schoeller Daten Service, Credit Suisse, Royal Bank of Canada, Banque Nationale de Paris, Norwest Technical Services, Norwest Mortgage, Standard Bank of South Africa, Bausparkasse Schwaebisch Hall, Den Danske Bank, and Banca Commerciale Italiana.

## General references

- J. Banbury, "Towards a Framework for Systems Analysis Practice," *Critical Issues in Information Systems Research*, R. J. Bolland, Jr. and R. A. Hirschheim, Editors, John Wiley & Sons Ltd., London (1987).
- M. A. Cusumano, *Japan's Software Factories—A Challenge to U.S. Management*, Oxford University Press, Oxford (1991).
- Domain Analysis and Software Systems Modeling*, R. Prieto-Diaz and G. Arango, Editors, IEEE Computer Society Press, Los Alamitos, CA (1991).
- Intellectual Foundations for Information Professionals*, H. K. Achleitner, Editor, Social Science Monographs, Boulder, Distributed by Columbia University Press, New York (1987).
- S. M. McKinnon and W. J. Bruns, Jr., *The Information Mosaic*, Harvard Business School Press, Boston (1992).
- P. M. Senge, *The Fifth Discipline—The Art and Practice of the Learning Organisation*, Random House, London (1992).
- D. Tasker, *Fourth Generation Data—A Guide to Data Analysis for New and Old Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1989).
- W. Tracz, *Tutorial: Software Reuse—Emerging Technology*, The Computer Society of the IEEE, Los Alamitos, CA (1990).
- Tutorial: Software Reusability*, P. Freeman, Editor, The Computer Society of the IEEE, Los Alamitos, CA (1987).

*Accepted for publication August 25, 1995.*

**Roger Evernden** *WorkSpace International Ltd., 124 Copsewood Road, Southampton, SO18 1QR, United Kingdom (electronic mail: 100035.3226@compuserve.com)*. Mr. Evernden is a senior consultant on information strategic architectures, model-based application development, and reusability. He has worked closely with IBM on the development of its Financial Application Architecture, and since initiating the Information Framework project, has given more than 200 presentations and seminars to financial institutions around the world. He has been involved with enterprise-wide models since the early 1980s. At the leading edge in his profession, Mr. Evernden has conducted detailed research and development into reuse, model-based methodologies, object-oriented techniques, knowledge bases, business process reengineering, framework engineering, and paradigm shifts. His contributions to the discipline of information systems have appeared in numerous articles and conference papers.

Reprint Order No. G321-5594.