

# Estrategia de Negocio y Arquitectura Empresarial

## Visión y Experiencia de IBM

Alonso Verdugo Medina, M.D.  
Advisor Architect  
IBM SOA Design, CoBIT, ITIL Certified  
WebSphere Tiger Team  
alonsov@co.ibm.com

## Agenda

- Visión y Entendimiento en IBM
- Por que su importancia en nuestra Región
- Cual es el Reto

## Visión y Entendimiento de IBM



John Zachman

En 1984 John Zachman (IBM) escribió un artículo (de uso interno en IBM) de arquitectura de la empresa para los sistemas de información, en el que propuso un framework para la EA

Zachman fue invitado a publicar un artículo publico, que se produjo en 1987, en **IBM Systems Journal**.

En 1990 se retiró de IBM creó su propia firma de consultoría en arquitectura de la empresa;



En 1992, una vez publicada en IBM Systems Journal, y en asociación con John Sowa (también se retiró recientemente de IBM) una versión ampliada del documento original, que contiene una versión nueva y definitiva del marco.

### A framework for information systems architecture

by J. A. Zachman

*With increasing size and complexity of the implementations of information systems, it is necessary to use some logical construct (or architecture) for defining and controlling the interfaces and the integration of all of the components of the system. This paper defines information systems architecture by creating a descriptive framework from disciplines quite independent of information systems, then by analogy specifies information systems architecture based upon the neutral, objective framework. Also, some preliminary conclusions about the implications of the resultant descriptive framework are drawn. The discussion is limited to architecture and does not include a strategic planning methodology.*

The subject of information systems architecture is beginning to receive considerable attention. The increased scope of design and levels of complexity of information systems implementations are forcing the use of some logical construct (or architecture) for defining and controlling the interfaces and the integration of all of the components of the system. Thirty years ago this issue was not at all significant because the technology itself did not provide for either breadth in scope or depth in complexity in information systems. The inherent limitations of the then-available 4K machines, for example, constrained design and necessitated suboptimal approaches for automating a business.

Current technology is rapidly removing both conceptual and financial constraints. It is not hard to speculate about, if not realize, very large, very complex systems implementations, extending in scope and complexity to encompass an entire enterprise. One can readily delineate the merits of the large, complex,

enterprise-oriented approaches. Such systems allow flexibility in managing business changes and coherency in the management of business resources. However, there also is merit in the more traditional, smaller, suboptimal systems design approach. Such systems are relatively economical, quickly implemented, and easier to design and manage.

In either case, since the technology permits "distributing" large amounts of computing facilities in small packages to remote locations, some kind of structure (or architecture) is imperative because decentralization without structure is chaos. Therefore, to keep the business from disintegrating, the concept of information systems architecture is becoming less an option and more a necessity for establishing some order and control in the investment of information systems resources. The cost involved and the success of the business depending increasingly on its information systems require a disciplined approach to the management of those systems.

On the assumption that an understanding of information systems architecture is important to the development of a disciplined approach, the question that naturally arises is "What, in fact, is information

\* Copyright 1987 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 without fee without further permission by computer-based and other information-service systems. Permission to reproduce any other portion of this paper must be obtained from the Editor.

### Extending and formalizing the framework for information systems architecture

by J. F. Sowa  
J. A. Zachman

*John Zachman introduced a framework for information systems architecture (ISA) that has been widely adopted by systems analysts and database designers. It provides a taxonomy for relating the concepts that describe the real world to the concepts that describe an information system and its implementation. The ISA framework has a simple elegance that makes it easy to remember, yet it draws attention to fundamental distinctions that are often overlooked in systems design. This paper presents the framework and its recent extensions and shows how it can be formalized in the notation of conceptual graphs.*

The world contains entities, processes, locations, people, times, and purposes. Computer systems are filled with bits, bytes, numbers, and the programs that manipulate them. If the computer is to do anything useful, the concrete things in the world must be related to the abstract bits in the computer. Zachman's framework for information systems architecture (ISA) makes that link.<sup>1</sup> It provides a systematic taxonomy of concepts for relating things in the world to the representations in the computer. It is not a replacement for other programming tools, techniques, or methodologies. Instead, it provides a way of viewing a system from many different perspectives and showing how they are all related.<sup>2</sup>

Most programming tools and techniques focus on one aspect or a few related aspects of a system. The details of the aspect they select are shown in utmost clarity, but other details may be obscured or forgotten. As examples, consider each of the following techniques:

- Flowcharts, which were introduced by John von Neumann in 1945, are the oldest and still most widely used programming aid. They focus on the operations performed by a computer and their temporal sequence. They are fine for showing algorithms, but the data structures processed by the algorithms are only mentioned incidentally as they are being operated upon.
- Entity-relationship diagrams are a popular graphic notation for showing entity types, their attributes, and the relations that connect them. They are fine for showing certain kinds of constraints, but they cannot show all constraints, and they ignore the operations performed by and on the entities.
- Relational databases emphasize tables and the operations for manipulating them to derive and

\* Copyright 1992 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 without fee without further permission by computer-based and other information-service systems. Permission to reproduce any other portion of this paper must be obtained from the Editor.

## Puntos Tener en Cuenta

- Facilita la agilidad para el Negocio
  - Arquitectura es el puente entre la estrategia de negocio y T.I
  - Define un modelo de gobierno ante los cambios de requerimientos
  - Permite definir trazabilidad, análisis de impacto , simular
  - Establece un modelo de maduración
- Prescriptiva
  - Mejores practicas
  - Como incorporar nuevas tecnologías

Captura la “esencia” de su compañía

Objetivos de Negocio => Procesos => Implementación

## Arquitectura e Ingeniería

- Fuente de insatisfacción
- 33% del presupuesto se ejecuta durante la fase de pre-desarrollo
- 60% inicia con la documentación



## La región y el Reto

- Realidad (que nos gusta y que no)
- Madurez
- No esta en nuestra cultura el medir => como hacer ciencia ?
  
- Arquitectura obliga a
  - Documentar
  - Medir → COMPETITIVIDAD
  - Optimizar
  
- Genera
  - Mejores Practicas
  - Modelos replicables
  - FRAMEWORKS → CONOCIMIENTO
  
- Ejemplo Unv. De Los Andes