

## **Exclusive Interview with John Zachman, President of Zachman International, CEO of Zachman Framework Associates**

by Roger Sessions, Editor-in-Chief,  
*Perspectives of the International Association of  
Software Architects.*

Editor's Note: John Zachman is the author of The Zachman Framework™ and, without doubt, the single most influential person in the field of Enterprise Architectures. Recently, he agreed to participate in an interview with Perspectives. We are honored to include this interview in our special issue dedicated to a twenty-year retrospective on the field that John Zachman began.

**ROGER SESSIONS:** John, Welcome to *Perspectives of the International Association of Software Architects*. As President of Zachman International, CEO of Zachman Framework Associates and a highly regarded speaker and seminar leader, your free time is very limited, and I appreciate you sharing it with us.

In this issue of *Perspectives*, we are celebrating the 20th anniversary of Enterprise Architectures. I believe this field effectively began in 1987 with the publication of your historic paper, "A Framework for Information Systems Architecture", in the *IBM Systems Journal*. Let me take you back two decades. What was the main message that you were delivering in that now famous paper?

**JOHN ZACHMAN:** Roger ... first of all, thank you for your interest in my work. Regarding history, I have to think a moment as it actually originated quite a few years ago. I wrote the original article in 1982. It was published internally in IBM in 1984 and then written for the *IBM Systems Journal* in 1987<sup>1</sup>.

There were two basic points in the original article.

First, Architecture is an ENTERPRISE issue, not a systems issue. Today, I would say, that the end object is to engineer and manufacture the Enterprise, not simply to build and run systems.

I wrote another article around 10 years ago "The Framework for Enterprise Architecture: Background, Description and Utility<sup>2</sup>" in which I explained that in the early '80's, the world was simply not able to grasp the concept of *Enterprise Architecture*. This is why I referred,

### Interview with John Zachman

(continued from page 2)

in the original article, to The Framework for *Information Systems Architecture* which was more understandable at that time.

Today, I wish I had not done that because there is a lot of misunderstanding, particularly in the Information Systems community, that Architecture, even *Enterprise Architecture* only relates to Information Systems. Clearly, that is a gross misunderstanding.

The second major point was that Enterprise Architecture is not some arcane, arbitrary concept. It is specifically definable based on the Architecture precedent of the much older disciplines of Architecture/Construction and Engineering/Manufacturing.

I recently wrote another article, “Architecture Is Architecture Is Architecture<sup>3</sup>.” The underlying structure of descriptive representations for anything and *everything* is the same ... everything has bills of material, everything has functional specs, everything has drawings, etc., etc., etc.

The descriptive representations of Enterprises have to have the same underlying conceptual structure as everything else, bills of material, functional specifications, drawings (i.e. “geometry”, or maybe “geography” in the case of Enterprises). Otherwise, I would argue that, whatever models, descriptive representations, you are producing are not architectural models... They are something else. They may be models, but they are not being used to do Architecture work.

**ROGER SESSIONS:** What were some of the events that led you to write your original paper?

**JOHN ZACHMAN:** By default, when P. Dewey Walker left IBM for a General Management position in the health industry, I ended up being the kind-of nominal heir to the methodology that he created and had employed internally in IBM. Dewey called the methodology “*Business Systems Planning*” (BSP) because he knew that it was NOT “*Information Systems Planning*”. It was a strategy-oriented methodology that produced and employed what I would now call the Framework, Row 1 models. That planning methodology was to be followed by an Architecture methodology because those of us who were practicing it knew that “architecture” was the structural bridge that connected the strategy with the implementation.

However, we didn’t know what architecture was at that point in time. I was responsible for Planning Programs in IBM’s Western Region where we were supporting the Airframe Manufacturing Industry with BSP and other planning programs, and it was then that I got the idea that



John Zachman

we could learn a lot about Enterprise Architecture if we could learn something about architecture for airplanes, or buildings, or locomotives, or super computers, that is, for complex engineering products.

I began to see that Architecture IS Architecture IS Architecture. I simply put Enterprise names on the engineering design artifacts that were the descriptive representations for anything. Unfortunately, some of the meta-entity names I chose for the Enterprise Architecture models had very strong Information Systems connotations since I came from the Information Systems industry and had an Information Systems vocabulary. Subsequently, I have lived to regret some of my meta-model name choices as I am sure they contributed somewhat to the misunderstanding that Enterprise Architecture is an Information Systems issue.

We have been working for the last 5 years with some linguists<sup>4</sup> who have been very helpful in order to find more business-oriented words that much more accurately convey the concepts of the Enterprise Architecture meta-model. We published that standard “Zachman Framework” meta-model in 2005, and hopefully, that will serve to clarify the orientation of Enterprise Architecture as an *Enterprise* issue.

**ROGER SESSIONS:** What hopes did you initially have for your paper?

**JOHN ZACHMAN:** In 1982, I was at a meeting in Tampa, Florida where a large company was conducting a series of strategy sessions for their management folks and

### Interview with John Zachman

(continued from page 3)

my Framework graphic showed up in one of their copyrighted documents. That's what precipitated the original article. I became aware that I had to get the Framework itself copyrighted, just to protect my ability to work on my own work. Had it not been for that meeting, I probably would never have made the effort to write the original article.

Subsequently, there were some other people who were encouraging me to rewrite the article for publication in the *IBM Systems Journal*. In 1981, as a result of a conference on Enterprise Modeling in Nashville, I had written an *IBM Systems Journal* article comparing Business Systems Planning (BSP) and Business Information Control System (BICS), a methodology authored by David Kerner of IBM's Santa Theresa Laboratory<sup>5</sup>. I think it may have been that earlier article that led the Systems Journal folks to encourage me to write the 1987 Framework article. So, my hope for the article was merely to get the concepts published in a copyrighted journal. I had no further intention or expectation at that point in time.

**ROGER SESSIONS:** How has the publication of this paper changed your life?

**JOHN ZACHMAN:** Roger ... that is a hard question. I never think of the article or the Framework in the context of my changing my life. I was just doing what I was doing. I will probably keep on doing it because I like doing it and I actually do believe that the Framework reflects some underlying truths, natural laws. I think we all have this kind-of innate desire to discover the underlying order of the universe. In fact, I call some of the seminars I do, "Enterprise Physics 101". The more I think about this, the more I am convinced that it IS "physics", the kind-of underlying laws of nature as related to Enterprises.

**ROGER SESSIONS:** Have you been surprised by the influence of this paper in the industry?

**JOHN ZACHMAN:** This is another hard question because I am not conscious of the influence of the paper on anyone, not to mention an industry! On the other hand, I consciously try to influence as many people as I can--- to get serious about Enterprise Architecture.

I think the United States dominated the Industrial Age because it got serious about architecture for industrial products. We learned how to create extremely complex engineering products and how to change them to keep them relevant over their useful life, that is, we learned about Architecture for industrial products.

My opinion is that in the Information Age, it is the Enterprise that is increasing in complexity and changing dramatically and that whoever figures out how to accommodate and exploit Enterprise Architecture concepts and formalisms, and therefore can accommodate extreme complexity and extreme change of Enterprises, is likely to dominate the Information Age.

**I think the United States dominated the Industrial Age because it got serious about architecture for industrial products.**

**ROGER SESSIONS:** When you wrote this paper, you described a "Framework for Information Systems Architecture". We now know this framework as The Zachman Framework for Enterprise Architecture<sup>6</sup>. Can you discuss the transition from a framework for "Information Systems" to a framework for "Enterprise Architectures"?

**JOHN ZACHMAN:** I think I answered this question earlier. The Framework has never been in a transition. The Framework has in a way been in existence forever. It is simply a schema.

It is kind of like the Periodic Table which is also a schema and also has been in existence forever. The Periodic Table classifies the elements of the universe in two dimensions: first, by the number of neutrons and protons in the nucleus of the atoms; and second, by the number of electron rings of the atoms. It was only after Mendeleev articulated that schema that Chemistry became a science, a discipline.

Up until that time, Chemistry was Alchemy. The alchemists could make compounds all right, but it was not a science. It was an art. Compound-making was based on their experience. It was not predictable and not repeatable. The Periodic Table had been there all the while. It simply had not been made explicit. It was the practitioners' understanding of it and how to use it that was changing.

Similarly, it is our understanding about the Framework that has been in a transition. The Framework is kind of like the Periodic Table in the sense that the schema has been there all the while. All I did was make it explicit, and it is only our understanding of it and how to use it that is changing.

Anyone who has attended one of my seminars knows that I, in no way, am trying to compare myself with Mendeleev. I point out that I am sure Mendeleev knew what he was looking for when he defined the Periodic Table. In contrast, I had no idea what I was looking for

### Interview with John Zachman

(continued from page 4)

when I defined the Framework. I was just hoping I would discover something when one day, it kind-of fell on my desk! It's kind of like a blind squirrel finds an acorn every now and then.

By the way, it was only 50 years after Mendeleev published the Periodic Table that the practice of Chemistry became predictable and repeatable such that the practitioners, the methodologists, the Chemists, could get sophisticated to the extent that they could split atoms. It occurs to me that once the underlying structure of a discipline is discovered, friction goes to zero! The processes (methodologies) become predictable and repeatable. We may have a similar experience in the Enterprise Engineering and Manufacturing discipline.

**ROGER SESSIONS:** How have you seen the field of Enterprise Architecture change over the last 20 years? What aspects of it have not changed?

**JOHN ZACHMAN:** I think that the principal change over the last 20 years is the perception of legitimacy of the subject, Enterprise Architecture. This is not insignificant! Because, if you don't think that Enterprise Architecture is a legitimate subject, the likelihood of doing something about it or investing in the research for making it a science or even learning about it is low to zero.

What has not changed is that we still have LOTS to learn about it. It was only January of this year that SIM, the Society of Information Management, launched a Working Group to study Enterprise Architecture and that was only because Leon Kappelman, a Professor at North Texas University, had sensed that it was time to establish some legitimacy to the subject.

In general, the information industry is still focused on building and running systems, that is, we are *manufacturing* pieces, islands of automation, stovepipes of the Enterprise. In general, the information industry presently is not thinking about *engineering* entire Enterprises.

**ROGER SESSIONS:** How do you define *Enterprise Architecture*?

**JOHN ZACHMAN:** This is another hard question for me as well because I have written a book on the subject, *The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing*<sup>1</sup>. However, in short, Architecture is a set of descriptive representations that are relevant for describing something you intend to create and that constitute the baseline for changing an instance of that thing once you have created it. Therefore, *Enterprise Architecture* is the set of descriptive representations

relevant for describing an Enterprise and that constitutes the baseline for changing the Enterprise once it is created.

Unfortunately, few (if any) Enterprises in 2007 have Enterprise Architecture. That is the problem. Up until now, Enterprises have happened ... somehow. They haven't been architected (engineered).

In contrast, by 2007, *every* airplane, *every* hundred story building, *every* locomotive, *every* battleship, *every* computer, *every* industrial product has Architecture. Otherwise, we couldn't have created these complex products and we couldn't make continuous engineering changes to them without catastrophic results.

Jay Forrester observed that "Organizations built by committee and intuition perform no better than would an airplane built by the same methods ... As in a bad airplane design, which no pilot can fly successfully, such badly designed corporations lie beyond the ability of real-life managers."<sup>8</sup>

**ROGER SESSIONS:** How do you define an *Enterprise Architecture Framework*?

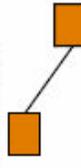
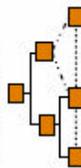
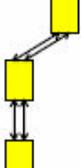
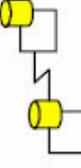
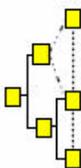
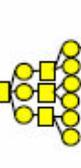
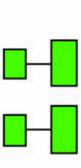
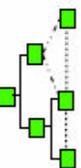
**JOHN ZACHMAN:** My definition of Framework is "schema", a classification structure. There are other people that use the word to mean things like, "a high level or generalized set of guidelines, typically for directing some kind of work, some process or methodology." A schema is quite different. A schema is used for analytical purposes, for "engineering". A good schema, or good classification system, is two dimensional, that is, it is "normalized", it is "clean". There can be no "apples and oranges". That is, you can't put two different things in the same category, nor can any one thing go in more than one category.

Why would one want to employ classification theory? For engineering purposes. To simplify. To examine one thing (one category) at a time, but at the same time, not lose sense of that one category of things in the context of the total set of categories.

Therefore, an Enterprise Architecture Framework would be a normalized classification of descriptive representations, of engineering design artifacts, of architectural representations for an Enterprise. The "Zachman Framework" is a schema, a classification of the total set of descriptive representations that are relevant for describing an Enterprise such that it is "normalized". One (meta) fact can only go in one Cell and one Cell can only contain one kind of (meta) fact.

Many people look at my Framework and miss this point. The metamodel for all the relevant meta-facts has always been found at the bottom of each Cell in the Framework.

ENTERPRISE ARCHITECTURE - A FRAMEWORK™

	Who	What	How	Where	Who	When	Motivation	Why	
SCOPE (CONTEXTUAL)	Planner	DATA List of Things Important to the Business  ENTITY - Class of Business Thing e.g. Semantic Model 	FUNCTION List of Processes the Business Performs  Process - Class of Business Process e.g. Business Process Model 	NETWORK List of Locations in which the Business Operates  Node - Major Business Location e.g. Business Logistics System 	PEOPLE List of Organizations Important to the Business  People - Major Organization Unit e.g. Work Flow Model 	TIME List of Events/Cycles Significant to the Business  Time - Major Business Event/Cycle e.g. Master Schedule 	MOTIVATION List of Business Goals/Strategies  Ends/Mean = Major Business Goal/Strategy e.g. Business Plan 		
BUSINESS MODEL (CONCEPTUAL)	Owner	Ent = Business Entity Reln = Business Relationship e.g. Logical Data Model 	Proc. = Business Process IO = Business Resources e.g. Application Architecture 	Node = Business Location Link = Business Linkage e.g. Distributed System Architecture 	People = Organization Unit Work = Work Product e.g. Human Interface Architecture 	Time = Business Event Cycle = Business Cycle e.g. Processing Structure 	End = Business Objective Means = Business Strategy e.g. Business Rule Model 		
SYSTEM MODEL (LOGICAL)	Designer	Ent = Data Entity Reln = Data Relationship e.g. Physical Data Model 	Proc. = Application Function IO = User Views e.g. System Design 	Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics e.g. Technology Architecture 	People = Role Work = Deliverable e.g. Presentation Architecture 	Time = System Event Cycle = Processing Cycle e.g. Control Structure 	End = Structural Assertion Means = Action Assertion e.g. Rule Design 		
TECHNOLOGY MODEL (PHYSICAL)	Builder	Ent = Segment/Table/etc. Reln = Pointer/Key/etc. e.g. Data Definition 	Proc. = Computer Function IO = Data Elements/Sets e.g. Program 	Node = Hardware/Systems Software Link = Line Specifications e.g. Network Architecture 	People = User Work = Screen Format e.g. Security Architecture 	Time = Execute Cycle = Component Cycle e.g. Timing Definition 	End = Condition Means = Action e.g. Rule Specification 		
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	Sub-Contractor	Ent = Field Reln = Address e.g. DATA 	Proc. = Language Statement IO = Control Block e.g. FUNCTION 	Node = Address Link = Protocol e.g. NETWORK 	People = Identity Work = Job e.g. ORGANIZATION 	Time = Interrupt Cycle = Machine Cycle e.g. SCHEDULE 	End = Sub-condition Means = Step e.g. STRATEGY 		
FUNCTIONING ENTERPRISE		e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY		FUNCTIONING ENTERPRISE

© 1986 - 2005 John A. Zachman, Zachman International

### Interview with John Zachman

(continued from page 5)

In fact, the meta-meta-entities are found at the bottom of each Cell as well. They have all been there from its origin. It is for these meta-entities that we have labored to find better names, more business-oriented words, words that more accurately reflect the normalized concepts of the schema.

Is the metamodel of the Framework Important? Clearly, if you want to store all of the Enterprise Models in a database, you would have to have a model of all the models and the defined relationship between all of the models, in which case you could store the total “knowledge-base” that was relevant for describing the Enterprise (and therefore managing it) and that would serve as the baseline for changing it in minimum time with minimum disruption and minimum cost.

In short, I use the word “framework” to mean a structural classification of single variable concepts, a two-dimensional semantic *structure* ... *not* a generic process, nor a methodology, nor even a categorization of outputs from processes or methods. The outputs of methods or processes I would suggest are “views” of an underlying semantic structure; and views are not the semantic structure; and semantic structures are not views.

Those are two vastly different, however related, things. My Framework is a semantic structure. The outputs (models) produced by a methodology are views ... that is, unless the methodology deliberately is designed to do engineering, that is, Architecture, producing single variable, normalized artifacts, what I would call, “primitive” models.

**ROGER SESSIONS:** Why should a large enterprise care about Enterprise Architectures?

**JOHN ZACHMAN:** There are basically three reasons for Architecture: Complexity, Change and Mass-Customization. If you want to create something so complex that you can't see it all at one time in its entirety in the excruciating level of detail required to create it, you will have to start writing things down ... descriptive representations ... Architecture.

This is a pretty simplistic metaphor, but if you are only trying to create something simple like a log cabin, then you don't need Architecture ... you need a tool (axe) and some raw material (a forest) and some time. The smaller the log cabin, the faster you can create it.

By the same token, if you are only trying to write a program, you don't need Enterprise Architecture ... you need a tool ... a compiler ... some raw material ... some data and some time. And, the smaller you make the

program, the faster you can write it. However, if you are trying to create a hundred-story building or an Enterprise, forget the tools and raw material ... now you are going to have to have Architecture.

Regarding change, if you retain the descriptive representations you used to create the complex thing and you maintain the representations consistent with the instantiation, then the representations constitute a baseline for managing change, that is, you can change the instance of the thing in minimum time, with minimum disruption at minimum cost. And conversely, if you don't ... you can't.

Regarding mass customization, if you engineer the “primitive” components (the single variable models, the “elements”, one category of the schema) such that they can be assembled into more than one product (implementation) and if you pre-fabricate them so they are in inventory before you ever get an order for a result (an implementation), then you can reduce the time to create the end result to only the time it takes to pick the parts, the “primitive” components, out of inventory and assemble them in response to the order, virtually, zero time-to-market.

Furthermore, if an Enterprise does not have the Column 1 (Inventory) models and has not kept them “aligned”, I don't think it can manage the *Enterprise* inventories of assets.

If it doesn't have the Column 2 (Transformation) models and has not kept them aligned, I don't think it can manage the yield on the *Enterprise* transformations.

If the Enterprise does not have the Column 3 (Distribution) models and has not kept them aligned, I don't believe it can manage the *Enterprise* storage and transportation (or transmission) capacities.

If an Enterprise does not have the Column 4 (Work Flow) models and has not kept them aligned, I don't believe it can manage the *Enterprise* work performance.

If an Enterprise doesn't have the Column 5 (Dynamics) models defined and has not kept them aligned, I don't believe it can manage the *Enterprise* cycle times.

If an Enterprise doesn't have the Column 6 (Motivation) models defined and has not kept them aligned, I don't believe it can manage the *Enterprise* objectives.

That is, in each case, without Enterprise Architecture as defined by my Framework, I don't believe the *Enterprise* can be managed without spending a lot of money and without risking a lot of errors, that is defects, in the Enterprise.

### Interview with John Zachman

(continued from page 7)

**ROGER SESSIONS:** Many CIO's struggle to justify the return on investment (ROI) of creating an Enterprise Architecture. What advice would you give them?

**JOHN ZACHMAN:** This is a BIG problem.

First, if you don't believe that Enterprise Architecture is a legitimate idea, the probability of investing in it is low to zero.

Second, writing a program and getting it to run is a lot easier and faster than engineering and manufacturing an Enterprise. And, systems of programs that are running are discernable results. You can count them. You can run them. You can calculate the costs/benefits. Etc.

However, the end object of Enterprise Architecture is not building and running systems, getting programs to run. The end objective is to engineer the Enterprise so it is as LEAN as possible (minimum possible complexity and minimum possible costs) and MEAN as possible, that is, so that it can dynamically accommodate external demands.

If this was easy, we probably wouldn't have to pay General Management exorbitant salaries and every Enterprise would be lean and mean. No ... this is really hard, and it won't happen overnight. It is actually a different way of life, a new paradigm. It will require engineering work (i.e. Enterprise Architecture).

I think it was Fred Brooks who observed that programming is manufacturing ... not engineering. Therefore, the CIO's of the world may not even have a lot of Enterprise Architecture (engineering) skills available upon which to draw.

Here's my advice ... we'd better start working on this pretty soon because to develop and/or acquire the skills, the learning curve may be a little flat; however, at the point in time that the Enterprise needs to have these Enterprise Architecture capabilities, it may be too late to acquire them. And, I don't think we will have until the sweet bye and bye to get some critical mass of Architecture artifacts in place.

Here's another piece of advice ... you don't have to have the entire Enterprise engineered, that is all 30 Cells of my Framework made explicit, Enterprise-wide, horizontally and vertically integrated at excruciating level of detail before you can derive benefit or before you can implement anything. That ought to be your end-state vision, but you can start manufacturing before you have the product completely engineered ... like all manufacturers do. You just have to do enough engineering to mitigate the downstream costs of scrap and rework and assume some

risks that the incurred scrap and rework costs will not exceed the value of the investment.

**Therefore, the CIO's of the world may not even have a lot of Enterprise Architecture (engineering) skills available upon which to draw.**

Also, there could be a LOT of short term benefits to be derived from doing a LITTLE engineering. That is, you don't actually have to produce manufactured results to derive benefits from some engineering.

Here is another thought ... the CIO actually cannot unilaterally solve this problem. It is an ENTERPRISE problem. The CIO's and those of us who have some understanding of Enterprise Architecture can do some work on this whether the Enterprise understands what we are doing or not.

At some point, it is only the Enterprise that can make the trade-off between the short term and the long term. Do they want to save money or do they want to grow? Do they want to cut expenses or do they want to build assets.? Do they want us to build and run systems or do they want us to engineer and manufacture the Enterprise?

I can't help but think of a July '06 *Fortune Magazine* article, "The New Rules"<sup>9</sup> in which the author argues that this may be a watershed time in the history of American Capitalism. The whole concept of Capitalism is to invest in the current time frame to reap benefits in the future, not to cut costs, sacrificing the future in order to jack up the stock price in the current accounting period. And, I can't help but remember a *Wall Street Journal* article a year or so ago that observed, "you can't cut your way to sustainable growth."<sup>10</sup>

**ROGER SESSIONS:** Can you give us basic overview of the Zachman Framework?

**JOHN ZACHMAN:** The Zachman Framework is a two dimensional classification scheme ... a normalized schema. It is the intersection of two classifications, the universal linguistic communication classification of primitive interrogatives: What, How, Where, Who, When and Why; and the classification of audience perspectives: Owner, Designer, Builder, bounded by the Scoping Perspective and the Implementation Perspective (Owner, Designer, Builder is more commonly known in the I/T community as Concepts – Computation Independent; Logic – Platform Independent; and Technology – Platform

### Interview with John Zachman

(continued from page 8)

Specific and the Scoping Perspective as the Boundary Definition; the Implementation perspective as Vendor Specific.)

On page 6 is the picture of the Framework graphic. This is the public domain version that has my original names for the meta-entities at the bottom of each Cell.

**ROGER SESSIONS:** Can you tell us about two or three major success stories in applying The Zachman Framework to a large enterprise?

**JOHN ZACHMAN:** This is another hard question for me to answer because I am not a methodologist, constrained by results and time. Practitioners (methodologists) have to do actual work, to produce definitive results within specified time frames. The methodologist is *not* constrained by theoretical purity. They will compromise theoretical purity in the interest of delivering results within some specified period of time.

I am a theoretician, kind of like a scientist. I am preoccupied with structure, classification. I search for the underlying natural laws and further, I am constrained by integrity. I have spent my lifetime trying to understand the schema and to ensure it has integrity internally within itself, that is, it is “normalized,” and that it has integrity with all other known natural laws (that is, known science). I suppose there is another sense of integrity in the fact that, I cannot falsify the schema (the natural laws) to prove my personal theories. (Tragically, we have seen some scientists falsify the data to prove their own theories, but my personal opinion is that this is a self-defeating strategy.) As a theoretician, I cannot with any integrity, be constrained by results and time (that is, I cannot be constrained by process, methodology.)

**I am preoccupied with structure, classification. I search for the underlying natural laws and further, I am constrained by integrity.**

Having said all of that, some of my best friends are methodologists. They tell me of successes they have had in employing the concepts of my Framework notably:

Doug Erickson of ENTARCO has had notable success at the State of Pennsylvania, Department of Labor and Commerce as well as the State of Ohio, Bureau of Workers Compensation.

Sam Holcman of Pinnacle Business Group has had notable success at General Motors as well as at Volkswagen.

John Hall of Model Systems has had notable success at Swisscom Mobile.

Stan Locke of Zachman Framework Associates has had notable success at Canadian Tire (which is kind-of the Wal-Mart of Canada ... with that comment, I am sure I have offended some people from Wal-Mart as well as some people from Canadian Tire) as well as success at SIL International, Wycliffe and JAARS.

Clive Finkelstein of Information Engineering Services Corporation has had notable successes at Kengii Korean Regional Bank and the Government of Canada. Clive has worked around the world for decades and has recently published a book mapping his Information Engineering methodology against my Framework, “Enterprise Architecture for Integration.”<sup>11</sup>

All of these methodologies are vastly different. The thing they have in common is they employ, to at least to some extent if not to a large extent, the engineering concepts embodied in the Framework schema.

**ROGER SESSIONS:** What are the greatest challenges that one can face in trying to apply The Zachman Framework?

**JOHN ZACHMAN:** There are several formidable challenges. First of all, the stress level in many Enterprises is extremely high. There is a great urgency to produce short term results.

The concept of Enterprise Engineering and Manufacturing (i.e. Enterprise Architecture) constitutes a new paradigm. If we could effect the new paradigm, we would likely relieve considerable stress in the Enterprise through managing complexity, change and reducing the time to create new manifestations of the Enterprise. If you have architectural assets, that is, assets that have been engineered to be reused, or “assembled” into virtually any manifestation of the Enterprise required by Management, you can reduce the time for new implementations to just the time required to assemble them, virtually zero, making new manifestations of the Enterprise dynamic.

The problem is much more of a cultural problem than a technical problem. There is nothing keeping us from doing this work short of doing the work. Unfortunately, we (the information community) are not long-term-oriented, engineering things to be reused in any implementation of the Enterprise, Enterprise-wide. We tend to be very short-term-oriented. That was the message applied more generally (not specifically to only the information community) in the Fortune article I referred to above, “The New Rules.”<sup>12</sup>

**Interview with John Zachman**

(continued from page 9)

To be fair, in 2007, there are still a lot of things about Enterprise Architecture that remain to be discovered. There is some legitimate discomfort in the uncertainty of what we yet do not know. The good news is that the body of knowledge at the moment is exploding as a result of the openness to acknowledge the legitimacy of the subject, and we can learn to do and actually do architecture work without having to know everything about the subject.

By the way, there are cultures in the world that are not intimidated by the specter of a long term, asset-based strategy.

**By the way, there are cultures in the world that are not intimidated by the specter of a long term, asset-based strategy.**

**ROGER SESSIONS:** Besides The Zachman Framework, TOGAF is probably the other well-known framework for Enterprise Architectures. What are the similarities and differences that you see between TOGAF and The Zachman Framework?

**JOHN ZACHMAN:** The “Zachman Framework” is a schema, an application of classification theory, a device amenable to engineering. Other frameworks tend to be focused on implementation or methodologies as opposed to classification, and therefore, facilitate manufacturing as opposed to facilitating engineering. A classification structure could be used to evaluate methodological formalisms (other frameworks) for completeness, for motivation, for underlying assumptions, etc.

TOGAF is completely different from my Framework. My Framework is totally neutral relative to methodologies ... Process. It has no methodological implications, that is, it does not imply anything about how you might do Enterprise Architecture or what “composite” artifacts you might produce methodologically. In contrast, TOGAF prescribes how you might do architecture. In that regard, TOGAF is similar to DODAF<sup>13</sup> or MODAF.<sup>14</sup>

Someone recently told me that there is a new TOGAF document that maps the TOGAF models to my Framework. I am not sure how they have done that, or whether they were meticulous about adhering to the underlying theoretical implications of the “primitive” engineering design artifacts. Someone also told me that the TOGAF artifacts map against multiple Cells of my

Framework. That should be expected since the orientation of a methodology is implementation, that is, manufacturing. Implementations must, by definition, be “composite” constructs, that is, made of components from more than one “primitive” Framework Cell.

My Framework would be useful to a methodologist for determining how complete the methodology might be, whether it was engineering-oriented or manufacturing-oriented, what compromises might be made or should be avoided in producing results, how to accommodate the meta-constructs required for implementations, etc, etc.

Most other frameworks, specifically TOGAF, are preoccupied with getting work done (manufacturing). They are not preoccupied with defining the nature, the inherent structure, of the work (engineering). Those are two completely different things.

**ROGER SESSIONS:** The Clinger/Cohen Act of 1996 effectively mandated the use of an enterprise architectural methodology throughout the U.S. Federal Government. This led directly to the Federal Enterprise Architecture. Many people credit your framework as an important influence on the Federal Enterprise Architecture. Can you discuss your involvement with the Federal Enterprise Architecture and the influence on it of The Zachman Framework?

**JOHN ZACHMAN:** The origin of the Federal Enterprise Architecture Framework<sup>15</sup> principally was a NIST research project in 1989. There were two Frameworks proposed in the resultant NIST Research document<sup>16</sup>, my Framework (three Columns of my two-dimensional schema addressing Enterprise engineering) and what later became known as the NIST Framework, a single dimensional classification of subject areas supporting Information Strategy. Since the issue was still perceived to be an Information issue, the “NIST Framework” was picked up by several Federal Agencies and used as the basis for their information strategy.

Subsequently, it became more apparent that the issue was much more an architecture issue and my Framework became somewhat influential in other agencies. In the FEAF document of 2001, a derivative of the NIST Framework plays a prominent part although if you look closely, one of the figures shows my Framework underlying the NIST graphic.

It is my understanding that the next release of the FEAF document was to include my complete six column Framework. Unfortunately, the next release of the FEAF document never materialized as the people who were instrumental in its impending production either left government service or were placed in other responsibilities.

### Interview with John Zachman

(continued from page 10)

I only attended one or two of the workshops that resulted in the FEAF document, so my contribution to that particular document was minimal. Also, since I was not in the employment of the government, I am only recounting this history as an external observer. I happened to be intimately involved in the NIST research project in 1989, so that portion of the history is likely fairly accurate. In fact, I wrote 60 of the 159 pages of the NIST project document.

I doubt that anyone else would remember or even associate the origins of the FEAF Framework with the NIST project. Although there are several folks who participated in the research project still active, I don't know any of them involved with the Federal Government or with FEAF, so I am fairly certain that connection has been lost.

In fact, I doubt that you could find many people that even know that the NIST research project ever occurred or that the resultant NIST document even exists anymore.

**ROGER SESSIONS:** Many new technologies have come into play since your paper came out. Most recently, web services is having a major impact. Soon, we expect to see software as a service, playing an important role in the enterprise. Do you see these technologies changing your view on enterprise architectures and if so, how?

**JOHN ZACHMAN:** No technology will ever affect the Framework schema. It is independent of technology. General technologies found in the marketplace would constitute alternative constraints employed in the Row 4 Enterprise instance models. Specific vendor product technologies constitute the constraints of the Row 5 instance Listings. The schema simply defines the templates for and orders the Enterprise engineering design



John Zachman (early photo)

artifacts. The contents of the artifacts are Enterprise-specific, which is where specific technologies would appear.

**ROGER SESSIONS:** What changes do you expect to make in The Zachman Framework over the next five years?

**JOHN ZACHMAN:** Once again, the Framework schema has been around for thousands of years and I am sure it will be around for a few more thousands of years. What changes is our understanding of it and how to use it for Enterprise engineering and manufacturing. It is kind of like the Periodic Table once again in the fact that the Periodic Table enables continued research and development in the science of Chemistry.

Similarly, the Framework for Enterprise Architecture would enable continued research and development in the science of Enterprise Engineering and Manufacturing if the practitioners, the methodologists, chose to use it for that purpose.

Within 50 years of the publication of the Periodic Table, the Chemists and Physicists were splitting atoms! Similarly, I suspect that if the Framework is used intensively by the methodologists, the state of the art in Enterprise Engineering and Manufacturing will explode in like fashion.

**ROGER SESSIONS:** How do you feel about how Computer Science is taught today? What changes would you like to see made?

**JOHN ZACHMAN:** This, clearly, is a gross generalization, but my observation is that the academic community is not helping us. Contrary to the general perception, the academic institutions seem to get very short-term-oriented. If they thought that the local corporations were hiring JAVA II programmers, they would have Masters Degree programs in JAVA II programming.

This global human penchant for “silver bullets” tends to search for the next technological panacea that will be the solution to all the world's problems. This tends to concentrate the academic focus on the technologies rather than on the Enterprise. I hope that the revelation that the end object is engineering and manufacturing of the Enterprise will at some point take precedence over technological silver bullets and building information systems. That would constitute a major breakthrough in academia in my estimation.

**ROGER SESSIONS:** I have noticed that your range of interest extends beyond technology. You are, for example, active in Native Hope International. Could you tell us something about your work with this organization?

**Interview with John Zachman**

(continued from page 11)

**JOHN ZACHMAN:** Native Hope International is a ministry that is supported out of the church I attend in Van Nuys, California, the Church on the Way. Native Hope International simply loves the people of the Native American community and tries to help them see that God has a future and a plan for their lives and to give them hope for a better life.

The focal point for the ministry is my long time friend, Bryan Bright Cloud. His approach to the Native American peoples is a pretty typical Christian ministry approach of not trying to change people but loving them and showing them through love that God loves them.

There is something of a relationship between this ministry and my work in Enterprise Architecture. I would observe that chaos happens by chance, by accident. Order happens by “intelligent design”, and intelligent design presumes a designer who is eternally consistent and benevolent. If I didn’t believe that there was order in the universe that was eternally consistent and benevolent, I would not likely be inclined to expend a lot energy looking for it.

**ROGER SESSIONS:** John, it has been a pleasure talking to you. Thank you very much for your legendary contributions to our industry, and for being with us today.

**JOHN ZACHMAN:** Roger ... once again, thank you for your interest in my work. I have enjoyed thinking through your thoughtful questions. I hope you find some useful insight in my answers.

**References**

- (1) “A Framework for Information Systems Architecture” by John A. Zachman. IBM Systems Journal Vol. 26, No. 3, 1987.
- (2) “The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing” by John A. Zachman. www.ZachmanInternational.com. Included in the Appendix of Articles by John A. Zachman
- (3) EIMInsight Magazine March 2007. www.EIMInstitute.org.
- (4) Notably Gary Simons, Associate Vice President for Academic Affairs, SIL International, Dallas, TX
- (5) “Business Systems Planning and Business Information Control Study: A Comparison” by John A. Zachman. IBM Systems Journal Vol. 21, No. 1 1982.
- (6) “Extending and Formalizing the Framework for Information Systems Architecture” by John F. Sowa and John A. Zachman. IBM Systems Journal Vol. 31, No. 3 1992.
- (7) i.b.i.d.
- (8) Jay W. Forrester is the author of “Industrial Dynamics”, M.I.T. Press 1961 – 1973. The above quote

- came from a talk, “Designing the Future” on December 15, 1998 at Universidad de Sevilla, Spain.
- (9) “The New Rules” by Betsy Morris. Fortune Magazine July 24, 2006.
- (10) “When Meeting Targets Becomes the Strategy, CEO is on Wrong Path” by Carol Hymowitz. Wall Street Journal March 8, 2005
- (11) “Enterprise Architecture for Integration: Rapid Delivery Methods and Technologies” by Clive Finkelstein. Artech House 2006. Journal March 8, 2005.
- (12) i.b.i.d.
- (13) Department of Defense Architecture Framework (U.S.)
- (14) Ministry of Defense Architecture Framework (U.K.)
- (15) Chief Information Officers Council “Federal Enterprise Architecture Framework” Version 1.1, September 1999. U.S. Department of Commerce, Technology Administration, National Technical Information Service.
- (16) NIST Special Publication 500-167 “Information Directions: The Integration Challenge” Elizabeth N. Fong and Alan H. Goldfine, Editors. Information Systems Engineering Division, National Computer Systems Laboratory, National Institute of Standards and Technology September 1989.

**I hope that the revelation that the end object is engineering and manufacturing of the Enterprise will at some point take precedence over technological silver bullets and building information systems.**



**Enterprise Architecture Consulting**

**Roger Sessions, CTO**

- High Business Value Focus
- Accelerated Time to Payoff

[www.objectwatch.com](http://www.objectwatch.com)