

Enterprise Architecture: Issues, Inhibitors and Incentives

By
John A. Zachman
© 1999 Zachman International

Recently, there has been quite a bit of discussion in the U.S. Federal Government on the subject of Enterprise Architecture. In fact, there is a Congressional order, the Clinger-Cohen Act, that requires each of the Federal Bureaucracies to produce an Architecture Strategy, which precipitated circulation of set of questions for discussion, as follows:

1. What are the most important issues raised by Architecture?
2. What are the biggest inhibitors to Architecture?
3. What are the business drivers for Architecture?

My response to these questions, of course, was based on my understanding of the Framework for Enterprise Architecture in which I hypothesized that Enterprise Architecture is comprised of a set of architectural representations (models). The Framework is typically depicted in matrix form whose rows represent the perspectives served by the representations, that is, the Owner, Designer, Builder, etc. and whose columns represent different abstractions, or independent variables, which correspond to the set of primitive interrogatives, that is: What (things), How (processes), Where (locations), Who (people), When (events) and Why (motivation.) In all, there are 30 different representations relevant for describing a complex object, like an Enterprise. (Some people count the “Implemented Enterprise” row of models in the total in which case the number would be 36.)

With that as a little background, here are the answers I posed to the three discussion questions:

Most Important Issues

There are three vital issues that become evident when you consider the set of models that constitute Enterprise Architecture.

ISSUE 1. If you don’t actually build the models, it doesn’t mean that the models don’t exist.

If the Enterprise exists, by definition, all of the models always exist ... you have the “Owner,” you have the “Designer,” you have the “Builder,” etc., etc. You have “Things,” you have “Processes,” you have “Locations,” etc., etc.

However, it takes time and costs money to build the models. If you don’t take the time and spend the money to build the models, it doesn’t mean that the models go away. It simply means that you didn’t take the time and spend the money to make the models EXPLICIT. That is, the models are IMPLICIT. That means, you are making ASSUMPTIONS about the models. In this case, the question becomes, “how good are your assumptions?”

For an indication of how good your assumptions have been over the last 50 years, how does the Enterprise feel about the current inventory of applications, the “legacy?”

In most cases, there is no architecture that makes explicit the assumptions that have been implicitly imbedded in the legacy applications and the high probability is that the assumptions that were initially made, even if they were good at the outset, are becoming increasingly inconsistent with the reality of the Enterprise as it presently is perceived, and the time and cost of changing the legacy applications is contributing heavily to the frustrations that the Enterprise is feeling toward the I/S community these days.

The implications of having no Architecture are that once the assumptions start to change, it is extremely difficult, extremely time consuming, and extremely costly to attempt to “reverse engineer” the assumptions out of the existing systems, change them and reconstruct new implementations.

In the long term, it is much cheaper and faster to build and maintain all the models if the assumptions start changing faster and faster.

ISSUE 2: The “system” IS the Enterprise.

Traditionally, people have not perceived that the systems of the Enterprise actually ARE the Enterprise. In most cases, it was thought that the systems merely “supported” the Enterprise. Although that may have been the case in times past, it clearly is not the case any longer. In fact, there are few cases where it even would be possible to exist as an Enterprise without “systems.” In fact, we have been replacing the people of the Enterprise with systems for 50 years and it is only logical that if the replaced people were the Enterprise, the systems that replaced them have become the Enterprise.

In fact, some of the systems of the Enterprise are still manual (that is, people) systems. The only thing that differentiates the manual systems from the automated systems is the technology constraints that are employed by the “Builder” (Row 4) in their implementation. If the technology employed is pencils, paper and file cabinets, the resultant system would be called a “manual” system. On the other hand, if the technology is stored programming devices and electronic media, the resultant systems would be called “automated” systems. In any case, the system, automated or manual, IS the Enterprise.

Enterprise Architecture is not a technical issue ... or an I/S issue. It is an ENTERPRISE issue!! If there is one good thing that has come out of the Y2K problem is a heightened awareness that Architecture is an Enterprise issue because if the system is broken, the Enterprise is broken! The system IS the Enterprise.

The systems community, on their own, will never be able to fix the problem as the problem is an Enterprise problem.

ISSUE 3: Enterprise Architecture is a physics problem. Actual work (not some kind of technological magic) will have to take place.

All the desirable characteristics you want for your systems (that is, for your Enterprise, since the system IS the Enterprise) including flexibility, interoperability, integration, seamlessness, alignment, adaptability, quality, reduced time-to-market, and so on are the result of engineering, Enterprise engineering, Enterprise Architecture Engineering. Nothing magic is going on. Neither is it technology that delivers all of these desirable characteristics. It is only actual work, engineering work that will enable the Enterprise to accommodate dramatic increases in complexity and to change with minimum time disruption and cost.

If you REALLY want the systems of the Enterprise to be aligned with the Enterprise intent; if you REALLY want the Enterprise to be flexible, adaptable; if you REALLY want systems on demand, to change the Enterprise dynamically with minimum time disruption and cost, and so on ... actual Enterprise Architecture engineering work is going to have to take place. You are going to have to build and manage all of those models.

Biggest Inhibitors

The three biggest inhibitors to Enterprise Architecture are the users, I/S and the vendors.

The Users.

The Users don't really want to deal with these Architectural issues because it requires actual WORK! They would rather just throw some money at the problem, that is, find some technological "silver bullet" solution, or just let I/S take care of it!

I/S.

I/S doesn't really want to solve the problem because the prevailing cultural inclination still tends to be "you start writing the code and I'll go find out what the users have in mind." "Actual work in not taking place until code is being produced." "Lines of code ... that is what we are being paid for!"

The Vendors.

The vendors don't really want to solve the problem because "a bird in the hand is worth two in the bush." It is better to take the revenue from a sub-optimal solution than risk

waiting for revenue from an optimal solution. And beyond that, optimal solutions mean less total accumulated revenue because optimal solutions remove redundancy and discontinuity!

Note that this is a “vicious cycle.” The users, I/S and the vendors all feed off of each other to inhibit Architecture progress. The only place it clearly is imperative to deal with the Architecture issue is at the level of the Enterprise itself, the ultimate level of optimization. Unfortunately, in 2000, the stress levels in the Enterprise are so high that it is extremely difficult to get General Management to hold still long enough to even understand the problem, let alone figure out what to do about it. It has all the earmarks of a “Catch 22.”

PLEASE!! DON'T MISUNDERSTAND ME!!! I am NOT saying that all we can do is give up until General Management figures out there is a problem and decides what to do about it. My opinion is, WHOEVER figures out there is a problem (user, I/S or vendor ... or, General Manager, for that matter) has to do everything in their power to break the vicious cycle and start working on Architecture.

If user, I/S, vendor or General Manager could think objectively for a moment, it becomes abundantly clear that their own, long term, self interests can only be served through employment of Architectural concepts as the health and well-being, that is, the survival of the Enterprise rests on it.

Business Drivers

There are two business trends that are driving the issue of Enterprise Architecture, complexity and change. These are the inherent characteristics of an information-based economy.

Complexity.

If you give everyone access to the same information at the same time, a “powershift” will take place. No longer will all the power be concentrated in two or three people at the top. The power goes outboard, de-central. In fact, if the customer has access to the same information at the same time as the supplier, the power will shift into the customer environment. It will become a “buyer’s market.” (This was elaborated in detail by Alvin Toffler in “Powershift.”)

In this case, the complexity of the Enterprise will increase dramatically. Business will no longer be as simple as, develop a good product or service and then go find a bunch of customers to sell it to. NO!! The game will change. Now the game becomes, go find yourself a good customer and develop and/or integrate the set of products and/or services required to keep that customer a good customer. This is orders of magnitude more complex because the burden of integration now falls on the supplier instead of, by default, on the customer.

When the complexity starts increasing by orders of magnitude, it will be simply impossible to mentally retain the entire knowledge-base, to remember how it all works without recording the detail in some fashion ... without the descriptive representations that constitute Enterprise Architecture.

Change.

“Knowledge is change. The ever-increasing body of knowledge feeding the great engine of technology creates ever-increasing change.” Alvin Toffler, “Future Shock.”

Not too many people argue anymore with the dramatic escalation of the rate of change as we are all feeling it in a very personal sense as well as a corporate sense. In fact, in a CEO survey that was taken in the 1995, 1996 time frame, to a person, all 108 CEO’s that were interviewed declared that the biggest problem facing the Enterprise was “change.”

If you were to search all of history for wisdom in how humanity has learned to manage change, you would discover that change management always starts out with the descriptive representations of the object you want to change. In Architecture and Construction, in Engineering and Manufacturing, you start with the drawings, the functional specs, the bills-of-material, etc. of the object you want to change. That is you start with the architectural representations, in Enterprise terms, the set of models that constitute the Architecture of the Enterprise.

Think about buildings. How do you change buildings? You start with the drawings, the functional specs, the bills-of-material, etc.

Think about airplanes. You start with the drawings, the functional specs, the bills-of-material.

Think about that PC on your desk. How do you change that PC? You don’t reach in and start swapping jumpers on the motherboard indiscriminately. No! You start with the drawings, the functional specs, the bills-of material.

If you don’t have the descriptive representations (the drawings, functional specs, the bills-of-material, etc. ... the “models”) of the thing you want to change, you have only three options.

You can change by trial and error. Swap out a few jumpers on the motherboard and see what happens. This is okay as long as you don’t mind if you end up with a boat anchor instead of a PC. This is the HIGH RISK option.

Second, you can recreate the drawings, the functional specs, the bills-of-material, etc. This takes time! This is the LONG TIME and HIGH COST option.

Third ... you can not change it. This is the DINOSAUR option.

Trial and error (high risk), reverse engineering (long time and high cost) or don't change it (the dinosaur) options. These are the only three options for changing anything, including Enterprises, if you don't have the descriptive representations of the thing you want to change.

If you REALLY want to change the Enterprise, particularly if you want to change it often and with minimum disruption, time and cost ... the day you want to change it, you are going to wish you had all of the descriptive representations, the models that constitute the Enterprise Architecture.

What are the business drivers that demand Enterprise architecture? Complexity and change. Someday, you are going to wish you had all of those models, every one of them Enterprise-wide, horizontally and vertically integrated at excruciating levels of detail.

PLEASE DON'T MISUNDERSTAND ME! I am NOT saying you will have to have all the models built before you can start delivering any implementations in the short term. However, I AM saying, somebody better start thinking about how you are going to satisfy short term demand and at the same time minimize doing things that will inhibit or worse, PROHIBIT you from achieving the long term, integrated, engineered, architecturally consistent Enterprise of the Information Age. This means, somebody is going to have to get a pretty good idea about what the long term architected Enterprise looks like in order to figure out how to keep from compromising it to the extent that it is unachievable ... (kind of like where we are today with the "legacy" Enterprise.) This also means, there are going to be times when you will choose NOT to satisfy short term demand because of the compromise of the long term position.

The end object is not simply to get the systems to run. That is an Industrial age idea.

The end object to produce an Enterprise that can accommodate orders of magnitude increases in complexity and orders of magnitude increases in the rate of change. This is an Information Age idea and this REQUIRES Enterprise Architecture.

Enterprise Architecture is no longer an option ... it is mandatory.

*John A. Zachman
Zachman International
2222 Foothill Blvd. Suite 337
La Cañada, Ca. 91011
818-244-3763 (Phone and Fax)
johnzachman@compuserve.com*