

The Data Administration Newsletter (TDAN.com)

[Robert S. Seiner - Publisher](#)

[Main Menu](#)

[New Articles](#)

[Article Archive](#)

[Special Features](#)

[What's New](#)

[Contact the Publisher](#)

FRAMEWORKS, META-DATA REPOSITORY & THE KNOWLEDGE WORKER

Michael Gorman - Whitemarsh Information Systems Corp.

[\[Return to the Article Archive\]](#)

Frameworks are a good thing. They help us organize, plan and manage. They help us set things in context. We can see interrelationships. Major “things” can be seen as the columns, and the unfolding transformations for the different actors or clients can be seen as the rows. Frameworks have a simpleness to them and can be set against tests of “common sense.”

For sure, it is logical to believe that John Zachman had all this in mind when he created the Information Systems Framework in the late 1980s. Key to this framework’s columns were the six interrogatories, What, How, When, Where, Who, and Why. Mapped to an Information Systems Architecture, the Data was seen as the What, Function was How, Network was Where, People were Who, Time was When, and Motivation was Why. It was clean and simple.

It would be nice if life was similarly clean and simple. However, life is not, and also, life seems to become more cluttered and more complex. Interestingly, the increased clutter and complexity seems to be accelerating.

Several years ago while at a WRAD (Warehouse Repository Architecture and Design, www.wrad.org) 2000 in Orlando, Florida there was a Business Rules Group (www.businessrulesgroup.org) pre-conference session. Within this session, an attendee asked what they felt was a perfectly innocent question, “Just what was the Zachman Framework?” Key persons, all of whom are all known well, like David Hay, Terry Moriarty, Warren Selkow, and Ron Ross all rose to help this person with the question. The most interesting part of the eight distinct sets of answers was that each gave a different rendition of just what the Zachman Framework was and what it meant to them. All the presentations overlapped somewhat, but in key areas they were different.

While somewhat on the topic of Business Rules, Ron Ross was overheard to suggest that there is really a special Zachman framework for just business rules. Thus, rather than have business rules live and die within the Motivation column, it would have it’s own set of six columns and possibly six rows.

So, upon reflection there was one burning question: “Just what is the Zachman Framework?” The more precise the definition one tries to use, the more the framework became useful for one situation but then unsuitable for another. Earlier, in the early 1990s, The MITRE Corporation was tasked to create a detailed plan and implementation strategy for the Zachman framework within a large Federal organization. It failed. Every time the Zachman framework was pushed in one direction it

became unglued from and too far astray from another.

In 1996, while attending a DAMA New Jersey conference at which Barbara von Halle spoke, she conveyed how she used the Zachman Framework. Her presentation was different from previous framework presentations heard at WRAD. Similar, but different. Goal of her presentation was to help identify areas within the enterprise that needed work. VonHalle would provide a brief list of the items inferred by the column-row intersection and then solicit the enterprise's assessment. Areas that seemed "challenged" then became opportunities for consulting.

In a recent book on systems analysis by David Hay, he employed the Zachman framework as a strategy for explaining his trek through systems analysis. Chapter 2 of a draft of the book describes the framework. Hay then says in this chapter that "While agreeing in general with the meaning of each row, this book respectfully takes a different view of the names of the rows and attempts to refine their definitions." While Hay's changes are not fundamental, they are changes nonetheless. And to reflect the changes properly, Hay renames the framework to be an "Architecture Framework."

A final example about the Zachman Framework is from the United States Federal Government's Chief Information Officer's council. The CIO Council started with the Zachman framework, they dropped a row, changed the names of the columns, and then redefined some of the contents of the cells. Notwithstanding these "small" changes, if you ask someone what the basis of the Federal's Enterprise Architecture Framework, they will state that they are using the Zachman Framework.

So the question then becomes, "Will the real Zachman Framework please stand up?" Are any one of the above cited frameworks more correct than the others? Is any one inherently wrong? Or is the Zachman framework simply a generalized organization strategy for the identification of and interrelationship among the artifacts of an information system? Both answers cannot be correct. Either it is a generalized strategy or it is a specific one. If it is specific then any deviation from it is not being faithful.

Since it seems quite acceptable to make modifications to the original Zachman framework and not be held up to ridicule or stake-burning, the first choice then seems more acceptable. That is, that the Zachman framework is merely a generalized strategy for organizing, planning and managing complex environments into a contextual set of interrelationship of artifacts. The major "things" become the columns, and the unfolding transformations become the rows. And, once complete, frameworks should have a simpleness to them and should be held up to a test of "common sense." The contents of the cells then are the artifacts that are created, evolved, and maintained as the overall intent of the framework is accomplished.

A bigger question then results. If a number of different teams within an enterprise all "do the Zachman" framework but have a different understanding as to the exact nature of the columns, rows, and cells, then how well can their work be integrated and exchanged. It would seem difficult then at best.

With all this as background, this paper proposes that there can be multiple classes of frameworks and that they can either be orthogonal or overlap, but that the entire set of artifacts from each must be well defined, and be able to be seamlessly interrelated into a single metadata repository. Each framework then becomes a rows-and-columns "window" into the underlying metadata repository. The cells of each framework identify one or more artifacts that would be captured into the overall product requirements, specification, design, and implementation.

Figure 1, for example contains three different frameworks. These could be for Hardware, Systems Software, Business Rules, Human processes, Organizations, or what ever. From this figure it is clear that the three frameworks, whatever they may be are not same design. One is 6x6, another

3x3, and a final one, 3x6. Regardless of their architecture, they do have one cell in common, and that this cell is I-C3, II-A6, and III-C2.

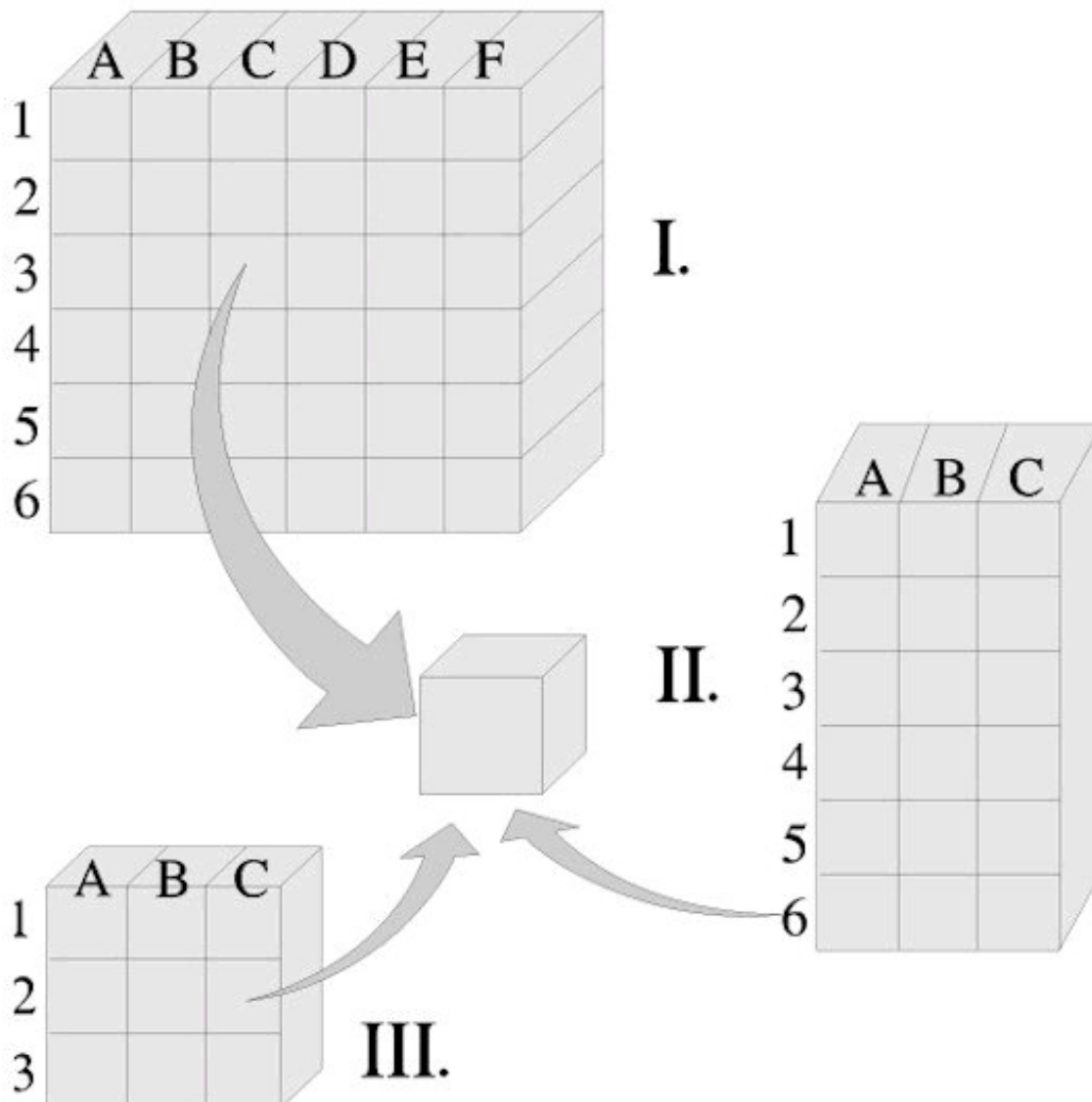
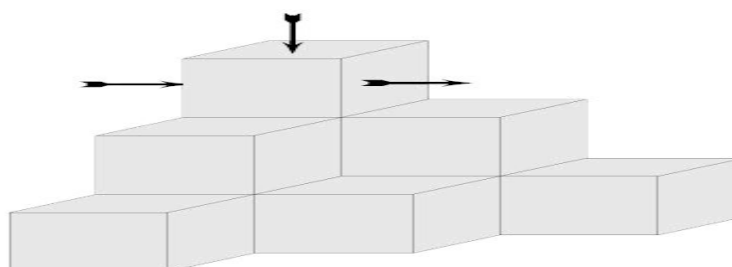


Figure 1. Three frameworks, their cells, and a cell common among all three.

But what is the cell? Is it simple, or complex. That is, does it represent just one artifact or multiple artifacts. Figure 2 illustrates that regardless of whether the cell is simple or complex, each cell really represents an object class. That is, each artifact in the cell has its own data structure, processes, inputs and outputs. In Idef-0 parlance, the notation is ICOM, that is, one or more inputs, one or more constraints, one or more outputs, and one or more mechanisms. And if the cell is complex, as illustrated in Figure 3, then each component of the cell too has its own set of ICOMs.



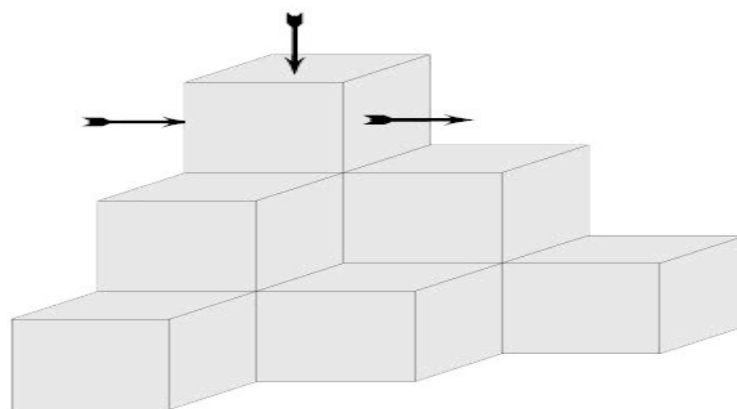


Figure 2. Complex cell containing nine distinct artifacts with their own set of ICOMs.

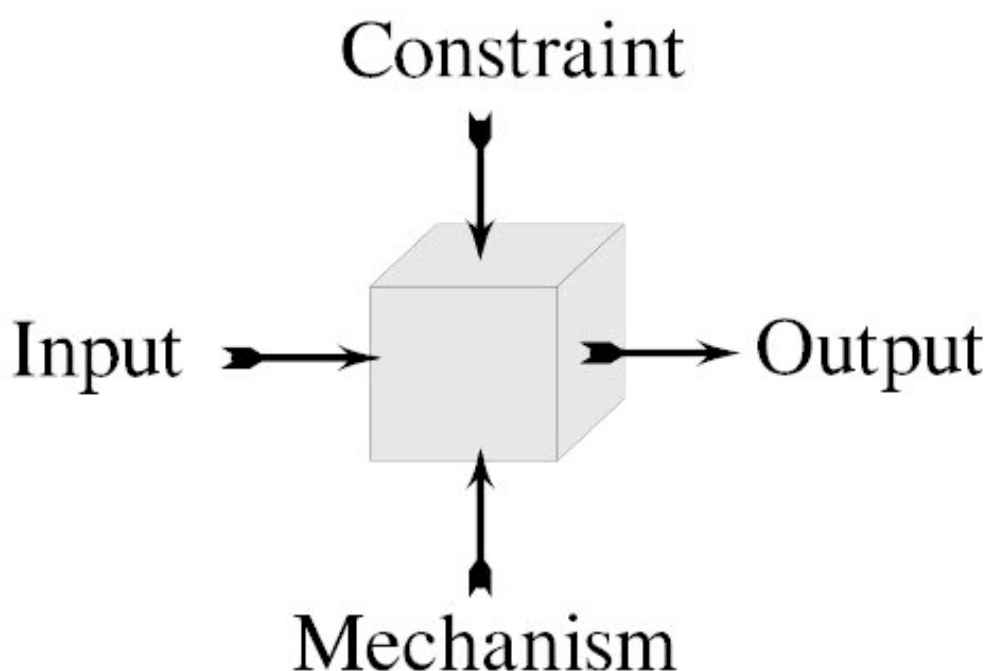


Figure 3. A given cell's artifact object class in terms of Idef-0 notation.

The final component of this is the construction is the metadata repository. Much has been written about these, and so it will not be repeated here, but if the three illustrated frameworks were hardware, systems software, and information systems, then maybe the common cell would be persistent data storage, and from the hardware point of view it would be disk drives and supporting disk networks, and from the systems software point of view it might be database management systems (DBMS) and access methods. And finally, from the information systems point of view it might be physical database designs.

Clearly, the sum total of all this metadata must be identified, collected, interrelated, and evolved as disk storage systems are related to DBMSs and access methods and are related to physical database designs. The metadata repository then serves as the integrator of these artifacts, and three different frameworks serve as the subject specific windows into their particular set of artifacts.

Figure 4 illustrates the relationship between some common cell from the three different frameworks and a small but integrated subset of meta entities from a metadata repository. It is useful to note that

the Form meta entities would properly derive from a framework column that involves human activities, while the View meta entities would derive from a framework column for information systems, while the DBMS meta entities from a physical schema model, while the schema-table-column meta entities would derive from a logical database framework column, and the 11179 meta entity would have been derived from a abstract business fact inventory framework object.

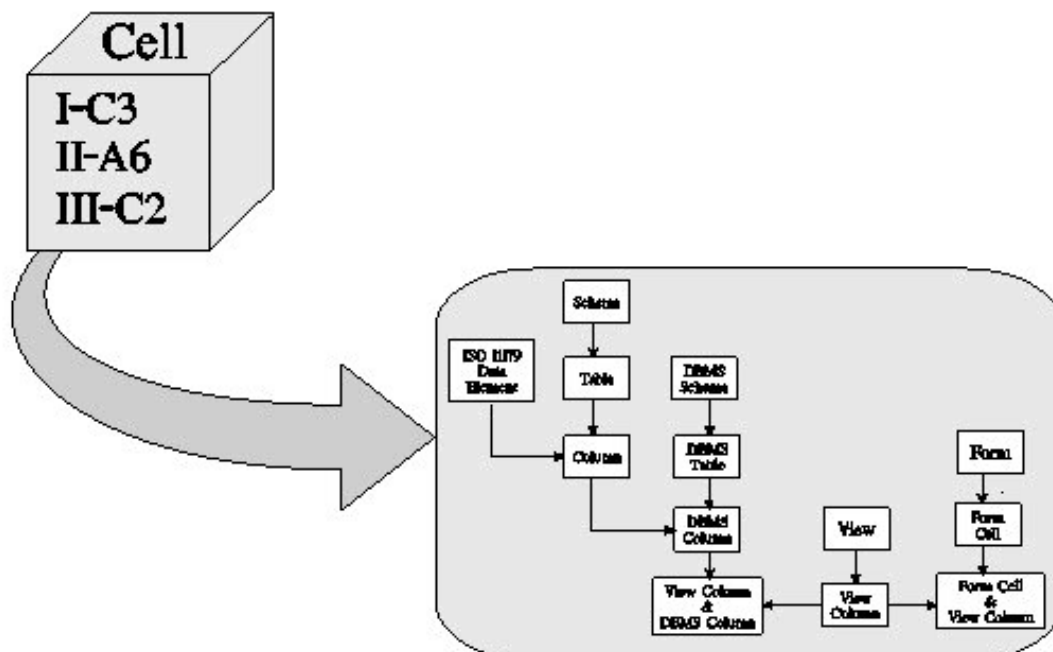


Figure 4. Interrelationship between a common framework-cell and a metadata repository meta entity set.

The point to all of this is that it is neither known nor does it really matter which framework is the progenitor of each meta entity set. It only matters that they are derived and that the source frameworks are well organized, designed, and truly support the effective planning and accomplishment of work. The same, of course can be said of the overall meta entity schema that then defines the metadata database. And the metadata database is, in turn, the super-type set for all data instances.

The various frameworks that have been created to accomplish the various components, for example, frameworks for hardware, systems software, application software, organizations, business scenarios and events, enterprise architectures, and of course Whitemarsh's pet framework, the Knowledge Worker Framework. These frameworks all contain cells, some common and some different. Each cell maps to one or more meta entities in the meta data repository schema, which in turn, form the schema for the database of the metadata repository. The metadata repository then is the persistent data component of the metadata repository system. Examples of that would be the Platinum Repository, and the Whitemarsh metabase. Another key use of the "cell" is that it infers the actual work methods of the knowledge worker. Given that each cell (or sub-cell within a complex are specified according to an IDEF-0 strategy then each cell or sub-cell has inputs, outputs, constraints and mechanisms. Simply put, each cell is a fully defined object class. Finally, there's the knowledge worker who is guided by the project's work plan and employs the metadata repository to identify, specify, design, implement, and maintain all the critical knowledge worker products essential for our environments.



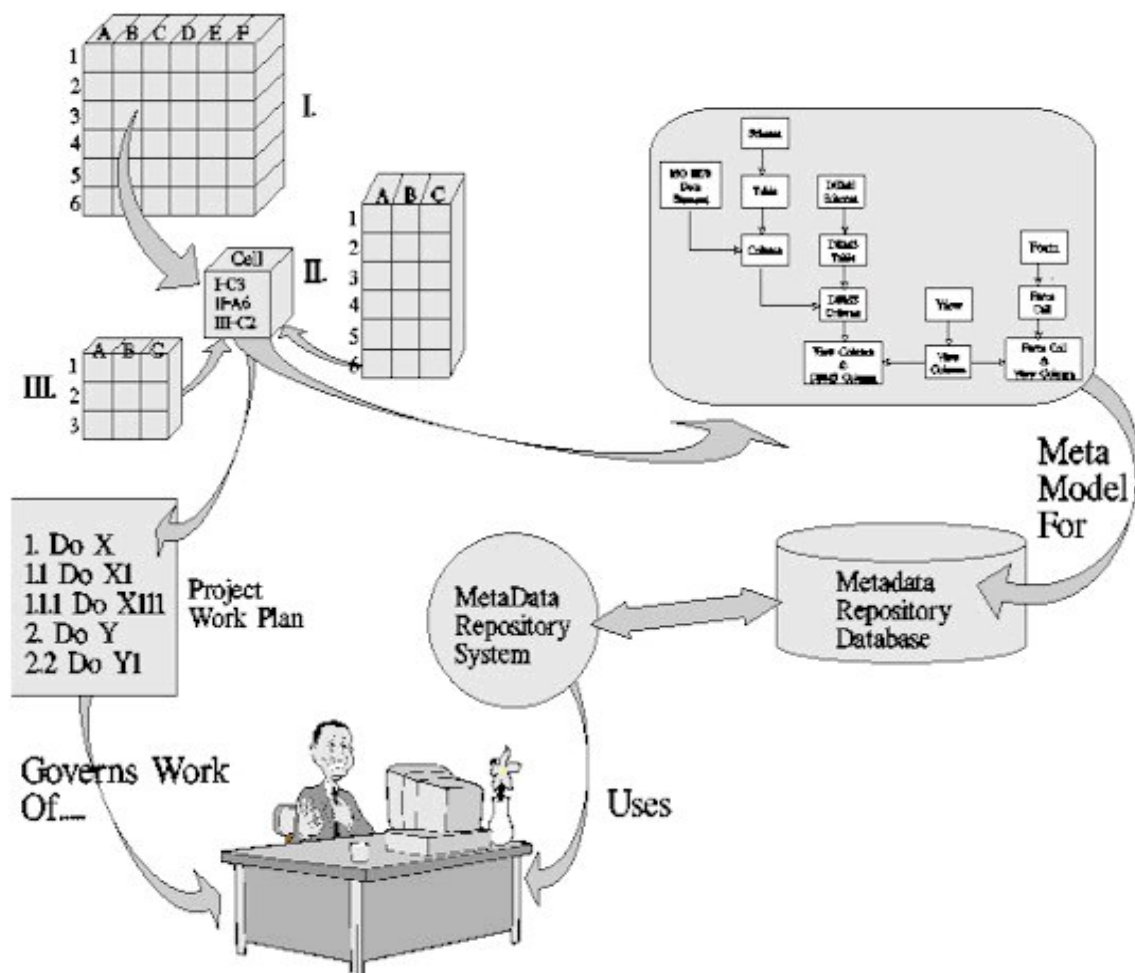


Figure 5 brings significant parts of all this together.

In summary then, and as John Zachman so often says, “Someday you are going to wish you had all those models, Enterprise-wide, horizontally and vertically integrated at an excruciating level of detail.” That “someday” probably occurred about 25 years ago when it first became possible to have a unified database of infrastructure work products. And that “someday you are going to wish” also probably occurred about 25 years ago when it became possible to have enterprise-wide IT systems and databases. Enterprises that acted on Zachman’s admonition often succeeded at their IT efforts, and those that do not almost always failed. The United States Government’s General Accounting Office’s library is a graveyard of failure tombstones (a.k.a., “GAO Reports”).

This paper suggests a completely closed loop environment for what John likely intended in his 1987 IBM Systems Journal paper on frameworks.

Michael M. Gorman, President of Whitmarsh Information Systems Corporation, has been involved in database and DBMS for almost 30 years. Mr. Gorman has been the Secretary of the ANSI Database Languages Committee, X3H2 for 19 years. X3H2 standardizes SQL. A full list of Whitmarsh’s clients and products can be found on the web site, www.wiscorp.com. The goal of the web site, WisWeb, is to make data management books, courses, methodologies, software, and metrics available to the database community through electronic publishing and downloading. WisWeb memberships are very reasonable and are designed for the individual, the ISD organization, universities/colleges, and professional training organizations.

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716-1518 USA
Phone: +1.301.249.1142
FAX: +1.301.249.8955
Email: mmgorman@wiscorp.com
WWWeb: <http://www.wiscorp.com>

[\[Return to the Article Archive\]](#)

The Data Administration Newsletter (TDAN.com)
Robert S. Seiner - Publisher - rseiner@tdan.com