a|EA JOURNAL

Case Study

The Zachman Framework Populated with Baseball Models

Terry Bahill, Rick Botta, and Jesse Daniels

ABSTRACT

Frameworks help people organize integrated models of their enterprises. This organization helps ensure interoperability of systems and helps control the cost of developing systems. The Zachman framework for enterprise architecture is a six by six classification schema, where the six rows represent different perspectives of the enterprise and the six columns illustrate different aspects. To ensure a complete and holistic understanding of the enterprise architecture, it is necessary to develop models that address the perspectives and aspects that constitute the rows and columns, respectively, of the framework. In this paper, a Zachman framework is populated with models for Baseball. These models should be easy to understand without a steep learning curve. Most of the cells in this example are filled with quantitative simulatable models that have been published in peer-reviewed journal papers. The other cells are filled with simple thought models. Jacques Barzun (1954) wrote, "Whoever wants to know the heart and mind of America had better learn baseball, the rules and realities of the game." From the perspective of the Zachman framework, the way to learn Baseball is to define the models within the framework, as presented in this paper.

KEYWORDS

enterprise architecture, systems engineering, systems analysis, science of baseball, modeling methodology, Zachman framework

FRAMEWORKS AND ORGANIZED MODELS

Frameworks help people organize and assess completeness of integrated models of their enterprises. Several popular frameworks have been used to architect enterprises, such as the Department of Defense Architecture Framework (DoDAF), the Federal Enterprise Architecture Framework (FEAF), the Treasury Enterprise Architecture Framework (TEAF), the ANSI/IEEE 1471 Standard (Maier, Emery & Hilliard, 2004) and the Zachman framework (1987). The Zachman framework, like many others, considers multiple perspectives of an enterprise. Unfortunately, we have few examples available in the public domain illustrating a complete set of models that comply with the Zachman framework. This article fills a complete framework for the game of Baseball. Use of the Baseball "enterprise" as an example allows us to use non-proprietary models that can be understood by a large number of people without a steep learning curve.

The concept of examining a system at many orders of magnitude was presented by Boeke (1957). Later Charles & Ray Eames (1968) created a film of this concept and finally Philip & Phylis Morrison (1977) popularized the idea with the book *Powers of Ten*. Rouse et al. (1992) added the other dimension (What, How and Why) to make it a matrix. However, all these authors treated the decomposition as more and more level of detail. Zachman made it different perspectives on the enterprise. He also added the other aspects of Who, When, and Where.

Frameworks provide an organizational schema by which enterprises can be developed and understood. However, they alone are not sufficient. A complete enterprise engineering development toolkit would be comprised of a framework, a process (such as the IBM Rational Unified Process or SIMILAR (Bahill & Gissing, 1998)), a method (Martin, 1997), a notation (such as UML) and tools (such as IBM Rational Rose, Enterprise Architect, or a mass spectrometer).

Today's businesses are complex. The systems that implement their business processes are equally complex. But people have a hard time agreeing on how to define complexity. One way to define complexity is as a measure of how well we can predict expected performance. The more complex a system is, the harder it is to predict its actual performance. We can deal with complexity and performance predictions using integrated models and simulations of a business. Frameworks can help organize these models into multiple levels of abstraction to better understand how the business rules, organizational strategies and resources are turned into a physical system. This paper has more to do with the concept of frameworks and integrated models for dealing with complexity than it does with the game of baseball. Then why should we model baseball using a (Zachman) Framework? When developing enterprise and system architectures, we can use frameworks to convey the concept of integrated models and simulations that is independent of any business domain: and baseball, which itself is a complex enterprise, is a business domain that most people already understand.

Different people use sets of models that comply with a framework for different reasons. High-level planners use them because they help ensure interoperability of systems and help control the cost of developing systems. Business people use them to help predict return on investment. Directors of Systems Engineering use them to better understand how the business rules, organizational strategies and resources are turned into a physical system. Low-level researchers, designers and engineers use them to understand their enterprises so that they can understand their organizations and the flow of money within and without.

We have found the Zachman framework to be useful for organizing the models of our business enterprise and of our customer. In this paper, we examine the Zachman framework, show what kinds of models go in each of its cells and look at its strengths and weaknesses. We want to help our readers to decide whether or not the Zachman framework would help them to organize the models in their businesses.

The structure of this paper is as follows. First we described the background and evolution of frameworks in general. In the next section, we will describe the Zachman framework in detail. Then we will populate a Zachman framework with models from Baseball. We hope that these models are easy to understand, without a steep learning curve. Then, we discuss lessons learned from researching this topic. Finally, the holistic nature of this project is presented in the Summary.

ZACHMAN FRAMEWORK

The Zachman framework is a normalized six by six classification schema for organizing descriptive representations of an enterprise. The rows represent different stakeholder perspectives of an enterprise, while the columns depict different areas of interest within those perspectives. The Zachman framework is simply a framework – it is not a process, a method, a notation or a tool. Because the framework is normalized, rows and columns cannot be added or omitted and still be called a Zachman framework. Everything fits into the Zachman-defined six by six schema. The forte of the Zachman framework is that it is a normalized schema; it provides an even coverage of important topics and does not have redundancy built into it. Each cell in the schema can be thought of as having two dimensions – scope (width) and level of detail (depth). Each cell in the schema contains at least one "primitive" model or artifact. A primitive model consists of information specific to a single column. Table 1 below shows an empty Zachman framework.

Name of Enterprise	1. What (data)	2. How (function)	3. Where (network)	4. Who (people)	5. When (time)	6. Why (motivation)
1. Scope (context)						
2. Business model (concept)						
3. System model (logical)						
4. Technology model (physical)						
5. Detailed representation (component)						
6. Real system						

Table 1. Empty Zachman Framework

The Rows

The following description of the Zachman framework rows is based on a Software Engineering Institute document (2006). The framework contains six rows and six columns yielding 36 unique cells as shown in Table 1. The rows represent different perspectives (views) and roles of the enterprise.

- **Scope** describes the vision, mission, context, boundaries, architecture and constraints of the enterprise. The scope states what the enterprise is to do. It is called a black box model, because we see only the inputs and outputs, not the inner workings. This row is also referred to as the Context row.
- **Business model** defines goals, strategies, structure and processes that are used to support the mission of the enterprise. This row is also referred to as the Concept row.
- **System model** contains system requirements, objects, activities and functions that implement the business model. The system model states how the system is to perform its functions. It is called a white box model, because we see its inner workings. This row is also referred to as the Logical row.
- **Technology model** considers the constraints of humans, tools, technology and materials. This row is also referred to as the Physical row.
- **Detailed representation** presents individual, independent components that can be allocated to contractors for implementation. This row is also referred to as the Out-of-Context row since models in this row are typically so detailed that they are essentially "out of context" with models in the upper rows of the framework.
- **Real system** depicts the operational system (or the sliver of that system) that is under consideration.

The rows do not represent a physical decomposition of the system. Nor do they portray finer and finer levels of detail. Rather the rows show different perspectives (or views) of the enterprise.

Table 2 below from the Zachman Institute web site (www.zifa.com) shows an example using common stakeholders as known from the building industry. The rows of the schema provide multiple perspectives of the overall architecture from the points of view of the planner (row 1), the home buyer (row 2), the designer or architect (row 3), the builder or general contractor (row 4) and subcontractors (row 5). The left column identifies the perspective and defines the role (or stakeholders) in the enterprise that would own that perspective. Some modelers ignore the perspective and focus on the roles. Others ignore the roles and focus on the perspective. In this paper, we will try to include both the perspective and the roles.

A House	Models	Nature or Purpose
1. Scope (Planner)	Bubble charts, i. e. rough sketches	Basic capabilities for the building Gross size, shape and relationships Architect-owner mutual understanding Initiate project
2. Business model (Home Owner)	Architect's drawings	Final building envisioned by the owner Floor plans, site plans Architect-owner agreement on the building Establish a contract
3. System model (Designer, Architect)	Architect's plans	Final building visualized by the designer Translation of owner's view into a product Detailed drawings in a dozen categories Basis for negotiation with general contractor
4. Technology model (Builder, General Contractor)	*Contractor's plans*	Final building as seen by the builder Architect's plans constrained by technology How to build descriptions Directs construction activities
5. Detailed representation (Subcontractor)	Electrical schematics, Plumbing blueprints, Communication network specifications	Subcontractors plans Detailed stand-alone model Specification of what is to be built Directs installation of specific items.
6. Real system (User, Maintainer)	House	Physical building

Table 2. Models for Building a House That Differ in Detail and Nature

Many roles are possible, such as Planner, Owner, Architect, Designer, Builder, Financier, Stockholder, Regulator, CEO, CIO, President, Director, Chief Systems Engineer, Program Manager, Information Systems Manager, Systems Engineering Team, Systems Analyst, Requirements Engineer, Use Case Engineer, User Interface Designer, Architect, Component Engineer, Test Engineer, System Integrator, Integration Tester, System Tester, Assembler, Technician, Machinist, User and Maintainer. The identified roles will be the primary stakeholders for that row (view).

For the Baseball example in this paper, the owner of (and the person most interested in) the scope models in row 1 will be an Executive, like the Commissioner of Baseball, or Major League Baseball Inc. The owner of the business models in row 2 will be an Owner of a baseball team. The owner of the system models in row 3 will be a General Manager of a baseball team. The owner of the technology models in row 4 will be a Team Manager. The owner of the detailed models in row 5 will be the Scientist, Engineer or Coach who created each model. And finally, the owner of the real system in row 6 will be an individual major-league Baseball Player. The owner should be the role that controls the flow of money in that row.

The Columns

"I Keep six honest serving men
(They taught me all I knew):
Their names are What and Why and When
And How and Where and Who."

From *The Elephant's Child*
by Rudyard Kipling (1902).

In journalism class, we were taught to start every news story with who, what, when, where, why and sometimes how. This is also good advice for understanding a system. The columns of a Zachman framework present these aspects of the enterprise.

- **What (data)** describes the entities that are considered important to the business, as viewed from each perspective. These entities are the things for which information is to be maintained. Examples include equipment, business objects and system data.
- **How (functions)** defines the functions, processes, operations and activities that the enterprise is concerned about relative to each perspective. Inputs and outputs are also considered in this column.
- **Where (networks)** shows geographical locations and interconnections between activities within the enterprise. This includes major business geographical locations, networks and the playing field.
- **Who (people)** represents the people within the enterprise and information for assessing their capabilities and performance. The design of the enterprise has to do with the allocation of work and the structure of authority and responsibility. This column also deals with human-machine interfaces and relationships between people and the work they perform.
- **When (time)** represents time, or the event relationships that establish performance criteria. This is useful for designing schedules, the processing architecture, the control architecture and timing systems.
- **Why (motivation)** describes the motivations of the people and the enterprise. This reveals the enterprise goals, objectives, business plan, knowledge architecture, and reasons behind making decisions and taking actions. It is concerned with how the goals and strategies are translated into specific ends and means.

Horizontal and Vertical Integration

The models organized by row and column in the Zachman framework should be horizontally and vertically integrated. This means that you should not work models in a given cell without considering impacts to other cells in the same row and in the same column. As an example of horizontal integration, consider a functional model developed in the How column. The inputs/outputs, resources, deployment, time constraints, and goals related to each function should be considered in the What, Who, Where, When, and Why columns, respectively, in the same row. As an example of vertical integration, consider a requirement in the Physical (Technology) row, Why column for the performance of some aspect of a software component. This requirement should be linked to other motivational models in higher rows, such as an objective requirements document in the Concept row, Why column. This is nothing more than requirements traceability and is a common best practice for systems engineering. Through this traceability, we call the requirement in the Physical row vertically integrated with the requirements in the Concept row. These impacts and traceabilities can be modeled with dependency relationships.

This means that no cell is an island. Each cell is fundamentally related with other cells in the same row and same column. From a practical standpoint, information in multiple columns is typically discovered simultaneously while constructing models in a given row. It is suggested that this information be captured in models in the appropriate cell.

Each row (view) has at least one role (stakeholder). That stakeholder's concerns would be the primary focus of the Why cell in that row.

BASEBALL EXAMPLE

Now we will take models from Baseball and show where they fit within the context of the Zachman framework. When possible, we selected models that have been published in peer-reviewed journals. Although in Tables 1 to 4 we print the cells from top to bottom, we think, for the purpose of this paper, they are best read and discussed from bottom to top.

Column 1, What (data). The physical product (data item) depicted in column 1 is the baseball bat; *column 1 (what), row 6 (real system)*.

There are many models for a baseball bat that explain the Center of Percussion (CoP), moment of inertia (MoI), coefficient of restitution (CoR), etc. (Adair, 1994; Cross, 1998; Nathan, 2000 and 2003; Sawicki, Hubbard & Stronge, 2003; Bahill, 2004). From the perspective of a bat manufacturer, this detailed representation of the bat can be represented as a model depicting the required length, the taper of the handle, the width of the barrel, the bat weight, etc. This model maps to column *1 (what), row 5 (detailed representation).*

The swing of a bat can be modeled with a translation and two rotations, one about the batter's spine and the other between the two hands (Brancazio, 1987; Watts & Bahill, 2000); *column 1 (what), row 5 (detailed representation).*

There is an ideal bat weight and a best weight distribution for each batter (Bahill & Karnavas, 1989 and 1991; Bahill & Morna Freitas, 1995; Bahill, 2004). The team helps the individual select and acquire the right bat; *column 1 (what), row 4 (technology model)*.

Each organization provides facilities for batting practice, conditioning and skills development; *column 1 (what), row 3 (system model)*.

The National Collegiate Athletic Association (NCAA) controls college sports. In this role, it has created rules governing the allowed dimensions and performance of aluminum bats. For example, the bat shall not weigh less (in ounces) than its length (in inches) (Crisco, 1997: Nathan, 2003). Financial models could be in every column. Those appropriate for column 1 include cost of equipment such as the bats, cost of training facilities and cost of physical conditioning equipment; *column 1 (what), row 2 (business model)*.

The creator of baseball, until recently believed to be Abner Doubleday, would have listed rules, bats, balls, players, and fields among the list of things important to the game. The rules of ball-and-stick games (baseball, softball, cricket, tennis, etc.) are written to challenge the physiological limits of the human in many dimensions (Regan, 1992). The bat is regulated to make the game exciting, but traditional; *column 1 (what), row 1 (scope)*. By considering other ball-and-stick games, we are testing and defining the scope of our chosen enterprise.

We do not have enough room in this paper to cover all of Baseball. That would take thousands of models. Therefore, we only show slivers of Baseball. For example, in column 1, we only looked at the baseball bat. In contrast, we could have looked at the ball, or the bat and the ball. We used a different sliver for each column, but even at that, we did not restrict our models to only that one sliver. For example, sometimes we talk about major league baseball and sometimes we talk about university baseball governed by the National Collegiate Athletic Association (NCAA).

The sliver used in column 1 (what) was the physical baseball bat. An alternate sliver for column 1 is information. Multiple TV cameras in major league stadiums pick up the flight of the pitch. These TV signals can be used to construct a computer model for the flight of the ball. These data can be used for many purposes. When they are used by the TV networks to display to the TV audience the location of the pitch relative to the strike zone, then they are being used as a Detailed Representation: *row 5, column 1, role TV audience.* These data could be used to determine if the ball passed through the strike zone and this Technology Model information could be transmitted in real time to the umpire to help him call balls and strikes, this would be *row 4, column 1, role umpire*. These data could also be put on the Internet for researchers to use to help determine the speed and spin of the ball, to allow them to model the movement of the ball; this would be *row 4, column 1, role researcher*. When this information is put on a CD and given to the umpire at the end of the game to give him feedback to improve the consistency of the set of umpires, then it is being used in a System Model (www.QuesTec.com): *row 3 column 1, role umpires*. When knowledge about the difference in the strike zones of American and National League umpires is used to regulate enforcement of baseball rules, then it is being used in the Business Model: *row 2, column 1, role commissioner and owners*. If such information were gathered and analyzed for cricket and tennis, then the derived wisdom would transcend baseball and become *row 1, column 1.*

Table 3 below shows a whole Zachman Framework for Baseball. We then describe the models cell-by-cell.

Baseball Model	1. What	2. How	3. Where	4. Who	5. When	6. Why
1. Scope (Commissioner of Baseball)	Equipment rules	Rules of baseball	Stadiums & TV, Leagues & divisions	Revenue sharing, Salary cap, Olympic teams	Chronology	Entertainment, Intellectual stimulation
2. Business Model (Team Owner)	NCAA bat rules	Define the strike zone	Shared use of stadiums	Alex Rodriguez' contract	Season schedules	Money, Power, Pride, Free bats
3. System Model (General Manager)	Batting practice facilities	Pitch tracker, Stadium instant replay	Placement of home plate	Fantasy baseball, Player contracts, 25-man roster	Pitching rotations	Wants a winning season
4. Technology Model (Team Manager)	Ideal bat weight & weight distribution	Teamwork & signals for hit & run, bunt	Mental rehearsal & understandings	Weekly statistics, Box scores, Sweet spot	Pitch count	Intentional walks, The blame game
5. Detailed Representation (Scientist, Engineer, Coach)	A swing, CoP, MoI, CoR	Rising fastball, Eye movement strategies, Speed & spin	Predict where & when, Fielders run curved paths	Players' physiological state, The count	Mental models, Work fast & change speeds	CFFF, Expect fastball with 3-0 count
6. Real system (Baseball Player)	Baseball Bat	One pitch & responses	Baseball Field	Major league baseball players	Pitch interval	Motivation for decisions

Table 3. A Zachman Framework Populated with Baseball Models

Column 2, How (function). The activity modeled in column 2 is one pitch and people's response to it; *column 2 (how), row 6 (real system).*

Once the ball is in the air, the movement of the pitch depends only on velocity, spin rate and spin axis (Watts & Bahill, 2000; Bahill & Baldwin, 2004; Bahill & Baldwin, 2006); *column 2 (how), row 5 (detailed representation).*

Two strategies are used by the batter for tracking the pitch using the saccadic and smooth pursuit eye movement systems (Bahill & LaRitz, 1984; McHugh & Bahill, 1985). A neurophysiologic model shows how the batter predicts where and when the ball will cross the plate (Bahill and. Karnavas, 1993; Bahill & Baldwin, 2004). The batter uses the spin of the ball and other visual clues to predict the ball's motion (Bahill, Baldwin & Venkateswaran, 2005;

Baldwin & Baldwin, 2006). Underestimating the pitch speed can induce the perceptual illusion of the rising fastball (Karnavas, Bahill & Regan, 1990; Bahill and. Karnavas, 1993); *column 2 (how), row 5 (detailed representation)*.

Teamwork and signals enable the manager, the batter and the runners to execute tactics such as hit and run, bunt, steal, take the pitch, swing away, etc.; *column 2 (how), row 4 (technology model).*

The pitcher pitches the ball. The batter swings and hits the ball. He runs toward first base, etc. Our activity diagram of Figure 1 shows one pitch and subsequent activities. It assumes a groundball hit into fair territory; the fielder on first base catches the throw; there are no other base runners; *column 2 (how), row 4 (technology model).*

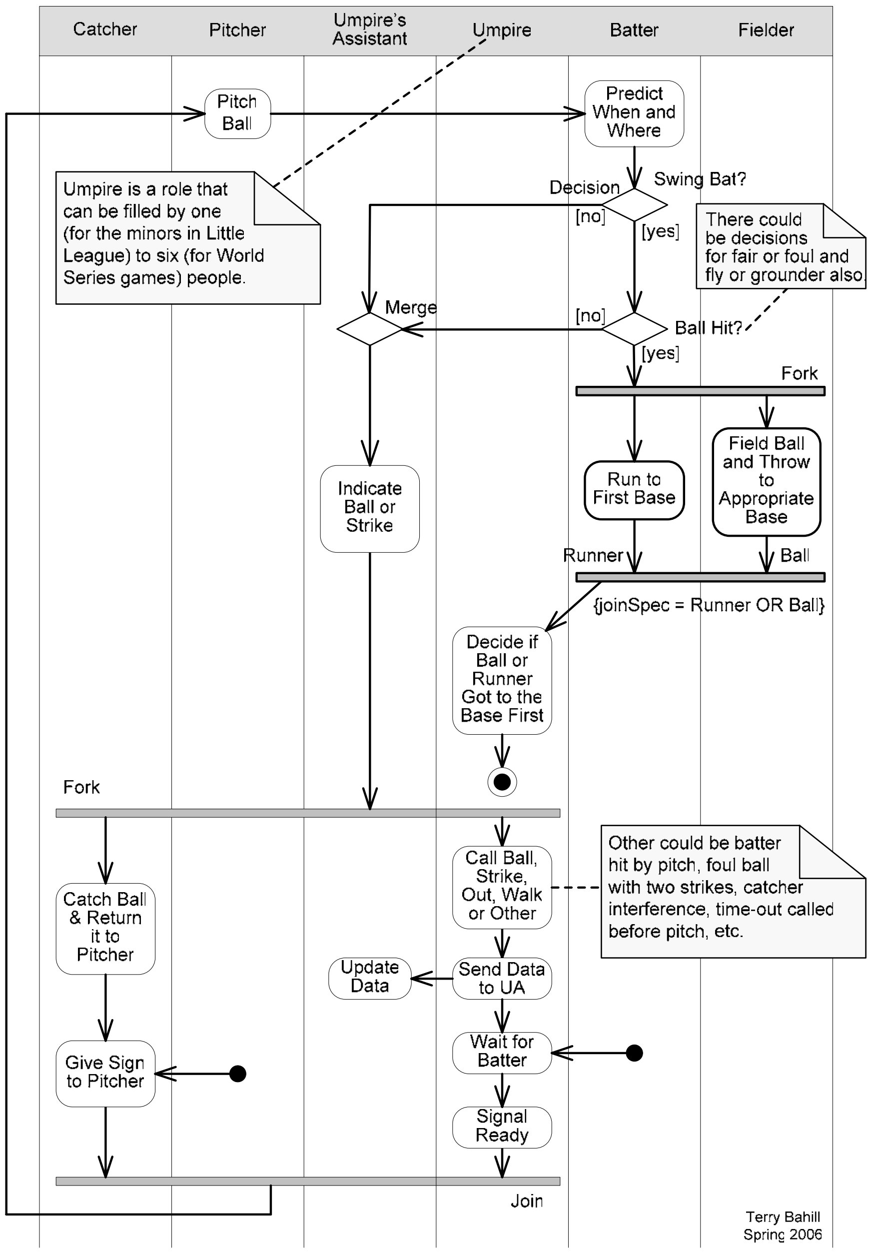

Figure 1. Activity Diagram for a Baseball Pitch and Subsequent Activity.

Stadiums can be equipped with a variety of optional equipment that can record and playback the pitch, such as the multiple television cameras used to aid the umpires (www.QuesTec.com) or entertain the TV audience (http://www.gueziec.org/kzone.html) and the stadium instant replay screens for the benefit of the players and spectators; *column 2 (how), row 3 (system model).*

Armstrong (1998) applied the Capability Maturity Model to improve the performance of his softball team; *column 2 (how), row 3 (system model).*

Major League Baseball Inc. defines the strike zone and manages umpires; *column 2 (how), row 2 (business model).*

The rules of baseball evolved over its first 50 years, but have been relatively constant over the last century (Gould, 2003), but that may change with the possible introduction of systems to aid the umpire in calling balls and strikes (Bahill, 2006) and with the introduction of instant replay; *column 2 (how), row 1 (scope).*

Column 3, Where (network). The sliver for column 3 is the baseball field; *column 3 (where), row 6 (real system).*

The human brain does not have x, y and z coordinates of objects. Humans must track objects using neurophysiological parameters. As a result outfielders run a curved path when tracking down fly balls (McBeath, Shaffer & Kaiser, 1995; Shaffer & McBeath, 2002); *column 3 (where), row 5 (detailed representation)*

Batters must predict where and when the ball will cross the plate (Karnavas, Bahill & Regan, 1990; Bahill & Baldwin, 2004); *column 3 (where), row 5 (detailed representation)*.

Before every pitch all fielders mentally rehearse where they will throw the ball if they receive a ground ball or a fly ball and they establish understandings with nearby fielders about where each player will go; *column 3 (where), row 4 (technology model)*

The placement in the stadium of home plate affects the area behind the plate, the design of protective netting, the orientation to the sun, the distance to the fences and therefore safety and playing performance; *column 3 (where), row 3 (system model)*.

Stadiums can be designed for baseball only or they may be shared by baseball, football and other events; *column 3 (where), row 2 (business model).*

Baseball is played in stadiums and broadcast on television; *column 3 (where), row 1 (scope).*
The teams are organized into leagues and divisions according to geography; *column 3 (where), row 1 (scope).*

Column 4, Who (people). Column 4 contains information about major-league baseball players. It should be possible for all this information about baseball players to be contained in one system, with the information rolled up from the bottom to the top; *column 4 (who), row 6 (real system)*.

The physiological state of individual players determines whether and how well they play an individual game, or if they are put on the disabled list. Some of this information is publicly available, most is not; *column 4 (who), row 5 (detailed representation)*.

The count is the model for how the batter is doing during an at-bat. Pitch by pitch information is usually recorded by the teams, but it is not generally available, although many people are trying to get access to this information; *column 4 (who), row 5 (detailed representation).*

Defining and locating the sweet spot of the bat is a human-machine interface problem: the teams help individuals understand this issue (Bahill, 2004). The sweet spot of the bat is four inches wide, but only one-third of an inch high (Baldwin & Bahill, 2004). This was determined using a new performance index of getting a hit, rather than the old performance index of getting a home run; *column 4 (who), row 4 (technology model)*.

Earlier, we discussed horizontal and vertical integration of models. The two previous paragraphs offer an opportunity to explore this for the baseball models. In basic research, one research group often uses the results of another group, but they seldom use the actual products. An exception to this was the model of Baldwin & Bahill, 2004. This model used new programs as well as the equations and the actual programs of Nathan, 2003; Sawicki, Hubbard & Stronge, 2003; and Bahill & Karnavas, 1993. Because the models were not designed to be used by others, we encountered the following interoperability problems: SI versus English units, right-handed versus left-handed coordinate systems, bat mass versus effective bat mass, different nomenclature, and most seriously lack of stated assumptions

The official score report for each baseball game contains on the order of 600 pieces of information (Major League Baseball Rules, 2005). Data for individual player performances are published daily in the box scores in the sports sections of newspapers. Player average performances are published weekly; *column 4 (who), row 4 (technology model).*

Individual player statistics, Markov models and manager decisions (such as batting order) are used to simulate games and seasons: this is called fantasy baseball (Burkiet, Harold & Palacios, 1997; www.stats.com). Scouts make observations and evaluations of players' performances and report this information back to their organizations; *column 4 (who), row 3 (system model)*.

The General Manager (GM) creates the 25-man roster and trades players to improve it. It must contain a balance of players at each position, including short relievers, long relievers, etc. The GM must consider player positions, salaries, performance, etc; *column 4 (who), row 3 (system model)*.

Billy Beane, General Manager of the Oakland Athletics, evaluates the worth of players with an innovative information system not reputations, and he has the most successful low-salary team in the major leagues. His system was based on the work of others. James collected data and evaluated hitting performance. AVM Systems collected data and evaluated fielding performance. McCracken collected data and evaluated pitching performance. Beane put all this information together. His goal was to win games. To win games you must produce runs. To produce runs you must get on base. Using this information system he drafted low-budget valuable players very successfully (Lewis, 2003); *column 4 (who), row 3 (system model)*.

George Steinbrenner, owner of the New York Yankees, evaluates the worth of his players traditionally, and he has the most successful high-salary team in the major leagues. The team owner must consider return on investment; *column 4 (who), row 2 (business model).*

Before the 2001 season began, the owners of the Texas Rangers created a ten-year $252 million contract for Alex Rodriguez (A-Rod). Their information system included projected team revenue, team performance and franchise value as well as salary, bonuses, taxes and contract insurance. Applying an 8% discount rate to the revenues and expenses, they obtained a net present value of negative $1.3 million, which is close to break-even. So they signed the contract (Cohen & Wallace, 2003; Cohen, 2003); *column 4 (who), row 2 (business model).*

Creating rules for selecting members of the Olympic baseball team effects the enterprise; *column 4 (who), row 1 (scope).* Because row 1 examines the system boundaries, it will often contain models that are outside of the organization.

The Commissioner of Baseball coordinates the teams; the Major League Baseball Players Association and players' agents orchestrate the activities of the players. The Commissioner must consider salary caps, retirement plans, drug testing, revenue sharing salary arbitration and free agency; *column 4 (who), row 1 (scope).*

Column 5, When (time). The fundamental unit of time in a baseball game is one pitch; *column 5 (when), row 6 (real system)*.

The batter's mental model for the pitch is based on the last one or two pitches or perhaps on the last 20 to 30 seconds (Bahill & Baldwin, 2004; Gray, 2002 and 2003); *column 5 (when), row 5 (detailed representation).*

A successful tactic of pitchers is "work fast and change speed on every pitch;" (Bahill & Baldwin, 2004); *column 5 (when), row 5 (detailed representation)*.

Pitch count - pitchers are often removed after, say, 120 pitches; *column 5 (when), row 4 (technology model).*

Sometimes pitching rotations are planned, but sometimes they are merely a default, e.g., "Spahn and Sain and pray for rain;" *column 5 (when), row 3 (system model)*

Television networks determine the starting times of many games; *column 5 (when), row 3 (system model).*

Season schedules for all of the teams are complex because of the many constraints (http://mat.gsia.cmu.edu/sports); *column 5 (when), row 2 (business model)*

Decisions must be made about interleague play, playoff structure, expansion teams, etc.; *column 5 (when), row 1 (scope).*

Column 6, Why (motivation). Column 6 concerns why people think the things they do and make the decisions they do; *column 6 (why), row 6 (real system)*.

Why does the pitcher decide to throw a fastball, a slider, a curveball or a changeup, *column 6 (why), row 5 (detailed representation).*

Critical flicker fusion frequency (CFFF) explains why pitchers think there is a difference between the two-seam and the four-seam fastballs although physics shows no difference (Bahill, Baldwin & Venkateswaran, 2005; Bahill, 2005); *column 6 (why), row 5 (detailed representation).*

Batters use many heuristics to decide what to do. Among them is, with a 3-0 count expect a fastball because the pitcher will have the greatest confidence in throwing it for a strike (Williams & Underwood, 1982; Bahill & Baldwin, 2004); *column 6 (why), row 5 (detailed representation)*.

The manager motivates his players by knowing when blame and when not to (Baldwin, 2001); *column 6 (why), row 4 (technical model).*

Why does a manager decide to pitch to a famous slugger rather than intentionally walk him? In making this decision the manager considers the score, runners on base, batting average, slugging average, etc. (Reiter, 2004)*; column 6 (why), row 4 (technical model).*

General Managers trade players in order to have a winning season within their constraints. They are motivated by their drive for success*; column 6 (why), row 3 (system model).*

What motivates major league baseball players? Money, prestige and pride. The top players can also get money from endorsements of clothing, equipment, video games, videos and books; *column 6 (motivation), row 2 (business model)*

What motivates baseball team owners? Power, ego, money; *column 6 (motivation), row 2 (business model)*

Easton Sports and Hillerich & Bradsby Company give wooden bats to major league players for free. Why? To build their brand image, so that they can sell more of their regular sports equipment; *column 6 (why), row 2 (business model)*.

The purpose of baseball is entertainment with two major subdivisions: television and baseball stadiums; *column 6 (why), row 1 (scope).*

Models in row 1 often test the scope of the enterprise. Angus (2006) uses baseball models to teach general management techniques; *column 6 (why), row 1 (scope)*.

Gould (2003) explains the intellectual complexity that causes sagacious Americans to be fascinated with baseball; *column 6 (why), row 1 (scope).*

LESSONS LEARNED

Level of Detail

Increasing detail. For practical applications, more detail is typically introduced in the lower rows of the framework than the upper rows, especially if a real system is to be developed and deployed. This is not a hard and fast rule and Zachman would argue that the level of detail is decided within a cell and not across rows; however, we have found it to be generally true in practice. We have observed that the number and size of the models in each cell generally increases from top to bottom. Thus, a row 5 cell could contain on the order of 100 times the mass (money, effort, pages of documentation, lines of code, number of diagrams, etc.) of a row 1 cell. Our tables do not show this mushrooming, because we concentrated on individual slivers of the problem. In contrast, instead of focusing on the bat in Table 3, the lower rows of column 1 could have included other equipment, such as balls, gloves, bases, masks, and the scoreboard.

They could have included system state, which is composed of total runs scored by the visiting and home teams, inning, whether it is the top or the bottom of the inning, last person who batted for each team, outs in this half of the present inning, balls and strikes on present batter, the names of the runners on first, second and third base, and a list of players who have been removed from the game: the system state contains the things that would be needed to restart a rain-delayed game. In addition, the lower rows of column 1 could have contained models for the pitcher such as the grip, the delivery (overhand, three-quarters, side arm), and wrist snap. Furthermore, they could have included models for teamwork such as signals (catcher-pitcher, manager-coach-batter, fielder-fielder) for hit and run, bunt, steal and executing double plays. And the lower rows of column 1 could also have included models for individual finances, batters' physiology, the minor league structure, TV time outs, weather, altitude, etc.

Morganwalp & Sage (2003) have handled this increasing detail by creating a three-dimensional Enterprise Architecture Framework (3-D EAF). Their rows and columns are similar to the Zachman framework, but they add another dimension to handle physical decomposition. This dimension contains Enterprise, Family of Systems, Systems, Subsystems, Components and Parts. They say that the number of models increases exponentially in this dimension.

Abstraction. Moves in the direction of decreasing detail (Bahill, Botta & Smith, 2005). In 1945, Pablo Picasso made a wash drawing of a bull. A month later, he finished his 11th state of abstraction, which was composed of six lines. Célestin said, "When you stand before his eleventh bull, it's hard to imagine the work that went into it."

Hierarchy. A Zachman framework is hierarchical: each cell could contain a framework of its own. For example, we could populate a framework for Baseball. Or we could populate a framework for the major leagues. Or we could populate a framework for the Arizona Diamondbacks. Or we could populate a framework for Randy Johnson, etc.

Granularity. The models in a row of a Zachman framework should be of equal granularity. Granularity refers to the level of detail or the level of abstraction of each model. The analogy is to rocks. Tiny rocks are called sand, small rocks are called gravel, larger rocks are called pebbles, even larger ones are river stones and big ones are called boulders. A driveway is often paved with gravel, where each rock is of the same size. Cockburn (2001) says that you raise the level by asking, "Why is the actor doing this?' and you lower the level by asking, "How is the actor going to do this?'

The following are the models in Table 3 row 6: What, the baseball bat; How, one pitch and responses; Where, the baseball field; Who: major league baseball players; Time: the pitch interval; and Why, motivation for decisions. These six things are at equivalent levels of detail. Examples of bad granularity would be a time scale based on (1) the orbital period of a 3d electron of an atom in the cork center or (2) the earth's orbital period.

The following are the models in Table 3 row 1: What, equipment rules; How, Baseball rules; Where, baseball is played in stadiums and broadcast on television; Who, Commissioner of Baseball coordinates the teams, Major League Baseball Players Association and players' agents orchestrate players' activities; When, chronology; decisions about interleague play, playoff structure, world series dates; and Why, the purpose of baseball is entertainment. These models are also at equivalent levels of abstraction. An example of bad granularity for row 1 of this framework would be using a Little Leaguer or the president of the U.S. for Who.

To help with horizontal granularity, you could consider time scales as well as perspectives and roles. In this Baseball example, the models in row 1 should run with time scales of a few years. The models in row 2 should deal with seasonal events. The models in row 3 should have time scales on the order of one game. The models in row 4 should have time scales of one at-bat. And the models in row 5 should have time scales of one swing.

It might be desirable to have the same real system in each column. In Table 3, we did not have room to cover all of Baseball: we only showed slivers of Baseball. For example, in column 1, we only looked at the baseball bat and this produced the rules about baseball bats in row 1. In contrast, we could have looked at the ball, or the bat and the ball. If we had looked at all possible slivers, then row 1 would have contained all baseball rules. Row 6 had even granularity, but each column looked at a different thing, e. g. column 1 had the bat, column 3 had the field and column 4 had the baseball players. For a more uniform framework we could have looked at What, the bat and ball; How, the pitch and collision; Where, the pitching rubber, the flight path, the plate, and the batter's box; Who, the batter; When, the duration of the pitch and swing; and Why, to outwit the opponent.

Examining the granularity of a framework is a way of validating the framework.

Modeling Generalities.
When architecting an enterprise, either generate models that address all of the cells, or document rationale for why certain cells were left out. This best practice ensures that there is complete coverage and that the modelers have at least considered each and every aspect and perspective of the problem in the context of the framework. In other words, this helps ensure that nothing has been inadvertently ignored.

Here are two additional generalizations about modeling. All elements in the same model should be at the same level. Models should exchange inputs and outputs only with other models of the same level, or maybe one level higher or lower.

Consider the Batter in the activity diagram of Figure 1. We could model the state of his mind with the following attributes and states:

experience: rookie, veteran, imminent free agent.

salary: considered too low, considered too high, commensurate with earned respect.

physiology: age, health, on disabled list.

competition: other players at his position.

Does this model fit with the activity diagram? No, because it is at a different level. If the Batter considered these factors while hitting, he may introduce confusion into his batting decision process.
Now let us reconsider the Batter in the activity diagram. We could model the state of his mind with the following attributes and states:

count (balls and strikes) and **outs.**

mental models: speed of last pitch, umpire's last call.

situation: runners on base.

signals: last signal from coach.

Does this model fit with the activity diagram? Yes, because it is at the same level. A batter considering these attributes and states is focusing on the level of model needed to increase his hitting performance.

Figure 2 below contains two class diagrams for the Batter of the activity diagram provided in Figure 1. The three compartments contain the name, attributes and functions of the class. The class diagram in the left column is at too high a level to fit with the activity diagram of Figure 1. The class diagram in the right column is at the correct level to fit with the activity diagram that was shown in Figure 1.

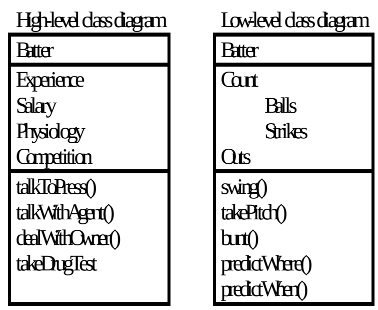

Figure 2. Class Diagrams for a Batter

A model with elements at different levels of detail will be hard to understand. In Bahill's experience, the most common student mistake in modeling is creating elements at different levels, for example writing a use case at one level and a creating a class diagram at another level.

All components in a model should be at the same level, but models can be broken into sub-models that are arranged hierarchically in levels. Models should only exchange inputs and outputs with other models of the same level, or maybe one level higher or lower (Bahill, Botta & Smith, 2005). In Fig. 3, models 0, 1 and 2 are at one level. They exchange information with each other. Model 1 has been decomposed into submodels 1.1, 1.2 and 1.3. These submodels exchange information with each other, with model 1 in the level above and with models 1.3.1, 1.3.2 and 1.3.3 in the level below. However, as indicated in Figure 3. Models should not skip levels in exchanging information. The structure and commentary on Figure 3 can also be applied to system design and organizational charts.

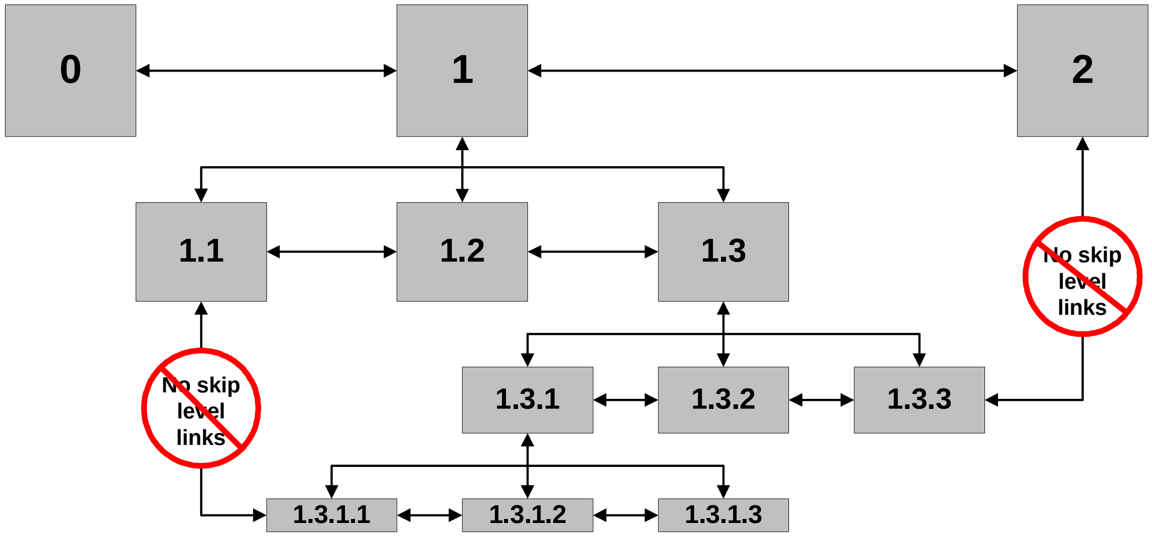

Figure 3. Example of Components and Decomposition

Linkages.

In going from the bottom to the top, row 5 models individual components; row 4 models the interfaces between these components and the resulting subsystems of row 4; and row 3 models the interfaces between the subsystems of row 4 and the resulting systems of row 3.

Top down or bottom up? In this paper, we discussed the models from bottom to top. That is the way many of these models were derived, because in basic research, new models are built on previous research. However, in designing new systems or modeling existing systems, better results will usually be obtained with a top-down approach. Architects and Systems Engineers start at the top and work down. Whereas discipline engineers and physicists usually start at the bottom and work up. Software engineers often start in the middle with use cases, objects and class diagrams. People usually start at the level that they are most familiar with. Unfortunately, this means different people may be working at different levels. Therefore, not everyone will be designing to the same requirements.

Methodology. A process or methodology should be followed when generating models according to a framework. The framework by itself is simply not enough for practical implementation. The framework can and should be used to tailor and structure a process so that the process prescribes that models be built to sufficiently address each aspect and perspective in the framework. Process also can enforce vertical and horizontal integration and consistency among models.

Using the Framework as an Assessment Tool

In addition to organizing the development of new architecture products, the Zachman framework is also useful as a general assessment tool. The framework, as a normalized schema for organizing a complete and holistic set of architecture descriptions can be effectively applied to assess existing artifact sets, or even determine an optimal skills mix for architecture development. When first embarking on an architecture development effort, it is useful to gather existing models and documentation about the subject of the architecture and arrange them into appropriate cells within the framework. This gives the architecture team an idea of the extent to which existing models, graphics, documents, etc., provide coverage in terms of a holistic description of the architecture. The results of this mapping of existing information into the framework can be depicted graphically and can be used to identify gaps in information and can further be used to focus artifact development efforts to fill the gaps. We have found that stakeholders of models in the top rows typically did not communicate with stakeholders of the bottom rows. Using a framework would help ameliorate this mistake.

Another example of using the framework to assess existing information is in the analysis of a set of customer requirements. Each requirement in the requirements set can be mapped to a particular cell within the framework. After mapping the requirements to the appropriate cell, the results can again be used to identify gaps in requirements definition. This information can be useful for prompting and guiding discussions with the customer to produce a more refined set of requirements for a system. In some cases, a given requirement may span multiple columns. This indicates that the requirement is compound and would be made more manageable through decomposition. The framework can also be used to determine the optimal skills mix for a company or architecting team. Mapping individual team member's skills against the Zachman schema can provide insight into the breadth and depth of the team/company skill mix. It is usually necessary to have a set of skills covering most of the framework – that implies that the team should be able to model from the Planner, Owner, Designer, etc., viewpoints with respect to motivation, data, function, people, network, and time, in order to effectively model all aspects of the enterprise.

Content and Arrangement of the Cells

The right tool. There is no correct modeling tool for any particular cell. For each cell, use which ever modeling tool is most appropriate, e. g., physical analogs, analytic equations, state machines, functional flow block diagrams, block diagrams of linear systems theory, transfer functions, state-space models, differential or difference equations, object-oriented models, UML diagrams, Monte Carlo simulations, statistical distributions, graphical (animated) simulations, mathematical programming, Markov processes, time-series models, financial models, Pert charts, Gantt charts, computer programs, use cases or mental models.

Each entity in a cell should be a model (a system) in its own right. That means it should have defined inputs, functions, outputs, states, objects, figures of merit, metrics, technical performance measures, interfaces, etc. If a different modeling tool is used for each model, then the interfaces will be harder to design. That is one advantage of the UML: it allows the models to communicate better.

Column order. A newspaper article should start with who, what, when, where, why and sometimes how, usually *in that order*, but we have seen Zachman frameworks with many different column orders. In fact, Zachman explicitly says that the ordering of the columns is not important. However, we think column order is important. (For the reasons given in the next two paragraphs) we suggest using the original Zachman column order: What, How, Where, Who, When and Why.

Which cells are most important? To increase the performance of baseball players, the lower-left cells are the most important and the upper-right cells are least important (with the column order that we have used) (Rouse, Cannon-Bowers & Salas, 1992). UML diagrams are also most useful in the lower-left corner. In contrast, a CEO would be interested in the top rows and in particular the upper-right corner. Models in rows 1 and 2 are the domain of the President, CEO, board of directors and rule making organizations. Models in rows 3 and 4 are owned by managers. Row 5 tasks are performed by the most engineers.

We selected four collections of research papers and categorized each paper into the cell that best represented its fundamental characteristics. The collections were the proceedings of a sports engineering conference held at the University of California at Davis (Hubbard, Mehta & Pallis, 2004) upper-left quadrant of each cell, the proceedings and keynote addresses of a sports engineering conference held in Tokyo (Subic & Ujihashi, 2005) lower-left quadrant, a biomedical engineering in sports collection (Hung & Pallis, 2004) upper-right quadrant, and the 2004 volume of the *Information Systems Journal*, lower-right quadrant. This categorization is shown in Table 4. The authors, for the most part, were research engineers and scientists. As can be seen, most of the papers fall into the lower left-hand corner of the framework. We would obviously find a different emphasis at meetings of sports agents, financial analysts or lawyers.

The sports engineering conference papers were five pages long and they were easy to categorize into a single cell. The biomedical engineering and information systems papers were longer and had to be divided into two or even three cells. The resulting numbers for each publication were rounded into integers and put into Table 4, shown on the next page.

	1. What		2. How		3. Where		4. Who		5. When		6. Why	
1. Scope												
	1						1					
2. Business model	2											
		3		1								
3. System model	3		4				2					
	1	1	1	4			2					1
4. Technology model	6	2	31				2	4			1	
	2	2	2	1			1	1				1
5. Detailed representation	78	4	37	5	6		2	4	1			
	51	2	27	1	2		2		1			1
6. Real system												

Table 4. Number of Papers with Primary Emphasis Indicated in the Related Cell

For the *Information Systems Journal* papers, we found that case studies and papers dealing with single software or a particular technique would be of interest to researchers and practitioners and we categorized them as Detailed Representation (row 5). Literature surveys, papers summarizing interviews with many companies and papers about the usage of software were categorized as Technology Models (row 4). Papers about frameworks, bodies of knowledge and general models for how to design and develop information systems would be of interest to educators and managers and they were categorized as System Models (row 3). Rows 1 and 2 needed no further elaboration.

The contract for Alex Rodriguez was cited in row-2 (Team Owner) and column-4 (People). The case study discussing this contract (Cohen & Wallace, 2003; Cohen, 2003) mentions dozens of things that were considered: all of them are in the upper-right corner; none of them are in the lower-left corner.

Given that we are studying Modeling and Simulation in the Enterprise, in which row and column should this paper be categorized? We think that it is row 3 (System Model), column 1 (What) and column 2 (How).

Primitives. A pure Zachman framework contains primitive models, that is, models that deal with only one column at a time. Most practical models, like Figure 1, are called composite models, because they form a composite of primitive elements from more than one Zachman column. Zachman says that composite models should be derived from primitive models: if composite models are built from the beginning, you will likely end up with point-in-time solutions that do not consider enterprise-wide implications.

The big picture. The purpose of identifying a set of models is to understand enterprises consisting of organizations and systems. The purpose of the individual models could be to (1) understand an existing system or organization, (2) create a new design or system, (3) control a system, (4) improve performance, (5) improve a system or an organization (6) increase return on investment, (7) plan system development, (8) point out important assumptions, (9) help formulate system structure, (10) suggest new experiments, (11) guide future data collection efforts, (12) suggest accuracy for calculating parameters, (13) adjust numerical values of parameters, (14) allocate resources, (15) detect critical criteria, (16) identify cost drivers, (17) identify bottlenecks, (18) help sell the product and (19) reduce risk.

Modeling business processes. To understand a business process such as design and management of information systems in a particular company, that company could create a database that contains data that answers these questions for each information system they have. Purpose? Owner? Architecture? Inputs? Outputs? Functions? Interfaces? Interacts with? Cost? Business case? Level? Who? What? When? Where? Why? How? Standard (e.g., UML, RUP, CORBA)? Modeling tool? Simulation tool? Zachman row? Zachman column? Then they could reduce the number of classes using affinity analysis. And finally, they could abstract this into a metamodel that shows how that company designs and manages information systems.

SUMMARY

The Zachman framework provides a general six-by-six schema that can be used to organize and assess completeness of descriptive representations for any complex enterprise such as an organization, your customer, a system or a sport. To ensure a complete and holistic understanding of the enterprise architecture, it is necessary to develop models that address the perspectives and aspects that constitute the rows and columns, respectively, of the framework. When constructing these models it is important to use whichever modeling notation and tool that is most appropriate for conveying the value of the information captured in the models. This may lead to a list of entities, a differential equation model, a set of DoDAF artifacts using UML or IDEF notation, or maybe a simple paragraph.

For systems engineering applications, a Zachman framework is typically used to organize models whose ultimate purpose is to better understand and communicate the context, requirements, and detailed design for a system or a system of systems that is to be built and deployed. As a result, there is typically more detail in the models in the lower rows of the framework. These detailed models are used to explain to engineers and customers how the system should be structured and implemented to most effectively realize the capabilities and requirements that are identified by analyzing models describing the Planner and Business Owner perspectives of the enterprise. Cells in the top rows typically contain models that express the organization’s vision and mission statements, hard technical problems, and concept of operation. Cells in the lower rows describe the logical, technical and physical solutions that ideally provide a better, more rewarding way to conduct business.

Models developed in a given row should contain information that is at the same level of detail (granularity) as other models in the same row. This helps keep the models understandable and allows for better correlation (integration) between models in the same row. Models organized within the Zachman framework should be horizontally and vertically integrated in terms of rows and columns of the schema. This implies that models in a given row should exhibit consistency in terms of how the primitive data, function, network, people, time and motivational elements are used, re-used, and depicted in the models. Vertical integration means that models in a given column should demonstrate traceability when moving from one row to the next. Finally, the Zachman framework can be used as a valuable assessment tool to determine the quality of coverage for existing artifacts such as requirements and diagrams, as well as identify gaps in skills within a company or architecture team.

We have found the Zachman framework to be useful. Our intention in this paper was to examine it, explain how to use it and look at its strengths and weaknesses. We wanted to help the reader to decide whether or not the Zachman framework would help him or her to organize the models in his or her business.

ACKNOWLEDGEMENT

The authors would like to thank Dave Baldwin, Mark Vriesenga, and Reuben Settergren for comments on the manuscript. The research for this paper was partially supported by the AFOSR/MURI F49620-03-1-0377 grant.

AUTHOR BIOGRAPHIES

A. Terry Bahill

Dr. Bahill is a Professor of Systems Engineering at the University of Arizona in Tucson. While on sabbatical from the University of Arizona, he did research with BAE Systems in San Diego. He received his Ph.D. in electrical engineering and computer science from the University of California, Berkeley, in 1975. He holds U.S. patent number 5,118,102 for the Bat Chooser™ a system that computes the Ideal Bat Weight™ for individual baseball and softball batters. He received the Sandia National Laboratories Gold President's Quality Award. He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE), of Raytheon and of the International Council on Systems Engineering (INCOSE). He is the Founding Chair Emeritus of the INCOSE Fellows Selection Committee. His picture is in the Baseball Hall of Fame's exhibition "Baseball as America." You can view this picture at http://www.sie.arizona.edu/sysengr/.

Rick Botta

Mr. Botta is the Director of Systems Engineering for BAE Systems in San Diego. He holds a

Bachelor of Science degree in Computer Science from California Polytechnic State University, San Luis Obispo. Rick has a quarter century experience in a wide variety of engineering, engineering management and program management roles involving development and integration of complex, software intensive systems. He is a member of INCOSE.

Jesse Daniels
Mr. Daniels is a systems engineer with BAE Systems in San Diego. He earned a B. S. and an M. S. in Systems Engineering in 1999 and 2000 from Department of Systems and Industrial Engineering at the University of Arizona. Jesse has been a key contributor involved in requirements analysis, design, development, test and deployment of numerous complex systems for US and foreign military. Jesse is currently a full-time member of the Systems Engineering and Integration Center of Excellence at BAE Systems.

REFERENCES

Adair, R. *The Physics of Baseball.* New York: Harper & Row, 1994.

Angus, J. *Management by baseball: the official rules for winning management in any field,* Harper Collins, 2006.

Armstrong, J. How maturity modeling saved my softball team, *Proceedings of the 8th Annual International Symposium of the International Council on Systems Engineering* (INCOSE), Vancouver, British Columbia, Canada, July 26-30, 1998

Bahill, A.T. The Umpire's Assistant, http://www.sie.arizona.edu/sysengr/sie577/UmpiresAssistant.doc accessed October 2006.

Bahill, A.T. The ideal moment of inertia for a baseball or softball bat, *IEEE Transactions on System, Man, and Cybernetics – Part A: Systems and Humans*, 34(2), (2004), 197-204.

Bahill, D. Baseball video. http://www.sie.arizona.edu/sysengr/baseball/2Seam-4Seam-Video.AVI, 2005.

Bahill, A.T. & Baldwin, D. Describing baseball pitch movement with right-hand rules, *Computers in Biology and Medicine*, in press 2006.

Bahill, A.T. & Baldwin, D. The rising fastball and other perceptual illusions of batters, in G. K. Hung & J. M. Pallis (Editors), *Biomedical Engineering Principles in Sports*, Kluwer Academic, 2004, pp. 257-287.

Bahill, A.T., Baldwin, D., & Venkateswaran, J. Predicting a Baseball's path, *American Scientist,* 93(3), pp. 218-225, 2005.

Bahill, A.T. & Botta, R. Baseball models populating a Zachman Framework, The Engineering of Sport 5, (Eds.) M. Hubbard, R. D. Mehta and J. M. Pallis, *Proceedings of the 5th International Engineering of Sport Conference*, September 13-16, 2004, Davis, CA, International Sports Engineering Association (ISEA), Sheffield, England, vol. 2, pp. 312-318.

Bahill, A.T., Botta, R., & Daniels, J. A systems engineering approach to organizing baseball models, keynote address at the Asia-Pacific Congress on Sports Technology, Tokyo, September 11-14, 2005.

Bahill, A.T., Botta, R., & Smith, E. What Are Levels? *Proceedings of the 15th Annual International Symposium of the International Council on Systems Engineering* (INCOSE), July 10-15, 2005, Rochester, NY

Bahill, A.T. & Daniels J. Using object-oriented and UML tools for hardware design: a case study, *Systems Engineering*, 6(1), (2003), 28-48.

Bahill, A.T. & Karnavas, W. Determining ideal baseball bat weights using muscle force-velocity relationships, *Biological Cybernetics*, 62: (1989), 89-97.

Bahill, A.T. & Karnavas, W. The ideal baseball bat, *New Scientist*, 130(1763): 6 April (1991), 26-31.

Bahill, A.T. & Karnavas, W. The perceptual illusion of baseball's rising fastball and breaking curve ball, *Journal of Experimental Psychology: Human Perception and Performance,* 19: (1993), 3-14.

Bahill, A.T. & LaRitz, T. Why can't batters keep their eyes on the ball, *American Scientist*, 72, May-June (1984), 249-253.

Bahill, A.T. & Morna Freitas, M. Two methods for recommending bat weights, *Annals of Biomedical Engineering*, 23/4, (1995), 436-444.

Baldwin, D. How to win the blame game, *Harvard Business Review*, July-August (2001), R0107C.

Baldwin, D. & Bahill, A. T. Nickel patterns on pitches, *Society for American Baseball Research (SABR),* 36th Annual Conference, June 29-30, 2006, Seattle, WA, available at http://www.sie.arizona.edu/sysengr/RedDot.ppt.

Baldwin, D. & Bahill, A. A model of the bat's vertical sweetness gradient, in M. Hubbard, R. D. Mehta & J. M. Pallis (Editors), *The Engineering of Sport 5*, vol. 2, Proceedings of the 5th International Engineering of Sport Conference, September 13-16, 2004, Davis, CA, International Sports Engineering Association (ISEA), Sheffield, UK, pp. 305-311.

Barzun, J. *God's Country and Mine; A declaration of love spiced with a few harsh words,* Atlantic-Little, Brown and Company, 1954.

Boeke, K. *Cosmic View, the Universe in 40 Jumps*, John Day Company, New York, 1957.

Brancazio, P. "Swinging for the Fences: The Physics of the Baseball Bat," paper presented at the New England section of the American Physical Society meeting, October 1987.

Burkiet, B., Harold, E. & Palacios, J. A Markov chain approach to baseball, *Operations Research,* 45(1) (1997), 14-23.

Cockburn, A. *Writing effective use cases*, Addison-Wesley, 2001.

Cohen, R. Teaching Note on A-Rod: Signing the best player in baseball, *Harvard Business School Case Study* 5-203-091, April 16, 2003.

Cohen, R. & Wallace, J. A-Rod: Signing the best player in baseball, *Harvard Business School Case Study* 9-203-047, January 27, 2003.

Crisco, J. *NCAA Research Program on Bat and Ball Performance*, final report, November 12, 1997.

Cross, R. The sweet spot of the baseball bat. *Am J Physics* 66, (1998), 772-779.

de Villiers, D. Using the Zachman framework to assess the Rational Unified Process, The Rational Edge, http://www.therationaledge.com/content/mar_01/t_zachman_dv.html, 2001.

Eames, C. & Eames, R. A Rough Sketch for a Proposed Film Dealing with the Powers of Ten and the Relative Size of the Universe, Presented at the annual meeting of the Commission on College Physics, 8 minutes, color, 1968.

Frankel, D. et al. The Zachman framework and the OMG's model driven architecture, *Business Process Trends*, www.omg.org/mda/mda-files/, September 2003.

Gould, S. *Triumph and Tragedy in Mudville,* W. W. Norton, NY, 2003.

Gray, R. Behavior of college baseball players in a virtual batting task, *Journal of experimental psychology: Human Performance and Perception*. 28, No. 5, (2002), 1131–1148.

Gray, R. Markov at the bat: a model of cognitive processing in baseball batters, *Psychological Science*, 13(6): (2003), 542-547.

Hubbard, M., Mehta, D. & Pallis, j. (Editors). *The Engineering of Sport 5*, International Sports Engineering Association (ISEA), Sheffield, UK, 2004.

Hung, G. & Pallis, J. (Editors). *Biomedical Engineering Principles in Sports*, Kluwer Academic, 2004.

IBM RUP, Rational Unified Process, http://www.rational.com/products/rup/index.jsp, 2003.

Karnavas, W. J., Bahill, A.T. & Regan, D. A model for the rising fastball and breaking curve ball, *Proceedings of the 1990 IEEE International Conference on System, Man, and Cybernetics*, 925-928.

Lewis, M. *Moneyball: The Art of Winning an Unfair Game*, W.W. Norton & Company, 2003.

Maier, M., Emery, D. & Hilliard, R. ANSI/IEEE 1471 and systems engineering, *Systems Engineering*, 7(3), 257-270, 2004.

Major League Baseball Rules, 2005, http://mlb.mlb.com/NASApp/mlb/mlb/official_info/official_rules/foreword.jsp

Martin, J. *Systems Engineering Guidebook: A process for developing systems and products*, CRC Press, 1997.

McHugh, D. & Bahill, A.T. Learning to track predictable target waveforms without a time delay, *Investigative Ophthalmology and Visual Science*, 26, (1985), 932-937.

McBeath, M., Shaffer, D. & Kaiser, M. How baseball outfielders determine where to run to catch fly balls, *Science*, 268, April (1995), 569-573,

Moody, J., Chapman, W., Van Voorhees, F. & Bahill, A.T. *Metrics and Case Studies for*

Evaluating Engineering Designs, Chapter 4, Bat Chooser, Prentice Hall PTR, Upper Saddle River, NJ, 1997.

Morganwalp, J. & Sage, A. A system of systems focused enterprise architecture framework and an associated architecture development process, *Information, Knowledge, Systems Management,* IOS Press, 3, (2002/2003), 87-105.

Morrison, P. & Morrison, P. *Powers of Ten*, Scientific American Books, W. H. Freeman, 1977. This book has been converted into many web sites including http://www.wordwizz.com/pwrsof10.htm

Nathan, A. Dynamics of the baseball-bat collision. *Am J Physics*, 68: (2000), 979-990.

Nathan, A. Characterizing the performance of baseball-bats. *Am J Physics*, 71: (2003), 134-143.

Regan, D. Visual judgments and misjudgments in cricket, and the art of flight, *Perception*, 21: (1992), 91-115.

Reiter, J. Should teams walk or pitch to Barry Bonds? *Baseball Research Journal,* 32, (2004), 63-69.

Rouse, W., Cannon-Bowers, J., & Salas, E. The role of mental models in team performance in complex systems, *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6), (1992), 1296-1308.

Sawicki, G., Hubbard, M. & Stronge, W. How to hit home runs: Optimum baseball bat swing parameters for maximum range trajectories, *Am. J. Phy*. 71(11), (2003), 1152-1162.

Shaffer, D. & McBeath, M. Baseball outfielders maintain a linear optical trajectory when tracking uncatchable fly balls, *Journal of Experimental Psychology: Human Perception and Performance*, 28, (2002), 335-348.

Software Engineering Institute, Carnegie Mellon University, Pittsburg, PA. Accessed July 2006 at http://www.software.org/pub/afc/zachman.asp.

Subic, A. & Ujihashi, S. (Editors). *The Impact of Technology on Sport*, Australasian Sports Technology Alliance Pty Ltd, Melbourne, Australia, 2005.

Watts, R. & Bahill, A.T. *Keep Your Eye on the Ball: Curve Balls, Knuckleballs and Fallacies of Baseball,* W.H. Freeman, 2000.

Williams, T. & Underwood, J. *The Science of Hitting,* New York: Simon and Schuster, 1982.

Zachman, J. A framework for information systems architecture, *IBM Systems Journal*, 26(3), 1987.