The Zachman Framework for Enterprise Architecture in Practice

Cason A. Konzer
CIS562
University of Michigan-Flint
Holly, USA
casonk@umich.edu

***Abstract*—ZFEA broke ground in the domain of EA, bringing to the IS community a 'seminal framework' best described as ontology and explanatory theory. In this work we describe in detail the historical evolution of ZFEA along with its use cases in practice. ZFEA is compared against competing EAs and found to be a foundational EAF in which the others draw from, with focus on specific abstractions. ZFEA is found to be the broadest EAF, a generic tool for rationalizing complex engineered objects. In practice the tool is considered too broad, best suited for ideation and EA exploration.**

***Index Terms*—Zachman, Enterprise, Architecture**

I. INTRODUCTION

ZFEA (Zachman Framework for Enterprise Architecture) was conceived by John A. Zachman in the early 1980s while working at *IBM (International Business Machines Corporation)* as an account executive. Inspired by 15+ years of experience in industry, Zachman's model provided a framework for business systems planning, today coined EAFs (Enterprise Architecture Frameworks). EA (Enterprise Architecture) encompasses information systems in organizations applicable to all employees, supporting business processes through hardware and software. Made clear by Zachman, ZFEA is a schema based on primitive interrogatives (what, how, when, who, where, why) and reification, the transformation of an abstract idea into an instantiation [2]. Building on this note, ZFEA can be applied not only to EA, but any objects (Figure 1).

The paper follows with a detailed history of ZFEA, related works on its academic study, how it's to be used in practice, a comparison against competitors, and concluding remarks. In practice today, there are a plethora of EAFs, although their business use cases are quite impractical, and can be viewed more generally from a theoretical lens.

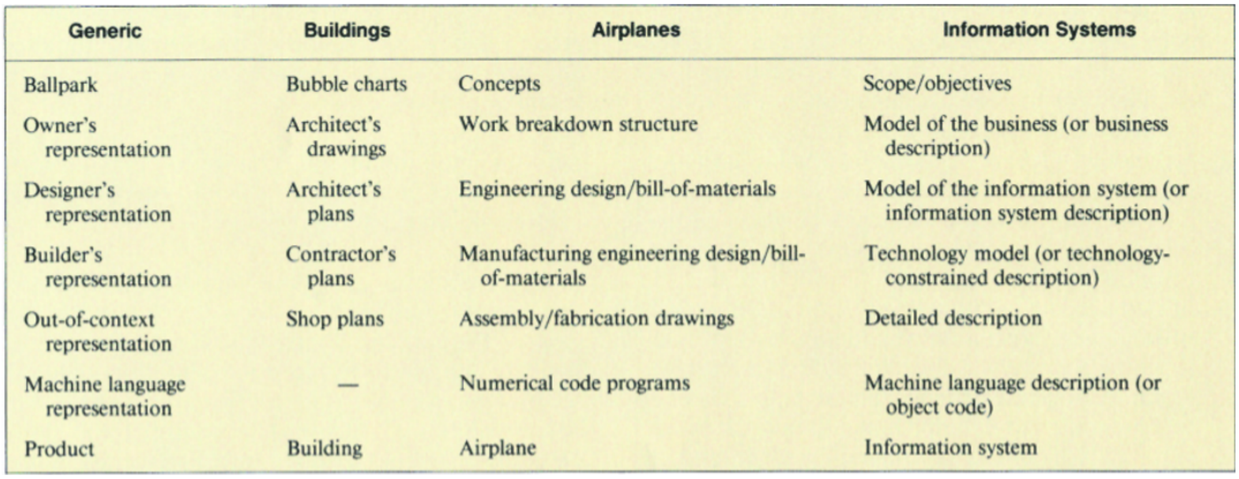
II. HISTORY

Prior to the first publication of his framework [1], Zachman created a crude representation in 1984 on transparencies for overhead projector use, both by himself and with the help from *IBM* graphics support [3]. Notable, is that at this time the row-view of the model was not quite distinguished into stakeholder perspectives, but rather their paralleled model inputs.

A. ZFEA 1987 (A Framework for Information Systems Architecture)

In his *IBM Systems Journal* seminal paper [1], Zachman brought the row-wise perspectives to life under the following nomenclature: Ballpark, Owner, Designer, Builder, Out-of-Context, Product/System (Figure 1,2). In a similar manner, it is at this time that column-wise interrogatives of what, how, and where were introduced as data, process, and network descriptors respectively. About the remaining interrogatives, e.g. who, when, why, Zachman had these developed at the time, but felt the framework was already quite complex and would take time for the community to digest. As a result, only a brief mention of the framework's extension to future state is made, and details provided in the paper appendix.

Summarizing the state upon this publication, ZFEA was a 6 × 3 matrix representing a set of architectural representations for IS (Information Systems), covering half of the cells realized today. The framework was directed at a subset of use cases within the IS community [1]:



Generic	Buildings	Airplanes	Information Systems
Ballpark	Bubble charts	Concepts	Scope/objectives
Owner's representation	Architect's drawings	Work breakdown structure	Model of the business (or business description)
Designer's representation	Architect's plans	Engineering design/bill-of-materials	Model of the information system (or information system description)
Builder's representation	Contractor's plans	Manufacturing engineering design/bill-of-materials	Technology model (or technology-constrained description)
Out-of-context representation	Shop plans	Assembly/fabrication drawings	Detailed description
Machine language representation	—	Numerical code programs	Machine language description (or object code)
Product	Building	Airplane	Information system

Fig. 1. 1987 Analog Representations from Core Views/Perspectives [1].

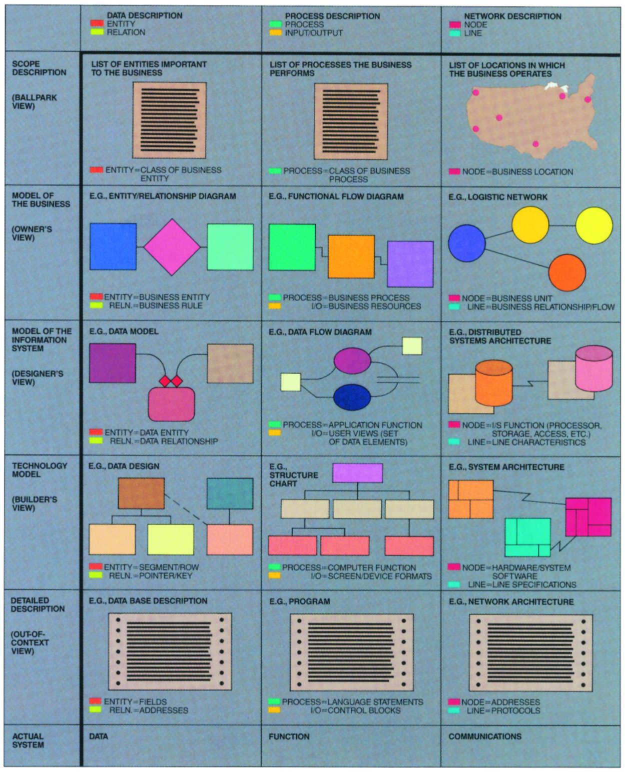
- Improving professional communications.
- Understanding purpose and risks of developing an architectural representation.
- Relating tools and methodologies.
- Improving approaches to produce architectural representations.
- Rethinking the nature of the application development process.

B. ZFEA 1992 (Extending and Formalizing the Framework for Information Systems Architecture)

After retiring from *IBM* in 1991, Zachman operated *ZI (Zachman International)*, an EA consulting and certification business. As a co-author, Zachman, published again in *IBM Systems Journal*, a follow up to the seminal paper with J.F. Sowa, bringing insights from the few years of widespread academic and industry exposure [4]. It is in this paper that the who, when, and why interrogatives were brought to fruition, despite their lack of clear empirical ties within IS. Additionally more granular examples were brought to the matrix entries, of which the seminal work did not have the time to cover in detail.

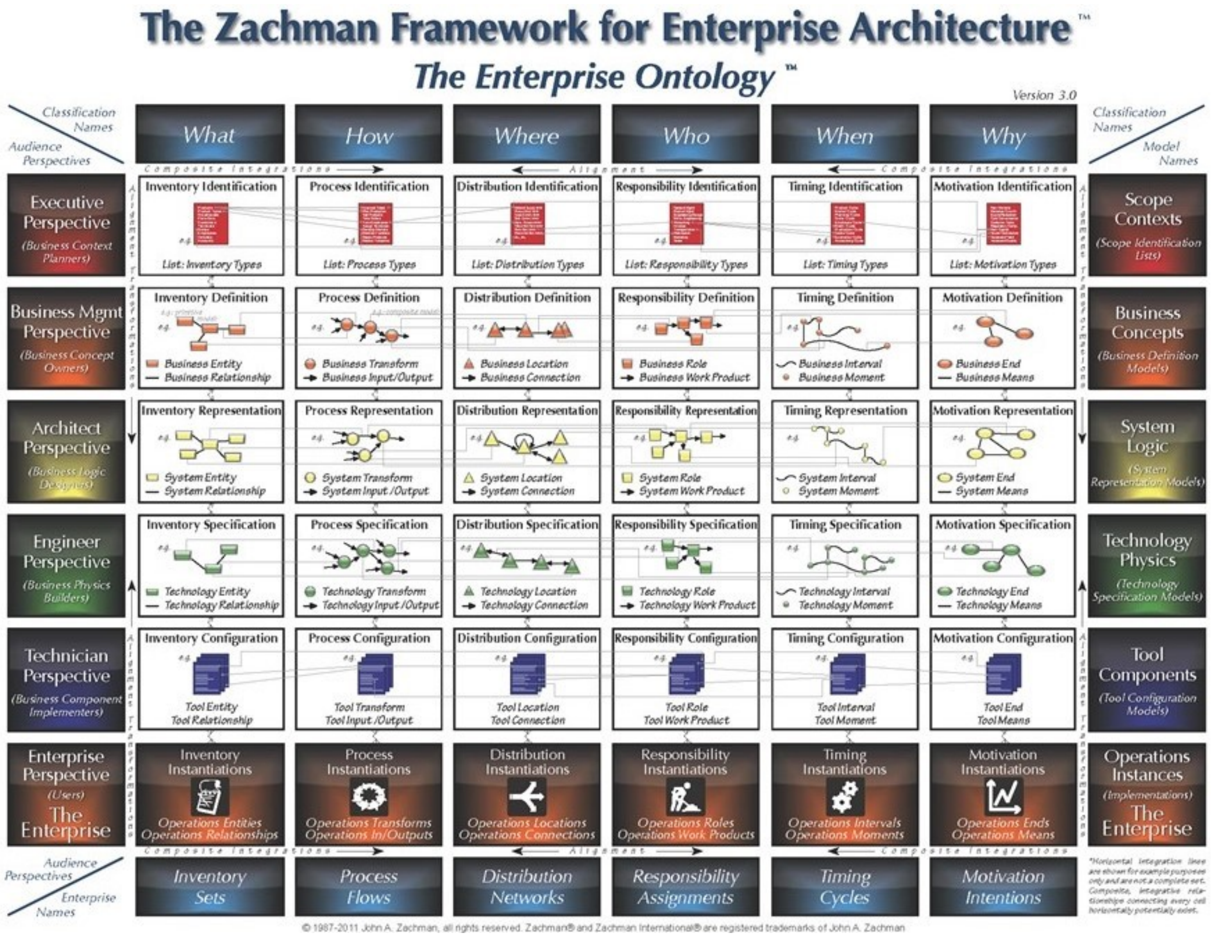
With the added complexity, the framework had an intrinsic chaos. Here, a clear set of rules are defined to bring clarity and order [4]:

1) The columns have no order.
2) Each column has a simple, basic model.
3) The basic model of each column must be unique.
4) Each row represents a distinct, unique perspective.
5) Each cell is unique.
6) The composite or integration of all cell models in one row constitutes a complete model from the perspective of that row.
7) The logic is recursive.



	DATA DESCRIPTION 🟥 ENTITY 🟨 RELATION	PROCESS DESCRIPTION 🟩 PROCESS 🟧 INPUT/OUTPUT	NETWORK DESCRIPTION 🟪 NODE 🟦 LINE
SCOPE DESCRIPTION (BALLPARK VIEW)	LIST OF ENTITIES IMPORTANT TO THE BUSINESS 🟥 ENTITY=CLASS OF BUSINESS ENTITY	LIST OF PROCESSES THE BUSINESS PERFORMS 🟩 PROCESS=CLASS OF BUSINESS PROCESS	LIST OF LOCATIONS IN WHICH THE BUSINESS OPERATES 🟪 NODE=BUSINESS LOCATION
MODEL OF THE BUSINESS (OWNER'S VIEW)	E.G., ENTITY/RELATIONSHIP DIAGRAM 🟥 ENTITY=BUSINESS ENTITY 🟨 RELN.=BUSINESS RULE	E.G., FUNCTIONAL FLOW DIAGRAM 🟩 PROCESS=BUSINESS PROCESS 🟧 I/O=BUSINESS RESOURCES	E.G., LOGISTIC NETWORK 🟪 NODE=BUSINESS UNIT 🟦 LINE=BUSINESS RELATIONSHIP/FLOW
MODEL OF THE INFORMATION SYSTEM (DESIGNER'S VIEW)	E.G., DATA MODEL 🟥 ENTITY=DATA ENTITY 🟨 RELN.=DATA RELATIONSHIP	E.G., DATA FLOW DIAGRAM 🟩 PROCESS=APPLICATION FUNCTION 🟧 I/O=USER VIEWS (SET OF DATA ELEMENTS)	E.G., DISTRIBUTED SYSTEMS ARCHITECTURE 🟪 NODE=I/S FUNCTION (PROCESSOR, STORAGE, ACCESS, ETC.) 🟦 LINE=LINE CHARACTERISTICS
TECHNOLOGY MODEL (BUILDER'S VIEW)	E.G., DATA DESIGN 🟥 ENTITY=SEGMENT/ROW 🟨 RELN.=POINTER/KEY	E.G., STRUCTURE CHART 🟩 PROCESS=COMPUTER FUNCTION 🟧 I/O=SCREEN/DEVICE FORMATS	E.G., SYSTEM ARCHITECTURE 🟪 NODE=HARDWARE/SYSTEM SOFTWARE 🟦 LINE=LINE SPECIFICATIONS
DETAILED DESCRIPTION (OUT-OF-CONTEXT VIEW)	E.G., DATA BASE DESCRIPTION 🟥 ENTITY=FIELDS 🟨 RELN.=ADDRESSES	E.G., PROGRAM 🟩 PROCESS=LANGUAGE STATEMENTS 🟧 I/O=CONTROL BLOCKS	E.G., NETWORK ARCHITECTURE 🟪 NODE=ADDRESSES 🟦 LINE=PROTOCOLS
ACTUAL SYSTEM	DATA	FUNCTION	COMMUNICATIONS

Fig. 2. 1987 Framework for Information Systems Architecture [1].



Classification Names Audience Perspectives	What	How	Where	Who	When	Why	Classification Names Model Names
Executive Perspective (Business Context Planners)	Inventory Identification List: Inventory Types	Process Identification List: Process Types	Distribution Identification List: Distribution Types	Responsibility Identification List: Responsibility Types	Timing Identification List: Timing Types	Motivation Identification List: Motivation Types	Scope Contexts (Scope Identification Lists)
Business Mgmt Perspective (Business Concept Owners)	Inventory Definition Business Entity Business Relationship	Process Definition Business Transform Business Input/Output	Distribution Definition Business Location Business Connection	Responsibility Definition Business Role Business Work Product	Timing Definition Business Interval Business Moment	Motivation Definition Business End Business Means	Business Concepts (Business Definition Models)
Architect Perspective (Business Logic Designers)	Inventory Representation System Entity System Relationship	Process Representation System Transform System Input/Output	Distribution Representation System Location System Connection	Responsibility Representation System Role System Work Product	Timing Representation System Interval System Moment	Motivation Representation System End System Means	System Logic (System Representation Models)
Engineer Perspective (Business Physics Builders)	Inventory Specification Technology Entity Technology Relationship	Process Specification Technology Transform Technology Input/Output	Distribution Specification Technology Location Technology Connection	Responsibility Specification Technology Role Technology Work Product	Timing Specification Technology Interval Technology Moment	Motivation Specification Technology End Technology Means	Technology Physics (Technology Specification Models)
Technician Perspective (Business Component Implementers)	Inventory Configuration Tool Entity Tool Relationship	Process Configuration Tool Transform Tool Input/Output	Distribution Configuration Tool Location Tool Connection	Responsibility Configuration Tool Role Tool Work Product	Timing Configuration Tool Interval Tool Moment	Motivation Configuration Tool End Tool Means	Tool Components (Tool Configuration Models)
Enterprise Perspective (Users) The Enterprise	Inventory Instantiations Operations Entities Operations Relationships	Process Instantiations Operations Transforms Operations In/Outputs	Distribution Instantiations Operations Locations Operations Connections	Responsibility Instantiations Operations Roles Operations Work Products	Timing Instantiations Operations Intervals Operations Moments	Motivation Instantiations Operations Ends Operations Means	Operations Instances (Implementations) The Enterprise
Audience Perspectives Enterprise Names	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	*Horizontal integration lines are shown for example purposes only and are not a complete set. Composite, integrative relationships connecting every cell horizontally potentially exist.

© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

Fig. 3. 2011 ZFEA V3.0 - The Enterprise Ontology [3].

Building upon rule 6, the paper provides toy cell models in which the rows include conceptual relations across columns to provide the holistic perspective view. Rule 7 is broken down into three unique dimensions, namely related frameworks, framework versions, and nesting frameworks. Of course, frameworks will be updated over time, driving versioning, more interesting is the concepts of nested frameworks. As ZFEA can truly be used to describe any object, the cell in on instantiation of ZFEA can become its own instance, with relations to the neighboring cells in the parent instance. This recursive nature is now clearly exhibited in repositories today, such as Web Enterprise Architect, GitHub, etc.

Lastly, conceptual graphs are given as a generic modeling framework of which can be uniquely developed for each cell. At this point in time, the model had evolved to include use cases for EA segmentation, variable separation across cells, component design, and formalizing business EA principals. With dependency knowledge across cells, the impacts of changes in a single cell become able to demonstrate influence across abstractions, perspectives, and holistically, the versions.

C. ZFEA Today

Since the initial release, ZFEA has stood the test of time and required only minor revisions. Comparing the 1992 version to the latest 2011 release, a few additions are brought, slight nomenclature has changed, and aesthetic subtleties are incorporated (Figure 3). Additions include enterprise and model names, aiming for presentation to general business management rather than strictly IS professionals. The current ZFEA, version 3.0, has been subtitled with *The Enterprise Ontology*, as it is truly more so an ontology than a framework [3].

III. Related Work

ZFEA has now 35+ years of industry and academic exposure, prompting a plethora of detailed literature on it's behalf. Within the research endeavors it is notable to classify these works based on their content, review (contextual) and implementation (case) studies.

A. Review Studies

From a review perspective, it is commonly noted that ZFEA alone is often misinterpreted, and lacks granularity for business practices. Specific concerns of the framework include the following [5], [6]:

- Cells in the framework lack explicit models.
- Dependencies between cells are only vaguely defined and lack detail.
- There is no process or methodology based in SWE (Software Engineering) prescribed.
- The framework is too complex to support communication.
- The framework is too abstract to capture architectural problems.

As a result of these concerns, [5] partitions ZFEA according to domain engineering phases and requirements engineering techniques from the SWE field. Additionally, they map ZFEA to more granular ontologies grounded in EA, namely the Bunge-Wand-Weber Ontology and The Enterprise Ontology from The Enterprise Project at Artificial Intelligence Applications Institute at the University of Edinburgh. When viewed independently, the two ontologies both fail to encompasses ZFEA in total, only this is achieved with their combination.

From an impetuous perspective, [6] regards ZFEA as practically useless. Characterizations of the popular 'futile' EAs (ZFEA, TOGAF, ArchiMate) are given as intentionally vague, continuously hyped, containing elusive explanations and empty promises. Speaking from both industry experience and academia, the author promotes individual judgement of architects and a focus on utilizing best practices from experience, suggesting the Business Capability Model as one example of a 'real tool'.

In a more positive light, [7] classifies ZFEA as an explanatory theory in IS. An explanatory theory is characterized as one which distinguishes what, how, why, when, and where without any aim to predict, and which is untestable by nature. Generally speaking such a theory looks to provide understanding alone, noting criticism often comes from the assumption that methodologies and processes should be provided as well. Here, ZFEA can be seen as a tool for contextualizing, not instantiating, an enterprise.

The last of the review studies, [8], approaches ZFEA from a SE (Systems Engineering) perspective. While ZFEA is noted as an ontology in and of itself, the authors main contribution is to provide an updated version, noting that due to their generality in nature, some EAFs simply cannot be applied within various environments. It is their perspective, that an ES (Enterprise System) is best viewed from the lens of MBSE (Model-Based Systems Engineering). Proposal is made to utilize the "Vee" Model consisting of a top-down definition & decomposition process, paired with integration & verification (Figure 4). Crucially elaborated is that through simulation, MBSE allows architects to begin validation prior to integration, a virtualized A/B test allowing to catch issues early on in development.

B. Implementation Studies

For the implementation studies an equivalent approach was taken by all works. In attempt to characterize aspects of an EA which were in need of improvement, a subset of ZFEA was applied with a paired questionnaire conduct a factor analysis. [9] analyzed reporting tools for business intelligence in the context of an enterprise plantation management system from the planner and owner perspectives; [10] studies knowledge management systems in banking from the planner and user perspectives; and [11] student an integrated data management system of student internships from the planner perspective. From all studies insights were gained as to how the EA could be best utilized and improved in their specific business context.

IV. Method

The methodology of the ZFEA is a rather simple task of filling out the matrix cells. As an explanatory theory, the implementation is rather open, posing only those restrictions described in II-B. In this manner, and as previously noted, ZFEA explicitly

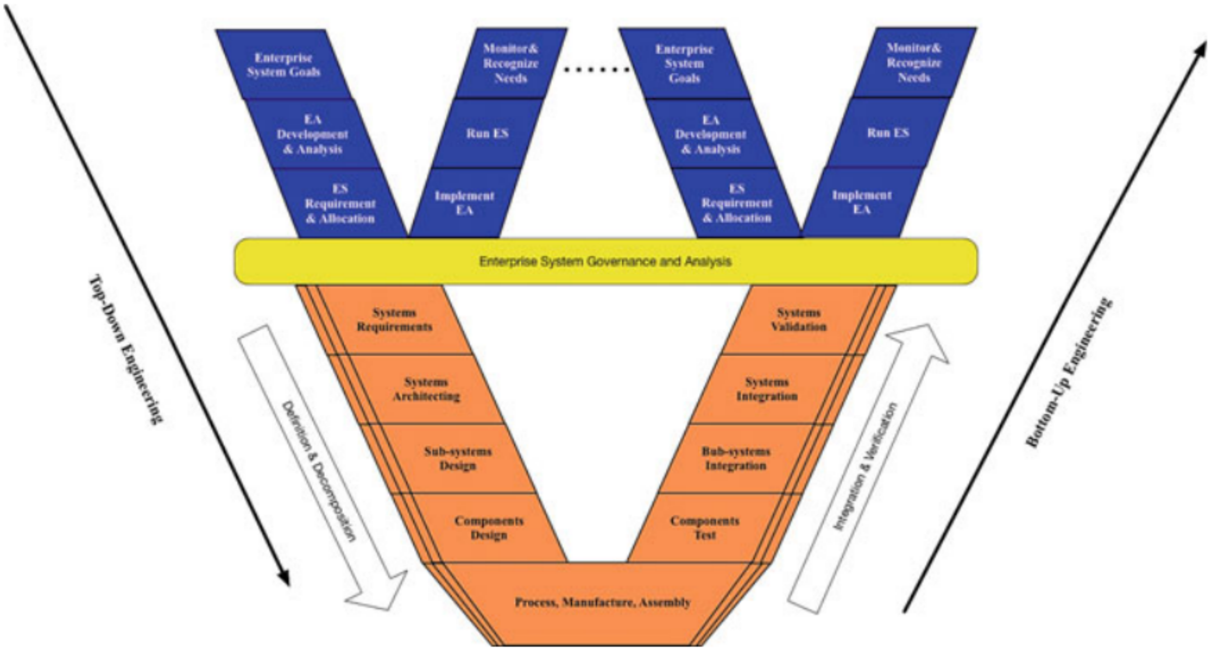


Fig. 4. The ES "Vee" Model [8].

lacks methods, processes, or tools for implementation. From an implementation perspective, the architect should abide by the rules and use the framework for ideation and exploratory analysis into the EA at hand. Similarly, conceptual graphs are suggested for modeling (or filling the cells), such that predicate logic is implied. Notably, the column and row headers for classification names and audience perspective provide a clear starting template to guide ZFEA's use (Figure 3).

V. Comparison

The EAFs in which we consider for comparison are as follows: DoDAF (Department of Defense Architecture Framework), FEAF (Federal Enterprise Architecture Framework), TEAF (Treasury Enterprise Architecture Framework), and TOGAF (The Open Group Architectural Framework). As grounds for comparison we will rely strictly upon the analysis of [12], providing a baseline across all EAFs.

In the work, the comparison is conducted along three dimensions: audience perspectives, classification names, and SDLC (Systems Development Life Cycle) phases. Upon all of the dimensions, except for that of the SDLC maintenance phase, ZFEA encompasses the abstraction. It is only FEAF in which the SDLC maintenance phase is incorporated into the EAF although lacking the user perspective as well as the who, when, and why characteristics.

Of the EAFs, ZFEA is the most comprehensive, in part due to its basis in classical architecture and roots in historical primitives. For context, DoDAF specializes is military operations, FEAF in sharing information across federal EAs, TEAF similarly in information sharing across treasury offices, and TOGAF in open systems integration of mission critical applications. Revisiting the earlier criticism of ZFEA, it is becomes evident that at a certain point an EAF can become too far abstracted from reality, in which case it makes sense to choose a limited framework which is more specific to the business context.

VI. Conclusion

In this paper we have provided an overview of the ZFEA EAF, from inception to today, consensus from research articles based in context and case, along with a comparison to other EAFs and its practical applications. We have seen that EA has evolved from previous literature on IS architecture, providing an explanatory theory. What the theories generally lack are methodologies and practices, to aid in their usage. Commonly accepted as an ontology, ZFEA lacks the required granularity for SE. Use of ZFEA in implementation studies focus primarily on factor analysis to determine prioritization of future changes. From the perspective of SE, a general paradigm shift has taken place moving from EAFs to MBSE. EAFs today are thus best viewed as an exploratory tool for architects, too vague for integration tasks. Looking to the future, EAFs shall continue to be used and taught in academia, while in EA practice, traditional managerial and engineering tools will be used.

References

[1] J. Zachman, “A framework for information systems architecture,” *IBM Systems Journal*, vol. 26, no. 3, pp. 276–292, 1987.

[2] J. A. Zachman, “About the zachman framework,” 2008. [Online]. Available: https://zachman-feac.com/zachman/about-the-zachman-framework

[3] J. P. Zachman, “The zachman framework evolution,” 2022. [Online]. Available: https://zachman-feac.com/the-zachman-framework-evolution

[4] J. F. Sowa and J. A. Zachman, “Extending and formalizing the framework for information systems architecture,” *IBM Systems Journal*, vol. 31, no. 3, pp. 590–616, 1992.

[5] Z. Chen and R. Pooley, “Rediscovering zachman framework using ontology from a requirement engineering perspective,” in *2009 33rd Annual IEEE International Computer Software and Applications Conference*, vol. 2, 2009, pp. 3–8.

[6] S. Kotusev, “Fake and real tools for enterprise architecture: the zachman framework and business capability model,” *Enterprise Architecture Professional Journal (EAPJ)*, 2019.

[7] A. Gerber, P. le Roux, C. Kearney, and A. van der Merwe, “The zachman framework for enterprise architecture: An explanatory is theory,” in *Responsible Design, Implementation and Use of Information and Communication Technology*, M. Hattingh, M. Matthee, H. Smuts, I. Pappas, Y. K. Dwivedi, and M. Mäntymäki, Eds. Cham: Springer International Publishing, 2020, pp. 383–396.

[8] X. Zhang, C. Wang, L. Li, and P. Wang, “An overview of complex enterprise systems engineering: Evolution and challenges,” in *Complex Systems Design & Management*, D. Krob, L. Li, J. Yao, H. Zhang, and X. Zhang, Eds. Cham: Springer International Publishing, 2021, pp. 175–188.

[9] J. Danny, Shanlunt, G. Wang, and H. Alianto, “The application of zachman framework in improving better decision making,” in *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*, 2018, pp. 245–249.

[10] W. Sardjono, A. Retnowardhani, R. Azizah, and Maryani, “Analysis of application of zachman framework for knowledge management systems success optimization,” in *2020 International Conference on Information Management and Technology (ICIMTech)*, 2020, pp. 277–282.

[11] M. Muslih, D. Nurjanah, F. S. Fauziah, Susanti, M. A. Permana, C. Irawan, and J. Asian, “Implementation of impact zachman framework on internship architecture data management,” in *2020 6th International Conference on Computing Engineering and Design (ICCED)*, 2020, pp. 1–6.

[12] L. Urbaczewski and S. Mrdalj, “A comparison of enterprise architecture frameworks,” *Issues in information systems*, vol. 7, no. 2, pp. 18–23, 2006.