

TOGAF and Zachman

Two approaches to large enterprise IT architecture

6/13/2025

Tony Shen

Data communications Labs

TOGAF and Zachman - Two Approaches to Large Enterprise IT

Contents

Introduction	2
TOGAF	2
Zachman Framework	2
TOGAF and Zachman.....	3
ServiceNow CMDB and Zachman 6x6 Matrix	4
Zachman 6x6 Matrix Generation Example - 1.....	5
Zachman 6x6 Matrix Generation Example – 2.....	7
Summary	8
References	9

Revision History
06/13/2025 Created

TOGAF and Zachman - Two Approaches to Large Enterprise IT

Introduction

Every day, large enterprise IT departments face challenges in operations, security, data protection, disaster prevention and recovery, and planning and implementing new solutions to meet ever-changing business requirements by innovation and efficiency while maintaining the entire IT system's coherence, integrity, and resilience to changes. Growing complexity in enterprise IT infrastructure and dispersing footprint when increasingly using cloud computing and leveraging AI exacerbate the challenges and hence demand IT departments to have right approaches to handle IT evolution right and handle it holistically. In this short article, let's take a brief look at TOGAF and Zachman and see how they work to address IT challenges.

TOGAF

TOGAF (The Open Group Architecture Framework) is an enterprise architecture methodology and framework used to improve business efficiency. It provides a structured approach to designing, planning, implementing, and governing an enterprise's information architecture.

Key Components of TOGAF:

1. **Architecture Development Method (ADM)** – A step-by-step process guiding the development and evolution of an enterprise architecture.
2. **Enterprise Continuum** – A framework for categorizing assets like models, patterns, and architectures that support development.
3. **Architecture Domains:**
 - **Business Architecture** – Defines business processes, governance, and organizational structure.
 - **Application Architecture** – Specifies software applications needed to support business goals.
 - **Data Architecture** – Focuses on data management and organization.
 - **Technology Architecture** – Covers infrastructure, hardware, and software.

Benefits of TOGAF:

- Provides a **structured** and repeatable approach to enterprise architecture.
- Enhances **alignment** between business strategy and IT infrastructure.
- Reduces costs and improves efficiency by **standardizing** architecture practices.
- Facilitates better **communication** across departments and stakeholders.

Zachman Framework

The Zachman Framework is a structured approach to enterprise architecture that helps organizations organize and understand complex business systems. It was developed by **John Zachman** in the 1980s and is widely used for classifying and managing enterprise architecture components.

TOGAF and Zachman - Two Approaches to Large Enterprise IT

Key Features of the Zachman Framework:

1. **Two-Dimensional Grid** – The framework is structured as a **6x6 matrix**, where:
 - **Rows** represent different stakeholder perspectives (e.g., business owners, architects, engineers).
 - **Columns** represent fundamental questions: **What, How, When, Who, Where, and Why**.
2. **Ontology, not a Methodology** – Unlike TOGAF, which provides a step-by-step process, Zachman is **not a methodology** but rather a classification system for organizing architectural artifacts.
3. **Reification Process** – It follows a structured transformation from abstract ideas to concrete implementations, ensuring clarity in enterprise architecture.

Benefits of the Zachman Framework:

- **Comprehensive View** – Helps organizations see their architecture from multiple perspectives.
- **Standardized Classification** – Provides a structured way to document enterprise components.
- **Improved Communication** – Ensures alignment between business and IT teams.

TOGAF and Zachman

Both **TOGAF** and **Zachman Framework** are widely used in enterprise IT architecture, but they serve different purposes. Here's a comparison based on key factors relevant to large enterprises:

1. Purpose & Approach

- **TOGAF**: A **methodology** providing a structured process (Architecture Development Method—ADM) for designing and implementing enterprise architecture.
- **Zachman**: An **ontology** — a classification framework that organizes architecture artifacts into a structured matrix.

2. Structure & Scope

- **TOGAF**: Covers **business, data, application, and technology architecture**, offering detailed implementation guidance.
- **Zachman**: Uses a **6x6 matrix** to categorize different enterprise perspectives but does not provide implementation methods.

3. Flexibility & Usage

- **TOGAF**: Designed for **large-scale enterprise IT**, helping organizations align **business strategy with IT architecture** through a repeatable process.
- **Zachman**: Acts as a **high-level classification tool**, useful for organizing existing architecture documentation.

4. Governance & Standardization

- **TOGAF**: Includes governance mechanisms, **enterprise-wide compliance**, and best practices for architectural standardization.

TOGAF and Zachman - Two Approaches to Large Enterprise IT

- **Zachman:** Lacks governance frameworks but ensures consistency in **documentation and classification**.

5. Best Fit for Large Enterprises

- **TOGAF:** Best for enterprises that need **step-by-step architecture development**, aligning IT systems with business needs.
- **Zachman:** More suited for enterprises that require a **structured way to organize architecture artifacts**, without prescribing implementation methods.

For **large enterprise IT**, TOGAF is more **practical** as it offers an **implementation roadmap**, making it easier to manage architecture evolution. Zachman is valuable for **documentation and classification** but needs to be paired with other methodologies for execution. Many enterprises use both: **Zachman for structuring architecture assets** and **TOGAF for guiding development**.

ServiceNow CMDB and Zachman 6x6 Matrix

In many large enterprises, IT departments use ServiceNow as the ticket system. ServiceNow CMDB records all IT assets as CIs (Configuration Items). CIs include servers, storage systems, network hardware gears, software, business applications and IT utility and/or special-purposed applications. ServiceNow CMDB essentially has all the data that you need to create your Zachman 6x6 matrix. The good news is that you can automatically generate the matrix from the CMDB. Here below is how you do it.

Generating a **Zachman 6x6 matrix** automatically from **ServiceNow CMDB** requires mapping CMDB data to the Zachman framework's six perspectives (**Planner, Owner, Designer, Builder, Implementer, and User**) and six fundamental questions (**What, How, When, Who, Where, Why**). Here's how you can approach it:

Steps to Automate Zachman Matrix Generation from CMDB

1. **Define CMDB Data Categories**
 - Identify Configuration Items (**CIs**) that align with Zachman's six perspectives.
 - Map CI attributes to Zachman's six fundamental questions.
2. **Extract Data from CMDB**
 - Use **ServiceNow Query Builder** or **Database Views** to retrieve relevant CI data.
 - Ensure relationships between assets (e.g., applications, infrastructure, business processes) are well-defined.
3. **Transform Data into Zachman Matrix Format**
 - Create a **structured database view** that pivots CMDB data into Zachman's rows and columns.
 - Use **CMDB schema models** to ensure consistency.
4. **Automate Matrix Generation**
 - Develop **ServiceNow reports** or **custom scripts** to format extracted data into a **6x6 matrix**.

TOGAF and Zachman - Two Approaches to Large Enterprise IT

- Use **Power BI** or **Tableau** for visualization.
- 5. **Validate & Refine**
 - Ensure completeness and accuracy of mappings.
 - Adjust data relationships to improve matrix clarity.

Zachman 6x6 Matrix Generation Example - 1

Here's a structured approach to **automating the generation of a Zachman 6x6 matrix** from **ServiceNow CMDB**:

1. Extract CMDB Data

Use **ServiceNow Query Builder** or **REST API** to retrieve Configuration Items (**CIs**) and their attributes. Example API call:

Python

```
import requests
import json
import pandas as pd
```

```
# ServiceNow instance details
INSTANCE = "your-instance.service-now.com"
TABLE = "cmdb_ci"
USERNAME = "your-username"
PASSWORD = "your-password"
```

```
# API request to fetch CMDB data
url = f"https://{INSTANCE}/api/now/table/{TABLE}"
headers = {"Accept": "application/json"}
response = requests.get(url, auth=(USERNAME, PASSWORD), headers=headers)
```

```
# Parse JSON response
data = response.json()
ci_list = data.get("result", [])
```

```
# Define Zachman matrix structure
zachman_matrix = {
    "What": [],
    "How": [],
    "When": [],
    "Who": [],
    "Where": [],
```

TOGAF and Zachman - Two Approaches to Large Enterprise IT

```
"Why": []
}

# Map CMDB attributes to Zachman matrix
for ci in ci_list:
    zachman_matrix["What"].append(ci.get("name", "Unknown"))
    zachman_matrix["How"].append(ci.get("sys_class_name", "Unknown"))
    zachman_matrix["When"].append(ci.get("install_date", "Unknown"))
    zachman_matrix["Who"].append(ci.get("manufacturer", "Unknown"))
    zachman_matrix["Where"].append(ci.get("location", "Unknown"))
    zachman_matrix["Why"].append(ci.get("description", "Unknown"))

# Convert to DataFrame for structured output
df = pd.DataFrame(zachman_matrix)
print(df)
```

2. Map CMDB Data to Zachman Framework

Define mappings between **CMDB attributes** and **Zachman's six perspectives**:

- **Planner (Executive View)** → Business Goals, Strategy
- **Owner (Business View)** → Business Processes, Services
- **Designer (Logical View)** → Application Architecture, Data Models
- **Builder (Physical View)** → Infrastructure, Servers, Networks
- **Implementer (Operational View)** → IT Operations, Monitoring
- **User (Functional View)** → End-user interactions, Service Requests

3. Transform Data into a 6x6 Matrix

Use **Python Pandas** to structure the extracted data:

Beginning of Python code example

```
import pandas as pd

# Sample data mapping
data = {
    "What": ["Business Goals", "Applications", "Data Models", "Infrastructure",
            "Operations", "User Requests"],
    "How": ["Strategy", "Process Design", "Data Flow", "Network Setup", "Monitoring",
            "Usage Patterns"],
    "When": ["Roadmap", "Workflow", "Data Lifecycle", "Deployment Schedule",
            "Incident Response", "Access Timing"],
    "Who": ["Stakeholders", "Business Owners", "Architects", "Engineers", "IT Admins",
            "End Users"],
    "Where": ["Corporate HQ", "Data Centers", "Cloud", "On-Prem Servers", "Remote
            Access", "User Devices"],
```

TOGAF and Zachman - Two Approaches to Large Enterprise IT

```
"Why": ["Business Value", "Efficiency", "Data Integrity", "Performance", "Security",  
"User Experience"]  
}
```

```
df = pd.DataFrame(data)  
print(df)
```

```
# End of code example
```

4. Automate Matrix Generation in ServiceNow

- Use **ServiceNow Reporting** to visualize the matrix.
- Export data to **Power BI or Tableau** for structured presentation.
- Implement **scheduled scripts** to update the matrix dynamically.

Zachman 6x6 Matrix Generation Example – 2

Here's a **detailed ServiceNow CMDB extraction script** that retrieves Configuration Items (CIs) and their attributes, then formats them into a **Zachman 6x6 matrix**.

1. Extract CMDB Data Using ServiceNow REST API

This script pulls data from the **CMDB_CI** table and organizes it into Zachman's framework.

```
# Beginning of Python code example
```

```
import requests
```

```
import json
```

```
import pandas as pd
```

```
# ServiceNow instance details
```

```
INSTANCE = "your-instance.service-now.com"
```

```
TABLE = "cmdb_ci"
```

```
USERNAME = "your-username"
```

```
PASSWORD = "your-password"
```

```
# API request to fetch CMDB data
```

```
url = f"https://{INSTANCE}/api/now/table/{TABLE}"
```

```
headers = {"Accept": "application/json"}
```

```
response = requests.get(url, auth=(USERNAME, PASSWORD), headers=headers)
```

```
# Parse JSON response
```

```
data = response.json()
```

TOGAF and Zachman - Two Approaches to Large Enterprise IT

```
ci_list = data.get("result", [])

# Define Zachman matrix structure
zachman_matrix = {
    "What": [],
    "How": [],
    "When": [],
    "Who": [],
    "Where": [],
    "Why": []
}

# Map CMDB attributes to Zachman matrix
for ci in ci_list:
    zachman_matrix["What"].append(ci.get("name", "Unknown"))
    zachman_matrix["How"].append(ci.get("sys_class_name", "Unknown"))
    zachman_matrix["When"].append(ci.get("install_date", "Unknown"))
    zachman_matrix["Who"].append(ci.get("manufacturer", "Unknown"))
    zachman_matrix["Where"].append(ci.get("location", "Unknown"))
    zachman_matrix["Why"].append(ci.get("description", "Unknown"))

# Convert to DataFrame for structured output
df = pd.DataFrame(zachman_matrix)
print(df)

# End of code example
```

2. Automate Matrix Generation in ServiceNow

- Use **Scheduled Jobs** to run this script periodically.
- Store results in a **custom table** for reporting.
- Export data to **Power BI or Tableau** for visualization.

3. Additional Enhancements

- Include **CMDB relationships** (e.g., cmdb_rel_ci) for deeper insights.
- Apply **filters** to extract only relevant CIs (e.g., servers, applications).
- Integrate with **ServiceNow Reporting** for dashboard visualization.

Summary

To address large enterprise IT challenges in terms of how to monitor, guide, and control IT development to support business, two approaches can be considered and used, i.e., TOGAF and the Zachman Framework. The two can be used as a combo, where TOGAF provides

TOGAF and Zachman - Two Approaches to Large Enterprise IT

methodology and Zachman does classification. Like pursuing an expedition, to achieve his goals, one must know what he has, where he starts, where he wants to end, and how to get there. In IT, Zachman allows the IT department to know what they have; TOGAF shows them what to do and how to do it right to reach their destinations.

References

[TOGAF | www.opengroup.org](http://www.opengroup.org)

[Zachman - A DEFINITIVE GUIDE – Enterprise Architecture Work](#)