The Zachman Framework For Enterprise Architecture and Rational Best Practices and Products

A Rational Software White paper

Rational
the e-development company™
Table of Contents

Introduction.................................................................................................................................................. 1

  WHAT is the Zachman Framework ? ............................................................................................................ 1
  WHO invented the Zachman Framework?.................................................................................................... 2
  WHEN was the Zachman Framework invented? .......................................................................................... 2
  HOW is the Zachman Framework used? ...................................................................................................... 2
  Where is the Zachman Framework used? .................................................................................................. 3
  WHY is the Zachman Framework relevant to Rational? ............................................................................. 3

Mapping Rational Best Practices and Products to the Zachman Framework ............................................... 3
  Differences between the Zachman Framework and the UML/RUP.............................................................. 3
  Zachman Framework Rows/ Roles ............................................................................................................ 4
  Zachman Framework Columns .................................................................................................................... 4
  Zachman Framework Cells ......................................................................................................................... 5
  Tables Notes............................................................................................................................................... 5

Critical Evaluation of the Zachman Framework and its impact on Rational..................................................... 5
  Support for Enterprise Asset Reuse ............................................................................................................ 6
  Bias Towards Data-Driven and Process-Decomposition Methods .............................................................. 6
  Tendency towards heavy-weight, ‘completist’, and detailed ‘top-down’ enterprise modeling......................... 6
  Divorced process, data and other views of the enterprise and its systems .................................................. 6
**Introduction**

The Zachman Framework for Enterprise Architecture, since its publication in the IBM Systems Journal in 1987\(^1\), has achieved a level of penetration in the domain of business and information systems architecture and modeling. This paper examines the key principles of the Zachman Framework and contrasts it with Rational best practices and products. It concludes that:

- The Zachman Framework operates in the same space as RUP\(^2\) and the UML, but does not compete directly with them, as it aims to provide neither a modeling language nor a development process;
- The Zachman Framework could act as a complementary partner to RUP, by supporting the establishment of a consistent and ordered set of reusable assets (patterns, frameworks, components etc.) off the back of one or more RUP projects;
- The Zachman Framework can have the effect of setting unhelpful agendas due its underlying philosophy and terminology which are steeped in a ‘process and data’ view of the world rather than a ‘use case, object and component’ view.


**WHAT is the Zachman Framework?**

The Zachman Framework is ‘a logical structure for classifying and organizing the descriptive representations of an Enterprise that are significant to the management of the Enterprise as well as to the development of the Enterprise’s systems\(^2\).’

It takes the form of the two-dimensional matrix, as shown in the diagram below.

---

**Figure 1: Zachman Enterprise Architecture – A Framework™ (John Zachman, Zachman International)**

---

The rows of the Zachman Framework matrix show different possible levels of abstraction, each of which is said to represent the perspective of a named role in the organisation. They move from a very high level of abstraction in the top of the matrix, which is called the ‘Scope’ or ‘Conceptual’ view and is said to represent the perspective of the ‘Planner’, to a very low level of abstraction in the bottom row of the matrix, which is called the ‘Detailed Representation’ view and is said to represent the perspective of the ‘Sub-Contractor’. (Note that Zachman heavily implies that rows 2 to 5 of the matrix (i.e. the perspectives of the ‘Owner’, ‘Designer’ and ‘Builder’) are the three pivotal model perspectives, and that the other two are of secondary importance.)

The columns of the Zachman matrix represent different types of abstractions, i.e. the different aspects of reality which a model can represent, such as ‘Data’, ‘Function’ etc. Each type of abstraction is mapped by Zachman to one of the six interrogatives: “WHAT?” (Data); “HOW?” (Function); “WHERE?” (Network); “WHO?” (People); “WHEN?” (Time) and “WHY?” (Motivation). (Note that Zachman seems to see three of these (Data, Function and Network) as being pivotal, and the others as being of only secondary importance.)

WHO invented the Zachman Framework?
The Zachman Framework was invented by John Zachman, who now operates Zachman International, an information systems planning and education consulting practice, and is the chairman of the Zachman Institute for Framework Advancement (ZIFA).

WHEN was the Zachman Framework invented?

Prior to the publication of the Zachman Framework, views of which models should be built were inextricably linked with the processes for building these models as part of monolithic, proprietary methods such as Information Engineering, SSADM and the like. The Zachman Framework, at the time, could therefore be said to represent a valuable unified view of the end-products of the modeling process, irrespective of the precise process which was used to deliver them.

Note that the Zachman Framework pre-dates the advent of all but the most prototypical object-oriented analysis and design methods and, in particular, it pre-dates the UML, which performed a similar service for the object community of disentangling a standard modeling language from the process of developing models expressed in that language.

HOW is the Zachman Framework used?
As the Zachman Framework is nothing more than a classification scheme – a static framework – the question arises as to what the framework should be used for. In general, John Zachman sees the framework as being:

- A PLANNING TOOL – which helps you position issues, see the total range of alternatives and therefore make informed choices;
- A PROBLEM-SOLVING TOOL – which tames the complexity of the business through abstraction, enabling the isolation of individual business variables.

Two more specific uses are implied by Zachman, both of which have immediate relevance for Rational, namely:

- Evaluating tools and methods by mapping them to the framework and therefore providing a neutral way of cataloguing what they do and do not support;
- A context for building up flexible, componentised architectures and systems, capable of supporting high rates of enterprise change, and replacing the ‘inventory of existing systems’ which are ‘not integrated’ as a result of being built ‘out-of-context’.

---

3 www.zifa.com
Where is the Zachman Framework used?
This is a question which I would like to enlist the help of the Rattle’s readership in answering. In the UK, we have come across a few large clients and prospects who espouse the Zachman Framework, most notably British Airways and the BBC. Past discussion threads within Rational technical e-mail groups suggest that there are others, but currently I have no clear picture of the level of penetration of the framework, nor the impact, for good or ill, which this tends to have on the take-up of Rational products and best practices. The author would therefore be grateful for any intelligence from the field on places where it has been encountered or is to known exist, and what impact it has had on us.

Why is the Zachman Framework relevant to Rational?
The Zachman Framework is relevant to Rational because it operates in exactly the same area as most of our best practices and products, namely models of the enterprise and its systems, from high-level business models to executing code models.

The good news should be that, where the Zachman Framework is found, there too should be a found a client that takes the need for modeling seriously, and which should therefore be potentially fertile ground for Rational messages and products.

The bad news is that, where the Zachman Framework is found, certain agendas will have been set concerning the nature of the modeling process and the concepts and terminology which should be used to describe it, and Rational may therefore be obliged to show conformance to, and/or justify divergence from, the principles and practices (both explicit and implicit) which underlie the Zachman Framework.

The rest of this article therefore examines how Rational best practices and products can be positioned within/ against the Zachman Framework.

**Mapping Rational Best Practices and Products to the Zachman Framework**

This section attempts to show objectively how Rational best practices and products map to the Zachman Framework, leaving critical evaluations of the merits and demerits of the framework as issues to be addressed in the final section.

**Differences between the Zachman Framework and the UML/RUP**

It should be remembered that the Zachman Framework is a classification mechanism for business and system models, which makes it a fundamentally different thing to both the Unified Modeling Language (a language for expressing business and system models) and the Rational Unified Process® (a process for delivering business and system models). Notice, in particular, that RUP explicitly supports projects, which produce specific deliverables for specific stakeholders within defined timescales and costs, whereas the Zachman Framework exists above the individual project level, and aims to accommodate and classify all the outputs of all the various modeling projects within an enterprise. (We might think of the Zachman Framework as the seabed where deposits of sediment are accumulated over time, and RUP as the river which shapes, carries and ultimately drops these deposits.)

We should, nonetheless, be able to perform a set of mappings from each cell in the Zachman Framework matrix to:

- The UML Elements which can be used to describe that particular abstraction at that particular level of abstraction;
- The RUP Artifacts which realize that particular abstraction at that particular level of abstraction.

Like any mapping of this kind it will inevitably be inexact and open to debate and interpretation, and as a result it is probably best described and justified initially in more general terms of how the rows and columns of the Zachman Framework matrix roughly map to areas of the UML and RUP, before a more specific set of mappings is attempted at the matrix cell level.

---

Zachman Framework Rows/Roles

The rows in the Zachman Framework represent levels of abstraction and are also defined in terms of the Roles in the enterprise whose perspective is defined by models of this level of abstraction. As such, they tend to map roughly to large-grained composite artifacts in RUP, such as the Business Model, Analysis Model, Design Model, Implementation Model etc., as detailed below:

1. Scope (Conceptual)/Planner – this model perspective does not map directly to any one RUP artifact, but some of the cells within it map roughly onto artifacts within the RUP Requirements Artifact Set, and others to high-level business model elements such as the RUP Business Use-Case Artifact. The Planner role perhaps best maps to a combination of the RUP Project Manager Worker and the Stakeholders identified by the RUP Vision Artifact. Note that this highest level mapping is by its nature the least meaningful, because it exists at a level above the individual project level at which the RUP operates.

2. Enterprise Model (Conceptual)/Owner – this perspective is roughly analogous to the RUP Business Model Artifact and the Owner role can therefore be said to be roughly analogous to the Business Analyst Worker that owns this model and the project Stakeholders which it primarily targets.

3. System Model (Logical)/Designer – this perspective is roughly analogous to a combination of the RUP Use-Case Model Artifact and Analysis Model Artifact and the Designer role can therefore perhaps be said to be roughly analogous to a combination of the Systems Analyst Worker and Architect Worker which respectively own these two models.

4. Technology Model (Physical)/Builder – this perspective is roughly analogous to a combination of the RUP Design Model Artifact and the Deployment Model Artifact, and the Builder role can be said to be roughly analogous to a combination of the Architect Worker that owns this model and Designer Worker that produces many of its constituent artifacts.

5. Detailed Representations (Out-Of-Context)/Sub-Contractor – this perspective is analogous to the RUP Implementation Model Artifact and the Sub-Contractor role can be said to be roughly analogous to the Designer Worker that produces many of this model’s constituent artifacts.

6. Functioning Enterprise – this represents the reality which is abstracted by the other rows of the matrix, and therefore is not strictly speaking part of the matrix of models or abstractions. It maps to the live systems within the enterprise which are the ultimate products of the software engineering business as described by the UML and delivered by the RUP.

Zachman Framework Columns

The columns in the Zachman Framework represent dimensions of abstraction, or aspects of reality being modeled. As such, they tend to map roughly to logical groupings of UML elements and diagrams, and will typically cut across many RUP activities and artifacts, as detailed below:

1. Data (What) – maps to structural UML constructs and related UML extensions and RUP artifacts, such as Business Objects, Classes, Objects, Attributes, Tables, etc.

2. Function (How) – maps to UML use-case model elements such as use cases, their business model stereotypes and their realizations.

3. Network (Where) – maps to a cross-section of model elements which are related to ‘where’ questions, such as physical deployment Nodes.

4. People (How) – maps to UML elements which describe humans and their interaction with the system, such as Actors and RUP User-Interface related artifacts.

5. Time (When) – maps to UML constructs such as Events, but often have no obvious UML or RUP equivalents in the higher levels of abstraction.

6. Motivation (Why) – maps to business rules as expressed in Stakeholder Requests, Supplementary and Special Requirements, Constraints and Notes, expressed in natural language or in the Object Constraint Language (OCL).
Zachman Framework Cells

The table below shows the RUP artifacts and/or Rational tool features which support each cell in the Zachman Framework. The mappings work better in some cases than in others, and in some cases the choice of individual mappings has been influenced by the desire to avoid repetition of Rational solutions across more than one matrix cell.

<table>
<thead>
<tr>
<th>Scope</th>
<th>Data</th>
<th>Function</th>
<th>Network</th>
<th>People</th>
<th>Time</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Rose-Analysis Class Model</td>
<td>ReqPro/ Rose - UseCase Model</td>
<td>Rose– Systems &amp; Subsystems</td>
<td>Rose- Actors &amp; User Interface Proto</td>
<td>Rose–Analysis Model Interaction Diagrams</td>
<td>ReqPro–Supplementary and Spec Requirements</td>
</tr>
<tr>
<td>Physical</td>
<td>Rose-Data Model</td>
<td>Rose – Design Class Model</td>
<td>Rose- Deployment Diagrams</td>
<td>Rose – User Interface Subsystem Design</td>
<td>Rose– Design Model Interaction Diagrams</td>
<td>Rose – Business Rules as Constraints</td>
</tr>
</tbody>
</table>

Tables Notes

The following notes relate to the table above:

1. Business Location is not a RUP or UML concept, but could easily be supported in Rational RequisitePro using a user-defined Requirement Type.
2. A less-than-perfect mapping.
3. Business Event is not a RUP or UML concept, but could easily be supported in Rational RequisitePro using a user-defined Requirement Type. Note that Major Business Events could alternatively be mapped to Business Actor / Business Use-Case association relationships.
4. Business Goal is not a RUP or UML concept, but could easily be supported in Rational RequisitePro using a user-defined Requirement Type.

Critical Evaluation of the Zachman Framework and its impact on Rational

This section provides a summary critical evaluation of the Zachman Framework and its positioning relative to Rational best practices and products.
**Support for Enterprise Asset Reuse**

One of the main strengths of the Zachman Framework is that it operates above and across the individual project level, and thus provides a framework within which to accumulate reusable assets such as patterns and components.

Thus, for example, if an application development project develops a set of business use-case and object models as part of the RUP Business Modeling Workflow, these might then be made available to a separate team responsible for refining project outputs into reusable models and patterns. This team could place these models inside an evolving ‘big picture’ of models of the business and its systems, so that they can be found and reused to support the scoping of new projects and the development of their business cases, requirements and analysis models.

Alternatively, projects might be justified, scoped and initiated purely to develop new models, such as high-level business use-case and business object models, using modified subsets of RUP, on the basis of the value that these will be able to deliver in the strategic planning and scoping processes for future business and IT change projects.

**Bias Towards Data-Driven and Process-Decomposition Methods**

The Zachman Framework positions itself as being neutral across methods, processes, modeling techniques and tools, and therefore as being able to offer a set of benchmarks against which an enterprise can evaluate these approaches and tools prior to their adoption. It is, however, questionable as to whether true neutrality of this kind can ever be achieved and, in fact, the Zachman Framework is based on a set of assumptions and terminology which implicitly align it with data-driven and process-decomposition methods and processes, rather than use-case-driven, object-oriented or component-based approaches. While this is mainly a question of presentation, it may act to bias users of the framework to be predisposed against recognition of the ability of UML and RUP based tools and processes to effectively support the models outlined in the Zachman Framework.

**Tendency towards heavy-weight, ‘completist’, and detailed ‘top-down’ enterprise modeling**

It is easy to fall into the trap of being seduced by the neatness and symmetry of the Zachman Framework, and the danger is that it can result in undertakings which become Quixotic quests to build and maintain a complete set of models which model every conceivable aspect of the enterprise at every conceivable level of abstraction, with no regard for how much these modeling enterprises cost nor what value they are delivering and to whom.

For example, the ‘Who, Where, Why, What, When and Why’ questions feel like a nice, neat and complete set of interrogatives, such that once you’ve used the first three or four you feel obliged to go on and complete the set despite the fact that the last two or three uses are increasingly artificial and serve more to distract than illuminate (witness the sub-headings within the first section of this article). Zachman himself implies that the What, How and Where are in general more important than the Who, When and Why, which are there more for completeness, but the Framework does not formalise this distinction between those elements that are genuinely valuable, and those that are largely there for completeness.

Since the Zachman Framework itself contains no process, it is devoid of advice in terms of what models should be built, when and why. The danger, therefore, is that adopters of the framework implicitly assume that the ultimate aim must be to build and maintain all levels of all types of model across the whole enterprise, and further conclude that a program should therefore be set in motion with the sole aim of doing just this. It need not be pointed out that a program such as this can easily become an endless, self-serving exercise which winds up delivering little by way of identifiable value to any specific business area or endeavour. Clearly, depending upon the domain and purpose of the models, particular abstractions may have more, less or no value, and again the Zachman Framework, in my experience, can act to discourage such a selective and deliverable-focused approach to modeling.

**Divorced process, data and other views of the enterprise and its systems**

In slicing up the set of models that can be built of the enterprise and its systems into a number of named columns, the Zachman Framework tends to encourage a high level divorce between these views of the enterprise, along the lines of those encouraged by data-driven and process-decomposition approaches, where models are progressed by drilling down these columns in the framework independently, and producing models which are highly stove-piped views of the enterprise.
The value of use-case driven, object-oriented and component-based models is that they show at any particular level of abstraction how function and form are interlinked and how they work together to achieve the goals of the system.

The Zachman Framework, in depicting the various model elements as being isolated within cells, has no way of depicting the links which must exist to integrate these model elements into useful views of the enterprise.