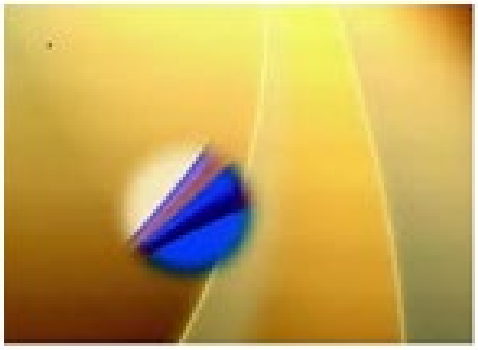
Visible® Analyst Tutorial

Visible Systems Corporation
24 School Street, 2nd floor
Boston, MA 02108
617-902-0767

www.visiblesystemscorp.com
https://twitter.com/VISIBLECorp

Email: contact@visiblesystemscorp.com

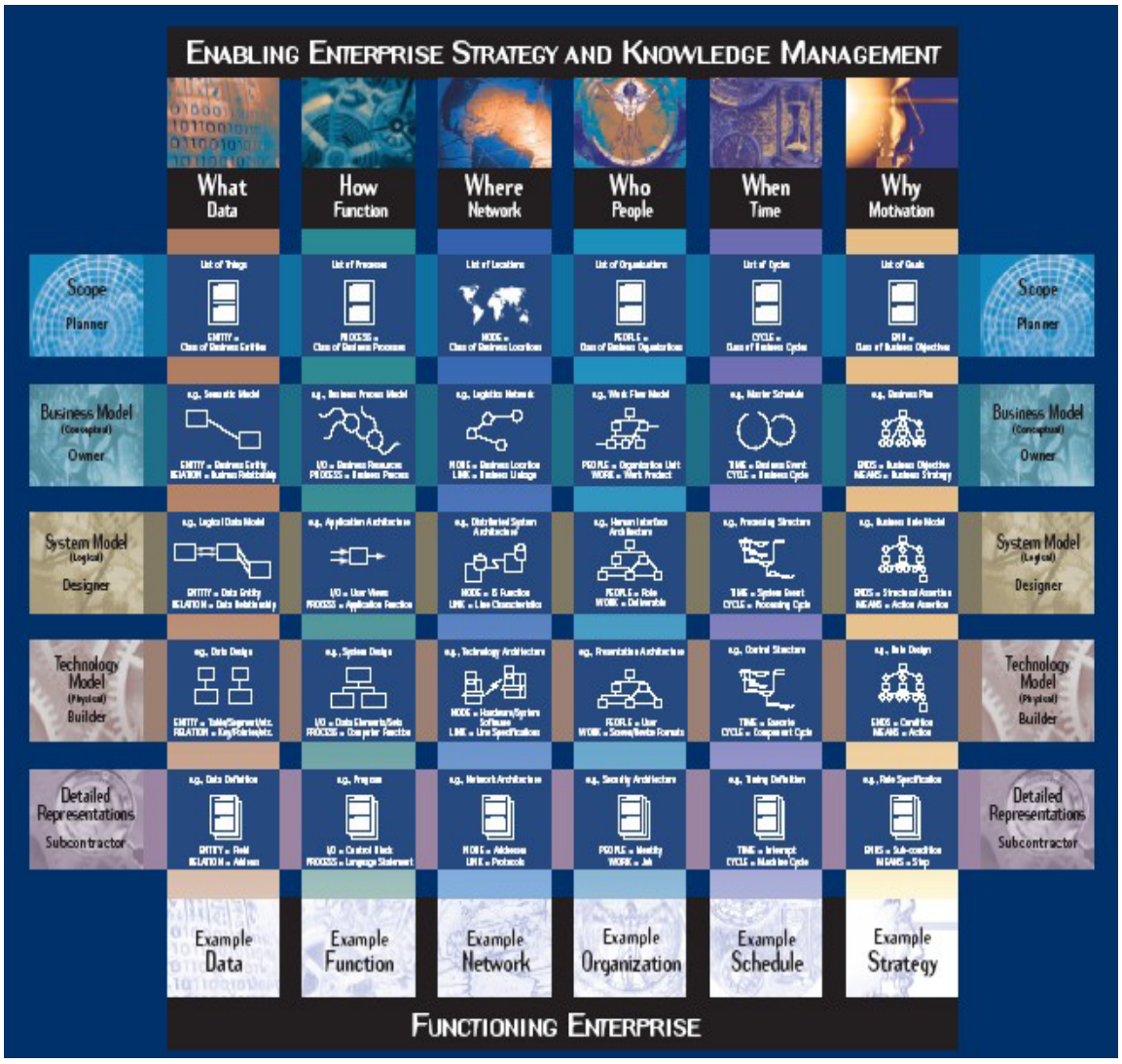


Visible®

Visualize. Align. Transform.™

Visualize patterns, align strategy and transform change into meaningful business outcomes.

Enterprise-wide Analysis, Design and Planning for Improvement.



Information in this document is subject to change without notice and does not represent a commitment on the part of Visible Systems Corporation. The software described in this document is furnished under a license agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of this agreement. It is against the law to copy the software onto any medium except as specifically allowed in the license or non-disclosure agreement.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or otherwise, including photocopying, reprinting, or recording, for any purpose without the express written permission of Visible Systems Corporation. Visible Systems Corporation makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Names, dates, and information used in examples in this manual are fictitious and only for examples.

Copyright 2008 – 2020 by Visible Systems Corporation, All rights

reserved. Printed and bound in the United States of America.

This manual was prepared using Microsoft Word for Windows.

Visible Analyst
Tutorial on Structured Methods, Repository Management and The Zachman Framework

Visible Analyst® is a registered trademark of Visible Systems Corporation.

The Zachman Framework illustration on the cover page of this tutorial was printed and used with the permission of the Intervista Institute © 2004 (www.intervista-institute.com). Microsoft and Windows are registered trademarks of Microsoft Corporation. Other product and company names are either trademarks or registered trademarks of their respective owners.

Dear Colleagues:

Thank you for your time in selecting our product, the Visible Analyst. At Visible, we take your time and effort seriously. To that end, we pride ourselves on delivering the most appropriate, value-oriented solutions. And, we feel that we offer the very best in product support that often differentiates us from our competitors.

As you read though the tutorial, please take the time to understand that our approach to software development is one of a model driven approach. Within the framework of this approach, Visible, in part, supports the Model Driven Architecture (MDA) as defined by the Object Management Group (OMG). This group, commonly referred to as the OMG, is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise wide applications. For more information about the OMG and in particular their MDA specification, please reference their web site at http://www.omg.org/mda/.

In conjunction with a model driven approach, Visible has incorporated a framework to enable you to better plan and manage your Enterprise Architecture effort. In this edition, The Zachman Framework, is the framework of choice. However, you can customize the Visible Analyst to implement other frameworks like, for example, the US Federal Enterprise Architecture Framework (FEAF).

Visible Systems Corporation. Visible Analyst, Visible Developer, Visible Data Governance, Visible Web Portal, Visible Self Service Data Discovery, Visible Sight (Context-driven business insights), Razor SCM, Polaris (Task Management).

The Zachman Framework

INTRODUCTION

It has been Visible Systems Corporation's experience that no matter where you start in your application development activities, you will soon find yourself making certain ―assumptions‖ about things not under your control or outside of your scope. To confirm or validate these assumptions, you find yourself addressing the artifacts up and down the Zachman Framework rows and/or across the columns to capture the true drivers for the system: who? what? where? when? why? and how?[1] This means coordinating with the affected or interested business experts, system users, and management.

In 1987 John Zachman wrote, ―To keep the business from *dis*integrating, the concept of an information systems architecture is becoming less of an option and more of a necessity.[2]‖ From that assertion over a decade ago, the Zachman Framework for Enterprise Architecture has evolved and become the model around which major organizations view and communicate their enterprise information infrastructure. The Zachman Framework draws upon the discipline of classical architecture to establish a common vocabulary and set perspectives--a framework--for defining and describing today's complex enterprise systems. Enterprise Architecture provides the blueprint--or architecture--for the organization's information infrastructure and provides a framework for managing information complexity and managing change.

Today the Zachman Framework has become a standard for Enterprise Architecture used by many of the most successful organizations in the world. Evidence of the acceptance of the Framework has been apparent at the annual forums conducted by the Zachman Institute for Framework Advancement (ZIFA, www.zifa.com). At each forum, attendees hear presentations on the many different aspects and practical uses of the Framework. *Visible* fully supports both the concept and philosophy of the Zachman Framework. *Visible* helps clients gain greater control of their information systems and technology requirements through development of an enterprise-wide architecture.

---

[1] ―Visible and the Zachman Framework for Enterprise Architecture‖ by Alan Perkins p. 2. Copyright © 1997-2001, Visible Systems Corporation.

[2] ―A framework for information system architecture‖ by J.A. Zachman p. 454 IBM Systems Journal, Vol. 26, Nos. 3, 1987, ©1987, 1999 IBM.

Visible takes an engineering approach to developing an enterprise architecture. We use a combination of forward and reverse engineering to establish the enterprise architecture. Forward engineering tasks include business planning and data and process modeling. Reverse engineering tasks include analysis and documentation of all existing structures for the organization. The result is a model that represents an integrated view of the enterprise architecture framework, with redundancies and discrepancies resolved and documented. All conceptual and logical architecture components can all be maintained in *Visible*'s proprietary modeling tool, *Visible Analyst®.*

The Visible Analyst supports the tasks and techniques involved in the creation and management of an enterprise architecture, with sufficient flexibility to integrate and support other approaches to software engineering. Visible Analyst captures business plans of multiple organization levels and maintains the hierarchy of planning components (mission, goals, strategies, measures, business rules, etc.).

Unlike many other modeling tools, Visible Analyst has the capability of directly linking each business plan component to the entities and attributes of a data model that support/implement the planning elements. This feature is used to control quality and completeness, and to ensure that process and system designs meet business requirements. Visible Analyst can also be used to specify physical information system designs based on the data model or import physical designs of existing data structures into the repository, and then link them back to the business plan component.

The following sections provide an overview of the Visible Analyst's repository and modeling capabilities, followed by an explanation of the Visible Analyst framework project. Each cell in the Zachman framework project is detailed in a cell-by-cell review including an explanation of the artifacts created for the cell in the Visible Analyst. A backup copy of the Zachman project is located in the VA\Zachman folder on the product CD. If you do not have a product CD, contact Visible Systems support at support@visiblesystemscorp.com for a copy of the document and backup file set. To access the project, see the Restore instructions at the beginning of this tutorial.

It is important to remember that the Visible Analyst Enterprise Project using the Zachman Framework is not a static one-time snapshot view of the enterprise. As mentioned in the cell explanations, the artifacts such as the business plan, physical data model, security architecture, strategic goals, etc. will change as the enterprise changes. Using the Visible Analyst and its repository to model the enterprise provides a one-stop location where all information about the enterprise is located. External documents may be changed, but the hyperlinks to the artifacts are maintained within the enterprise project, allowing for both a birds eye and physical implementation perspective of the enterprise.

Figure 2-1 Zachman Framework

Image provided courtesy of the Intervista Institute, Copyright © Intervista Institute (www.intervist-institute.com)

THE ZACHMAN FRAMEWORK PROJECT AND CELL DEFINITIONS

When implementing an Enterprise Architecture Framework, it is important where you begin. In our white paper, ―Enterprise Architecture Engineering‖[3] by Alan Perkins and Clive Finkelstein, available on our web site at www.visiblesystemscorp.com, they state, ―A well documented Enterprise Architecture is a logical organization of information pertaining to the following multi-level, multi-dimensional, enterprise wide elements.

- Strategic goals, objectives, strategies
- Business rules and measures
- Information requirements
- Processes, systems and applications
- Relationships between architecture elements
- Technology infrastructure‖

They emphasize that *the most important starting point* is that establishing the right sponsorship helps to insure successful development and deployment. Alan also explains, ―…all potential users of the applications and systems based upon the architecture must be involved in the process. Without both management sponsorship and near universal involvement, enterprise-wide architecture engineering projects usually fail.‖[4] Additional white papers are available on our web site at www.visiblesystemscorp.com that help explain the ―Critical Success Factors for Enterprise Architecture Engineering‖, ―Business Rules ARE Metadata‖, etc.

Each cell of the framework is described using the following format beginning with the ―What column Planner perspective‖ and proceeding down the column in a top-to-bottom left-to-right order.

- Cell location, label, perspective and descriptive type
- An explanation of the cell definition
- The artifact created in the project to implement the cell. The name and location of the cell is included in the artifact label where appropriate, such as "Row 4 Column 1 Physical Data Model".
- A detailed explanation of the project artifact
- Alternative artifacts that could be created in the project to implement the cell

---

[3] ―Enterprise Architecture Engineering‖ by Alan Perkins and Clive Finkelstein p.3. Copyright © 2000, Visible Systems Corporation.

[4] ―Critical Success Factors for Enterprise Architecture Engineering‖ by Alan Perkins p. 4. Copyright © 2000 Visible Systems Corporation.

This project contains many different types of artifacts, consisting of diagrams, strategic planning statements, lists, user defined objects, etc., and each was created to document a specific cell of the framework. Only one artifact was created for each cell, and additional artifact option types are included in this document and the cells' repository definition when appropriate. An actual enterprise project will have multiple artifacts representing a particular cell. When using the Visible Analyst in a real-world implementation of the framework, users should consider using the Enterprise Modeling feature (described in the online help system) to eliminate any naming conflicts, to maintain logical and physical data model separation, program specification, etc. The Enterprise Modeling feature will maintain the linkage between the artifacts in the projects, promoting object re-use.

Note

Appendix A in the ―Visible Analyst Framework Document‖, located in the VA\Zachman folder on the product CD, contains additional resources and modeling capabilities available in the Visible Analyst when creating artifacts for the various cells. If you do not have a product CD, contact Visible Systems support at support@visiblesystemscorp.com for a copy of the document.

Framework Rules

Before beginning the enterprise project and creating the cell artifacts, it is important that users know and understand the rules of the framework as described by John Zachman and John Sowa.[5]

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.

Accessing the Visible Analyst Project Artifacts

For users unfamiliar with the Visible Analyst we have included instructions to access the project artifacts maintained in the Visible Analyst Framework project.

[5] ―Extending and formalizing the framework for information systems architecture‖ by J. A. Sowa and J. A. Zachman, IBM Systems Journal, Vol. 31, No 3, 1992. Pages 599-603

Accessing the diagrams

Clicking the File | Open Diagram menu item displays a list of diagram types supported by the Visible Analyst in alphabetical order. If a diagram has been created a plus sign is displayed next to the diagram type and clicking the plus sign displays the diagram list. Double click on the diagram label to open the diagram.
When viewing the repository entry of a diagram symbol, double clicking on a diagram label listed in the Locations field on the Locations tab of the entry opens the diagram in the background and the selected item is highlighted on the diagram.

Accessing the repository

There are a number of ways to display the repository Define Item screen for an object even when a diagram is not displayed:

- Select the Repository | Define menu item to directly access the repository. If no diagram object or planning statement was highlighted, a blank Define item screen is displayed. Clicking the Next button at the bottom of the screen displays all repository entries in alphabetical order. Entering a letter into the label field and clicking the Search button displays a list of all repository entries beginning with that letter. Click the Search button and then the F1 key on the keyboard to display the ―Repository Searches – Overview‖ help file.
- Double click on the label of the item in the Object Browser.
- For diagram symbols and strategic planning statements, double clicking on the diagram symbol or statement opens the item‘s Define Item screen.
- Right click on the diagram object and select Define from the object menu, or left click on the object so it is highlighted and choose the Repository | Define menu item.
- Non-diagram repository object entries, such as data elements, data structures, user defined objects, etc., can be accessed by double clicking on the label of the item in the Object Browser or using the Repository | Define and Search feature mentioned above. When viewing a parent object‘s Define Item screen, such as an entity, class, data flow, etc., double click on the attribute item or select the attribute in the Attributes field and click the Jump button located at the bottom of the Define Item screen.

Accessing the strategic planning statements

Select the File | Strategic Planning menu item or click the Strategic Planning icon which is the second icon on the first menu bar. Use the up and down, left and right arrows on the strategic planning icon bar to position the statements within the statement hierarchy. The other strategic planning icons allow you to display the levels, branches, priority, type and description of the statements on the screen.

Column 1: The "Data" or "What" column

Provides an understanding of the data important to the business with finer amounts of detail shown at each succeeding perspective.

Row 1, Column 1 - "List of Things Important to the Business"

Objectives/Scope (Contextual)
Data column, Planner role
Entity = Class of Business Thing

Cell explanation[6]
A list of items, objects, assets, etc. important to the business and defined at a high level of aggregation. The list is dependent on the enterprise modeled, and ―…defines the scope, or boundaries, of Rows 2 – 5 models of things significant to the Enterprise‖[7]. A software or manufacturing company would include Vendors, Products, Clients, Product Facilities, etc. A law firm would include specific knowledge areas, trial experience, etc., while educators would include a curriculum, educational levels, specific teaching disciplines, etc.

Project implementation
Implemented as an Strategic Planning Statement

Artifact explanation
The strategic planning statement's Detailed Description repository field contains the list of items important to the business. This list includes Employees, Financial Resources, Accounting Procedures, Equipment and Technology, Profits etc. Using the Links tab of the statement's repository entry, this statement was linked to the Row 1 Column 1 ―List of things important to the business‖ cell.

Each item in the list could have been be added as a separate sub-statement, and the repository fields of the individual statements populated with the discrete explanation of the items. Hyperlinks to external documents, which further describe this high-level aggregation of the business, can be created if necessary.

---

[6] The cell definitions are based on the cell explanations as described in the following documents:
"The Framework for Enterprise Architecture Cell Definitions" ZIFA 03.doc Copyright © Zachman Institute for Framework Advancement www.zifa.com
and
"A different Kind of Life Cycle: The Zachman Framework" by David C. Hay, Essential Strategies Copyright © 2000, Essential Strategies, Inc www.essentialstrategies.com

[7] "The Framework for Enterprise Architecture Cell Definitions" ZIFA 03.doc Copyright © Zachman Institute for Framework Advancement www.zifa.com

Alternative project implementations

- A User Defined Object of type ―Business Object‖ can be created and implemented as an item in the list, with each item maintaining separate repository entries that can be linked to other cell artifacts.
- The list can be maintained in any word processing application and a hyperlink to the file created in any one of the cells descriptive-type fields (notes, detailed description, etc.) in the repository.

Row 2, Column 1 - "Semantic Model"

Enterprise Model (Conceptual)
Data column, Owner role
Entity = Business Entity
Relation = Business Relationship

Cell explanation:
Contains a model of the things[8] important to the business, as seen by the participants in the business, and is modeled as a high-level entity relationship diagram. These relationships are later implemented as business rules.[9]

Project implementation:
Implemented as an Entity Relationship diagram.

Artifact explanation:
This conceptual data model diagram contains a model of the high-level business objects and the relationships maintained between the objects. Entities include Company Employees, Company Management, Company Business Relationships, Products, etc. The relationships model the business concepts between the entities, such as Employees - *design* - Products; Employees – *produce* – Products; Company Management – *acquires* – Capital Resources, etc.

Alternative project artifact implementations:
A class diagram could be used to model this cell with the classes identifying the business objects and the relationships between these objects defining the business concepts.

---

[8] "A different Kind of Life Cycle: The Zachman Framework" by David C. Hay, Essential Strategies Copyright © 2000, Essential Strategies, Inc. www.essentialstrategies.com

[9] "The Framework for Enterprise Architecture Cell Definitions" ZIFA 03.doc Copyright © Zachman Institute for Framework Advancement www.zifa.com

Row 3, Column 1 - “Logical Data Model”

System Model (Logical)
Data column, Designer role
Entity = Data Entity
Relation = Data Relationship

Cell explanation
The Technology neutral fully normalized logical data model with attributes and unique identifiers defined to record information important to the business.

Project implementation
Implemented as an Entity Relationship diagram

Artifact explanation
The entities involved in the logical data model can be modeled on one global diagram, and then separate subset entity relationship diagrams created if desirable. Note that key relationships between entities in the Visible Analyst extend across the diagrams for purposes of model analysis to provide additional analysis / and verification of the models.

The subset area diagrams can be copied to satellite projects using the Enterprise Copy feature and implemented as physical data models while maintaining the relationships to the logical data model.

Note

- A physical data model can be reverse engineered from any ODBC compliant RDBMS, and the physical model used as the basis for creating a new logical data model.

Alternative project artifact implementation
A class diagram can also be used to model this cell.

Row 4, Column 1 - “Physical Data Model”

Technology Model (Physical)
Data column, Builder role
Entity = Segment/Table
Relation = Pointer/Key

Cell explanation
The entities in the subject areas are converted into table definitions of a technology constrained fully attributed entity relationship model. All keys, indexes, table and column check constraints, database storage information, database views, stored procedures, etc., are defined for implementation into a specific RDBMS.

Project implementation
Implemented as an Entity Relationship diagram

Artifact explanation
The fully attributed entities and relationships are added to entity relationship diagram(s) with corresponding Visible Analyst repository entries. All physical information about the entities and elements is defined, including primary, foreign and alternate keys; unique and non-unique indexes; table and column check constraints; database storage information; database views; stored procedures; triggers; etc. Each diagram can be modeled to correspond to specific business subject areas, such as Accounting, Shipping, Sales, etc. and these individual diagrams used as the basis of the generated SQL DDL.

Note

- The RDBMS tables, attributes, keys, index, trigger, stored procedure, tablespace information, database views etc., can be reverse engineered from the existing RDBMS and used to populate a Visible Analyst project. Diagrams can automatically be generated to display the imported tables, views and relationships, and data elements. Foreign keys can be inferred during the reverse engineering procedure to auto generate relationships if none are defined.

Alternative project artifact implementation
A class diagram can be used to model the physical information, and once the classes are copied to an entity diagram, SQL DDL can be generated.

Row 5, Column 1 - “Data Definition”

Detailed Representations (Out-of-Context)
Data column, Sub-Contractor role
Entity = Field
Relation = Address

Cell explanation
The artifact is the implementation and data definition of the tables and column in the specific RDBMS, as well as the SQL DDL script.

Project implementation
A User Defined Object of type ―Database‖ was created and implemented as the repository object ―SQL Server Database‖.

Artifact explanation
This ―SQL Server Database‖ user-defined object functions in 2 ways:

1. As a container object to list all of the entities associated with a business area implemented in a specific database(s). Entities can be listed in many of these user defined database objects.
2. As the Visible Analyst repository entry linked to the implemented code, which can be stored in a source control application or stored in an external file. When a source code control application is used, the objects Links To field on the Links tab lists the connection to the source code control application. Otherwise, a hyperlink is created to connect the object to the external file containing the SQL DDL script.

The generated SQL DDL code could be pasted into the objects Notes field or a user-defined attribute could be created to store the SQL DDL code as part of this object‘s repository entry.

Alternative project artifact implementation(s)
A pre-defined Visible Analyst ―Cluster‖ repository object is used to maintain a listing of entities that can be displayed as one symbol on a diagram. Its purpose is to reduce the amount of displayed detail. The cluster object would be used to define the entities implemented in a specific database based on a specific diagram. The External Link to the source code control application would be entered in the ―Links To‖ field on the cluster‘s Links tab. Note that entities can only exist within one cluster.

Column 2: The “Function” or “How” column

Describes the process and functions performed by the business. Additional detail is displayed for each succeeding perspective.

Row 1, Column 2 - “List of Processes the Business Performs”

Objectives/Scope (Contextual)
How column, Planner role
Function = Class of Business Processes

Cell explanation
This cell lists the processes /activities the business performs.

Project implementation
Implemented as a Functional Decomposition diagram.

Artifact explanation
The Functional Decomposition Diagram was chosen because the symbols allow the user to display the high-level business functions and processes in a hierarchical relationship. Each methodology symbol maintains a separate repository entry allowing the user to fully describe the function/process, and include hyperlinks to external documents if necessary. Through the use of off-page connectors, each function and its sub-processes can be modeled on separate multi page diagrams and copied to a satellite project if necessary for further decomposition. The Functional Decomposition Diagrams also can be used to spawn a high-level data flow diagram that segues into the next cell in the column, Business Process Modeling.

Alternative project artifact implementation(s)

- The Strategic Planning Statements can be used to define the business functions and high-level child processes in the statement hierarchy.
- A hyperlink from this cell to an external document listing the functions and processes can be used to link the cell to the document.

Row 2, Column 2 - “Business Process Model”

Enterprise Model (Conceptual)
How column, Owner role
Process = Business Process
I/O = Business Resource

Cell explanation
The activities of the business function and processes are described independent of system implementation. The inputs and outputs describe the business resources.

Project implementation
Implemented as the Data Flow diagram
"Row 2 Column 2 Business Process Model"

Artifact explanation
The data flow diagram is specifically suited for modeling the business processes, external influences and the input and outputs of the processes. Nest relationships are created when a process is exploded to a child diagram where more detailed information is defined. The split data flow feature is useful for decomposing the high-level inputs and outputs to show granular details on the lower level diagrams. The repository entries for the diagram symbols capture the process and data details in excruciating detail. The model balancing analysis confirms the integration of the lower level processes with the parent processes.

Alternative project artifact implementation(s)

- The Business Process Model (BPMN) diagram can also be used to model the business processes.
- The functional decomposition diagram can be used to model the business processes
- The activity diagram can be used to model the business processes.

Row 3, Column 2 - "Application Architecture"

System Model (Logical)
How column, Designer role
Process = Application Function
I/O = User Views

Cell explanation
An information perspective of the business processes explaining the controls and mechanisms and conversion of input data to output data.

Project implementation
Implemented as a Business Process Model diagram.

Artifact explanation
The Business Process Model diagram, using the BPMN notation developed by the Business Process Management Initiative (www.bpmi.org) and the Object Management Group (OMG www.omg.org) is specifically suited for modeling the business processes. These diagram models communicate the business processes including the business processes, the events (triggers) that begin, end or interrupt the processes, and the information

(artifacts) used and developed by the processes. The BPMN notation supports Private (internal), Abstract (public) and Collaboration (global processes).

The repository entries for the diagram symbols capture the processes, events and data details in excruciating detail. The model analysis confirms the integration of the lower level processes with the associated model items.

Alternative project artifact implementation(s)

- The data flow diagram can be used as the artifact to define this cell.
- An activity diagram can be used to show the high-level inputs, processes and synchronization of the application architecture.
- A class diagram could also be used to define the business users, the methods of the business, and the relationships between the business users.
- A Use Case diagram can also be used and then ―nested‖ to an Activity diagram, where the inputs and outputs can be shown interacting with the business processes.

Row 4, Column 2 - "System Design"

Technology Constrained Model (Physical)
How column, Builder role
Process = Computer Function
I/O = Data Element Sets

Cell explanation

This system design is converted into to the module definitions or class methods, or BPEL if a BPMN diagram is used. A high level of abstraction is necessary to model this cell. BPEL generation is not yet available in the Visible Analyst.

Project implementation

Implemented as a Business Process Model Diagram.

Artifact explanation

The Business Process model was selected for its ability to model the activities, decisions, roles, as well as events and states of the system design. Concurrency of actions to be completed before processing can continue can be modeled and BPEL could be generated based on the design. BPEL generation is not yet available in the Visible Analyst.

Alternative project artifact implementation(s)

- Structure Chart diagrams could be used to model the programs architecture, i.e. calling structure and information passed from module to module.
- A data flow diagram can be used as an alternative diagram artifact

- An activity diagram can also be used because of its capabilities to include the events and concurrency of the processes.
- A Sequence diagram could also be used to define the calling structure and methods.
- A class diagram can also be used.

Row 5, Column 2 - “Program”

Detailed Representations (Out-of-Context)
How column, Sub-Contractor role
Process = Language Statement
I/O = Control Block

Cell explanation
The programs designed in the above columns are converted / compiled into the actual running programs

Project implementation
The Visible Analyst repository has a predefined repository object type of "Program", which is used to link to the program code stored in a source code control application.

Artifact explanation
This "program" object can be linked to the external code maintained in a source code control application such as RAZOR or Visual Source Safe using the Links To field on the programs Links tab. All methods associated with classes are stored in the Visible Analyst repository with an entry type of "module". These modules can be added to the composition field of the program object, detailing which modules are used in the program. Additionally, structure chart diagrams or sequence diagrams can be used to model the modules and calling structure of the program.

Inclusion of a hyperlink to sections of the code such as header files or code files written in C, C++, C#, VB files, .Net .sln files, etc. can also be created.

Alternative project artifact implementation(s)
- Creation of a user defined object similar to the program repository object mentioned above to maintain a link to the source code control application storing the generated program code.
- Link to Visible Developer, which creates the 3-tier business object program as ASP, VB6 or .Net code.
- A structure chart diagram can also be used to model the program and be the sequence object linked to the code.

Column 3: The “Network” or “Where” column

Describes the geographical distribution of the enterprise‘s activities.

Row 1, Column 3 - “List of Locations in Which the Business Operates”

Objectives/Scope (Contextual)
Where column, Planner role
Node = Major Business Location

Cell explanation
A list of locations where the business operates.

Project implementation
Implemented as a Functional Decomposition diagram.

Artifact explanation
The functional decomposition diagram was chosen to create a hierarchy of the business architecture with the corresponding repository entries providing fields to maintain a detailed description of the location. Hyperlinks to external items for each location can also be included as part of the locations repository entry, to record contract information, rules and regulations specific to the location, etc.

Alternative project artifact implementation(s)

- Strategic planning statements could be used to describe each location, with subsidiary locations defined as sub-statements.
- A _locations‘ user defined object could be created in the repository, and hyperlinks created to reference the external contracts, rules and regulations, etc as noted above.

Row 2 Column 3 “Logistics Network”

Enterprise Model (Conceptual)
Where column, Owner role
Node = Business Location
Link = Business Linkage

Cell explanation
The detailed communications chart, listing the communications network and the protocols used, such as voice, data, post, rail, shipping, etc. and how the locations interact.
Project implementation
Implemented as a Structure Chart diagram.

Artifact explanation

The structure chart diagram type was selected so that the nodes could be modeled as modules and the links signifying the individual communications between the modules defined as data couples. These couples as well as the modules maintain repository entries allowing for a detailed description of the communications nodes and links. Hyperlinks to external information can also be included in the repository definitions. Additional details of the diagram symbols and the objects they represent can be defined in the repository using user-defined attributes as necessary.

Each location can be modeled independently but connected to the main location diagram via on-page or off-page connections.

Alternative project artifact implementation(s)

- A planning statement or user-defined object can be created to reference this cell, and a hyperlink to an external application supporting a network diagram can be created.
- Hyperlinks to other documents or artifacts associated with this cell but modeled externally can be created.

Row 3, Column 3 - "Distributed System Architecture"

System Model (Logical)
Where column, Designer role
Node = I/S Function (Processor, Storage, etc.)
Link = Line Characteristics

Cell explanation:
The architecture of the data distribution, where it is created, and where used. Technology neutral, it would contain the descriptions of the system facilities, ―…controlling software at the nodes and lines (processors/operating systems, storage devices/DBMS', peripherals/drivers, lines/line operation systems, etc)‖.[10]

Project implementation:
Implemented as a Structure Chart diagram.

Artifact explanation
The structure chart diagram was used so that the data couples signifying the Links show the transfer of the information between the module symbols as Nodes on the diagram. Additional

[10] The cell definitions are based on the cell explanations as described in the following documents:
"The Framework for Enterprise Architecture Cell Definitions" ZIFA 03.doc Copyright © Zachman Institute for Framework Advancement www.zifa.com
and
"A different Kind of Life Cycle: The Zachman Framework" by David C. Hay, Essential Strategies Copyright © 2000, Essential Strategies, Inc www.essentialstrategies.com

details of the diagram symbols and the objects they represent can be defined in the repository using user-defined attributes.

Alternative project artifact implementation(s)

- A planning statement or user-defined object can be created to reference this cell, and a hyperlink to an external application supporting a system architecture diagram can be created.
- Hyperlinks to other documents or artifacts associated with this cell but modeled externally can be created.

Row 4, Column 3 - "Technology Architecture"

Technology Constrained Model (Physical)
Where column, Builder role
Node = Hardware/System Software
Link = Line Specification

Cell explanation

Shows the physical design of the computer facilities including the details of the hardware and software used at the business locations.

Project implementation

Implemented as a Structure Chart diagram.

Artifact explanation

The structure chart diagram was used so that the data couples signifying the Links show the transfer of the information between the module symbols as Nodes on the diagram. Additional details of the diagram symbols and the objects they represent can be defined in the repository using user-defined attributes.

Alternative project artifact implementation(s)

- A planning statement or user-defined object can be created to reference this cell, and a hyperlink to an external application supporting a technology architecture diagram can be created.
- Hyperlinks from the cell to other documents or artifacts associated with this cell but modeled externally can be created.

Row 5 Column 3 “Network Architecture”

Detailed Representations (Out-of-Context)
Where column, Sub-Contractor role
Node= Address
Link = Protocol

Cell explanation
The definitions of the node address and line specification, which are translated into specifications of particular protocols, communication facilities, etc.,[8] are defined in this cell.

Project implementation
Implemented as the User-Defined Object of type‖ Architecture‖ and the repository entry "Row 5 Column 3 Network Architecture Implementation".

Artifact explanation
This user defined objects‘ text fields are used to maintain the network architecture information. It can be hyperlinked to an external application that models network architecture, or hyperlinked to external documents describing the architecture.

Alternative project artifact implementation
A planning statement could be used as a ―container‖ object to maintain information about the network implementation.

[8] "A different Kind of Life Cycle: The Zachman Framework" by David C. Hay, Essential Strategies Copyright © 2000, Essential Strategies, Inc. www.essentialstrategies.com

Column 4: The “People” or “Who” column

Those involved in the business and their relationship to the technology.

Row 1, Column 4 - “List of Organizations Important to the Business”

Objectives/Scope (Contextual)
Who column, Planner role
People = Class of Agent

Cell explanation
A list of people and organizations important to the business, including organizational units and their scope and boundaries is the artifact created for this cell.

Project implementation
Implemented using a Strategic Planning Hierarchy statement.

Artifact explanation
Only one planning statement was used to identify the people and organizations important to the business. Practically, each person, organization and organizational unit should be entered as sub-statements in the statement hierarchy to maintain individual repository entries. This procedure facilitates the definition of the person / unit especially when hyperlinks are created to external documents describing the relationship. Contact information with vendors; venture capital contracts; rental agreements; technology contracts; shipping agreements are some simple examples of the additional documentation associated with the people and organizations important to the business.

Note

- Not all users should be granted access to the sensitive business documents. In some cases a listing of the documents may be sufficient to define the artifact rather than a hyperlink to the actual documents themselves.

Alternative project artifact implementation(s)

- A functional decomposition diagram could also be used to identify the business units and the individuals, organizations and organizational units in a hierarchical diagram.
- A user-defined object could also be used to identify the people and organizations important to the business.

Row 2, Column 4 - "The WorkFlow Model"

Enterprise Model (Conceptual)
Who column, Owner role
People = Organizational Unit
Work = Work Product

Cell explanation
Allocation of responsibilities as described in an organizational chart with secondary documents defining the products. Security requirements are also included within this cell.

Project implementation
Implemented as a Business Process Model diagram.

Artifact explanation
The Business Process Model diagram type was chosen because it models concurrent actions to be completed before the next action begins along with the inputs and outputs. The model also includes the use of Swimlanes (Pools) to categorize the activities performed by the respective roles of the business users.

Alternative project artifact implementation(s)

- Data flow diagrams can be used to model the organizations, organizational units and processes performed.
- An Activity diagram can also be used to model the workflow.
- A functional decomposition diagram can be used to model the Organization Chart.
- A Use Case could also be used, with links to a nested activity or collaboration diagram modeling the work products.

Row 3, Column 4 - "Human Interface Architecture"

System Model (Logical)
Who column, Designer role
People = Role
Work = Deliverable

Cell explanation
Defines the people, their roles and responsibilities and interacting with the technology to create the deliverables.

Project implementation
Implemented as a Use Case diagram.

Artifact explanation
The Use Case diagram captures the interaction of the people and the work deliverables. Nested links to an activity diagram including the use of user-defined attributes and the use of hyperlinks to the deliverables can be modeled to show additional detail.

Alternative project artifact implementation(s)
- Data flow diagrams can be used to model the processes performed by the organizations interacting with the technology and resulting deliverables.
- A functional decomposition diagram can be used to model the interactions and the deliverables.

Row 4, Column 4 - "Presentation Architecture"

Technology Constrained Model (Physical)
Who column, Builder role
People = User
Work = Screen Format

Cell explanation
The actual interface is modeled with presentation formats including screens, navigation paths, security rules, etc.

Project implementation
This cell was implemented as a Use Case diagram.

Artifact explanation
The Use Case diagram can also be nested to an activity diagram. Each of the repository entries can be tied to the implementation code, such as the screen design as shown in the user interface code generated by Visible Developer. The security considerations can be modeled as user-defined attributes, separate user defined objects, or as planning statements and each of these repository objects linked to the appropriate Use Case symbol artifact. Hyperlinks to some external tools can also be created as necessary.

Alternative project artifact implementation(s)
- A database view object can be used to list the data elements used in the menu screens, and the Extended Attributes tab of the elements repository definition used to store the presentation information.
- A hyperlink from this cell can be used to a human interface architecture diagram or the screen configuration files developed in an external application.

Row 5, Column 4 - “Security Architecture”

Detailed Representations (Out-of-Context)
Who column, Sub-Contractor role
People = Identity
Work = Job

Cell explanation
Individual‘s program access permissions and work they are authorized to perform.

Project implementation
Implemented as a Class diagram.

Artifact explanation
Implemented as a class diagram with the class representing the users, programs, and the elements defining the class data such as permissions, security mechanisms, etc. Methods can also be defined for classes as an additional level of detail. Hyperlinks to the code stored in a configuration management application can also be created.

Alternative project artifact implementation
An entity diagram can be used with a user-defined attribute or user-defined object substituting for the method‘s definition.

Column 5: The "Time" or "When" column

Used to describe the effects of time on the business, and interacts with column 2, the How column.

Row 1, Column 5 - "List of Events Significant to the Business"

Objectives/Scope (Contextual)
When column, Planner role
Time = Major Business Event

Cell explanation:
A description of the business cycle and when events significant to the business occur.

Project implementation:
Implemented as a Planning Statement.

Artifact explanation:
The events are modeled as subset planning statements allowing for further definition and linkage to other artifact items listed in subsequent How columns.

Alternative project artifact implementation(s):

- A used-defined object could contain this list.
- Hyperlinks from the cell's definition to external documents describing the event.

Row 2, Column 5 - "Master Schedule"

Enterprise Model (Conceptual)
When column, Owner role
Time = Business Event
Cycle = Business Cycle

Cell explanation
When the business functions occur, including the initiating event and the processing order.

Project implementation
Implemented as a Business Process Model diagram.

Artifact explanation
The Business Process Model diagram models the business events, processes and when functions are to happen and under what circumstances.

Alternative project artifact implementation(s)

- A state transition diagram can be used to model this cell.
- An Entity Life History diagram can be used to model this cell.
- An activity diagram can be used to model this cell.
- A list of events and time lines can be defined as a user defined object or as external documents hyperlinked to the cell.

Row 3, Column 5 - "Processing Structure"

System Model (Logical)
When column, Designer role
Time = System Event
Cycle = Processing Cycle

Cell explanation

Model of the system events and times to complete the data transformation processes and entity state changes.

Project implementation

Implemented as a State Transition diagram.

Artifact explanation

The state transition diagram works well to show the states of the system and the events causing the change in state. Detailed information is documented in the appropriate repository fields with additional user defined attributes added as necessary.

The Business Process Model diagram, if used, can display the business system processes and the events causing the change in state. Detailed information is documented in the appropriate repository fields with additional user defined attributes added as necessary.

Alternative project artifact implementation(s)

- A data flow diagram can be used to model this cell.
- A state transition diagram can be used to model this cell.
- The activity diagram can be used to model this cell.
- The collaboration diagram can be used to model this cell.
- The sequence diagram can be used to model this cell.

Row 4, Column 5 - "Control Structure"

Technology Constrained Model (Physical)
When Column, Builder role
Time = Execute

Cycle = Component Cycle

Cell explanation
Triggers, messages, responses etc, described as system events with physical properties and processing cycles detailed.

Project implementation:
Implemented as a Sequence Diagram.

Artifact explanation:
The sequence diagram models the calling structure of the programs and the returns, etc. The details are stored in the appropriate repository fields with additional user defined attributes added as necessary.

Alternative project artifact implementation(s)

- The state transition diagram can be used to model this cell.
- A business process model can be used to model this cell.
- The structure chart diagram can be used to model this cell.
- The collaboration diagram can be used to model this cell.

Row 5, Column 5 - "Timing Definition"

Detailed Representations (Out-of-Context)
When Column, Sub-Contractor role
Time = Interrupt
Cycle = Machine Cycle

Cell explanation
Schedule Online and Batch applications (Function Details), showing the interrupts and machine cycles.

Project implementation
Implemented as a Collaboration diagram.

Artifact explanation
The collaboration diagram shows the timing of the application through the use of the messages that implement the business scenario.

Alternative project artifact implementation
A sequence diagram can also be used to model this cell.

Column 6: The “Motivation” or “Why” column

Translation of the business goals and strategies into the ends and means of the business.

Row 1, Column 6 - “List of Business Goals/Strategies”

Objectives/Scope (Contextual)
Why column, Planner role
Ends/Means = Major Business Goal / Critical Success Factor

Cell explanation
The goals and strategies of the business are identified.

Project implementation
Implemented as a Strategic Planning Statement.

Artifact explanation
The strategic planning statement hierarchy is specifically suited to create the artifacts necessary for this cell. The users can extend the statement types, and an editable priority field is available for assignment to each statement in addition to the predefined repository fields. Links to the other artifacts can be defined in the Links To field on the Links tab in the repository. Hyperlinks to external documents can be created as necessary.

Alternative project artifact implementation(s)

- The functional decomposition diagram can be used to define the hierarchy.
- Hyperlinks to an external document or another statement hierarchy application can be used.

Row 2, Column 6 - “Business Plan”

Enterprise Model (Conceptual)
Why column, Owner role
End = Business Objective
Means = Business Strategy

Cell explanation
The business plan contains the strategies, goals, financial considerations and motivation of the company. These artifacts can include both textual descriptions as well as financial documents.

Project implementation
Implemented as a Planning Statement hierarchy.

Artifact explanation
Individual strategic planning statements should be defined and can include references to external documents and artifacts hyperlinked to the statement. The ―Cost Structure‖, ―Capital Funding‖ statements might reference MS Excel spreadsheets, while the textual description of the ―Business Plan‖ statement contains a hyperlink(s) to MS Word document(s).
Note that these statements are linked to the functional decomposition diagram symbols repository entries as an example of the artifact integration available in the Visible Analyst.

Alternative project artifact implementation(s)

- An alternative implementation is the functional decomposition diagram "Row 2 Column 6 Business Plan Hierarchy". The decomposition diagram allows the artifacts to be listed as a hierarchy and would include hyperlinks to the external documents the symbols represent. While each artifact can be represented within a symbols Notes field or as a user-defined attribute, maintaining them external to the application allows these artifacts to be updated and maintained in one place while still linking to the symbol in the enterprise project. Note that the decomposition diagram symbols are linked to the individual planning statements to demonstrate the cross artifact reference capability in the Visible Analyst.
- A class diagram could be used to diagram the business plan and the methods used to detail the business constraints.

Row 3, Column 6 - “Business Rule Model”

System Model (Logical)
Why column, Designer role
End = Structural Assertion
Means = Action Assertion

Cell explanation
Business rules can be considered as the metadata of the enterprise to include the intents and means of the business, and are part of the information implemented as checks on the database and enterprise information. Examples of these business rules as metadata include Definitions of Business Terms; Data integrity constraints; Mathematical and functional derivations; Logical inferences; Processing sequences; Relationships among facts about the business, etc.

Project implementation
Implemented as a Planning Statement hierarchy.

Artifact explanation
The strategic planning hierarchy provides the structured hierarchy allowing the metadata to be defined, and most importantly, to be linked to the implementation of these rules to the data.

The tables, columns, check constraints, business processes, database access security rules, etc. that implement the business rules are linked to the business rule statement.

Alternative project artifact implementation
Create a hyperlink from this cell to a User Defined Object created to store these business rules.

Row 4, Column 6 - “Rule Design”

Technology Constrained Model (Physical)
Why column, Builder role
End = Condition
Means = Action

Cell explanation
This cell describes the physical specifications of the Business Rules.

Project implementation
Implemented as a User Defined Object of type ―Rule Design‖ labeled ―Row 4 Column 6 Rule Design‖, linked to the repository entries that implement the rule design, such as the relationship cardinality between entities / classes, column or table check constraints, a business process that enforces these rules, etc.

Artifact explanation
Since this cell describes the physical specifications of the Business Rules, implementation of these rules can also be enforced as part of the relationship cardinality or as table or column check constraints on the Class | Entity in the repository. (Remember, entities can be used on class diagrams and methods, keys, constraints, etc., can be defined and the class/entity used for SQL DDL and code generation). Additional user-defined attributes can be added to the project as necessary to store specific textual descriptions of the specifications.

Note

- There is some agreement in the enterprise community that the UML OCL Language be used to represent the artifacts of this cell. Enter the following links into a web browser for an explanation of the OCL language. The artifacts referenced by the OCL can natively be created in the Visible Analyst using the supported diagram types or identified as a user defined attribute(s).

Alternative project artifact implementation(s)

- Program rule design can also be detailed in the methods associated with a class, either defined on the Class or Sequence diagram.

- The cell‘s repository entry may also contain hyperlinks to appropriate external applications.

Row 5, Column 6 - “Rule Specification”

Detailed Representations (Out-of-Context)
Why column, Sub-Contractor role
End = Sub-condition
Means = Step

Cell explanation
Enforcement of the business rules in the programs are the artifacts.

Project implementation
Implemented as Application modules and Database tables, Data and Function Details. Hyperlinks from these repository entries to the implementation artifacts (programs) from the cell can be created.

Artifact explanation
The Rule Design artifacts as defined in the previous cell in this column are implemented in the code and applications as part of the Application modules, Database tables, Data and Function Details, etc. These programs or code should be hyperlinked to the Rule Design artifacts.

Alternative project artifact implementation
Application modules could be detailed as sequence or structure chart diagrams, but it is more appropriate to link to the code implementing the rules.