Presented at INCOSE 2001
Eleventh Annual International Symposium of the
International Council on Systems Engineering (INCOSE)
Melbourne, Victoria, AUSTRALIA 1 - 5 July 2001

Definition of Enterprise Architecture-centric Models for the Systems Engineer

Gundars Osvalds
Northrop Grumman/TASC
2701 Technology Drive
Annapolis Junction, MD 20701
(240) 456-2062
gosvalds@tasc.com

Abstract. Enterprise Architecture definition provides guidance to the systems engineer so that the enterprise system better reflects the needs of the customer. With the current systems engineering approach, the systems engineer can allocate requirements to systems without considering enterprise-wide considerations. The architecture-centric approach for system design provides enterprise-wide information for systems engineers, who can utilize it to affect systems design. This paper describes Enterprise Architecture and related concepts, and explores the state of the art of Enterprise Architecture and Enterprise Architecture frameworks. The Zachman Framework is used as the basis for creating models of the architecture. This paper proposes a bridging concept to define the models for the architectural perspectives. The bridging process provides a traceable methodology for developing architectural models of the enterprise that are directly traceable to the Zachman Framework primitives, thereby ensuring that the architecture models all elements that are necessary to completely describe the enterprise.

Keywords. Architecture, Architecture-centric, Architecture Framework, Enterprise, Enterprise Archi-tecture, Zachman Framework.

INTRODUCTION

Within an architecture-centric approach, the architect models the stakeholders' vision (an individual or organization that has with an interest in the enterprise) and creates an architecture definition that provides design guidelines for the systems engineer.

In order to describe architecture-centric concepts, it is necessary to define some basic terms. The enterprise wide context (system-of-systems) is being used in these definitions. The architecture is scaleable in that all concepts applied to the enterprise can also be applied to a system that is part of the enterprise.

Enterprise—an organization that uses Information Technology to perform its mission.

Architecture—"the fundamental organization of a enterprise (or system) embodied in its components, their relationships to each other, and to the enviornement, and the principals guiding its design and evolution" (IEEE 2000).

Enterprise Architecture—the description of the stakeholders mission including information, functionality, location, organization, and performance parameters. Enterprise Architecture describes the plan for building a system or set of systems.

Why Define Enterprise Architecture? Systems engineering is an interdisciplinary approach and a means to enable the realization of successful systems (INCOSE 1999). Our experience has shown that in many cases the systems engineer does not use a formal approach to resolve enterprise wide concerns. The architecture-centric approach provides the systems engineer a way to model the enterprise (system-of-systems) in order to formally address enterprise wide concerns. Thus with an architecture-centric approach the architecture becomes the center for enterprise-wide design decisions (Batman 1999). The architecture presents information needed to develop a enterprise design that meets the business and technical needs of the customer.

COMPARISON OF DESIGN APPROACHES

The design-centric approach and the architecture-centric approach are two common approaches for designing an Information Technology based enterprise. A design-centric approach allocates the requirements directly to the enterprises systems and its subsystems. A major flaw in this approach is the lack of a process where enterprise-wide tradeoffs are addressed. Without these decisions and tradeoffs, the design may not address all enterprise-wide concerns.

The architecture-centric approach mitigates this risk because it uses the architecture to define enterprise-wide information. This approach develops models of the enterprise. The resulting enterprise design includes all of the stakeholders' visions and concerns.

ARCHITECTURE-CENTRIC CONCEPTUAL MODEL

The architecture Working Group of the IEEE Software Engineering Standards Group has developed the

IEEE 1471 standard (IEEE 2000), which documents terminology used in architecture definitions. A part of this standard is a model that presents the Architecture Description concept. This model describes the architecture and related components using the Object Management Group Unified Modeling Language Class diagram (OMG 1999). This concept model was revised to include the Architecture Framework and to add other items to fully describe the architecture-centric concepts.

An Enterprise is an organization that uses Information Technology, which one or more systems then implement. Figure 1 presents in a UML Class diagram the Enterprise Architecture concepts. Each component is described below.

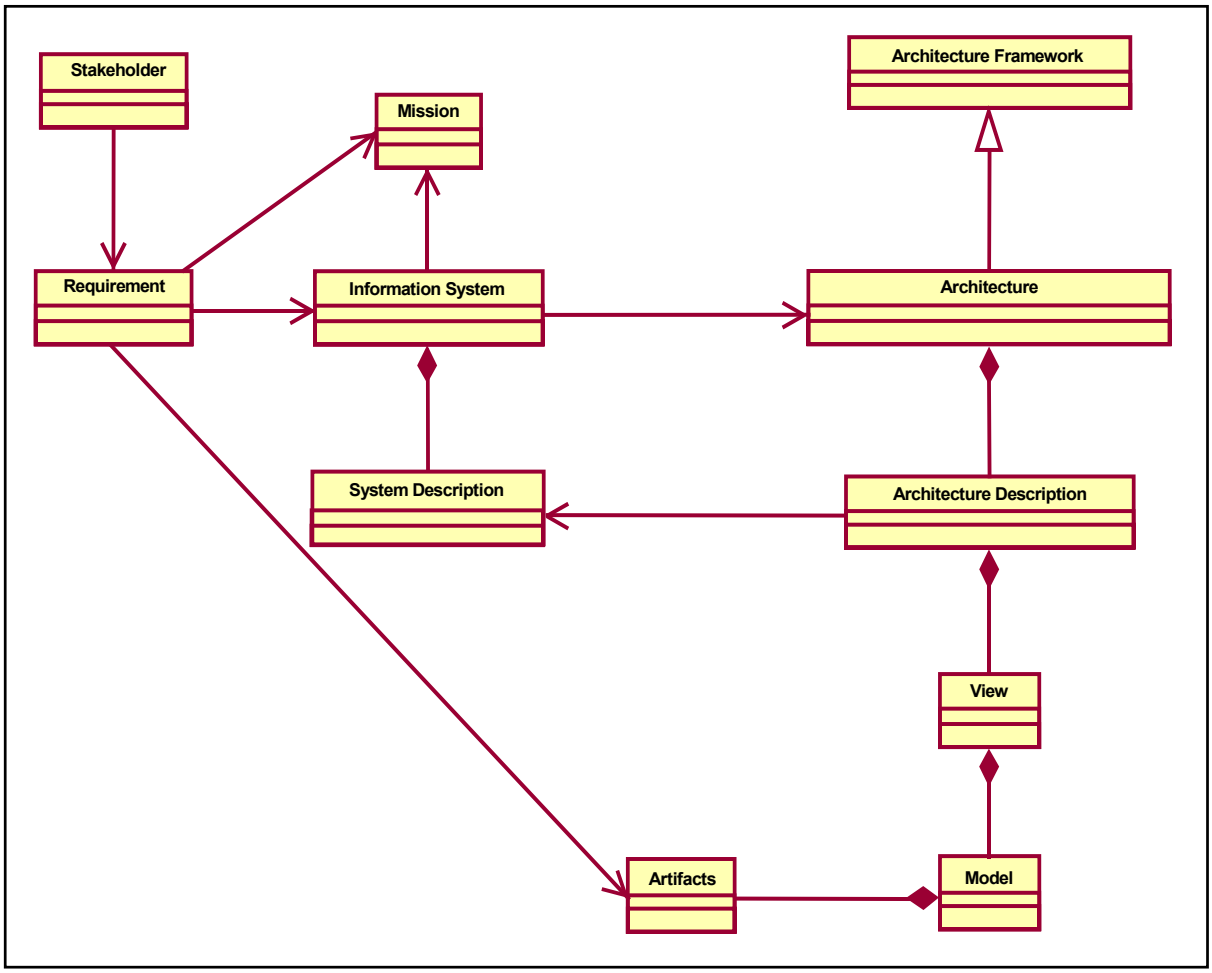

Figure 1. Enterprise Architecture Conceptual Model

Stakeholder—The Stakeholder is an individual, team or organization with interest in or concerns relative to an enterprise. The Stakeholder uses its vision, concerns (interest which pertain to the enterprises development, operation and any other aspects that are critical to the stakeholder), and goals to describe an enterprise (IEEE 2000).

Requirement—The Stakeholder documents the vision, needs, and goals of an enterprise in a requirement document. Requirements address overall system operation as well as detailed functionality.

There can also be multiple enterprises within an overarching enterprise. For example, if one company has many divisions, each division implements its enterprise missions, and each division enterprise is a subset of the company enterprise and its missions.

Mission—A Mission is the objective used to implement the stakeholders' vision.

Information System—An Information System is a "collection of different things which together produce results unachievable by the elements alone" (Rechtin et al 1997). A mission may be performed by one or more systems, such as a system-of-systems (or Information System): "A system-of-systems is a set of different systems so connected as to produce results unachievable by the individual system alone." (Krygiel 1999). The implementation of an Information System (or a system-of-systems) fulfills one or more missions to achieve a set of objectives. The Stakeholder defines the Mission to be performed in the enterprise. The Stakeholders' Mission require-

ments are enterprise wide.

System Description—A System Description is a design of the Information System.

Architecture Framework—An Architecture Framework provides guidance for the description of the information system via the use of views and models. "The Framework, as it applies to enterprises, 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." (Zachman, 1998).

Architecture—An Architecture is a realization of the Architecture Framework. The Architecture is a formal description of an Information Technology system (Open Group 1999). Therefore, a concrete representation of the Architecture Framework is the Architecture.

Architecture Description—An Architecture Description documents the views and models that the Architecture Framework uses to describe the system. Each view consists of one or more architectural models, which comprise the products that describe the model, e.g., data, function, and motivation to describe the system. Each view and model presents a different description of the system.

View—An Architecture View is one or more abstractions of the stakeholder perspectives. A view may be from the point of the owner, user, developer, designer and implementer. These views may contain Physical, Logical, Conceptual and Contextual representations (Zachman 2000).

Model—An Architecture Model contains data, process, hardware, personnel, performance, organization, and interface descriptions that define the system. The model may use any format that is appropriate to the documentation of the mission.

Artifacts—Architecture Artifacts are the basic elements used to develop architectural models.

ARCHITECTURE FRAMEWORK SURVEY

Industry has many frameworks that are customized for specialized applications, as well as a few frameworks that are generic. A survey of existing frameworks was performed to ascertain the state of art relating to frameworks in industry and government. Many frameworks were examined including the C4ISR/DoD Architecture Framework (C4ISR 1997), Federal Architecture Framework Enterprise Architecture Process (OIT 1999), The Open Group Architecture Framework (Open Group, 1999), The Rational Unified Process (Kruchten 2000) and the Zachman Framework (Zachman 1987). The goal was to find a framework definition that could be used as a common reference framework. The results of the survey indicated that the Zachman Framework meets this goal because it defines the basic elements that can be used to describe any enterprise.

THE ZACHMAN FRAMEWORK

The Zachman Framework (Zachman 2000) defines the artifacts that are needed to fully define an enterprise. Figure 2 depicts the Zachman Framework re-

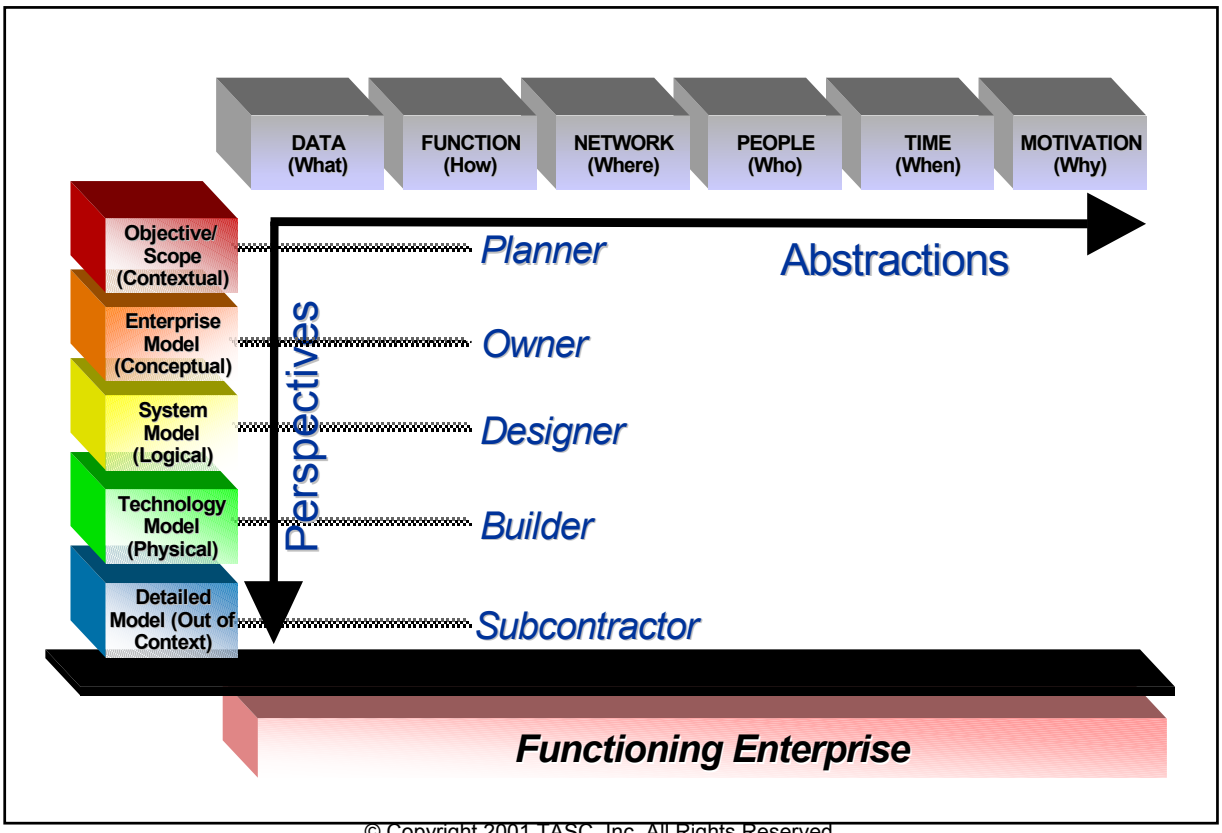

Figure 2. Zachman Framework Conceptual Description

lationships between the Abstractions and Perspectives.

An Abstraction consists of one of six abstracts that the Zachman Framework defines to be Data (What), Function (How), Network (Where), People (Who), Time (When), and Motivation (Why). These Abstractions represent different ways to describe the enterprise.

There are five Perspectives defined in the Zachman Framework. Each Perspective reflects a different level of Abstraction. The Perspectives for the enterprise are from the point of view of the Planner, Owner, Designer, Builder, and Subcontractor. These five Perspectives are also referred to as views into the Architecture.

The elements of the Zachman Framework are referred to as cells. The Framework defines a sample single-variable model for each cell. This information (i.e., primitive) can be considered to be an artifact. The Abstraction artifacts can then be used to create Views and Models by transforming between the perspectives. (Figure 3). The concept of developing transformations between the artifacts provides a way to create traceable composite models that are the bridges between the perspectives.

BRIDGING THE PERSPECTIVES

Bridging the perspectives through the use of models provides a way for the architect to develop models that have a direct relationship to the base framework (Zachman Framework). Through this modeling (Figure 4), the Perspectives are formalized by creating model views. These models are then used to populate

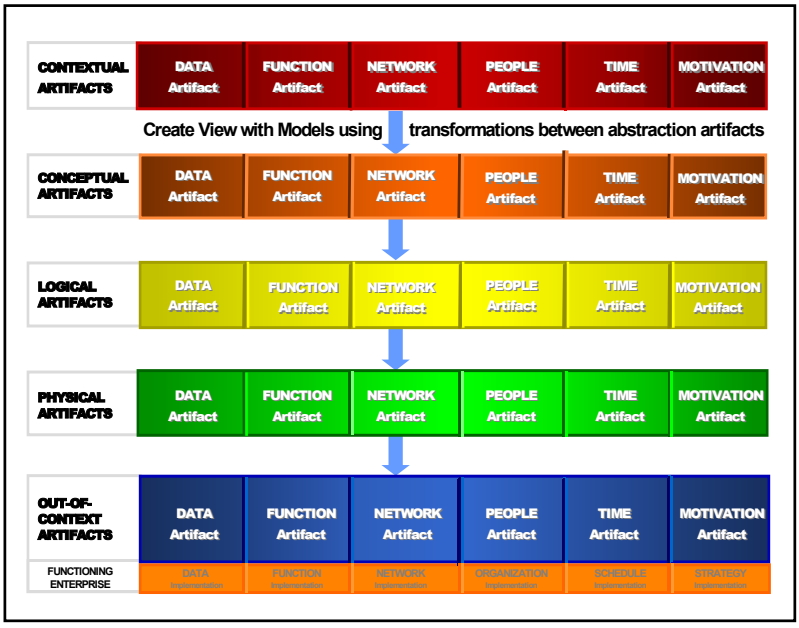

Figure 3. Bridging the Perspectives

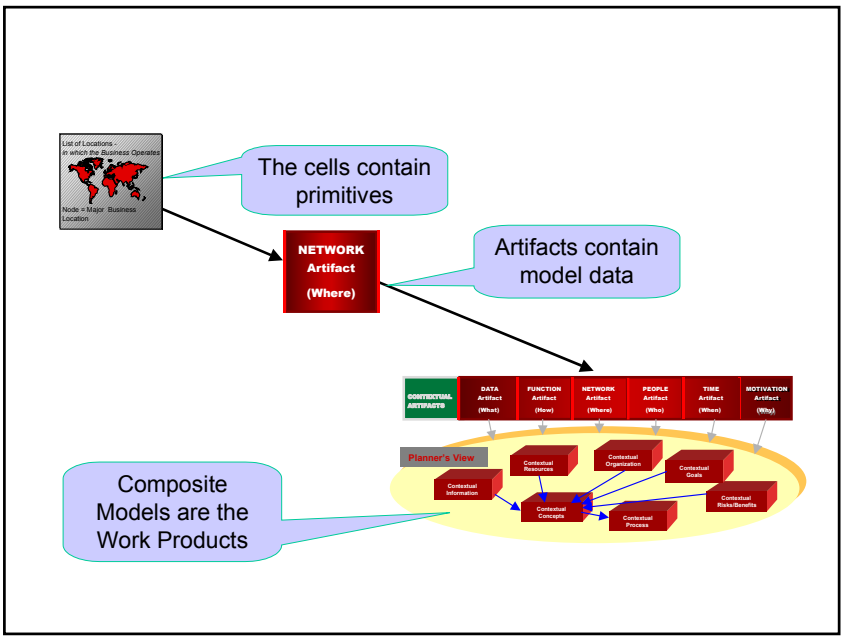

Figure 4. Primitives to Work Product Mapping

the next row of the Perspective artifacts. Using and populating Perspective artifacts one row at a time, ensures that all artifacts are addressed and therefore that the Enterprise Architecture is fully described as defined by the Zachman Framework.

Each Perspective row's artifacts are used as inputs to the Perspective models. A transformation between the Artifacts is used to develop models that represent the Perspectives' view. The developed model data is then used to populate the next row of Artifacts. The perspective modeling is the bridging concept presented here that provides a traceable modeling process from conceptual to implementation.

Figure 5 is an example of the models that are defined for the description of the Planner's View. The models utilize data that the previous view has placed in the abstraction artifacts (i.e., Data, Function, Network, People, Time, and Motivation). The diagram shows conceptual population of the models by the Abstraction artifacts. The models are methodology-independent (i.e., Functional, Object Oriented). After the models are created, the data in the models is used to populate the next row of Abstraction artifacts. Table 1 provides the definition of the models for all of the Perspectives (Osvalds 2000).

CONCLUSION

Systems engineers must approach the designing of systems with information that enables them to develop a solution that meets the vision of the stakeholder. The use of the architecture for modeling the enterprise functionality provides a way to verify the system design constraints before making and implementing design decisions. This set of design decisions addresses important system-wide issues affecting how the system will be engineered, as well as the cost, maintainability, and interoperability of the enterprise. Therefore, the better the enterprise is defined, the better the systems engineer's solution.

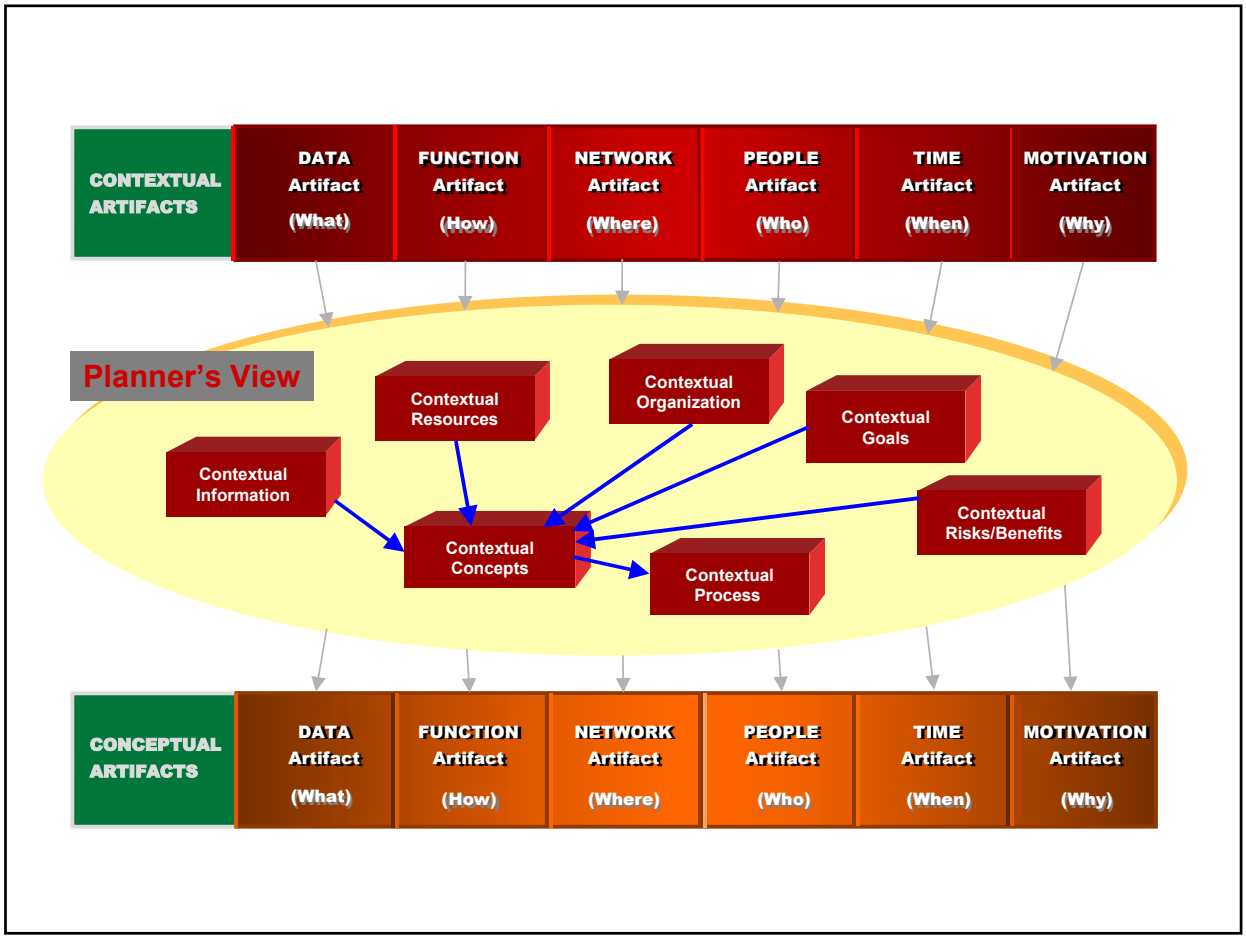

Figure 5. Example of Bridging the Zachman Framework Perspectives

Table 1. Bridging Model Descriptions

Perspective	Artifact Input	Model	Artifact Output
Planner's View	Contextual	Information	Conceptual
Planner's View	Contextual	Resources	Conceptual
Planner's View	Contextual	Organization	Conceptual
Planner's View	Contextual	Goals	Conceptual
Planner's View	Contextual	Risks/Benefits	Conceptual
Planner's View	Contextual	Concepts	Conceptual
Planner's View	Contextual	Process	Conceptual
Owner's View	Conceptual	Components	Logical
Owner's View	Conceptual	Process	Logical
Owner's View	Conceptual	Process States	Logical
Owner's View	Conceptual	Resource Interactions	Logical
Designer's View	Logical	Scenarios	Physical
Designer's View	Logical	Classes	Physical
Designer's View	Logical	Packages	Physical
Designer's View	Logical	Interactions	Physical
Designer's View	Logical	States	Physical
Builder's View	Physical	Elements	Out-of-context
Builder's View	Physical	Activities	Out-of-context
Builder's View	Physical	Interactions	Out-of-context
Builder's View	Physical	States	Out-of-context
Builder's View	Physical	Behaviors	Out-of-context
Builder's View	Physical	Deployment	Out-of-context

REFERENCES

Abowd, G., Bass, L., Clements, P., Kazman, R., Northrop, L., and Zaremski, A., *Recommended Best Industrial Practice for Software Architecture Evolution*, CMU/SEI-96-TR-025, ESC-TR-96-025, Software Engineering Institute, Carnegie Mellon University, 1996. www.sei.cmu.edu/publications/documents/96.reports/96.tr.025.html

Bass, L., Clements, P., and Kazman, R., *Software Architecture in Practice*. Addison-Wesley, 1998. cseng.aw.com/

Batman, J., "Characteristics of an Organization with Mature Architecture Practices." Essays on Software Architecture, Software Engineering Institute, 1999. www.sei.cmu.edu/architecture/essays.html

C4ISR Architecture Working Group, "C4ISR Architecture Framework." Version 2.0, C4ISR Architecture Working Group, Framework Panel, December, 1997. www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library/index.htm

Clements, P.C. and Northrop, L.M., *Software Architecture: An Executive Overview*, CMU/SEI-96-TR-003, ESC-TR-96-003, Software Engineering Institute, Carnegie Mellon University, 1996. www.sei.cmu.edu/publications/documents/96.reports/96.tr.003.html

Cook, M.A., *Building Enterprise Information Architectures*. Prentice Hall PTR, 1996. www.phptr.com/

IEEE Computer Society, IEEE Std 1471-2000, "IEEE Recommendation Practice for Architectural Description of Software-Intensive Systems," IEEE Computer Society, October, 2000. www.ieee.org

International Council on Systems Engineering (INCOSE), "What is Systems Engineering?" International Council on Systems Engineering, June, 1999.
www.incose.org

Kruchten, P., *The Rational Unified Process, An Introduction*. Addison-Wesley, March, 2000.
www.awl.com/cseng

Krygiel, A.J., "Behind the Wizard's Curtain: An Integrated Environment for a System of Systems," National Defense University and the DoD C4ISR Cooperative Research Program, July, 1999.
www.dodccrp.org/pubs/wizard.pdf

Motley, A.E., III, *Goal Analysis Procedure: Guidelines for Applying the Goals Analysis Process*, INCOSE Mid-Atlantic Regional Conference, 5 April, 2000.
www.incose.org

Object Management Group, "Unified Modeling Language Specification," Version 1.3, OMG, 1999.
www.omg.org/techprocess/meetings/schedule/UML_RTF.html

Office of Information Technology, The Chief Information Officers Council, "Federal Enterprise Architecture Framework," Version 1.1, Federal Agencies Information Architecture Working Group, Federal Architecture Framework Subgroup, September, 1999.
www.itpolicy.gsa.gov/mke/archplus/cmodel.htm

Open Group, "The Open Group Architectural Framework (TOGAF)," Version 5, The Open Group, December, 1999.
www.opengroup.org/togaf/

Osvalds, G., "The Use of Zachman Framework Primitives for Enterprise Modeling," Enterprise Architecture Conference, 26 October, 2000.
www.itpolicy.gsa.gov/mke/archplus/presents.htm

Rechtin, E. and Maier, M.W., *The Art of Systems Architecting*. CRC Press, 1997.
www.crcpress.com

SEI et al, "How Do You Define Software Architecture?" Software Engineering Institute.
www.sei.cmu.edu/architecture/definitions.html

Sowa, J.F. and Zachman, J.A., "Extending and Formalizing the Framework for Information System Architecture." *IBM Systems Journal*, 31(3),1992.
www.research.ibm.com/journal/

Spewak, S.H. and Hill, S.C., *Enterprise Architecture Planning*. John Wiley & Sons, 1192.
http://catalog.wiley.com/

Zachman, J.A., "A Framework for Information Systems Architecture." *IBM Systems Journal*, 26(3), 1987.
www.research.ibm.com/journal/

Zachman, J.A., "The Framework for Enterprise Architecture," *DM Review*, September, 1998.
www.dmreview.com/master.cfm?NavID=198&EdID=422

Zachman, J.A., "Enterprise Architecture - A Framework," Zachman International. January, 2000.
www.zifa.com/frmwork2.htm

BIOGRAPHY

Gundars Osvalds is a Senior Principal Member of the Technical Staff with the Cryptologic Systems Division of the Northrop Grunnman Information Technology Sector located in Annapolis Junction, Maryland. He leads company-sponsored efforts to define architecture and architecture processes. These efforts are the culmination of his 32 years of systems engineering experience in architecture design, requirements, performance analysis, and system hardware and software implementation.