System Architectures and Evolvability:

Definitions and Perspective

Rick Steiner
Rayteon Naval and Maritime Systems
1901 W. Malvern Ave, Fullerton, California 92833
v. 714 732-8312 email: fsteiner@mail.hac.com

ABSTRACT

The term "System Architecture" has been widely used in the systems engineering community for at least three decades. Even today, however, the use of this term often elicits more confusion than understanding! In particular, "System Architecture" has been used to describe, at various times, both the evolutionary system framework (Rechtin 91) (Rechtin/Maier 97), and the specific physical design or component interrelationship (Hatley 88). Even when it is agreed that the "System Architecture" represents a framework in which detailed design is performed, it is not generally agreed what aspects of behavior and structure should be captured in such a framework, how it should be represented, and how it relates to the specifics of system design. This paper examines current definitions of "systems architecture", and proposes a taxonomy of terms to distinguish "single use" from "enduring" applications of architecture. Particular attention is paid to enduring architectures and their relationship to systems engineering.

DEFINITION OF ARCHITECTURE

The term "architecture" is of course, borrowed from the common usage; usually in the sense of civil architecture. System architectures are a subclass of all architectures, as depicted in Figure 1 (OMT/UML class notation used).

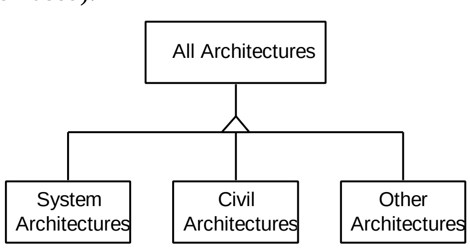

Figure 1: OMT representation of classes of architectures

One of the fundamental attributes of any architecture is consistency of approach in development of form and function. This consistency is the result of standard methods or "schools" of architecting, and it results in desirable and easily identifiable characteristics that endure from one design to the next. The enduring nature of architectures will be discussed later in this paper.

The INCOSE System Architecture Working Group (SAWG 94) defines the term "system architecture" as "the aggregation of decomposed system functions into interacting system elements who's requirements include those associated with the aggregated system functions and their interfaces requirements/definition". This is clarified when it is also stated that "when used as a noun the System Design is the same as the System Architecture".

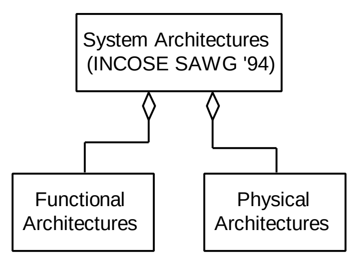

Figure 2: INCOSE SAWG '94 System Architecture Definition (OMT notation)

The Hatley-Pirbhai school of systems design uses the term "architecture" to represent the structure or physical design of the system, i.e. the complete collection of and relationship between system components. This definition will be referred to as the physical architecture (Lykins 97) (see Figure 2). The compliment of physical architecture is system behavior or functional architecture. The system functions of the functional architecture (requirements model components, in Hatley-Pirbhai parlance (Hatley 88)) can be allocated to elements of the physical architecture. Both the functional architecture model (requirements model) and the physical architecture model should be developed hierarchically, consistent with an iterative or spiral model of system development.

The above definitions, however, say nothing about system growth or evolution. Recent interest in evolutionary system development (Isaac, et al 94), inter-system impact assessment (McCay 94), standard architectures (Percivall 97), and design re-use (DISA 95) would suggest that a refinement or tailoring of this definition may be necessary. It seems appropriate to distinguish architectures that are developed without intent of reuse or evolution from architectures that have incorporated enduring attributes from one system design to the next (see Figure 3).

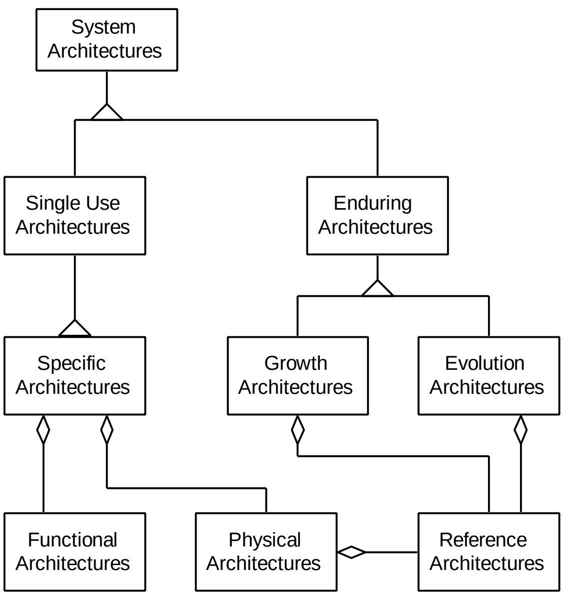

Figure 3: Enduring and Single Use Architectures

One may argue that this is unnecessary... since design growth and reuse cannot be predicted, all that is really required is a comprehensive library of proven design elements which are invoked appropriately for each system design. While such a design library approach may be appropriate in some traditional market segments, it has serious limitations in the development of large, complex, COTS based systems. These systems are typically composed of elements which are known beforehand to change through the life of the system. Sophisticated customers are now considering system growth and system evolvability as primary requirements, and as such they need to be considered from the start in a way that is comprehensive and explicit. It is proposed that the development of enduring architectures provides the mechanism to meet this need. This is totally consistent with the opening paragraph, which relates the term "architecture" to an implied enduring aspect of design, as well as a consistency of product.

THE CASE FOR ENDURING ARCHITECTURES

An enduring architecture, for the purpose of this paper, is not the physical or functional architecture. It is not necessarily even the product of the systems engineering or design process! This paper suggest that the term "enduring architecture" represents a set of constraints placed on the design for the purpose of:

1) Ensuring consistency of key characteristics as a product design matures or evolves, and

2) Exemplifying a strategy for continuing customer satisfaction and communication.

An enduring architecture, in this sense, can contain both structure and behavior, but at an abstract level. It does not contain the entire system structure, but only those aspects of the structure necessary to accommodate 1) and 2) above. Likewise for system behavior.

In doing this, the system architecture provides a framework in which the design is performed. It also serves to set the system context, constrain the system concept, and bound the interfaces of the resulting system, thus providing a common system lifecycle focus around which the design team, suppliers, and customer can rally. The relationship between system architecture and system design is represented in Figure 4. Square boxes represent functions, and rounded boxes represent information. Note that this diagram does not include the control flow or iterative loops normally experienced.

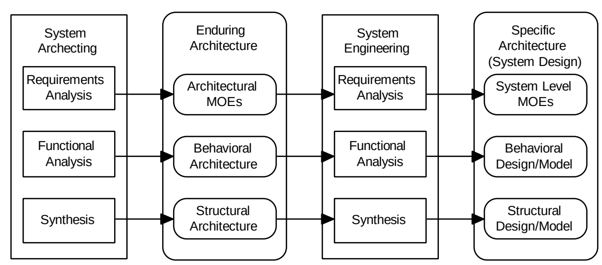

Figure 4: Relationship of System Architecture to System Design.

One might argue that the enduring architecture is really nothing more than the first level of system design, or that it represents nothing more that a set of standards. It is important to distinguish at all levels, however, enduring system aspects from non-enduring system aspects. An enduring architecture may well contain multiple levels of system hierarchy, each of which can represent an enduring function, interface, or component to be imposed on a given specific system design. This is much more specific than a typical standard, and in fact must be designed in the same sense that the system itself must be designed.

Many systems have been built without a preplanned enduring architecture and they have on the whole been rather successful! Analogs from civil architecture include traditional log cabins, beach shacks, and the Winchester Mystery House (Note 1). All of these structures have served their intended purpose, but it should be noted that they are always either difficult to reproduce, or of inconsistent quality. Table 1 provides some suggested guidelines for when an enduring architecture may or may not be desirable.

Enduring Architecture NOT desirable: (systems unchanging or tolerant to change)	**Enduring Architecture desirable:** (systems intolerant to change)
• highly expedient, short life systems (beach shack, or emergency response) • systems insensitive to change over time (no reliance on computer technology) • single use process employed (Winchester Mystery House) • fundamental nature of system is not well understood in advance (R&D, exploratory development) • highly precidented systems in slowly changing markets (building materials & tools)	• long life, distributed ownership systems (TV/radio/telecommunications systems) • series of reliable, complex systems with long service lifetimes (military weapons systems) • highly interconnected systems with need for significant future growth (business IT systems) • systems requiring a large degree of infrastructure support (personal transportation solution systems, a.k.a. automobile dealers & service centers)

Table 1: Desirability of Enduring Architectures in Various Applications.

Ideally, the enduring architecture should minimally constrain the systems engineer, thus providing adequate room for system tradeoffs at each level of design. For example, the enduring architecture might contain just enough system functionality to ensure interface consistency and preplanned functional growth. It also might contain just enough system structure to accommodate specific corporate hardware or software development strategies, and key alliances with other vendors.

DESIGN MATURATION VS. EVOLUTION

A principal reason for developing an enduring architecture is to manage the change in the system design. Sometimes, this change in system design is referred to as "evolution". It is not always clear when talking about system evolution, however, if one is considering the changes in design of that specific instance of the system, or rather the changes in design between generations of a family of related systems. Taking a cue from the biological sciences, this paper draws the distinction between 1) change encountered in a single generation of a system (maturation), and 2) change encountered between subsequent generations of a family (or genus) of systems (evolution). Maturation, in this context, refers to the development of a specific system design, from initial context & concept through production & disposal. There may be many individual systems produced as part of the production run, but they are usually all from a single generation of the design. Evolution refers to how the system design changes from one generation of a product design to the next, such as specifying which elements of the design are passed down/reused, and which elements of the design are new to the latest generation. Table 2 introduces the concept of Evolutionary and Growth architectures, and relates them to previous concepts of system architecture and design re-use.

WHAT IS A GROWTH ARCHITECTURE?

It is proposed to use the term "growth architecture" to address management of the changes in a design while it matures (per above definition, within a single generation). A growth architecture needs to specify the set of system characteristics that should endure through a single generation of products. Note that the growth architecture may address system growth, technology insertion, and preplanned product improvements (P^3I), since these aren't really part of a new generation of design. Open systems aspects (addition, extension, and adaption for use) can be addressed in the growth architecture only in a limited sense, since it cannot address "inheritance" of system characteristics to the next generation of systems. Likewise, design re-use cannot be addressed in a growth architecture; cross-generational aspects of system design families must rely on evolutionary architectures.

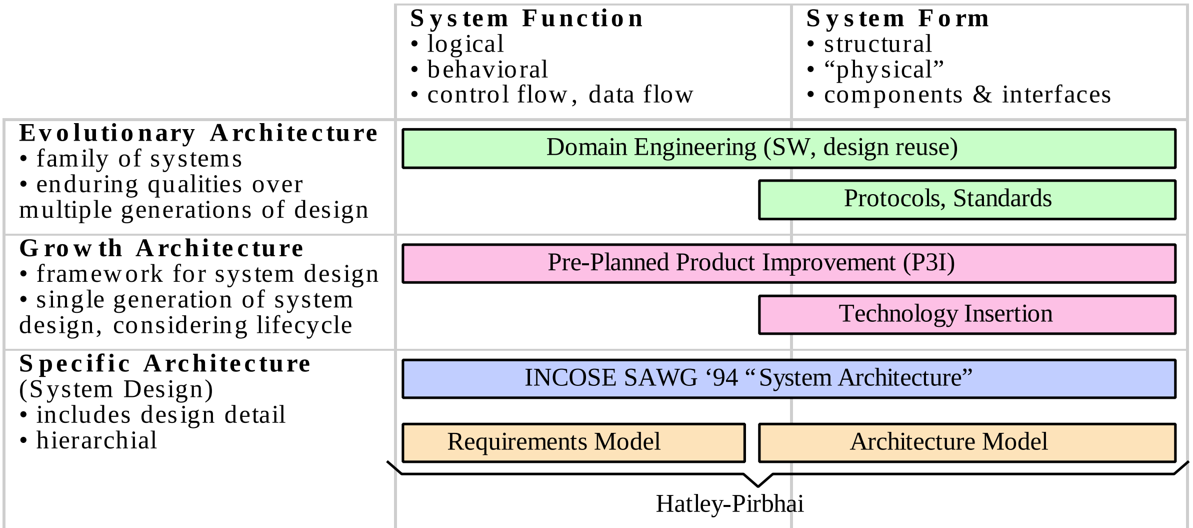

Table 2: Evolutionary, Growth, and Specific Architecture Definitions

WHAT IS AN EVOLUTIONARY ARCHITECTURE?

It is proposed to use the term “evolutionary architecture” to address management of the changes in design while it evolves (from one generation to another). The evolutionary architecture needs to specify the set of enduring characteristics to be designed into a family (or genus) of products. This requires the architect to set in advance (hopefully!) those things about a family products that he/she doesn’t want to change over time. In particular, the evolutionary architecture helps to define the “openness” of a family of systems, or their ability to accommodate “addition, extension, and adaption for use” from one generation to the next. An evolutionary architecture may not look much like a growth architecture... it will probably be much more sketchy, and not directly usable as a framework for specific system design. Examples of evolutionary and growth architectures are shown in table 3 below.

	Evolutionary Architecture (families of structures)	**Growth Architecture** (specific structure, lifecycle consideration)	**Specific Architecture** (system design)
Product System (deliverables and support)	TCP/IP, SMTP, WWW Honda “world car” USN “Surface Combatant ‘21”	Netscape Navigator family 2005 Honda line USN “Destroyer ‘21”	Netscape 3.1 2005 Honda Accord DDG (hull specific)
Producing System (processes and infrastructure)	Hammamtrack facility long range vision DoD High Level Architecture (HLA) Enterprise Business Management System	Cadillac production line ‘98-00 Consolidated engagement Simulation Program Management Plan (including planned reassessment)	‘99 Cadillac Seville production line specific simulation exercise Program Management Plan version 1.0

Table 3: Examples of Architecture for Product and Producing Systems

Obviously, an evolutionary architecture focuses on cross-generational attributes of products. In a way, the evolutionary architecture can be considered a kind meta-architecture, or the architecture of growth architectures... It provides a framework for the specification of desirable “genetic” or inherited characteristics from one growth architecture to the next. The kind of long-term, enduring characteristics that need to be considered in an evolutionary architecture may include:

- those things about the system which are hard to change
- human-system interface (form & function)
- distributed interfaces
- data transport protocols

    - those things which are distributed/tailored to each site
    - client software/hardware (unless have rapid update mechanism, e.g. JAVA)
- logistics/support concept
- lifecycle support mechanisms and infrastructure
- product distribution network
- service/support network and facilities
- production facilities, processes, and organizational constraints
- enterprise process improvement strategy
- design reuse (Domain Engineering)

An example of how the evolutionary architecture can relate to the specific architecture is show in Figure 5. Unfortunately, further discussion must wait for a subsequent paper.

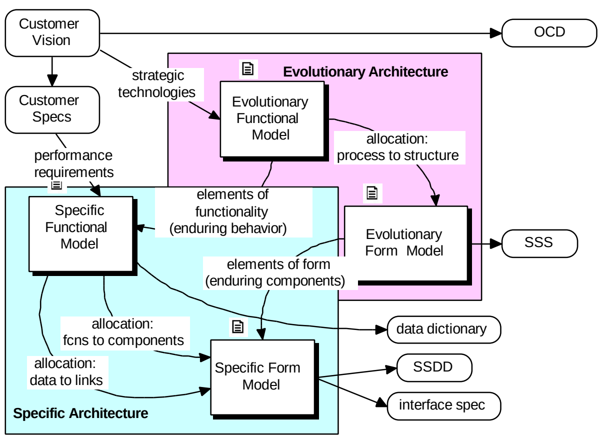

Figure 5: Interaction between Evolutionary and Specific architectures

SUMMARY

This paper has attempted to put a few of the various definitions of the term "system architecture" into perspective. By focusing on enduring and specific aspects of system architecture, useful distinctions can be made. A case has been established for the need to consider enduring architectures independently but in conjunction with specific architectures, especially in the development of complex, distributed, computer based systems of long effective lifespan. Future efforts will focus on content and form for specifying evolutionary architectures and growth architectures in a model-based systems engineering environment.

AUTHOR BIOGRAPHY

For the past 14 years, Mr. Steiner has been a key participant in application of information technology and systems engineering tools for Naval and maritime electronics systems development at Raytheon in Fullerton, California. He has been a member of INCOSE since 1993, and has participated in various working groups of the Modeling and Tools Technical Committee during that time.

NOTES

Note 1: The Winchester Mystery House in San Jose, California, was built by Sarah Winchester, an heiress to the Winchester repeating rifle fortune. She was a notorious mystic, and believed that as long as the house was still under construction, evil spirits would stay away and she would not die. As she advanced in age, the house acquired several interesting "features" that could not possibly be foreseen by the original architect!

REFERENCES

Defense Information Systems Agency (DISA), *Domain Engineering Process*, Version 2. April 1995

Hatley, Derek and Pirbhai, Imtiaz, *Strategies for Real-Time System Specification*. Dorset House 1988, Chapter 17 page 193, also page 217

Isaac, D. and McConaughy, G., *The Role of Architecture and Evolutionary Development in Accommodating Change*. 1994 INCOSE Proceedings

Lykins, H., *A Framework for Research into Model-Driven System Design*. 1997 INCOSE Proceedings Vol 1

McCay, B., *Design of a System-of-Systems Architecture Model and Its Use to Identify Cross-System Impacts*. 1994 INCOSE Proceedings.

Percivall, G., *Application of a System Architecture Standard to a Federated Information System*. 1997 INCOSE Proceedings

Rechtin, Eberhart, *Systems Architecting Creating & Building Complex Systems*. Prentice Hall, 1991

Rechtin, Eberhart and Maier, Mark, *The Art of Systems Architecting*. CRC Press, 1997

System Architecture Working Group (SAWG)*: Progress and Aspirations*. 1994 INCOSE Proceedings Vol. 1 page 563.