Interpreting ‘Systems Architecting’

M. R. Emes[1,*], P. A. Bryant[2], M. K. Wilkinson[3], P. King[4], A. M. James[1] and S. Arnold[5]

*Author to whom correspondence should be addressed

[1] UCL Centre for Systems Engineering, Mullard Space Science Laboratory, Holmbury St. Mary, Dorking, RH5 6NT, UK, Tel: 01483 204100, m.emes@ucl.ac.uk

[2] Logica, The Office Park, Springfield Drive, Leatherhead, Surrey KT22 7LP, UK

[3] Niteworks, PO Box 87, Farnborough Aerospace Centre, Farnborough, Hampshire, GU14 6YU, UK,

[4] Detica, Gloucester Business Park, Gloucester, GL3 4AB, UK

[5] University of Hertfordshire, School of Engineering and Technology, Hatfield, Hertfordshire, AL10 9AB

This is the peer reviewed version of the following article: [Emes, M. R., Bryant, P. A., Wilkinson, M. K., King, P., James, A. M. and Arnold, S. (2012), Interpreting “systems architecting”. Syst. Engin., 15: 369–395. doi: 10.1002/sys.21202], which has been published in final form at [http://dx.doi.org/10.1002/sys.21202]. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving.

Abstract

The UK Chapter of the International Council on Systems Engineering (INCOSE UK) has commissioned research to illustrate the variety of usage of the terms architecture and architecting in the systems engineering community. These terms, though widely used, are rarely strictly defined, and the meaning attributed to the terms is not consistent even in formal publications. Using soft systems methodology, this research has analysed three published sources (MODAF, The Art of Systems Architecting by Maier and Rechtin, and ISO/IEC 42010), and conducted a series of interviews with systems architecting practitioners.

Twelve contentious questions in systems architecting are discussed, and six perspectives on systems architecting presented, including three basic worldviews of the relationship between systems engineering and systems architecting. One model sees systems architecting as simply a rebranding of systems engineering to broaden its appeal with no change in content. Another model sees systems engineering restricted to its traditional processes, with systems architecting adding to systems engineering through external processes. The final model, and the most popular amongst the systems engineering community, sees systems architecting addressing shortcomings in traditional sequential lifecycle models by stretching the content of systems engineering to include new elements under the banner of systems architecting.

Keywords: systems architecting, architecture, systems terminology, belief systems, soft systems

Introduction

The INCOSE UK Architecture Working Group

The INCOSE UK Architecture Working Group (UKAWG) was formed at the end of 2006 at the request of the INCOSE UK Board with the primary aim of providing guidance and advice on UK Architecture practice to the UK Systems Engineering community. Contributing organizations include Atkins, BAE Systems, Cranfield University, Detica, John Boardman Associates, Logica, London Underground, MBDA, PA Consulting, Parsons Brinckerhoff, Rolls Royce, Thales, Ultra Electronics and University College London (UCL). Further details about the activities and membership of the UKAWG can be found on the UKAWG's Wiki www.ukawg.org (which is open to all INCOSE membership: registration can be achieved by e-mailing register@ukawg.org) [UKAWG].

The UKAWG has been working to develop a robust understanding of system architecture and architecting concepts. This has proved to be a challenging task, primarily because of the wide diversity of practices and viewpoints relating to architecture. Whilst the initial focus was on usage of terms by UK practitioners, UKAWG has latterly contributed to international Architecture Working Group initiatives including the reviewing of draft architecting standards (in particular ISO/IEC 42010).

Belief systems approach

Rather than attempting directly to harmonise interpretations of 'systems architecting', say through the production of a single normative standard, the UKAWG has taken the approach of recognizing the existence and potential validity of overlapping 'Belief Systems'. 'Belief Systems' are defined to be sociological world views, i.e. collective world views held within a community that permit a coherent and useful interpretation of specialist terms (such as 'system' and 'system architecture') within that community [Aerts, et al., 1994]. Typically, a Belief System emerges within a community of practitioners according to their mutual understanding of concepts relating to key terms and according to the utility of such terms within their practice.

Within each of these Belief Systems we have then attempted to tackle the smaller problem of providing normative definitions specific to the Belief System. As there are reportedly more than 130 international standards with the word architecture in their title or abstract [Bendz, 2008], this approach is more tractable and productive than attempting directly to develop a single normative standard.

The Belief Systems Method

Seeking a systems approach for managing conflicting perspectives, UKAWG identified Soft Systems Methodology [Checkland, 1999; Checkland and Scholes, 1999; Wilson, 2001]. Soft Systems Methodology is "an organized way of tackling perceived problematical (social) situations … [which arise] … because different people have different taken-as-given (and often unexamined) assumptions about the world" [Checkland and Poulter, 2006: xv]. UKAWG therefore developed an application of Soft Systems Methodology called the 'Belief systems approach' to explore different 'worldviews' in the systems architecting community. This was first presented at the 2008 INCOSE International Symposium [King and Bryant, 2008] and was later elaborated in a draft INCOSE Technical Paper [Wilkinson, King and Bryant, 2009].

Once a belief system has been defined, a shorthand statement of the belief is developed and then elaborated in a Root Definition before finally creating a conceptual activity model. With the concepts precisely described in this way, it becomes possible to identify equivalences and dissonances between the interpretations of the key terms [Wilkinson, et al., 2010]. Such comparisons have utility when attempting to understand the diversity of practice from a theoretical point of view and, more practically, to establish a baseline for communication when different architecting communities interact. Such situations occur frequently, including when distinct communities associated with each of the individual systems in a system of systems need to achieve mutual understanding.

Following the conception of the Belief Systems Method, the INCOSE UK Chapter funded a research project that would test the robustness and utility of the methodology. The project consisted of two phases. In the first phase, the belief systems methodology was applied to three architectural sources – draft standard ISO/IEC 42010 [2011](Systems and software engineering — Architecture description, Final Draft International Standard[1]), MODAF (the Ministry of Defence Architecture Framework version 1.2 [Ministry of Defence, 2008]), and the book "the Art of Systems Architecting" by Maier and Rechtin [2009]. These were selected because the time available for the project limited the review to three sources, and the sponsor wanted to include one international architecture standard (ISO/IEC 42010 was the most up to date and applicable), one architecture framework standard (MODAF is the most relevant to INCOSE UK[2]), and one systems architecting textbook that would have had wide readership amongst practitioners in a range of domains. Limiting the study to three sources gave only a partial view of the range of perspectives on systems architecting, but it did reveal some interesting results and perhaps more importantly provided a useful source of questions for the second phase of the project. Here, the belief systems methodology was used to investigate interpretation of the term 'systems architecting' amongst selected practitioners in UK industry. Again, time was a limiting factor, and only a small number of practitioners could be involved in this survey. This was not seen as a problem, however, since the survey was not intended to provide quantitative data, rather a selection of perspectives on systems architecting from different industries.

UKAWG also conducted a literature review to explore the origins of the terms 'systems engineering' and 'systems architecting' and to highlight the questions of interpretation that prove the most contentious.

Origins of the terms systems engineering and systems architecting

The terms 'system' and 'engineering' had been in common use in the English language for over three hundred years before the concept of combining them arose. The same is true of 'system' and 'architecting'. The words system, architect and engineer have a long heritage, as shown in Table 1.

[1] ISO/IEC 42010 draft CD0.8 was analyzed initially, with analysis updated using final draft ISO/IEC FDIS 42010

[2] Other sources such as the Zachman Framework and the Open Group Architecture Framework (TOGAF) were considered but felt to be less relevant to the mainstream systems engineering community represented by INCOSE members.

Term	Primary current definition	First defined in English language	OED Main Meanings (main meanings plus sub-meanings)	Complexity score[3] (OED words in definition/1000)
Architect	A skilled professor of the art of building	John Shute [1563]	3 (5)	0.444
Engineer	A person who makes engines, structures or systems	1380 [Herrtage, 1879]	6 (12)	3.404
System	An organized or connected group of objects	1638 [Mede, 1641]	10 (30)	8.704

Table 1: Maturity of words architect, engineer, system [OED, 2010]

Architect (and architecture) is the best understood of the three words (three different main meanings and a total of five sub-meanings in the Oxford English Dictionary (OED)), perhaps unsurprisingly since works on the principles of architecture have been published since Roman times [Vitruvius, 15BC]. The word derives [OED, 2010] from the Greek *archi-* meaning chief, principal, first in authority or order and *tekton*, meaning builder or craftsman (related to *tekhne* meaning art or skill). Hence the word architect is associated with technical leadership and connotes precedence as well as skill.

Engineer is more broadly interpreted than architect (six main meanings and twelve sub-meanings), and system has yet more complexity (ten different main meanings and thirty sub-meanings) in its definition as listed in the OED [2010]. Despite probably being the newest of the three words, the word system has around three times the semantic richness of the term engineer and around twenty times the semantic richness of the word architect (measured coarsely by the length of the OED dictionary definition). This means that interpretation of the unqualified word system is much more variable than is the word architect.

This interpretation of the terms architect and engineer indicates that an architect (if present at all) is responsible for the vision, and for overseeing the work to be done, and an engineer is responsible for creating something in its entirety. While an architect on his/her own would be unlikely to be responsible for hands-on implementation of a project, an engineer will be responsible for this. Whether engineering should include establishing the vision (i.e. whether architecting is a subset of engineering), or whether the vision must exist before the more practical engineering begins, is ambiguous – and one of the many questions that this research sought to address. The etymology of the word engineer suggests that design has always been part of the role of the engineer, since engineer is derived from the Anglo-Norman word engineor, meaning "a person who designs and

[3] This was a convenient measure devised by the authors for the purposes of comparing variety of interpretation of the words in question using the Oxford English Dictionary (OED)

constructs military works for attack and defence" [OED, 2010], but the extent to which design includes original conception is unclear.

So with shaky foundations it should not surprise us that the compound terms 'systems engineering' and 'systems architecting' are poorly understood. The OED sees systems engineering as "the investigation of complex, man-made systems in relation to the apparatus that is or might be involved in them" [OED, 2010]. It is interesting that the OED does not reference the definition of the international authority for systems engineering – INCOSE (International Council on Systems Engineering). INCOSE defines systems engineering (SE) as "an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: Operations, Performance, Test, Manufacturing, Cost & Schedule, Training & Support, Disposal. SE integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs" [INCOSE, 2004]. The earliest references to the term SE come from research at Bell Telephone Laboratories in the United States in the 1940s: "In our organization (Bell Telephone Labs) extensive use is made of an analytical procedure which we call Systems Engineering" [OED, 2010]. From the OED definition of engineering in general, we can postulate that SE is: 'The branch of science and technology concerned with the development and modification of complicated systems and processes using specialized knowledge or skills, typically for public or commercial use" [OED, 2010].

The emergence of systems architecting

In the 1950s, the practice of systems engineering was becoming more formally recognised and the foundations of systems architecting were arguably laid with the publication of 'System Engineering: an introduction to the design of large-scale systems' by Goode and Machol [1957]. Although the term architecture was not used, the issue of complexity and a sense of structure are implicit throughout the discussion of a structured design process. The earliest published definition of architecture in reference to technology is by Brooks: "computer architecture, like other architecture, is the art of determining the needs of the user of a structure and then designing to meet those needs as effectively as possible within economic and technological constraints. Architecture must include engineering considerations, so that the design will be economical and feasible; but the emphasis in architecture is upon the needs of the user, whereas in engineering the emphasis is upon the needs of the fabricator"[Brooks, 1962: 5]. An early reference to architecture that includes the term system is "the term architecture is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation" [Amdahl, Blaauw and Brooks, 1964: 87]. From the OED definition of (general) architecting we can infer that systems architecting is "The art or science of building or constructing systems for human use".

Ultimately, practitioners with particular design knowledge and skills are the source of architecture decisions and their rationale. The longstanding use of the terms architecture and architect used with regard to buildings and similar engineering structures is consistent with this general understanding. In the last few decades, however, the use of the term architecture in other branches of engineering has been growing, with the result that it has been subsuming, or at least refining, a region of general engineering design practice. In SE, software engineering and information technology, this has been a clear and mostly positive trend. However, different engineering communities, sometimes with implied professional distinction, have treated this focus on architecture (and on the specific engineering practice associated with it) with different emphasis.

Outside of civil engineering the explicit engineering use of the term architecture and its associated methods has been a comparatively recent phenomenon. In many branches of engineering these terms remain largely unused, with a range of synonyms seemingly communicating in an acceptable way the concepts, actions and information items that equate to system architecture and architecting. The same could, indeed, have been said about SE in its early decades. After over half a century of evolution, however, the scope and language of SE has now been internationally defined in a workable way by the International Organization for Standardization (ISO) in terms of process transformations ISO/IEC 15288 (Systems Engineering – Systems life cycle processes)[ISO/IEC 15288, 2002]. Preceding and subsequent to this watershed definition, other notable and influential process standards (successive issues of MIL-STD 499 [1969], IEEE 1220 [1994] and ANSI/EIA-632 [1999]) have focused on more specific regions of the system life cycle in a related way. SE has thus traditionally been defined as a discipline in terms of its process transformations, i.e. the actions undertaken or the procedural flow followed by practitioners. The SE work products – the process inputs and outputs in the form of material and energy, and the information they can convey – have been less clearly articulated by the formalisms governing the discipline.

The nature and content of work products, including the models that describe system architecture, can be inferred from the description of transformations in the ISO SE process reference model. However, a set of SE work products is explicitly defined in the ISO process assessment model ISO/IEC 15504 [2004] that complements ISO/IEC 15288. An alternative set of SE work products is also found in the relevant parts of CMMI (Capability Maturity Model Integration [Carnegie Mellon Software Engineering Institute, 2011]). Nevertheless, neither of these assessment standards provides more than a cursory definition of the system models that describe architecture, for example, the system functional model (2.08) and architectural design description (2.09) in ISO 15504.

This weakness in models of a system's architecture was recognized and addressed to an extent by developments termed model-based systems engineering (MBSE). MBSE may be seen as the formalized application of modelling to support SE, thereby providing a well-structured approach to designing and describing system architecture. Although SE has always depended on models, in practice MBSE was a move towards modelling techniques used in software engineering; this may have limited its popularity with those systems engineers and architects that do not have a software focus.

At the start of ISO/IEC 15288's development in 1995 the British Standards Institution (BSI) proposed a hierarchical model (depicted in Figure 1). This model can be seen in terms of a service level uppermost, technology-based implementation at the lowest level[4], and one or more layers of architecture design and build in between. However, in the 1990s organizational commitment to existing sequential representations, plus obligatory legacy titling of processes from the ISO standard on software engineering [ISO/IEC 12207, 1998], meant that only one process was perceived by most readers to address system architecture and this process emphasized the physical view of a system. ISO/IEC 15288 was “a product of fragmented, often discordant contributions”[Arnold, 2008: 4.27], and favouring greater harmonization of ISO software and systems standards, the 2008 revision of ISO/IEC 15288 did not take the opportunity to overcome this cardinal weakness and no change was made to this area of the model. Thus, despite a mostly beneficial influence on SE, ISO/IEC 15288 can in some measure be seen to have unfavourably conditioned SE minds regarding system architecture.

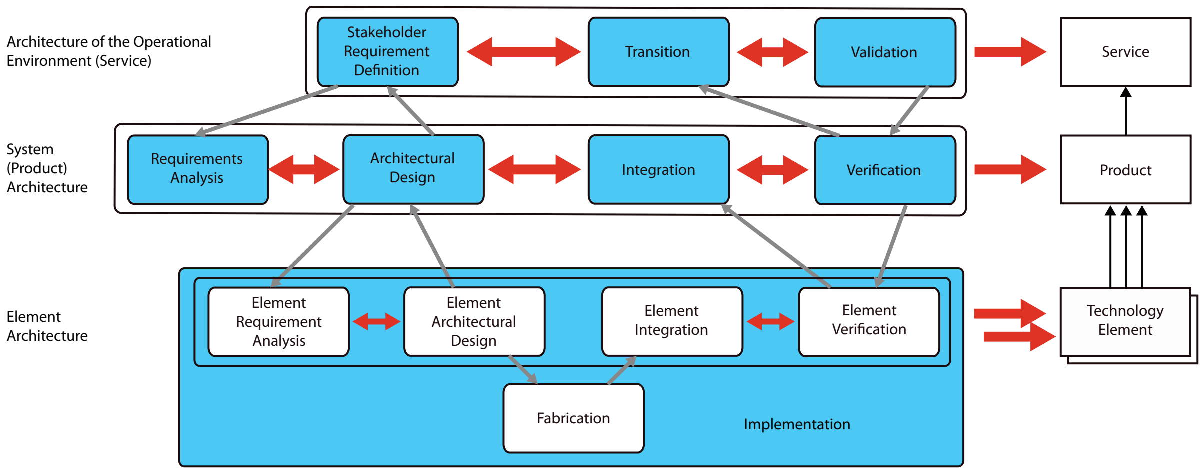


Figure 1: A hierarchical process-based representation of systems engineering based on ISO/IEC 15288

System Architecture in Software Engineering and Information Technology.

Unlike the SE community, the software engineering and information technology communities have been strongly influenced by the notion of architecture. They have evolved stronger modelling conventions for describing system architecture, and have juxtaposed these with actions and responsibilities that have customarily been central to SE. In consequence, they have established a measure of disciplinary jurisdiction in SE’s heartland.

Since around 1980, dictionary definitions of architecture have reflected this software/IT influence. The OED entry for architecture includes a ‘Computing’ sub-meaning, namely “the conceptual structure and overall logical organization of a computer or computer-based system from the point of view of its use or design; a particular realization of this” [OED, 2010]. By 1990, the definition of architecture offered by the IEEE was “the organizational structure of a system or component” [IEEE 610.12, 1990: 21], that is, a listing of parts and their organization or relationship in a system of interest. In the same standard, the ‘architectural design’ was described as “the process of defining a

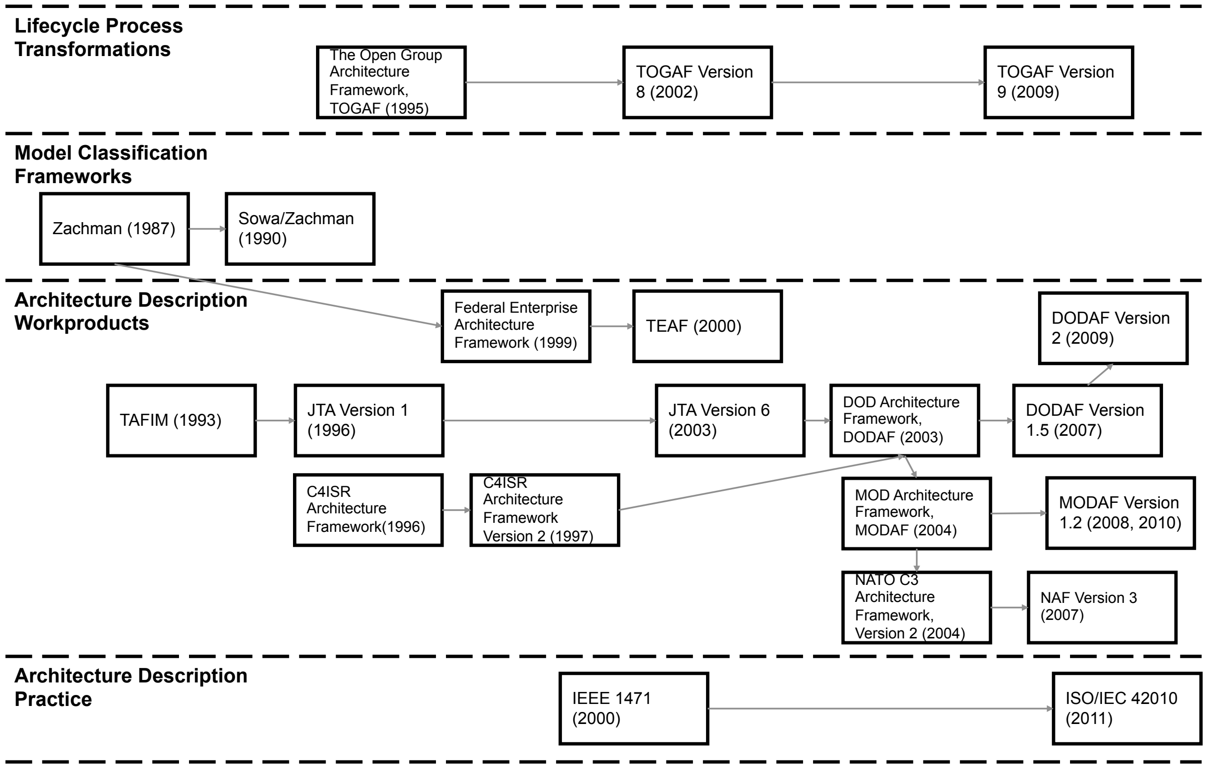
[4] At the lowest level, responsibility is transferred to the engineering skills and technology of particular implementation media. Depending on one’s persuasion, systems engineers may delegate responsibility to technology-specific engineering practitioners, or architects may delegate responsibility to systems engineers.

collection of hardware and software components and their interfaces to establish the framework for the development of a computer system" [IEEE 610.12, 1990: 21], or the result of this process. This blending of the concepts of architecture and design which has persisted through subsequent standards like ISO/IEC 15288 [2002] hampers attempts to draw meaningful distinctions between the terms architecture and design.

By 2000, the influential standard IEEE 1471 [ANSI/IEEE 1471, 2000: 3] had evolved the definition of architecture into "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment and the principles guiding its design and evolution". This definition proved contentious on several levels. Firstly, the term 'fundamental' had been introduced with no explanation. Secondly, the definition had now moved beyond descriptions of the system of interest by extending it to include its setting, and thus its operational behaviour and by implication the services it is intended to provide. Thirdly, it introduced the notion of the design rationale behind these descriptions. In doing so, it began to equate architecture with a region of strategic design at the heart of SE, and to endorse a distinguishable discipline that governs this: systems architecting (SA). This exposed a looseness in the definition of the scope and boundaries of the practice of SE [Emes, Smith and Cowper, 2005], and the potential overlap between SE and SA represented a battleground for the control of knowledge. "Shared abstractions are a means of systematizing knowledge and controlling it … abstractions are the means by which professionals define new problems, redefine existing problems, or defend against competing definitions" [Abbott, 1988: 98].

Architectural frameworks

Demand for a more pragmatic approach to architecting led to a formalization of the conventions and structuring of models that communicated the IEEE 1471 meaning of architecture. Three related but distinguishable strands of methodology appeared (shown in Figure 2



) and led to the publication of a number of influential architectural frameworks.

Firstly, out of an empirical assessment of different engineering domains, Zachman [1987] identified a repeated pattern of model types that relate to the roles, responsibilities and concerns of different parties. This separation of concerns tackled complexity by structuring and navigating models of a system architecture according to common, intuitive categories: essentially, the primary interrogatives: who, what, when, where, why and how. This work clarified that disciplined model construction, organization and management is a central tenet of effective, team-based design and of how architecture is communicated. Other workers in software and IT looked to how process transformations led to the design of architecture. Initially TOGAF [The Open Group, 1995] and related developments basically recast SE process models under the banner of architecture frameworks. However, since TOGAF's 2002 version this process model has been allied with definitions of models of architecture when a system is viewed from different concerns. It thus more closely resembles what is now recognized as an architecture framework. Yet others embarked on standardizing how to structure and populate system models that effectively communicated architecture to different stakeholders with their contrasting concerns throughout the system life cycle. Termed architecture frameworks, these standards address different domains of system application and/or different classes of system, and they provide collective, pretested and reusable templates and modelling rules for different communities.

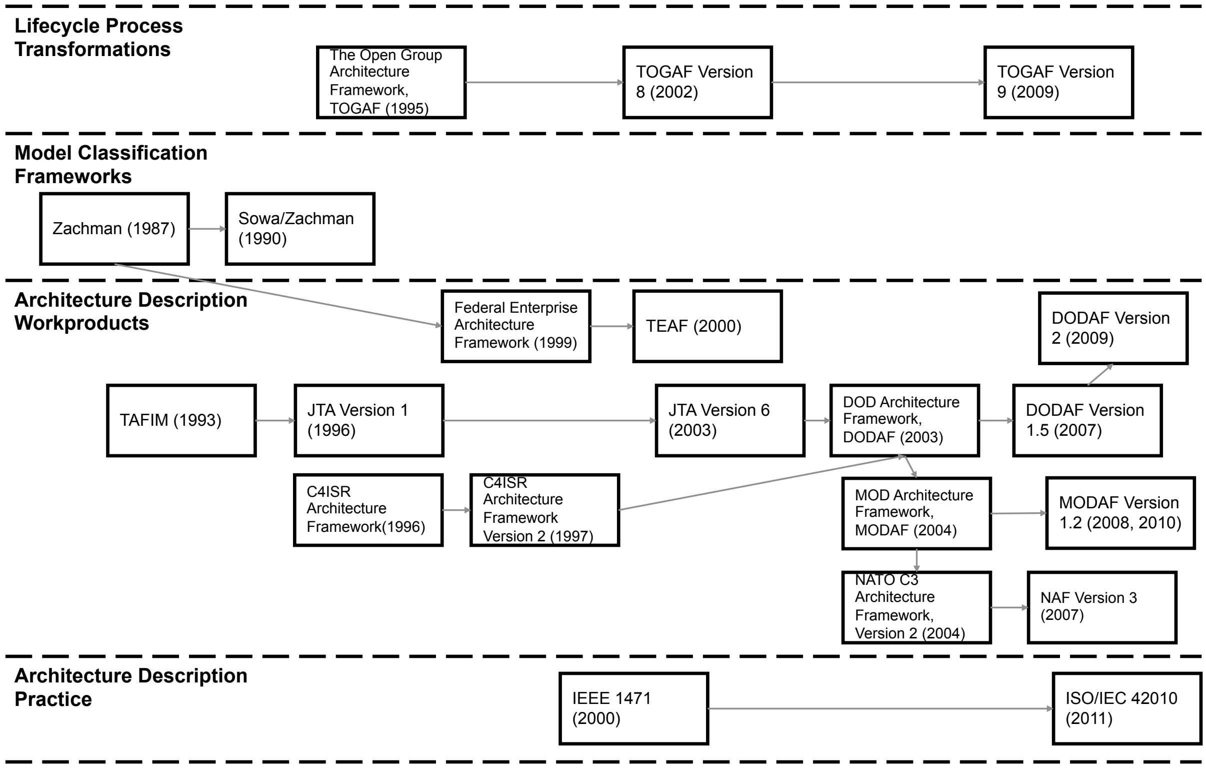


Figure 2: Evolution of Architecture Frameworks

Contentious questions in systems architecting

Despite the apparent unanimity necessary to subscribe to an architectural framework, it has been discovered through numerous debates in UK and international SA forums that little consensus exists over some basic questions relating to the practice of SA. Whilst a range of interpretations is expected from community to community, it is particularly surprising to find little commonality of interpretation of concepts even within communities, with a number of questions proving controversial.

Question 1: Is architecture different from architecture description?

Structure may be considered to exist when it is just a vision in the mind of the architect (before it has been described or built), but others associate architecture with formal description: “by the *architecture* of the system, I mean the complete and detailed specification of the user interface.” [Brooks, 1995: 45]. Whether or not we differentiate between architecture and architecture description, it is clear that neither of the definitions above requires the output of the architecting activity (the architecture) to include a design rationale (although the design rationale will surely have been considered in conceiving the architecture). ISO/IEC 42010 makes a formal distinction between architecture and architecture description, but some people find problematic the idea of an architecture having a separate existence to its description.

Question 2: Should the output of the architecting activity include design rationale or just structure?

In general terms, a system’s architecture is often associated with the perception of order in the composition of a system and the information that brings meaning to this order. In the case of man-made systems, this information may relate to design rationale, including the need for a system, any constraints on its solution, the technological opportunities for system creation, and the circumstances of the system's use. Others understand the output of the architecting activity to be merely the structure of a system – the building blocks and their arrangement: "the essence of architecting is structuring" [Rechtin, 1991: 1].

Question 3: Do natural systems have architecture, and if so, can they be architected?

Architecture is implicit in engineering design: to the observer, man-made systems possess architecture. Whether natural systems can have architecture is probably less obvious. It is clearly possible to describe the structure and arrangement of a rock formation, and to list the constituent bodies of the solar system and the interactions between them. In both of these cases, though, we are defining a system. The architect here is not a design authority but a definition authority. Yet for a system to exist, it seems reasonable to assert that someone or some group must define membership of that system, including the elements that make up the system (or alternatively to identify the system boundary that expresses the limit of system membership) and the interactions of interest. We can call that person the system architect. In a sense, the architect here has built a system for human use (consistent with the OED definition above), since to describe something as a system is to look through a particular lens. A system is a human mechanism for asserting order, or "a way of looking at the world"[Martin, 2008; Weinberg, 2001]: "I had realised before now that it is only a clumsy and erroneous form of perception which places everything in the object, when really everything is in the mind" [Proust, 2000: 275]. Similar, the interpretation of architecture in the built environment may be subjective. "Buildings are solid objects, there is no doubt about that, but they are never in themselves architecture. Architecture is dependent on the observer's culture, and the ideas that are brought to bear on the building … architecture is in the mind of the beholder" [Ballantyne, 2002: 49].

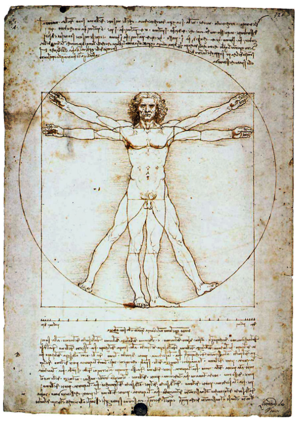


Figure 3: Leonardo da Vinci's Vitruvian Man

The practice of investigating and learning from a system that already exists has been defined as 'reverse architecting' by Rechtin [1991: 208], and more recently by UKAWG [Wilkinson, King and Bryant, 2009; Wilkinson, et al., 2010] based on the view that "the purpose of architecting is to understand existing parts of the environment as systems".

Understanding the architecture of the human body has been a great source of inspiration for artists and scientists since the European Renaissance, with Leonardo da Vinci's Vitruvian Man (1452-1519) shown in Figure 3 a famous early example of the analysis of the structure of the body[5].

[5] This drawing reinforces the Vitruvian principle that the length of a man's outspread arms is equal to his height, and also shows that a circle can be traced with the navel as the midpoint and the raised hands and feet touching the circumference.

Albrecht Dürer (1471-1528), Michelangelo Buonarroti (1475-1564) and Andreas Vesalius (1514-1564) were other famous students of the "architecture of the human body" [McArdle, Katch and Katch, 2009: xxviii]. Architectural patterns and the idea that successful designs should be repeated or inherited is also prevalent in nature, and is the basis of the theory of natural selection [Darwin, 1859]. Indeed, whilst Darwin doesn't use the term architecture, he refers throughout this book to the 'structure' of organisms. More recently, Rechtin refers to biological architectures and the architecture of the brain [Rechtin, 1991: 216-242], and Hersey defines architecture to include non-human buildings and objects such as anthills, beehives, body parts and molecules [Hersey, 1999].

Question 4: How can architectures be evaluated?

What constitutes a good architecture? Can the quality of architecture be predicted or quantified during a system's development? Being able to prove that a complex system's architecture is optimised, balanced and robust[6] would seem to be of great value. In reality, this is possible only insofar as a system can be proven to satisfy a set of requirements, and insofar as a set of requirements if satisfied can be proven to lead to customer (and other stakeholder) satisfaction. In an iterative project development lifecycle, an architecture evaluation needs to recognize the dynamic nature of a system's architecture and the role of good architecting in shaping user and system requirements.

In practice, the quality of an architecture is likely to boil down to a series of numbers (many of which will be subjective) which describe some measure of the system's effectiveness in meeting its requirements, some measure of its cost to develop, some measure of the timeliness of development, and some measure of risk (Maier and Rechtin [2009: 132] identify forces of performance, risk, schedule and cost as tensions to be balanced in the architecting process). How to choose between different architectures with different scores then becomes a value judgment. ISO/IEC is currently investigating the possibility of developing an international standard for Architecture Evaluation (ISO/IEC 42030), which should focus opinions in this area.

Question 5: What is the logical relationship between a system and its' architecture?

Can 'an architecture' apply to more than one system? How many architectures must or can a system have? Can a system exist without an architecture, or can architecture exist without a system? One interpretation is that a system is a human construct and that every system by definition has one and only one architecture, which is subjective and may change at any time through redefinition. Another interpretation is that a system is a real object that can have many architectures – each one looking at the system from a different perspective. Architectural frameworks have been developed for defining rules for creating architectures in various domains. These frameworks and international standards like ISO/IEC 42010 have rules that dictate the relationship between a system and its architecture. Some frameworks use the concepts of an architecture's viewpoints and views, but these terms are applied inconsistently within and across these frameworks.

[6] In the general case of systems with multiple stakeholders with divergent concerns, it will not be possible to optimize for each stakeholder's concerns. Instead we seek a balance that optimally serves the different stakeholders concerns, given the relative importance we attach to each stakeholder.

Question 6: Is systems architecting art or science?

The extent to which architecture (in general) should be considered an artistic endeavour is unclear. The confusion is fuelled by the dual senses of the word ‘art’. In its traditional sense, ‘art’ refers to skill in a particular craft, but in modern usage, the unqualified word ‘art’ refers more often to the (typically visual) creative expression of skill. An architect (of buildings) can be viewed as “a master-builder … a skilled professor of the art of building, whose business it is to prepare the plans of edifices, and exercise a general superintendence over the course of their erection”[OED, 2010].

Engineering often reflects a concern for the economy as well as the effectiveness of proposed designs, a concern that architects of buildings have historically lacked. “The world’s great monuments have been built by consuming resources which were considered vast by the standards of the day … In these buildings function and economy are not the principal forces at work, still less is sustainability a consideration. It is much more important that the building is magnificent than it was done for the minimum cost, and it is in this realm of ruinous consumption that architects have traditionally worked” [Ballantyne, 2002: 37]. Many definitions of architecting like this include at least a suggestion that the role goes beyond analytical decision-making and includes an element of artistic expression or creativity. In the 19th century, this suggestion was reflected in Ruskin’s definition of an architect [Ruskin, 1854: 61] “no person who is not a great sculptor or painter can be an architect. If he is not a sculptor or painter, he can only be a builder”. A modern translation of this view into the realm of systems might be “no person who is not creative can be a system architect”.

Maier and Rechtin [2009: xix] suggest that the role of architecting is weighted towards creativity and the role of engineering is weighted towards analysis. Perhaps one could therefore recast the Ruskin definition “no person who is not creative can be a systems architect. If he is not creative, he can only be an engineer”. Yet many engineers and scientists would be uncomfortable with the idea of major architectural decisions being based on creativity and heuristics rather than analysis. As Lawson points out, design in engineering and design in fashion both require both creativity and technical know-how. “Good engineering requires considerable imagination and can often be unpredictable in its outcome, and good fashion is unlikely to be achieved without considerable technical knowledge. Many forms of design, then, deal with both precise and vague ideas, call for systematic and chaotic thinking, need both imaginative thought and mechanical calculation” [Lawson, 1997: 4].

Question 7: How many system architects should a project have?

Is SA best done by an individual or a team? As systems become increasingly complex, the job of designing and manufacturing them increasingly requires a ‘divide and conquer’ approach of partitioning a system into manageable chunks before integrating the manufactured pieces into a working whole. There is a question as to how many architects should operate in a development process like this. One school of thought argues that the best designs are the work of a single mind. Brooks believes that “conceptual integrity in turn dictates that the design must proceed from one mind, or from a very small number of agreeing resonant minds” [Brooks, 1995: 44]. Two person teams can be particularly effective, but larger teams are inefficient, other than for design reviews, where it is essential that a large number of reviewers are present [Brooks, 2010: 77-82]. Of course, given the hierarchical nature of systems, one or two architects will be required for each system in a top-level system of interest. Consider, for example, a system of interest which is a train with five subsystems, and each subsystem is itself a system with a further five subsystems, below which we are at component level. In this case, the overall system development could have thirty-one architects or teams of architects (one at the system level, five at the subsystem level, twenty-five at the subsubsystem level). It is unclear whether we should consider the top, system-level architects to be special in any way.

Question 8: What is the relationship between systems architecting and systems engineering?

Is SA a subset of SE, or a superset of SE, or are they logically completely distinct? Brooks contends that "the entire system also must have conceptual integrity, and that requires a system architect to design it all, from the top down. To make that job manageable, a sharp distinction must be made between architecture and implementation, and the system architect must confine himself scrupulously to architecture" [Brooks, 1995: 37]. This interpretation, and the definition of an engineer from Table 1 "a person who makes engines, structures or systems" suggests a practical orientation, but many would expect a systems engineer to be responsible for both conception and construction of a system.

The term architecting would seem to have been fuelled by some degree of practice distinction vis-a-vis SE, and also for that matter mainstream software engineering. Rechtin suggested "the architect, therefore, is not a 'general engineer', but a specialist in reducing complexity, uncertainty and ambiguity to workable concepts. The systems engineer, in contrast, is the master of making feasible concepts work" [Rechtin, 1991: 13]. He also proposed that "architecting is working for a client and with a builder, helping determine the preferred architecture, that is, helping determine relative requirement priorities, acceptable performance, cost, and schedule – taking into account such factors as technology risk, projected market size, likely competitive moves, economic trends, political regulatory requirements, project organization, and the appropriate 'ilities' (availability, operability, manufacturability, survivability, etc.) Toward the end of the project, architecting is also certifying completion and satisfactory operation of the system" [Rechtin, 1991: 13]. This should be contrasted with engineering, which is "working with the architect and for a builder, applying the best engineering practices to assure compliance at the system level with the designated architecture and with applicable specifications, standards, and contracts. Toward the end of the project, engineering is certifying such compliance" [Rechtin, 1991: 13]. Rechtin thus promotes the idea that architecting is 'the front end of SE', setting the scene for SE and certifying its results. This relationship is implied in Kruchten's [1995] influential '4+1' view model (Figure 4) which relegates SE to physical concerns. It should however be noted (and is conveyed in Figure 1) that the terminating rule which for architects distinguishes between SA and system design is identical to that applied by SE practitioners in their distinction between design and implementation: that is, the transfer of responsibility to another party to design (and build and supply) a subordinate and more technologically-grounded part of the system.

Maier and Rechtin make the following distinction between the challenges of engineering and architecting. "Generally speaking, engineering deals almost entirely with measurable using analytic tools derived from mathematics and the hard sciences; that is, engineering is a deductive process. Architecting deals largely with unmeasurables using nonquantitative tools and guidelines based on practical lessons learned; that is, architecting is an inductive process." [Maier and Rechtin, 2009: xvii].

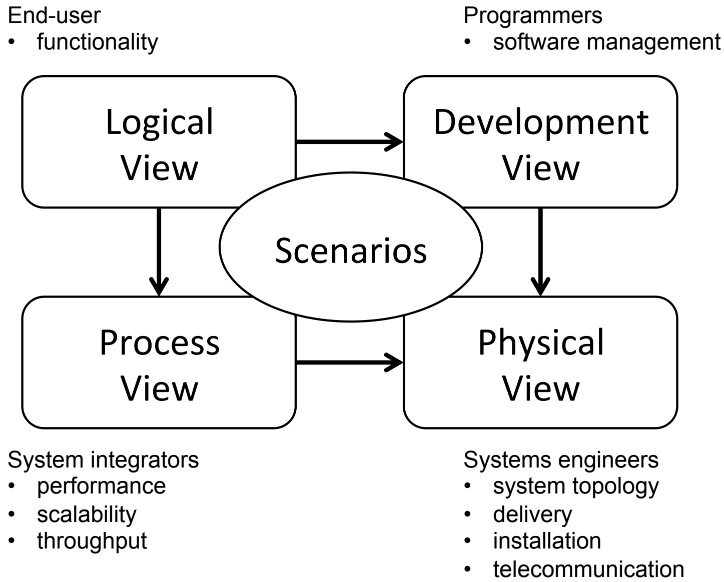


Figure 4: Kruchten's 4+1 View Model

Question 9: What is the relationship between systems architecting and systems design?

Many people make no formal distinction between SA and top-level system design. Some view SA as a form of job-title inflation with the title 'systems architect' now given to those previously called 'system designer' or perhaps 'senior design engineer'. This increase in the use of the title 'architect' is recognised in the built environment as well as in the development of complex systems. "It used to be the case that architects only ever designed temples and palaces, and conferred exalted status on buildings through their involvement. They also, and not incidentally, controlled the expenditure of vast resources … Nowadays most architects are involved with much more pedestrian buildings which need have only modest aims, and that is mostly the purely logical consequence of the fact that far more people today practise as architects."[Ballantyne, 2002: 40]. This suggests that people must have a conscious or subconscious judgment that architecture is more valuable or more important than design or engineering. It is sometimes suggested that architecting is a more creative process than design; as discussed above, Maier and Rechtin [2009] certainly contend that architecture is more creative than engineering. However, Brooks feels that "the setting of external specifications is not more creative work than the designing of implementations. It is just different creative work. The design of an implementation, given an architecture, requires and allows as much design creativity, as many new ideas, and as much technical brilliance as the design of the external specifications" [Brooks, 1995: 46].

Question 10: Who is the architect's customer?

Whom does the architect seek to serve in conducting his or her work? There seem to be two schools of thought here. One view is that the architect represents the user of the system: "the architect of a system, like the architect of a building, is the user's agent. It is his job to bring professional and technical knowledge to bear in the unalloyed interest of the user, as opposed to the interests of the salesman, the fabricator, etc." [Brooks, 1995: 45]. Another common view is that the architect's job is to specify a system to the level of detail necessary to enable detailed design work to be done. In this case, it is reasonable to assert that the primary customer of the architect is the design team.

Others see architects as providing a linking role between user and builder: "The architect is concerned with devising a form that the builder can build and which the user can use. This giving of form is the architectural profession's central … skill. The operation of this skill … very often seems to be a matter of common-sense problem solving. Nevertheless the process of design is not the kind of logical process which would take us step-by-step from initial premises to an inevitable outcome." [Ballantyne, 2002: 32]. Building on this, another important question is how much information the architecture should contain, or to what level of detail it should go.

Question 11: What transformation occurs when architecting is performed?

It is not obvious what change occurs when a system is architected. Should we consider architecting to have been performed when a vision for a system is conceived? Or is it performed when the system's architecture is formally described? Or is it performed throughout the conception and building of a system? When is the architecting task finished? ISO/IEC 15288 suggests that Architectural Design occurs at the third stage of eleven sequential steps in managing a system's lifecycle. Maier and Rechtin consider architecting to occur predominantly at the conceptualization and certification stages of a system's development [Maier and Rechtin, 2009: xviii]), but also describe the architecting process as 'episodic', seeing the architect's design role not restricted to 'high-level' considerations: "architects dig down into specific subsystem and domain details where necessary to establish feasibility and determine client-significant performance" [Maier and Rechtin, 2009: 254].

Question 12: What is the purpose of systems architecting?

There is a basic question that is rarely asked: why do we architect systems at all? Related to Question 5, can a system exist without being architected? Does it matter if we are talking about built or natural systems? If a system can be built without being architected, then in what respects is an architected system superior to a non-architected system?

These contentious questions have been explored through the research project, whose method and findings are described below.

Research Phase 1: Analysis of published sources

Method

For each source, the following process was followed:

1. Read source
   On and after reading each source, an initial interpretation of the concept of architecting in the source is developed

2. Suggest shorthand statement
   A concise summary statement is written, which articulates the underlying beliefs in the source with respect to SA

3. Identify Customers, Actors, Transformation, Worldview, Owner and Environment (CATWOE)
   The CATWOE investigation (following Soft Systems Methodology [Checkland, 1999]) is performed for the process of architecting according to each source. The CATWOE elements are explained below:
   Customers. The beneficiaries of the transformation (these may benefit or lose from the transformation)
   Actors. Those agents responsible for effecting the transformation
   Transformation. The proposed or observed change that is the focus of the soft systems analysis
   Worldview. The beliefs directly relevant to the transformation that explain its purpose
   Owners. Those responsible for the outcomes of the process and with the power to stop it.
   Environment. The constraints or influences imposed by the outside world on the transformation

4. Develop Root Definition
   A Root Definition is developed that describes the major transformation occurring in the architecting process according to each source. This takes the form "do P by Q in order to achieve R", with P, Q and R enriched with the CATWOE information. Here P refers to what the system or process is seeking to do (what is the transformation), Q refers to how the transformation is effected, and R refers to the objective of the transformation.

5. Develop Conceptual Model
   A Conceptual Model is developed which describes the logical sequence of steps that must be followed in order to achieve the transformation as described in the Root Definition, and identifies the control activities necessary to ensure integrity of the process.

6. Analyse terms/concepts in the Root Definition and Conceptual Model
   Having created the Conceptual Model, the terms used in each source are analysed to understand where the sources differ in their use of terms or the concepts to which the terms refer.

It was found to be necessary to iterate this process, and to review the shorthand statement after completing the conceptual model. It was also found helpful to compare the conceptual model developed between the different standards to check consistency, and to review steps 2-5 above once more in the light of this.

Results

The shorthand statements, root definitions (derived from the CATWOE investigation) and conceptual models are compared below. Note that of the three sources, ISO/IEC 42010 and MODAF are primarily concerned with architecture description, whilst Maier and Rechtin is primarily concerned with architecting.

Shorthand statements

ISO/IEC 42010

The primary purposes of architecting a system[7] are (1) to identify, (2) to describe and (3) to improve the fundamental concepts and properties of a system. This in turn facilitates analysis and evaluation of alternative architectures, and communication and co-operation between parties that create, utilize and manage modern systems.

MODAF

Architecting using MODAF ultimately enables a coherent portfolio of military capability and better-integrated systems, whilst avoiding unnecessary costs in the overall investment programme. MODAF achieves this by enabling standardized abstractions of complex real world situations that are amenable to detailed analysis, improving communication between parties that create, utilize and manage modern systems.

Maier and Rechtin

The purpose of architecting a system is to ensure that the system delivers maximum value for its client. This is achieved by applying a mix of heuristics and analysis to the system development process, and by ensuring that the architect represents the client's interests throughout the project lifecycle.

Root definitions

ISO/IEC 42010

A design-authority owned activity, in which a party that creates, utilizes and manages modern, increasingly-complex systems conceives, defines, expresses, documents, communicates, certifies proper implementation of, maintains and improves the "fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" (i.e. architecture) in the context of an organization, driven by a number of stakeholder concerns and technological, business, operational, organizational, political, regulatory, social and other influences, in order to improve system performance and understanding of a system throughout its lifecycle, to manage complexity, and to improve communication and cooperation between the system's stakeholders, thereby helping the system to work in an integrated, coherent fashion.

MODAF

A human activity system, owned by enterprise (military or non-military) owners/managers having responsibility for the delivery of effective, integrated systems, in which system architects (i.e. those responsible for producing the system architectures within the system development teams) produce descriptions of system architectures using MODAF, which provides definitions of common views using standard elements, with the belief that this will allow coherent investigation of present and

[7] ISO/IEC 42010 describes architecture as "fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution". An architecture description is "a work product used to express an architecture" ISO/IEC 42010, Systems and software engineering — Architecture description, Final Draft International Standard, 2011..

future capability of systems in isolation and as part of larger systems, thereby enabling coherent and efficient definition and development of future systems by system development teams and efficient and de-risked integration of these systems into the operational environment.

Maier and Rechtin

A client-employed design-authority owned human activity system, in which one or more individuals with the necessary skills and experience (i.e. the systems architects) works with the client applying background knowledge, heuristics, standards and regulations and constrained by time, cost and risk to turn a vague or poorly understood set of client needs into a coherent set of requirements that can be used as a basis for technical optimization through SE[8], with the understanding that an overarching vision of the structure of the system allowing for feedback between stages of the development lifecycle and the use of heuristics will lead to a better outcome for the client than a linear or waterfall approach to developing the system using SE or analytical techniques alone.

Distinctions between the sources

An integrated conceptual model is shown in Figure 5, which seeks to identify aspects specific to the three sources. The key differences between the sources are summarized in Table 2.

One general observation is that there are few direct contradictions between the sources – they focus on different aspects of the architecting process and are therefore somewhat complementary (as Figure 5 highlights).

ISO/IEC 42010 is the only standard to make a formal distinction between a system's architecture and the system's architecture description. This is significant, because ISO/IEC 42010 also suggests that the process of architecting involves formulation of both architectures and architecture descriptions. In other words, architecting involves both establishing the fundamental concepts and properties of a system (architecture), and describing these concepts and properties in work products (architecture descriptions).

[8] Maier and Rechtin distinguish between the roles of SE and SA. The former is more of a scientific, analytical approach aimed at technical optimization of a system with clear requirements; the latter is an inductive, artistic approach aiming to ensure that a system delivers qualitative worth and client satisfaction: "engineering is concerned more with quantifiable costs, architecting more with qualitative worth. Engineering aims for technical optimization, architecting for client satisfaction. Engineering is more of a science, and architecting is more of an art." [Maier and Rechtin, 2009: xvii].

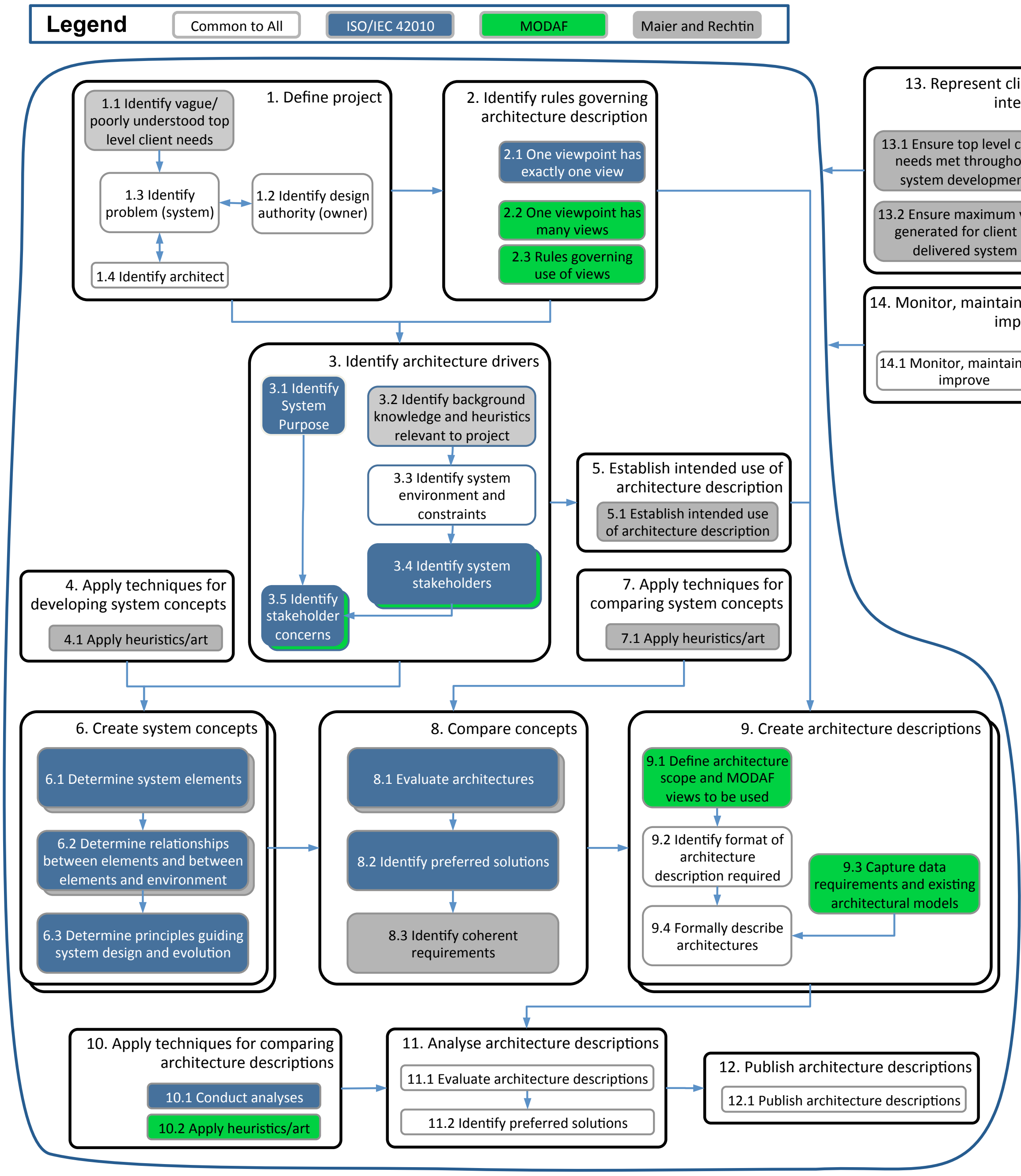


Figure 5: Comparison of the three standards

Maier and Rechtin suggest that architecture concerns only system structure[9]: "architecting is creating and building structures – that is, 'structuring'. *Systems* architecting is creating and building *systems*. It strives for fit, balance, and compromise among the tensions of client needs and resources, technology, and multiple stakeholder interests"[Maier and Rechtin, 2009: 27]. ISO/IEC 42010 sees a broader scope for architecture, suggesting that the relationship with the environment and principles guiding system design and evolution are also important parts of an architecture.

[9] This is not limited to physical structure, and may include information structures, software protocols, etc.

	ISO/IEC 42010	MODAF	Maier and Rechtin
Architecture vs architecture description	Formal definitions	Undefined	Informal definitions
Scope of architecture	System structure, environment, and principles guiding system design and evolution; natural systems can be architected	System structure and behaviour; as an enterprise architecture framework, natural systems are out of scope	System structure only; architecting part of acquisition process so natural systems cannot be architected
How to evaluate architectures	Architecture description must address stakeholder concerns; how to evaluate architecture itself is undefined	Undefined	Apply art/heuristics based on experience
Viewpoints/Views	1 Viewpoint has 1 View (in a given architecture description)	Viewpoint is collection of Views	Refers to ISO/IEC 42010 and MODAF
Architecting vs systems engineering	Undefined	Undefined	Architecting is more of an art, and one of its deliverables is a clear set of requirements; engineering is more of a science, optimizing a system starting from a clear set of requirements.
Customers or beneficiaries	All stakeholders	MOD and others in the supply chain through Viewpoints	Client
Architecting transformations	1. No Architecture → Architecture 2. Architecture but no Architecture Description → Architecture and Architecture Description 3. No Architecture or Architecture Description→ Architecture and Architecture Description 4. Architecture → Improved Architecture	Producing architecture models using MODAF starting from a statement of capability requirements	1. Fuzzy client needs →clear requirements 2. No architecture description → architecture description 3. No system→ system built
Purpose of architecting	To identify and to describe a fundamental system concepts and properties. This facilitates analysis and evaluation of alternative architectures, and communication and co-operation between parties that create, utilize and manage modern systems	To enable a coherent portfolio of military capability and better integrated systems, whilst avoiding unnecessary costs in the overall investment programme, by enabling standardized abstractions of complex real world situations that are amenable to detailed analysis	To ensure that the system delivers maximum value for its client

Table 2: Comparison of beliefs in architecting publications

Maier and Rechtin believe that "architecting takes place within the context of an acquisition process"[Maier and Rechtin, 2009: 20]. This precludes the application of architecting to understand natural systems; ISO/IEC 42010 does not preclude application to natural systems. MODAF is intended to be an enterprise architecture framework, so natural systems are not within its scope.

Maier and Rechtin suggest that the purpose of architecting is ultimately to deliver value for the client – the client is taken to be the predominant stakeholder. MODAF is not inconsistent with this, but is framed in the context of a particular client (MOD) with particular needs (interoperability, through a coherent portfolio of military capability and better integrated systems, whilst avoiding unnecessary costs). ISO/IEC 42010 is also fairly consistent with this, suggesting the value of architecting is derived from improvements in the feasibility, utility and maintainability of the systems to be built. More specifically, ISO/IEC 42010 believes that architecting will lead to improved communication and co-operation between systems, helping them to work in an integrated, coherent fashion. ISO/IEC 42010 is more general than MODAF in its consideration of stakeholders. MODAF implicitly takes the primary stakeholder to be MOD, although different Viewpoints address the needs of different stakeholders. ISO/IEC 42010 suggests that all stakeholders need to be identified formally and their needs/concerns assessed as part of the architecting process. However, no direct link is suggested in ISO/IEC 42010 between a system's stakeholders and its architecture. Furthermore, whilst it is suggested that architecture description identifies stakeholders, there is no recognition of influence in the other direction – i.e. that stakeholders may shape architecture description.

ISO/IEC 42010 requires a one-to-one relationship between Architecture Views and Architecture Viewpoints for a given architecture description (although across multiple architecture descriptions viewpoints may be reused, producing a view for each architecture description). MODAF defines a Viewpoint as a collection of Views, usually grouped by theme. For example, the Strategic Viewpoint consists of a set of Views of a System Architecture dealing with strategic aspects of a system. Since a View in ISO/IEC 42010 is made up of a number of Models, the term Model in ISO/IEC 42010 is analogous to the term View in MODAF.

Maier and Rechtin is the only source to discuss the innovative side of architecting; ISO/IEC 42010 describes architecting as "activities of conceiving, defining, describing, documenting, communicating, certifying proper implementation of, maintaining and improving an architecture throughout a system's life cycle" [ISO/IEC 42010, 2011: 1], which suggests that architecting is not simply the top-level design that occurs in the definition phase of a project. No guidance is offered in ISO/IEC 42010 as to the way in which these activities should be performed (other than the process of architecture description). Maier and Rechtin is the only source to suggest that the role of architect involves a significant amount of heuristic, inductive work as opposed to merely analytical, deductive work.

Note, though, that Maier and Rechtin do not deny the role of analysis in architecting, suggesting "an architect's design role is not restricted solely to 'high-level' considerations. Architects dig down into specific subsystem and domain details where necessary to establish feasibility and determine client-significant performance … The overall process is one of high-level structuring and synthesis (based on heuristic insight) followed by rational analysis of selected details" [Maier and Rechtin, 2009: 254]. MODAF mentions nothing about the role of the architect (the word "architect" is absent from MODAF). Maier and Rechtin is also the only source to discuss the relationship between SE and SA, and the only source to offer suggestions on how to evaluate architectures (primarily using art/heuristics based on experience).

Maier and Rechtin see the specification of requirements as a key output of the architecting process, stating: "architecting is characterized by dealing with ill-structured situations, situations where neither goals nor means are known with much certainty. In SE terms, the requirements for the system have not been stated more than vaguely, and the architect cannot appeal to the client for a resolution as the client has engaged the architect precisely to assist and advise in such a resolution. The architect engages in a joint exploration of requirements and design, in contrast to the classic engineering approach of seeking an optimal design solution to a clearly defined set of objectives" [Maier and Rechtin, 2009: xviii]. Neither ISO/IEC 42010 nor MODAF mention the development of requirements as being a key component of the architecting process.

Research Phase 2: Interviews

Interview Method

In order to explore some of the issues which had been uncovered from the analysis of the published sources, to try to explore some of the contentious questions that had not been addressed through the analysis of the published sources, and to gauge the diversity of opinion that existed across UK industry, we interviewed a total of seven people, representing the rail, defence, aerospace, and communications sectors in the UK.

Each interview lasted for approximately two hours. In each interview, although various topics were discussed depending on the viewpoints of the interviewee, there was a common spine to the interview, including the questions:

1. What is your background?
2. What does the term 'systems architecting' mean to you?
3. What is the ultimate purpose of systems architecting?
4. Thinking of systems architecting as a transformation process, what do you see as the inputs and outputs
5. Who do you see as the beneficiaries of the architecting process, and how do they benefit?
6. Who performs the architecting process?
7. Who has the authority to start and stop the architecting process?
8. Within what constraints does the architecting process have to operate?
9. Do you see the architecting process as art or science?
10. Do you feel there is value in distinguishing between architecture and architecture description?

11. Does the output of the architecting process include the purpose of the system or just its structure?

12. What are the processes that precede and follow systems architecting?

13. How would you measure the efficacy, efficiency and effectiveness of the architecting process?

The questions above enabled us to define the Customer, Actor, Transformation, *Weltanschauung*, Owner, and Environment (collectively referred to using the mnemonic 'CATWOE'), which are required to define a robust root definition in Soft Systems Methodology. Based on Brian Wilson's interpretation of Soft Systems Methodology [Wilson, 2001], the *Weltanschauung* or 'worldview' is a belief that, although not explicitly stated in the root definition, must be true in order for the root definition to make sense [Wilson, 2001: 22]. We can infer that there is a logical link between the worldview and the elements present in the root definition. The nature of this link is not made explicit by Wilson, but 'do X by Y in order to achieve Z' is offered by both Wilson [Wilson, 2001: 24] and Checkland and Scholes [1999: 36] as a check for building root definitions. Indeed, Checkland and Scholes [1999: A22-A24] identify that 'do X by Y in order to achieve Z' ensures that systems thinking occurs on three levels – that of the system (X, describing the doing activity), that of the supersystem (Z, describing the higher level purpose) and that of the subsystem (Y, describing the means by which the transformation is achieved).

The existence of these three levels in a root definition allows us to infer that there are two levels of 'logical' worldview. The first of these is a 'how' worldview, which we will call W1, which is the belief that doing X by Y will enable the transformation to be achieved. The second logical worldview is a 'why?' worldview: the belief that the transformation T will enable the higher level objective Z to be achieved. Note that these logical worldviews W1 and W2 are not additional statements offered by interviewees to support their interpretation of the architecting concept. They are derived automatically from the other elements of the root definition as captured in CATWOE. In fact, further elements from the CATWOE can be incorporated into W1 and W2. W1 could be expressed as the belief that 'A doing X by Y under the authority of O and subject to the context and constraints E will enable transformation T to be achieved'. Similarly, W2 can be expressed as the belief that 'Transformation T achieved within context E will enable Z to be achieved for the benefit of C'.

The usual method of defining a root definition is to construct a single sentence capturing all of the CATWOE elements. Wilson suggests a generic template could be "A system owned by O and operated by A, to do X by Y to customers C in order to achieve Z within the constraints E" [Wilson, 2001: 24]. There are a number of problems with this formulation. Firstly, as we will show below, it leads to the creation of extremely long sentences. Whilst this may seem to elegantly reflect the complexity of the underlying situation we are trying to model, it in fact makes any further analysis rather cumbersome. Secondly, attempting to capture the essence of some quite complex concepts in a single sentence means that some shortcuts and simplifications are necessarily made. For example, 'A system owned by O' suggests ownership rather than start/stop authority for the system. The two will sometimes coincide, but sometimes will not. The subtlety of the distinction is difficult to make within the single sentence formulation. A similar problem is encountered with the 'do X by Y to customers C in order to achieve Z' part of the sentence. These words suggest that the

Customers are having the transformation done to them directly, which will rarely be the case. The Customers identified in CATWOE should properly be all those affected by the transformation. Some of these will be beneficiaries of the transformation; others will be victims of the transformation.

A more accurate expression would be 'A system operated by A under the authority of O to do X by Y causing a number of benefits and detriments to be experienced by a range of stakeholders C (some of whom (C1) we want to maximise the benefits for relative to the detriments, others of whom we want to minimise the harm to (C2) and others of whom we choose simply to ignore (C3)), in order to achieve value Z1 for beneficiaries C1 and possibly some additional higher-level benefits Z2 that are not specific to the principal beneficiaries C1'. This is clearly quite a lengthy sentence even before we try to replace the different CATWOE elements with the relevant words.

The role of the logical worldviews discussed above can be identified by adding them to the standard sentence as follows:

'A system operated by A under the authority of O to do X by Y causing a number of benefits and detriments to be experienced by a range of stakeholders C (some of whom (C1) we want to maximise the benefits for relative to the detriments, others of whom we want to minimise the harm to (C2) and others of whom we choose simply to ignore (C3)), in order to achieve value Z1 for beneficiaries C1 and possibly some additional higher-level benefits Z2 that are not specific to the principal beneficiaries C1, subject to the belief W1 that doing X by Y will enable T_0 to be transformed into T_1 and the belief W2 that transforming T_0 into T_1 will enable Z1 and Z2 to be achieved'.

Given the length of the sentence that would be generated by this template, we decided to present the root definitions in a more visual way that would enable easy comparison between different elements of the worldview. This is shown in Figure 6.

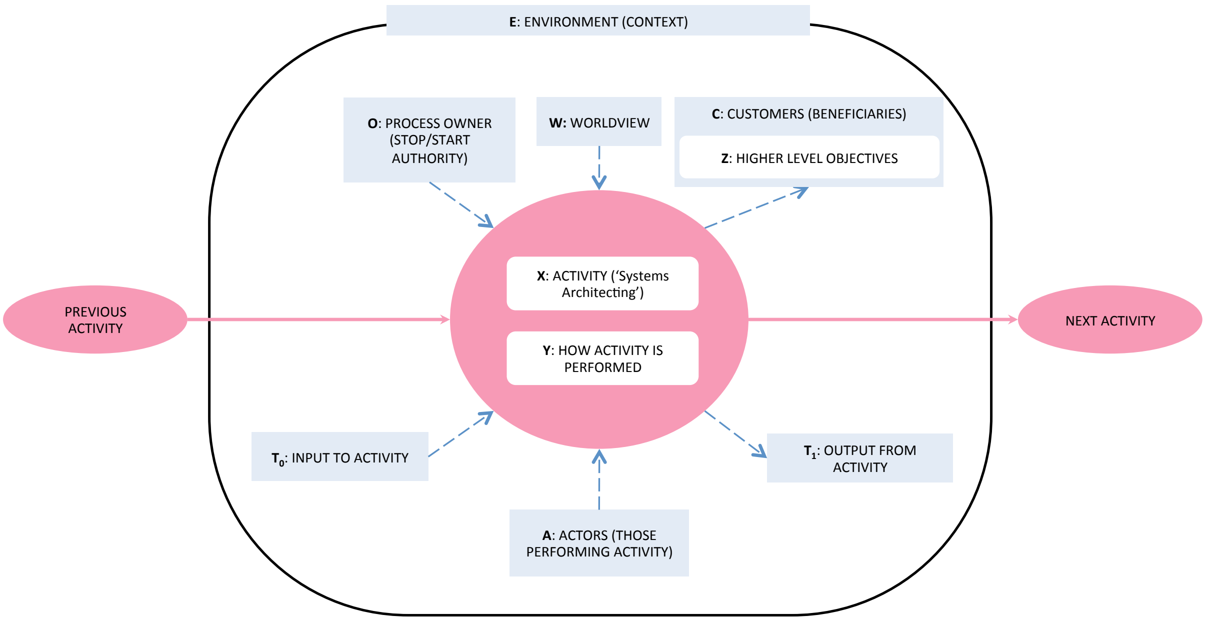


Figure 6: Template for summarizing root definitions

The associated implied worldviews derived from this are:

W1 – How? The belief that A doing X by Y under the authority of O and under context and constraints E will enable T_0 to be transformed to T_1.

W2 – Why? The belief that transforming T_0 to T_1 in context E will enable higher-level objective Z to be achieved benefiting customers C.

Note that in addition to the elements of CATWOE and 'do X by Y in order to achieve Z', we have added the concept of preceding and following steps to the transformation process, to facilitate the development of a conceptual model.

Results of Interviews

The perspectives derived from the interviews are shown in Figure 7 to Figure 12.

Discussion of interview findings

W2 is a shorthand statement that summarises a belief about the purpose of SA according to the root definition in question. We have not developed full conceptual models for each perspective, since we felt that most of the interesting differences between perspectives were captured in the template shown in Figure 6. W2, supplemented with W1 (explaining how the transformation is achieved) encapsulate the essence of each belief system. The six perspectives on SA are summarized in Table 3.

Perspective SA-1

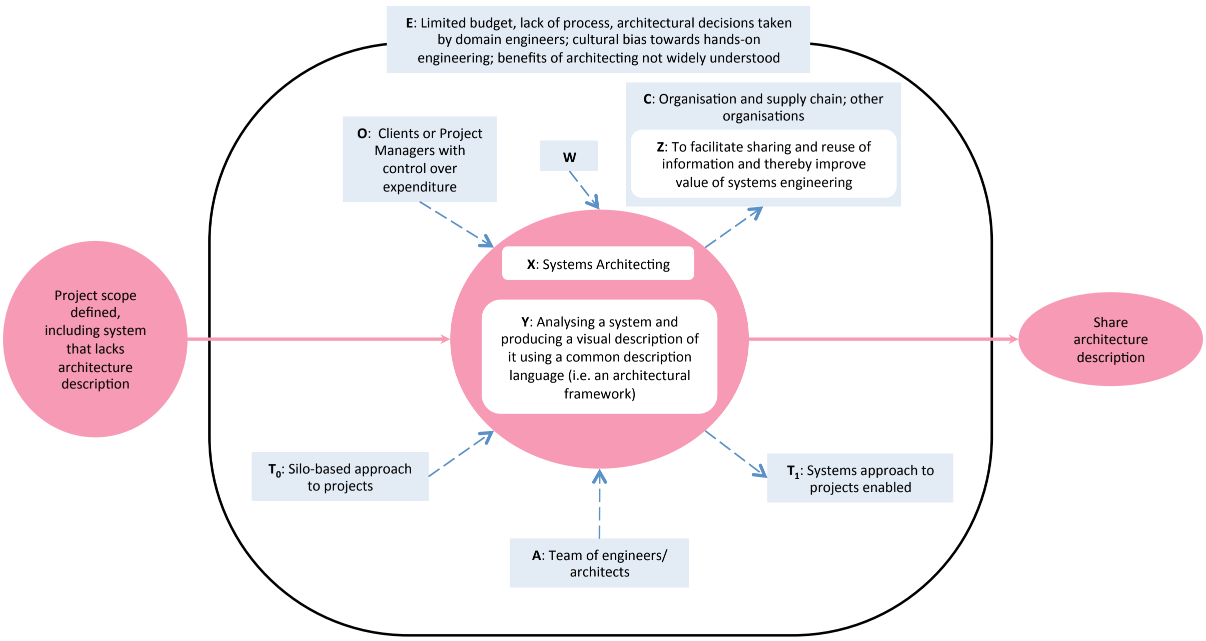


Figure 7: Schematic for Perspective SA-1

Perspective SA-2

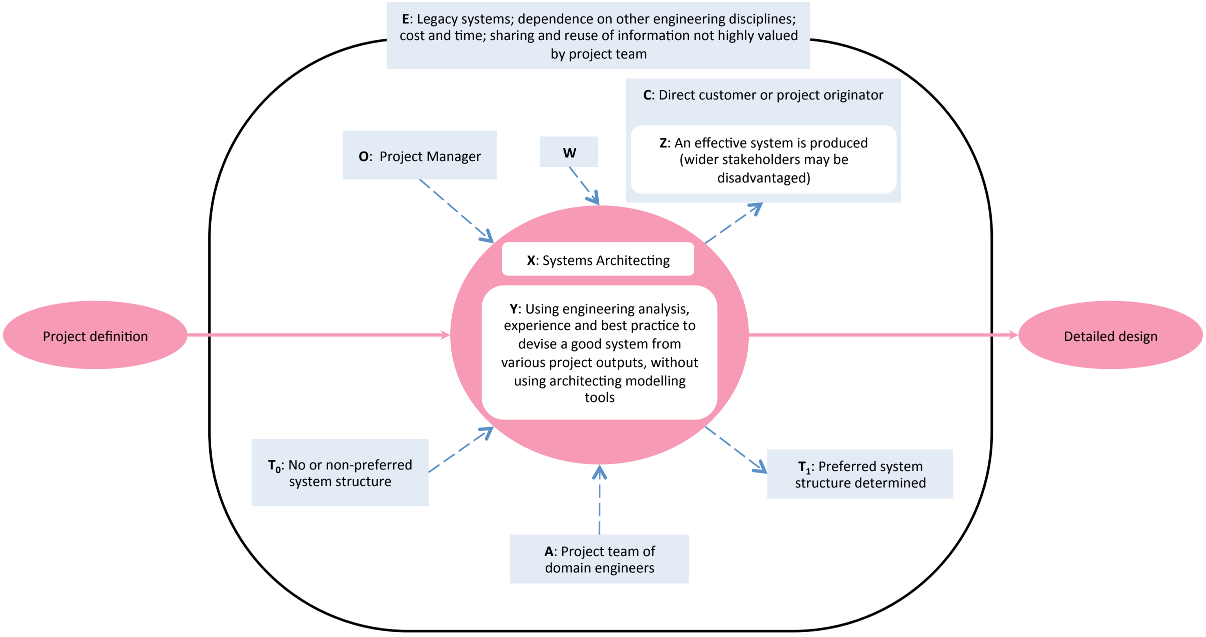


Figure 8: Schematic for Perspective SA-2

Perspective SA-3

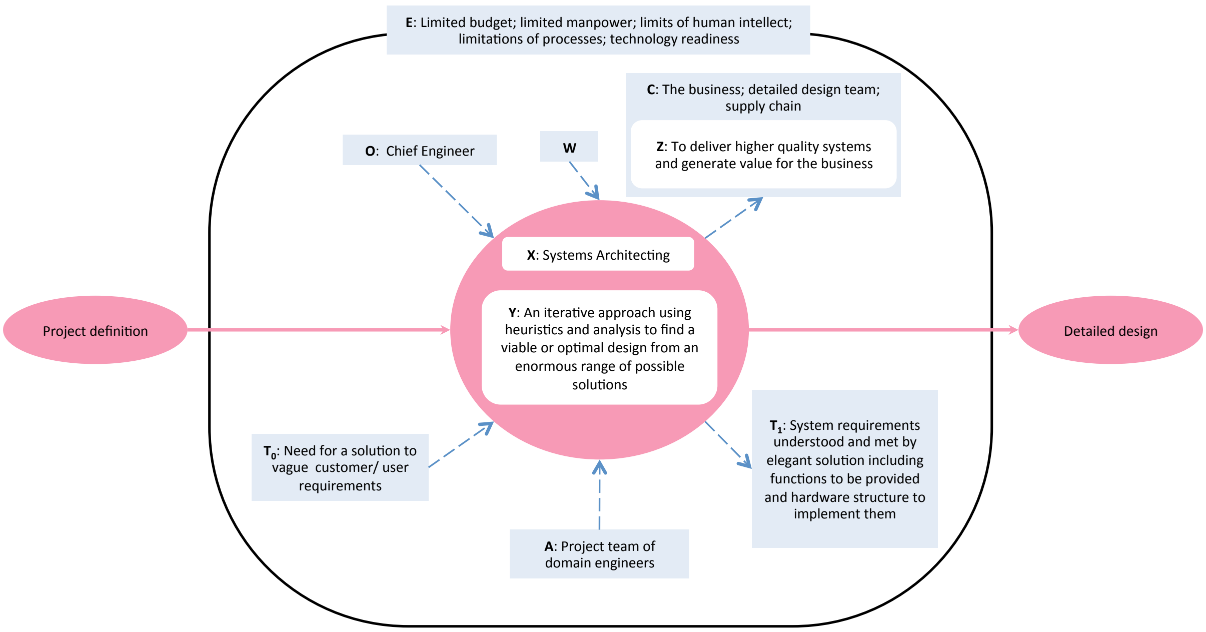


Figure 9: Schematic for Perspective SA-3

Perspective SA-4

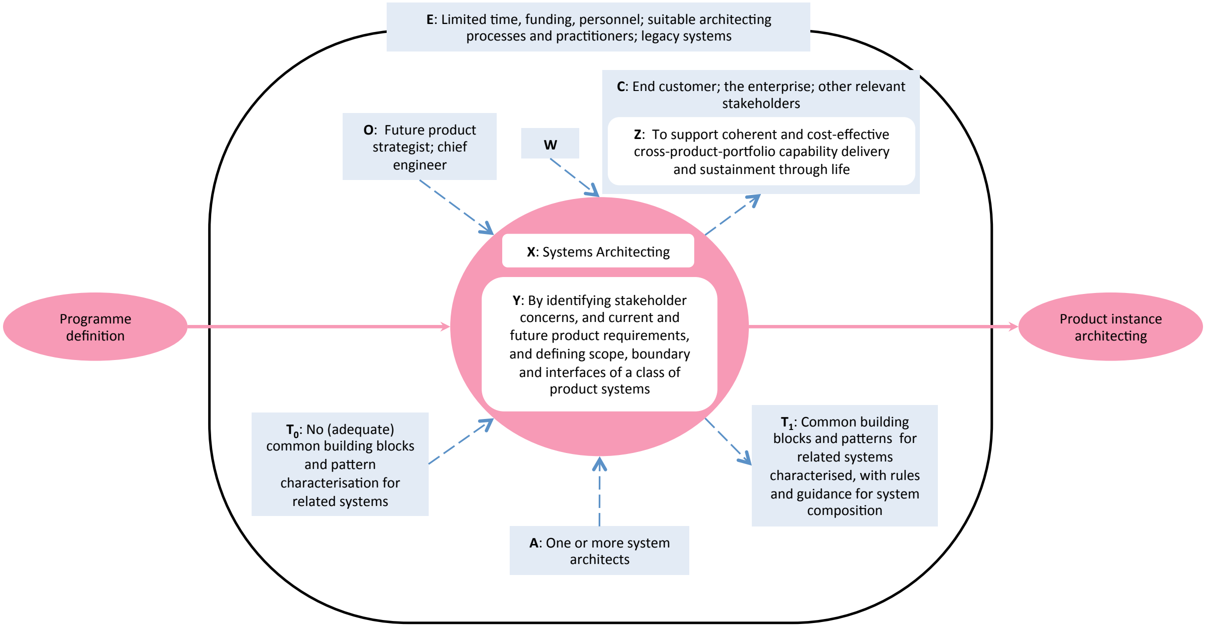


Figure 10: Schematic for Perspective SA-4

Perspective SA-5

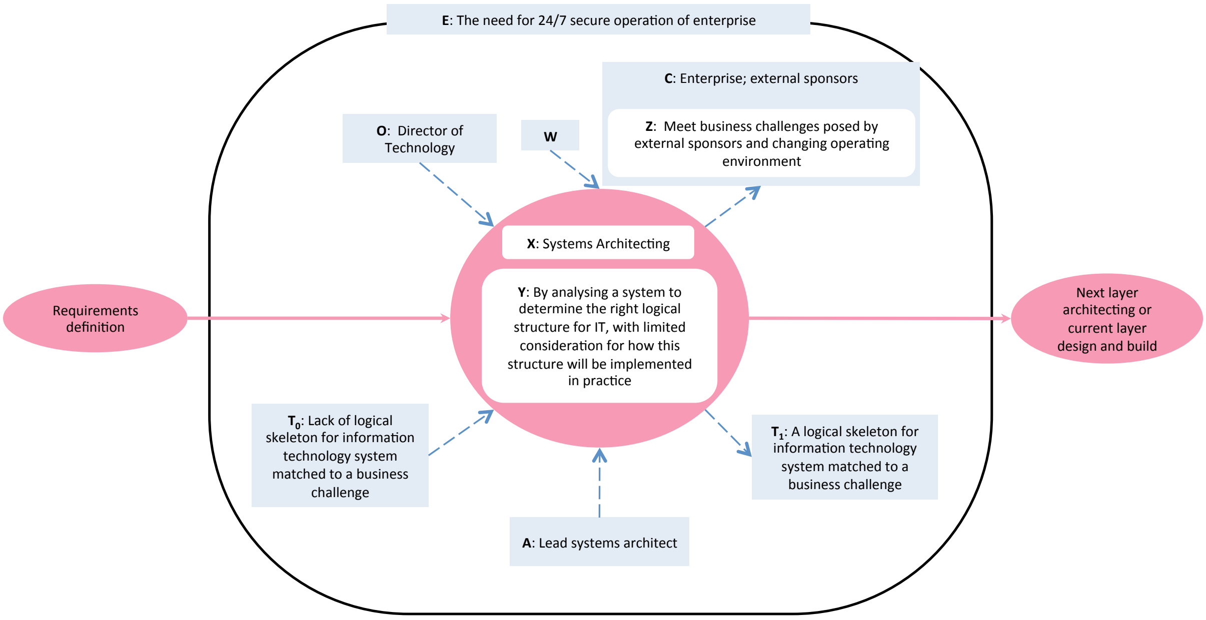


Figure 11: Schematic for Perspective SA-5

Perspective SA-6

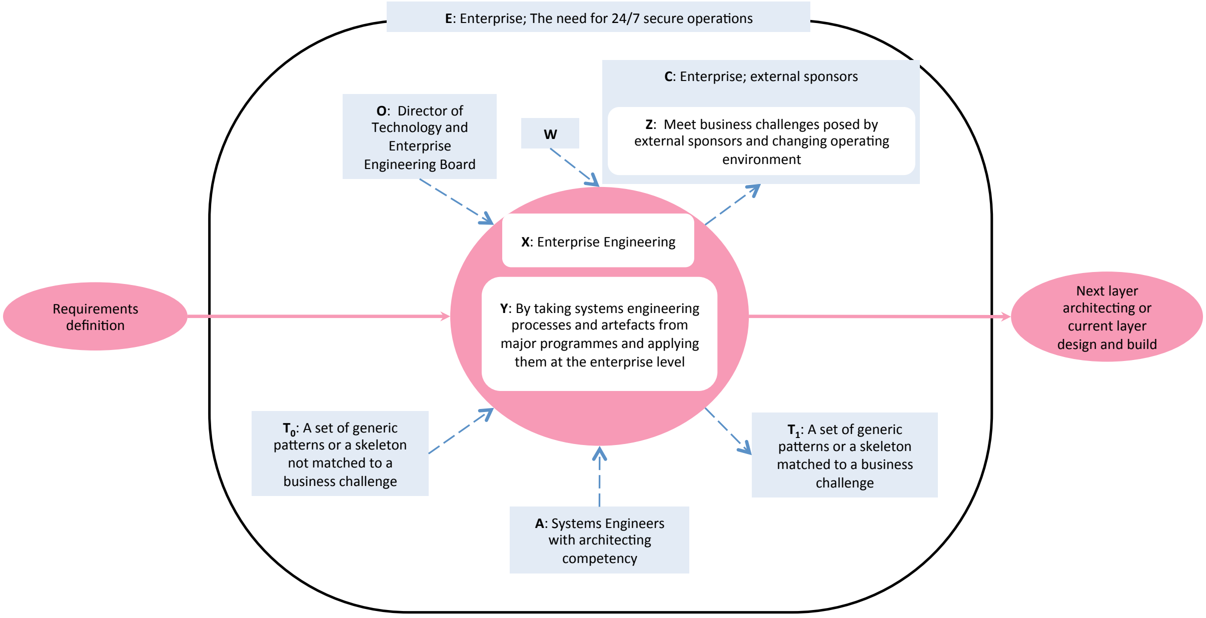


Figure 12: Schematic for Perspective SA-6

The approach adopted for the first phase of the research project (analysis of belief systems in architecture publications) was to start with a shorthand statement, capture the root definition using CATWOE, and then to develop a conceptual model. Although we did try to obtain a shorthand statement by asking interviewees near the start of the interview to describe their interpretation of the term 'systems architecting', we found that the answers received were not sufficiently consistent in structure and scope to permit comparison with other interviewees.

For example, one interviewee suggested that architecting is simply "the design of the design". Another described it as "capture of a system design in a form that is useful for this project and for subsequent projects, and then finding a viable system design from the enormous range of possibilities. To find this you need to capture the requirements and somehow search the solution space. Viable is a minimum, but you really want optimal". We therefore felt that a more useful approach was to build a shorthand statement from the logical worldviews that came from the CATWOE analysis. We used the template shown in Figure 6 to capture concisely the essence of the beliefs, and felt that developing a full conceptual model for each perspective would not add significantly to the findings. Note that each interviewee may have offered zero, one or several of the perspectives captured in Table 3.

General observations

There was an interesting range of opinions on who performs architecting (i.e. who the 'actors' were), and the relative roles of domain engineers and system architects. Some saw SA as an activity that occurred primarily within one engineering specialty (as in SA-2 in Table 3), as these were seen as the only individuals having the authority to make architectural decisions (albeit from a single domain perspective). This view reflects the reality that many organizations face when trying to promote SE – that silos of expertise can dominate over the broader, less detailed, top-down approach that a systems view of a project brings.

	W1 – How?	W2 – Why?
SA-1	Having a team of engineers with limited budget and process (answerable to clients or project managers with control over expenditure) analysing a system and producing a visual description of it using a common description language enables a shift from a silo-based approach to projects to a systems approach in a world where most architectural decisions are taken by domain engineers, there is a cultural bias towards hands-on engineering, and the benefits of architecting are not widely understood.	Going from having a silo-based approach to projects to a systems approach to projects will facilitate sharing and reuse of information and thereby improve the value of the SE activities of the organization its supply chain, and other organizations.
SA-2	Domain engineers acting under the authority of project managers and constrained by legacy systems, cost and time and depending on other engineering disciplines can devise an ideal structure of a system formed from various project outputs, by using engineering analysis, experience and best practice without using architectural modelling tools.	Determining the ideal structure of a system will enable a system to be made that is valued by the project team and the direct customer, but wider stakeholders may be disadvantaged, in a world where reuse and sharing of information is not highly valued.
SA-3	A lead architect and his/her supporting team working with limited budget, manpower (including limits of human intellect) and technology under the authority of a chief engineer can use an iterative approach using heuristics and analysis to find a viable or optimal design from an enormous range of possible solutions, understanding and meeting initially vague user/customer requirements with and elegant solution including functions to be provided and hardware structure to implement them.	Turning a need for a solution to vague user/customer requirements into an elegant solution to a clearly specified set of system requirements helps the team responsible for detailed design and ultimately delivers value to the business and the supply chain by delivering higher quality systems.
SA-4	A single system architect or team of architects under the authority of a future product strategist or chief engineer, operating with limited funding, time and staff and constrained by legacy systems and availability of suitable architecting processes and practitioners, can characterise a set of common building blocks and patterns for related systems, with rules and guidance for system composition, when no adequate characterization previously existed, by identifying stakeholder concerns and current and future product requirements, and defining scope, boundary and interfaces of a class of product systems.	Characterizing common building blocks and patterns for related systems with rules and guidance for system composition will benefit the end customer, the enterprise and other relevant stakeholders by supporting coherent and cost-effective cross-product-portfolio capability delivery and sustainment through life.
SA-5	By analysing a system to determine the right logical structure for information technology, with limited consideration for how this structure will be implemented in practice, a Lead System architect answerable to the Director of Technology can develop a logical skeleton for an information technology system matched to a business challenge whilst needing to maintain 24/7 secure operation of the enterprise as a whole.	Having a logical skeleton for an information technology system matched to a business challenge will benefit the enterprise by meeting business challenges posed by external sponsors and the changing operating environment.
SA-6	By taking SE processes and artefacts from major programmes and applying them at the enterprise level, systems engineers with architecting competency answerable to the Director of Technology create a set of generic patterns or a system skeleton matched to the current business challenge, under the constraints of having to maintain 24/7 secure operations.	Generating a set of generic patterns or a system skeleton matched to the current business challenge will ensure that performance targets as set out by the customer or sponsor are met or exceeded.

Table 3: Summary of Perspectives on Systems Architecting

The extent to which power is in the hands of domain specialists may be industry specific, depending on the history of an industry and the extent to which new approaches are constrained by needing to maintain compatibility with existing infrastructure. It may also be the case that industries with a longer association with the practice of SE are more likely to see the role of system architect as being responsible for whole system performance, rather than just having responsibility for a part of the system. Unfortunately, there were not enough interviews to uncover definitive answers to these questions.

The words ‘system’ and ‘enterprise’, and the words ‘architecting’ and ‘engineering’ could be considered to be alternatives that can be combined into four pairs: ‘system architecting’, ‘system engineering’, ‘enterprise architecting’, and ‘enterprise engineering’. The adoption of these terms in an organization seems somewhat arbitrary, with a failed initiative under a different label sometimes precipitating the adoption of a new label for a seemingly similar activity. Although the general trend when a change has occurred has been for SA to be added to the language to supplement or supplant SE, at least one interviewee felt that SA was a deprecated term which had been replaced by SE. It should be noted that the interviews focused on the use of the term ‘systems architecting’; any other terms (such as ‘enterprise architecting’) encountered came from interviewees rather than the interviewers.

All of the interviewees saw architecting as a blend of art and science [Maier and Rechtin, 2009: xx], or more specifically as creativity and heuristics (based on experience and best practice), and analysis based on data and logic.

According to Maier and Rechtin, “An initial architecture is a vision. An architecture description is a set of specific models” [Maier and Rechtin, 2009: 20]. Only one of the seven interviewees saw the value in distinguishing between architecture as vision and architecture description as a set of models. The main international standard on SA also makes the distinction between architecture and architecture description [ISO/IEC 42010 CD0.8, 2009].

For two out of the four industry sectors, more than one person was interviewed from the same organization. In both cases, there was significant consistency between views expressed both when the interviews were conducted in isolation and on the one occasion when two people from the same organization were interviewed together.

Interviewees believed that architecture should include description of purpose, not just blueprints showing structure.

Few people interviewed could answer the question of who owns the architecting process, i.e. who has the authority to start it and stop it? When we investigated the question of how the effectiveness of architecting could be measured, there was again very little understanding of how that could be achieved. There was even surprisingly little understanding of who the main beneficiaries of the architecting process would be in any given situation – in other words, for whose benefit the architecting was being performed.

The relationship between Systems Engineering and Systems Architecting

From our research we have also been able to explore the range of perspectives on the relationship between SE and SA in different communities and different publications.

Perspective 1: Systems Architecting is a subset of Systems Engineering

SE is an interdisciplinary approach and means to enable the realization of successful systems; SE creates systems. SA is a subset of this, focusing on the top-level structure (or top-level design) of the system; this activity follows the activity of requirements analysis as shown in a v-diagram (see Figure 13).

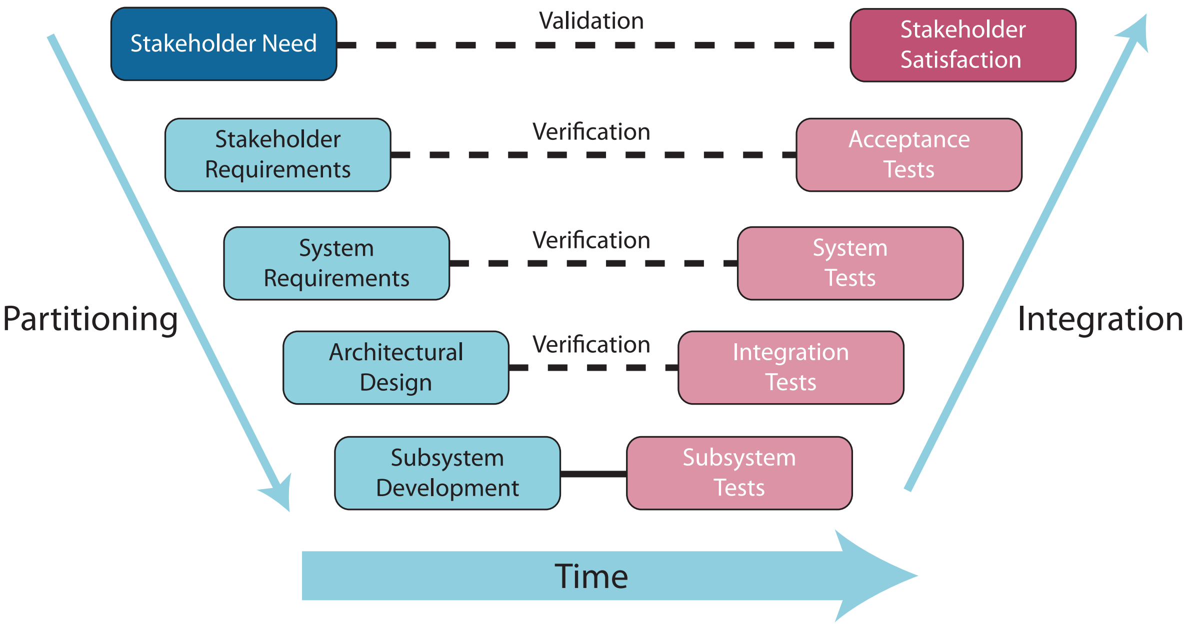


Figure 13: Systems Engineering V-diagram including Architectural Design process

In fact, it is a common view that System Design is a subset of SE, and that SA is simply a more fashionable term for the outmoded term Systems Design (or possibly top-level System Design) as shown in Figure 14.

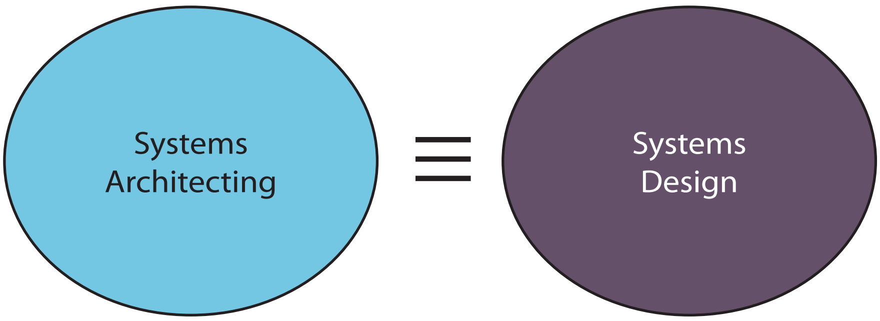


Figure 14: Systems Architecting and Systems Design are the same thing

Perspective 2: Systems Engineering is a subset of Systems Architecting

SA is the art and science of designing and building systems. Systems Architects are involved from concept to delivery, to ensure client needs are met. SE is a key subset of this, concerned with analysis and optimization, and successful integration of system elements. The requirements for the system are defined by the SA process and are an input to the SE activity.

Perspective 3: Systems Engineering is a distinct activity following Systems Architecting

SA is the first step in developing a system, and is concerned with defining the top-level blueprint for the system and specifying the system's requirements. SE is concerned with delivering a solution that will meet the system requirements according to this blueprint. SA and SE are two distinct stages in the system development process

Perspective 4: Systems Architecting and Systems Engineering have areas of overlap and areas of distinct focus

There are aspects of SE that are not contained in the activity of SA. There are aspects of SA that are not contained in the activity of SE. There are some areas of SA that are also aspects of SE

Perspective 5: Systems Architecting and Systems Engineering are distinct activities with no logical relationship

SA is an imaginative, creative and analytical endeavour concerned with defining a system's actual or ideal structure in terms of building blocks and relationships. SE is a practical endeavour concerned with delivering complex systems in a cost-effective, constrained manner. SA can be performed without SE (e.g. concept definition or 'reverse architecting' an existing system), and SE can be performed without SA.

Perspective 6: Systems Engineering and Systems Architecting are the same thing

There are three variants to this perspective.

1. SA is simply a more fashionable term for the outmoded term SE. 'Systems Engineering' suggests an unhelpful focus on mechanical or other specialist engineering, whilst 'Systems Architecting' is domain neutral and can be applied in any industry (including to non-engineering disciplines like health, retail and banking). To all intents and purposes, though, the two terms are interchangeable. The replacement of SE with SA may seem a trivial matter, but for some, the job title 'systems architect' suggests a greater status than 'systems engineer'. Herein lies a potential problem, since the use of the title 'architect' is regulated under law to preserve the professional role of the architect in the construction industry. This means that the titles 'systems architect' and 'systems engineer' can technically only be used by licensed architects: "a person shall not practise or carry on business under any name, style or title containing the word 'architect' unless he is a person registered under this Act" [Parliament of the United Kingdom, 1997: 10]. As the value of SE becomes increasingly recognised in the construction industry, the legitimacy of the term SA will come under greater scrutiny.

2. SA has replaced SE as state of the art for designing and building complex systems. SA is focused on a shared concept of system structure and purpose from the outset (improving communication), and a constant focus on meeting the customer's needs throughout the development lifecycle. SE is a traditional, sequential approach to developing simple systems, which cannot deliver effective systems in a rapidly changing marketplace.

3. SA was a fad that failed to produce practical solutions and has been replaced by the more holistic practice of SE. SA was too often the activity of an isolated group who hadn't thought through implementation, and was primarily concerned with information technology. SE is now recognized as a more holistic approach to system development that focuses not just on delivering functional requirements but on ensuring that the system as envisaged can be built in a cost-effective and timely manner.

Clearly, there is a significant range of interpretations of the relationship between SE and SA. These interpretations are summarized graphically in Table 4.

Perspectives on the Relationship between Systems Engineering and Systems Architecting

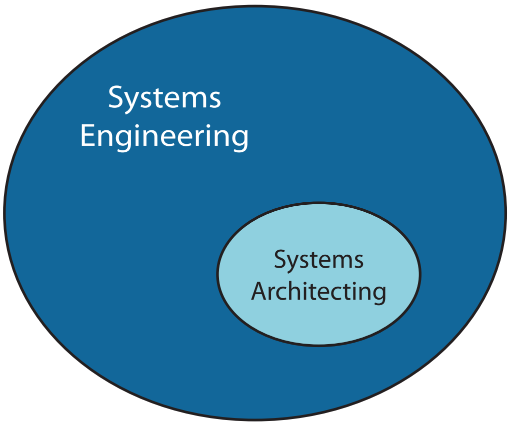


Figure 15: SA as a subset of SE

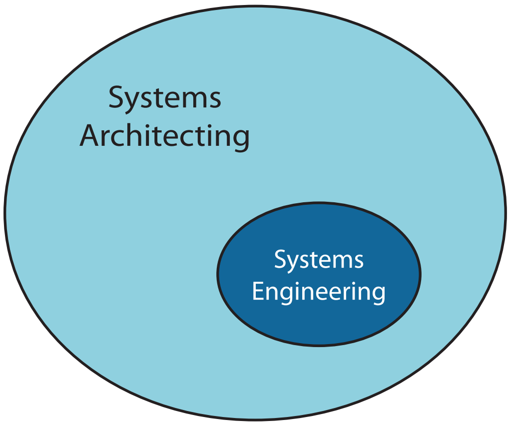


Figure 16: SE as a subset of SA

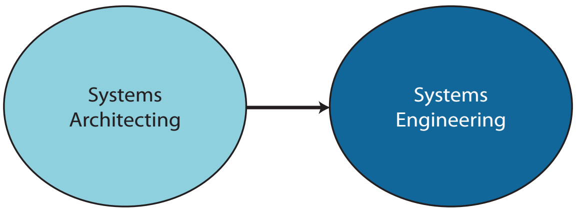


Figure 17: SE as a separate activity following SA

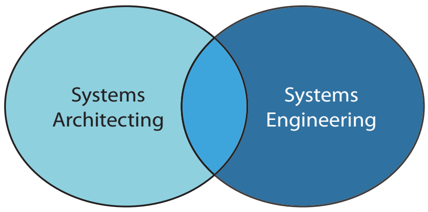


Figure 18: SE as an activity that overlaps with SA

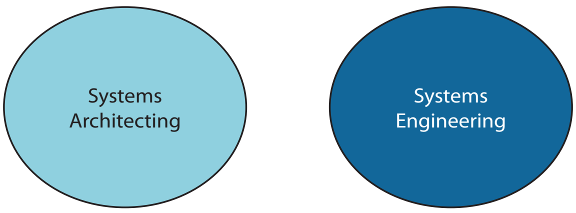


Figure 19: SA and SE as logically independent activities

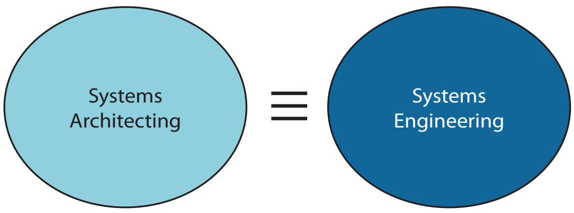


Figure 20: SA and SE as the same thing

Table 4: Summary of perspectives on relationship between Systems Engineering and Systems Architecting

When presenting the six perspectives to practising architects, we have generally found that two of the perspectives are supported, two are acceptable in theory but not necessarily adhered to, and two are rejected. The problem is that even within industries, there is little consistency over which models are supported and which are rejected. To explore this issue further, it would be useful to send questionnaires to a wider community of systems engineers and non-systems engineers. This would provide a quantitative assessment of the popularity of different perspectives in different industries.

Ad hoc discussion within the community of systems engineers already interviewed suggests that easily the most widely held perspective is that shown in Figure 15 – that SA is a subset of the larger activity of SE (or a specialization of SE). The second most commonly held view was that SA and SE are overlapping disciplines, with each having some unique features (as shown in Figure 18). Identifying the nature of these unique contributions in any general sense proves a challenge, however. Specific industry examples can be used to show how certain practices are followed as 'systems architecting' and other practices as 'systems engineering', but when we look at other industries, and sometimes just different organizations within an industry, the patterns can be lost.

Those that recognise SA as something new (i.e. those that believe it is more than just a more fashionable label for SE) generally identify that it has emerged to address a perceived shortcoming in traditional system development approaches (in particular the simple sequential waterfall model of system development with limited interaction with the customer during the development and little consideration of socio-political or economic factors as described in [Rechtin, 1991: 3]). In recognition of these shortcomings within 'traditional SE', two main worldviews have been adopted. In the first worldview, adopted by most advocates of systems engineering, best practice in SE is modified to incorporate some new ideas that could be labelled SA. This is equivalent to suggesting that traditional SE was missing something that is important in the delivery of successful systems, and SA can plug the hole (see Figure 21), with the result that SA is a subset of SE as in Figure 15.

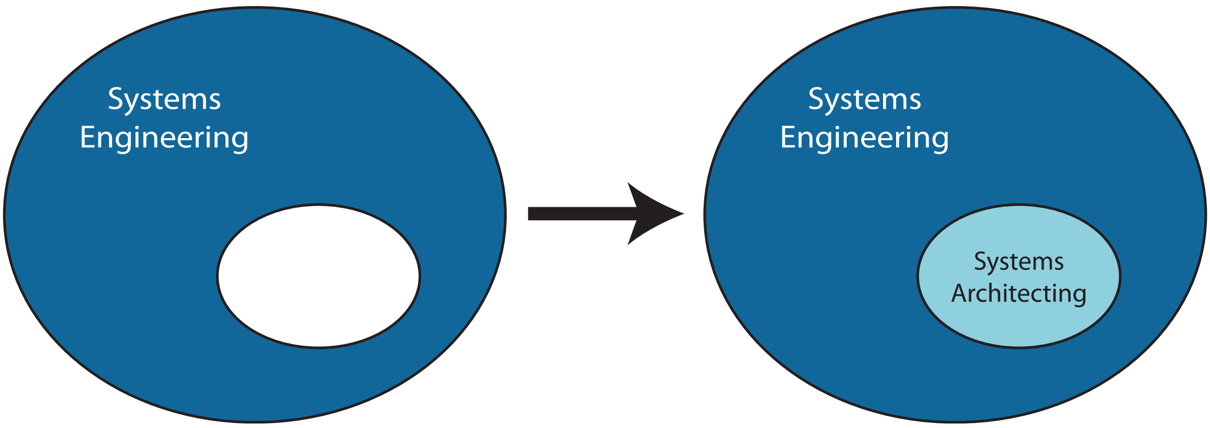


Figure 21: Systems Architecting has been added to Systems Engineering to improve it

The second worldview, more favoured by those outside the systems engineering community, sees systems engineering as fixed and unchanging, and supplements this with a separate activity called SA. SA may be seen as a precursor to SE as in Figure 17, or may happen before, during and after traditional SE is completed as in Figure 16. Either way, in this worldview, the focus and responsibilities of SE remain unchanged, but by adding SA to SE, more effective systems can be delivered. Both of these worldviews are theoretically defensible, but to view SE as unchanging requires us to believe that the body of knowledge in SE is essentially static, at least in those areas that pertain to SA.

SA is a relatively recent innovation, and it would be interesting to track the adoption of different perspectives on SA over time. According to Rogers, around half of the variance in innovations' rates of adoption is accounted for by the perceived attributes of the innovation, namely, relative advantage, compatibility, complexity, trialability and observability. The other half is accounted for by the nature of the innovation decision, the communication channels, the nature of the social system (interconnectedness, etc.), and the extent of change agents' promotional efforts [Rogers, 2003: 222]. Novel terminology scores poorly against many of these factors unless there is a strong change agent – in this case, standards bodies. With the publication of and adherence to architectural frameworks and ISO/IEC 42010, we would expect to see significant convergence in attitudes towards SA over time.

Conclusions

The application of the belief systems approach based on Soft Systems Methodology to the three sources proved useful in identifying belief systems including commonalities and distinctions between the three standards. A secondary benefit of the belief systems approach was that it enabled a more focused discussion on the terms and concepts in systems architecting than had been possible previously. The six-step method initially intended was iterated in considering each source, and having considered all three sources, to ensure overall consistency in application of the approach.

Soft Systems Methodology and the Belief Systems Approach developed by the UKAWG has proved a useful tool for exposing the range of interpretations of the term systems architecting that exist in the UK systems engineering community. We found the conventional single-sentence expression of a root definition unhelpful for comparing different perspectives on the same system, though, and therefore developed a standard visual template for expressing views relating to systems architecting integrating the CATWOE elements of soft systems methodology, the 'Do X by Y in order to achieve Z' formulation, and the ideas of preceding and subsequent activities. For a textual expression of a root definition, we proposed using the pair of logically derived worldviews W1 (explaining how the transformation is performed) and W2 (explaining how the execution of the transformation leads to value for one or more beneficiaries). These two sentences seem to be easier to digest than a single-sentence root definition, but capture all of the same information necessary to summarize the essence of the belief.

Through the interviews we encountered a broad range of interpretations on the meaning of the term systems architecting, despite the fact that most of the interviewees were known to have dedicated significant time contemplating and discussing the nature of systems architecting prior to the interviews. In fact, over five years of discussion in the UK Architecture Working Group has established nothing more emphatically than the fact that people have different views on what the term systems architecting means. Language is not science. Combinations of words can be seen as systems with emergent meaning; attempting to define expressions unambiguously or to formally establish that one interpretation of a term is better than another is therefore problematic. Perhaps the diversity of views reflects the fact that different communities have at different times independently identified various shortcomings in their planning of technology projects, and felt that the words 'architecture' or 'architecting' could describe the fix that was needed. In this case, it is natural that a broad range of cultures and organizations would use the terms and that many interpretations would exist. It is also no surprise that international standards organizations have found difficulty in agreeing standard terms in the areas of systems engineering and systems architecting, since the term 'system' has no universally accepted definition, and "architecting is an invented word" [Maier and Rechtin, 2009: xx]. Systems engineering is also a relatively immature discipline, and lacks the scientific foundation of other engineering disciplines.

While the lack of consensus impedes the establishment of clear international standards in the areas of systems engineering and systems architecting, the *value* of a standard with explicit definitions of the central terms becomes all the greater. Future revisions of ISO/IEC 15288 should clarify the role of architecting. At present, architecting is merely encapsulated within the 'architectural design' process; at the very least, the standard should acknowledge the recursive, hierarchical nature of architecting as shown in Figure 1, but it would be more helpful to see the standard going further and expressing a position on the relationship between system architecting and system design. Most perspectives on systems architecting are founded on the premise that traditional systems engineering lacks something, and that adding systems architecting to systems engineering leads to the delivery of better systems. One of the main areas of contention is whether systems architecting is seen as a practice within an improved systems engineering process, or whether systems architecting is a distinct activity that occurs outside the boundaries of systems engineering (as discussed in Table 4). Here, again, future revisions of ISO/IEC 15288 should provide guidance.

The difficulty of definition is recognized in the System Architecture 'Z-Guide' published by INCOSE UK as an introductory guide to systems architecting, particularly as practised in the UK: "architecture is a popular and evidently useful concept, with many practical benefits … unfortunately for the novice and the unwary there are many different interpretations in widespread use" [Wilkinson, 2010: 1]. The Z-guide attempts to synthesize these interpretations, suggesting, "The architecture of a system is its fundamental structure – which may include principles applying to the structure as well as specific structures". Even this definition concedes that some questions remain unanswered or context dependent, in particular just what qualifies as 'fundamental structure'.

Further investigation needs to be done to extend the set of perspectives on systems architecting, both in the UK and overseas. We suggest that a two-pronged approach could be useful here: firstly, to request members of the UKAWG to develop their own perspectives on systems architecting using the template shown in Figure 6, and secondly, to develop a questionnaire for members of INCOSE (UK and global) to ascertain the prevalence of the beliefs uncovered through this research (and to identify additional beliefs not yet discovered). Based on this wider information set, it might be possible to correlate particular beliefs with specific industry sectors, which would give further insight into the practice of system architecting across sectors. Ideally, this questionnaire could also be distributed more broadly outside INCOSE, to those practising in areas that INCOSE members would recognize as systems engineering or systems architecting. One potential obstacle to achieving a cross-sector consensus view of systems architecting will come from the construction industry. Here, use of the title 'architect' is strictly regulated (in the UK, for example, it is technically illegal to use any title containing the word architect without being registered as an architect with the relevant authority). This tension will need to be addressed in the future as the construction industry, with increasingly 'intelligent' buildings, seeks to embrace practices, processes and standards from systems engineering.

We have not investigated the use of the term systems architecting in the area of system-of-systems engineering; neither have we formally discussed the term enterprise architecting and sought to distinguish this from systems architecting. These may also prove interesting avenues for further research.

Acknowledgements

This project was funded by INCOSE UK. In addition to the research carried out by UCL, this paper draws significantly from UKAWG meetings, workshops and personal communications with UKAWG members, and the authors would like to thank those members and the organizations named in the introduction for their contributions. Thanks also to the seven interviewees for kindly giving up their time, and to Mark Maier for providing many useful insights in reviewing this paper.

References

A. Abbott, The System of Professions: An Essay on the Division of Expert Labor, University of Chicago Press, Chicago, 1988.

D. Aerts, L. Apostel, B. De Moor, S. Hellemans, E. Maex, H. Van Belle and J. Van der Veken, World views: From fragmentation to integration, VUB Press, Brussels, 1994.

G. M. Amdahl, G. A. Blaauw and F. P. Brooks, Architecture of the IBM System/360, IBM Journal of Research and Development 8(2) (April 1964), 87-101.

ANSI/EIA-632, Processes for Engineering a System, 1999.

ANSI/IEEE 1471, Recommended Practice for Architecture Description of Software-Intensive Systems, 2000.

S. Arnold, Transforming Systems Engineering Principles into Integrated Project Team Practice, Cranfield University, 2008

A. Ballantyne (Editor), What is Architecture?, Routledge, London, 2002.

J. Bendz, Presentation to INCOSE Special Architecture Workshop, IW08, Albuquerque, New Mexico, 2008.

F. P. Brooks, "Architectural philosophy," Planning a computer system - project stretch, W. Buchholz (Editor), McGraw-Hill, New York, 1962, 5-16.

F. P. Brooks, The Mythical Man-Month, Addison Wesley, Boston, 1995.

F. P. Brooks, The Design of Design, Addison Wesley, Upper Saddle River NJ, 2010.

Carnegie Mellon Software Engineering Institute, CMMI, http://www.sei.cmu.edu/cmmi/, 2011 [cited].

P. Checkland, Systems Thinking, Systems Practice, Wiley, Chichester, 1999.

P. Checkland and J. Poulter, Learning for Action: A Short Definitive Account of Soft Systems Methodology and its use for Practitioners, Teachers and Students, Wiley, Chichester, 2006.

P. Checkland and J. Scholes, Soft Systems Methodology in Action, Wiley, Chichester, 1999.

C. Darwin, The Origin of Species, John Murray, London, 1859.

M. R. Emes, A. Smith and D. D. Cowper, Confronting an identity crisis: how to "brand" Systems Engineering, Systems Engineering 8(2) (2005), 164-186.

H. H. Goode and R. E. Machol, System Engineering. An introduction to the design of large-scale systems, McGraw Hill, New York, 1957.
S. J. H. Herrtage, Sir Ferumbras, Kegan Paul, Trench, Trubner and Co, London, 1879.
G. Hersey, The Monumental Impulse: Architecture's Biological Roots MIT Press, Cambridge, MA, 1999.
IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology, 1990.
IEEE 1220, Application and Management of the Systems Engineering Process, 1994.
INCOSE, What is Systems Engineering?, http://www.incose.org/practice/whatissystemeng.aspx, 2004 [cited February 2011].
ISO/IEC 12207, Software Life Cycle Processes, 1998.
ISO/IEC 15288, Systems Engineering - Systems Life Cycle Processes, 2002.
ISO/IEC 15504-1, Information technology - Process assessment, 2004.
ISO/IEC 42010, Systems and software engineering — Architecture description, Final Draft International Standard, 2011.
ISO/IEC 42010 CD0.8, Systems and software engineering — Architecture description, 2009.
P. King and P. Bryant, Working Towards a New Characterisation of Architecture, INCOSE International Symposium, Utrecht, Netherlands, June 2008
P. B. Kruchten, The 4+1 View Model of Architecture, IEEE Software 12(6) (November 1995), 42-50.
B. Lawson, How Designers Think, Architectural Press, Oxford, 1997.
M. W. Maier and E. Rechtin, The Art of Systems Architecting, CRC Press, Boca Raton FL, 2009.
J. N. Martin, Using the PICARD Theory of Systems to Facilitate Better Systems Thinking, INCOSE Insight 11(1) (January 2008), 37-41.
W. D. McArdle, F. L. Katch and V. L. Katch, Exercise Physiology: Nutrition, Energy, and Human Performance, Lippincott Williams & Wilkins, Baltimore, 2009.
J. Mede, The apostasy of the latter times, London, 1641.
MIL-STD 499, Department of Defense, Military Standard: Systems Engineering Management, 1969.
Ministry of Defence, MODAF- Ministry of Defence Architectural Framework v1.2, http://www.modaf.org.uk, 2008 [cited 30 April 2010].
OED, Oxford English Dictionary: The definitive record of the English language, http://oed.com:80/Entry/10408, 2010 [cited 07 February 2011].
OED, systems engineering, n., http://www.oed.com/view/Entry/196665?redirectedFrom=systems engineering - eid19392743, 2010 [cited 15-Nov 2011].
Parliament of the United Kingdom, Architects Act, The Stationery Office Limited, 1997.
M. Proust, In Search of Lost Time, Vol. 6: Time Regained, Vintage, London, 2000.
E. Rechtin, Systems Architecting: Creating and Building Complex Systems, Prentice Hall, Englewood Cliffs NJ, 1991.
E. M. Rogers, Diffusion of Innovations, Simon & Schuster, London, 2003.
J. Ruskin, Lectures on architecture and painting, Smith Elder, London, 1854.
J. Shute, The first and chief groundes of architecture, London, 1563.
The Open Group, TOGAF, http://www3.opengroup.org/subjectareas/enterprise/togaf, 1995 [cited 15 Nov 2011].
UKAWG, UK Architecture Working Group, http://www.ukawg.org [cited 9 February 2011].
Vitruvius, De Architectura, 15BC.
G. M. Weinberg, An Introduction to General Systems Thinking, Dorset House, New York, 2001.
M. Wilkinson, Z8 - System Architecture, Z-Guides, H. Woodcock (Editor), 2010.
M. Wilkinson, P. King and P. Bryant, INCOSE UKAWG, Belief Systems in Architecting, 2009.
M. Wilkinson, P. King, A. James, M. Emes and P. Bryant, Belief Systems in Systems Architecting: Method and Preliminary Applications, IEEE Systems of Systems Engineering Conference, Loughborough, 22-24 June 2010

B. Wilson, Soft Systems Methodology: Conceptual Model Building and its Contribution, John Wiley & Sons Ltd, Chichester, 2001.

J. Zachman, A framework for information systems archtiecture, IBM Systems Journal 26(3) (1987), 276-292.