Systems architecture

From Wikipedia, the free encyclopedia

A **system architecture** or **systems architecture** is the conceptual model that defines the structure, behavior, and more views of a system.[1] An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).[2][3]

Contents

Overview

There is no universally agreed upon definition of what aspects constitute a system architecture, and various organizations define it in different ways, including:

- The fundamental organization of a system, embodied in its components, their relationships to each other and to the environment, and the principles governing its design and evolution.[4]
- The composite of the design architectures for products and their life-cycle processes.[5]
- A representation of a system, including a mapping of functionality onto hardware and software components, a mapping of the software architecture onto the hardware architecture, and human interaction with these components.[6]
- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.[7]

- An architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.[8]
- A description of the design and contents of a computer system. If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software.[9]
- A formal description of a system, or a detailed plan of the system at component level to guide its implementation.[10]
- The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time.[11]

One can think of system architecture as a set of representations of an existing (or future) system. It conveys the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that an architecture description may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people.

A system architecture primarily concentrates on the internal interfaces among the system's components or subsystems, and on the interface(s) between the system and its external environment, especially the user. (In the specific case of computer systems, this latter, special, interface is known as the computer human interface, *AKA* human computer interface, or CHI; formerly called the man-machine interface.)

One can contrast a system architecture with system architecture engineering (SAE) - the method and discipline for effectively implementing the architecture of a system:[12]

- SAE is a *method* because a sequence of steps is prescribed to produce or to change the architecture of a system within a set of constraints.
- SAE is a *discipline* because a body of knowledge is used to inform practitioners as to the most effective way to architect the system within a set of constraints.

History

It is important to keep in mind that the modern systems architecture did not appear out of nowhere. Systems architecture depends heavily on practices and techniques which were developed over thousands of years in many other fields, perhaps the most important being civil architecture.

Prior to the advent of digital computers, the electronics and other engineering disciplines used the term "system" as it is still commonly used today. However, with the arrival of digital computers and the development of software engineering as a separate discipline, it was often necessary to distinguish among engineered hardware artifacts, software artifacts, and the combined artifacts. A programmable hardware artifact, or computing machine, that lacks its software program is impotent; even as a software artifact, or

program, is equally impotent unless it can be used to alter the sequential states of a suitable (hardware) machine. However, a hardware machine and its software program can be designed to perform an almost illimitable number of abstract and physical tasks. Within the computer and software engineering disciplines (and, often, other engineering disciplines, such as communications), then, the term system came to be defined as containing all of the elements necessary (which generally includes both hardware and software) to perform a useful function.

Consequently, within these engineering disciplines, a system generally refers to a programmable hardware machine and its included program. And a systems engineer is defined as one concerned with the complete device, both hardware and software and, more particularly, all of the interfaces of the device, including that between hardware and software, and especially between the complete device and its user (the CHI). The hardware engineer deals (more or less) exclusively with the hardware device; the software engineer deals (more or less) exclusively with the software program; and the systems engineer is responsible for seeing that the software program is capable of properly running within the hardware device, and that the system composed of the two entities is capable of properly interacting with its external environment, especially the user, and performing its intended function.

By analogy, then, a systems architecture makes use of elements of both software and hardware and is used to enable design of such a composite system. A good architecture may be viewed as a 'partitioning scheme,' or algorithm, which partitions all of the system's present and foreseeable requirements into a workable set of cleanly bounded subsystems with nothing left over. That is, it is a partitioning scheme which is exclusive, inclusive, and exhaustive. A major purpose of the partitioning is to arrange the elements in the sub systems so that there is a minimum of interdependencies needed among them. In both software and hardware, a good sub system tends to be seen to be a meaningful "object". Moreover, a good architecture provides for an easy mapping to the user's requirements and the validation tests of the user's requirements. Ideally, a mapping also exists from every least element to every requirement and test.

Types

Several types of systems architectures (underlain by the same fundamental principles[13]) have been identified as follows:[14]

- Hardware architecture
- Software architecture
- Enterprise architecture
- Collaborative systems architectures(such as the Internet, intelligent transportation systems, and joint air defense systems)
- Manufacturing systems architectures
- Strategic systems architecture[15]

See also

- Architectural pattern (computer science)
- Department of Defense Architecture Framework

- Enterprise architecture framework
- Enterprise information security architecture
- Method Framework for Engineering System Architectures
- Process architecture
- Requirements analysis
- Software engineering
- Systems architect
- Systems analysis
- Systems design
- Systems engineering

References

1. ^ Hannu Jaakkola and Bernhard Thalheim. (2011) "Architecture-driven modelling methodologies." In: *Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII*. Anneli Heimbürger et al. (eds). IOS Press. p. 98
2. ^ Paul C. Clements (1996) "A survey of architecture description languages (http://www.sei.cmu.edu/library/assets/Survey_of_ADLs.pdf)." Proceedings of the 8th international workshop on software specification and design. IEEE Computer Society, 1996.
3. ^ Nenad Medvidovic and Richard N. Taylor (2000). "A classification and comparison framework for software architecture description languages. (ftp://ftp.ics.uci.edu/pub/arch/papers/TR-UCI-ICS-97-02.pdf)" Software Engineering, IEEE Transactions on 26.1 (2000): 70-93.
4. ^ From ANSI/IEEE 1471-2000.
5. ^ From IEEE 1220-1998 as found at their glossary (http://www.acq.osd.mil/osjtf/termsdef.html).
6. ^ From the Carnegie Mellon University (http://www.cmu.edu)'s Software Engineering Institute (http://www.sei.cmu.edu).
7. ^ From The Human Engineering Home Page's Glossary (http://www.manningaffordability.com/s&tweb/HEResource/Other/Definitions.htm).
8. ^ From OPEN Process Framework (OPF) Repository (http://www.opfro.org/Components/WorkProducts/ArchitectureSet/Architectures/Architectures.html).
9. ^ From The National Center for Education Statistics glossary (http://nces.ed.gov/pubs98/tech/glossary.asp).
10. ^ TOGAF
11. ^ TOGAF
12. ^ The Method Framework for Engineering System Architectures, Donald Firesmith et al., 2008
13. ^ *The fundamental principles of Systems Architecture* (http://www.lix.polytechnique.fr/~golden/systems_architecture.html#principles), by Boris Golden
14. ^ The Art of Systems Architecture, Mark Maier and Eberhardt Rechtin, 2nd ed 2002
15. ^ *Choosing A Strategic Systems Architecture* (http://www.alacrastore.com/research/forrester-Choosing_A_Strategic_Systems_Architecture-39702), by Brad Day

Sources

- Muller, G. "Systems architecting: A business perspective," CRC Press, (2012).

External links

- Principles of system architecture (http://sysarch.pbwiki.com)
- What is Systems Architecture ? (http://www.lix.polytechnique.fr/~golden/systems_architecture.html)
- INCOSE Systems Architecture Working Group (http://www.incose.org/practice/techactivities/wg/sysarch/)
- Journal of Systems Architecture (http://www.elsevier.com/wps/find/journaldescription.cws_home/505616/description#description)

Categories: Systems architecture | Systems engineering | Software architecture

---