Summary of IEEE 1471

By Jan Øyvind Aagedal, SINTEF Telecom and Informatics

Even if it is used by many, the term "architecture" has no well established definition. Nevertheless, in the field of software engineering there is no shortage of more or less overlapping definitions (see for instance www.sei.cmu.edu/architecture/definitions.html). Here we present one consistent set of definitions targeting architectural descriptions for software-intensive systems, namely the IEEE Std 1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. This recommended practice seeks to become a common frame of reference within which to codify common elements between different architectural description initiatives, and has become influential and used as a baseline for architectural description frameworks, for instance within OMG. It reflects generally accepted trends in practices for architectural description and provides a frame of reference within which future developments in software architectural technology can be deployed.

According to the recommended practice, software-intensive systems are those complex systems "where software contributes essential influences to the design, construction, deployment and evolution of the system as a whole". The purpose of IEEE Std 1471-2000 is to facilitate the expression and communication of architectures and thereby lay a foundation for quality and cost gains through standardisation of elements and practices for architectural description of software-intensive systems. This is in contrast to past attempts at architecture description where only hardware-related architectural aspects were addressed. With increasingly complex software, architectural integrity of the software should also be addressed and the recommended practice facilitates this.

Figure 1 shows the conceptual model of architectural description as defined in IEEE Std 1471-2000.

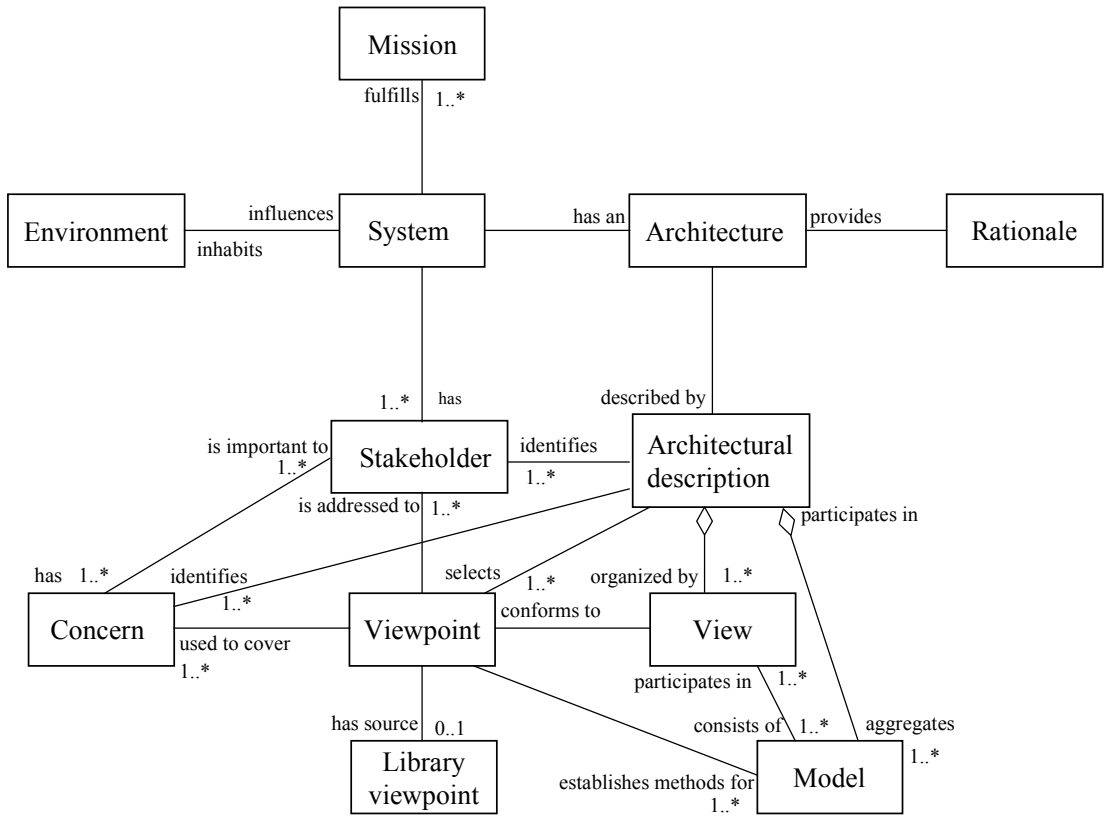

Figure 1: Conceptual model of architectural description from IEEE Std 1471-2000

Starting with *system*, it is defined to be "a collection of components organized to accomplish a specific function or set of functions." For the purposes of the recommended practice, "the term system encompasses individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, whole enterprises, and other aggregations of interest." From this it follows that anything can be a system as long as it fulfils some purpose (i.e., accomplishes function(s)) and one chooses to view it as a whole.

A system inhabits an *environment*, while the environment of a system can influence that system. The environment, sometimes referred to as the context, "determines the settings and circumstances of developmental, operational, political, and other influences upon that system. The environment can include other systems that interact with the system of interest, either directly via interfaces or indirectly in other ways. The environment determines the boundaries that define the scope of the system of interest relative to other systems". Essentially, one draws a line between the system of interest and anything outside that system that influences it in some way. This line is the interface between the system and its environment.

A system has one or more *stakeholders*. A stakeholder has one or more *concerns* relative to the system. Concerns are "those interests which pertains to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders." Typical concerns a stakeholder can have relative to a system are functionality, performance, security, reliability, safety, etc.

A system exists to fulfil one or more *missions* in its environment. The existence of a system has a purpose; it should meet one or more objectives of one or more stakeholders. Often some of these objectives coincide with enterprise objectives so that using the system is an efficient use of resources in the enterprise.

So far, the terminology presented has only been related to systems and their environments. However, most of IEEE Std 1471-2000 is concerned with architectural descriptions, and in the following terminology related to this are presented.

A system has an *architecture* and this can be described in an *architectural description*. Note the distinction between the architecture of a system, which is conceptual, from the description of this architecture, which is concrete. Architectural description is defined as "a collection of products to document an architecture". The architectural description can be divided into one or several *views*. Each view covers one or more stakeholder concerns. View is defined as "a representation of a whole system from the perspective of a related set of concerns". A view is created according to rules and conventions defined in a *viewpoint*. Viewpoint is defined as "a specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis". The distinction between view and viewpoint is analogous to that between a searchlight and what one sees using the searchlight as shown in Figure 2. A view expresses those concerns of a system architecture that are defined in its viewpoint definition. The viewpoint defines the languages and methods to use when describing such views.

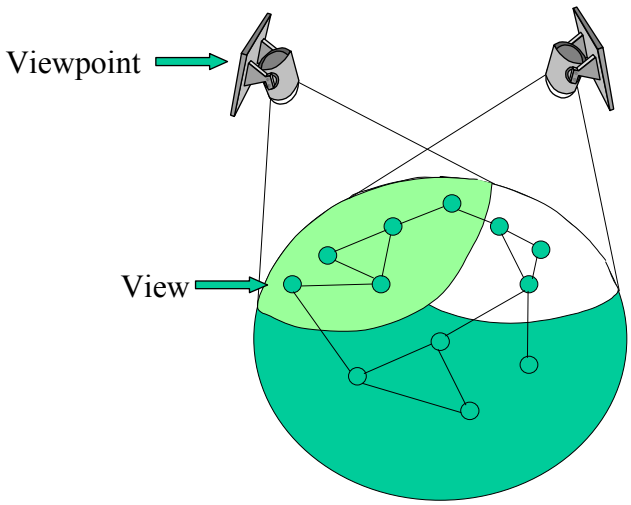

Figure 2: Views and viewpoints

In addition to information described in views, an architectural description may contain other information such as system overview and system rationale. This information is not described according to a viewpoint definition, but may follow other organisational documentation practices.

An architectural description selects one or more viewpoints for use. This choice depends on the concerns of the stakeholders that need to be addressed by the architectural description. The ISO Reference Model of Open Distributed Processing (RM-ODP) has selected five such viewpoints, but IEEE Std 1471-2000 does not prescribe any particular viewpoints. A viewpoint may be defined with the architectural description, but it may also be defined elsewhere and only used in the architectural description. Such externally defined viewpoints are termed *library viewpoints*. The five viewpoints of RM-ODP are examples of such library viewpoints.

A view may consist of one or more *models* and a model may participate in one or more views. Each such model is defined according to the methods established in the corresponding viewpoint definition. The architectural description aggregates the models, organised into views.