On System, Conceptual part and Part

Erik Herzog

References

1. Erik Herzog, System physical architecture modelling alternatives within AP-233, 26-4-2005.
2. David Oliver, e-mail titled “AP-233 approach to structure”, 2005-05-06.
3. Tom Thurman, e-mail “Re: AP233 approach to structure”, 2005-05-09.

Background

It appears that my previous note [1] on the use of the part entity to represent system physical or conceptual architecture has raised more questions than it answered. David Oliver made an interpretation in [2] that the “conceptual_part” concept used in [1] was a notation for capturing alternate realisation alternatives. A separate mail has been sent to David Oliver in order to clarify the concepts in [1].

Based on the content of [1] Tom Thurman has asked me to clarify why I use separate conceptual_part and system entities. In [3] Tom states that he does not believe there is a need for both concepts. This document captures the rationale for including both entities (the “conceptual_part” was labelled “physical_element”) in the AP-233 WD4 and WD5 developed in 2000 and 2001 respectively[1]. It shall be noted that no proofs are provided for the correctness of the approach. The reader need to formulate his/her own views.

It is also important to note that AP-233 WD5 precedes PLCS, and may not be compatible with the basic assumptions therein. For me PLCS compatibility is a priority so I am definitely open to alternate implementations. It is however important that the rationale is understood prior to the removal of any capabilities.

Assumptions

The following assumptions guided the development of the said versions of AP-233.

1. There exists no single accepted approach to model based systems engineering. Tool support is patchy and individual organisations may employ tools supporting different parts of the engineering process, e.g., only use a requirements management tool, a functional architecture tool, a physical architecture tool, or any combination of the mentioned tools.
    a. In individual organisation there may exist a need to represent and exchange a functional model with no allocation data and no reference to any physical implementation of a system. I.e., the functional model of the system may be considered to be the complete system specification.

    So, our individual tool environments may not be able to manage the whole data set defined for AP-233, and may even not be aware that there exist data elements of interest outside those handled by the tool environment itself. Likewise, individual organisation may consider a data set not including a particular kind of specification view present to be complete.

[1] The use of the term conceptual_part in [1] may have been unfortunate. The intention was to capture the fact that components in high level specifications are not the identical components from a PDM perspective as the ones considered by design engineers, i.e., the same items to be realised (or already realised) are considered at different levels of abstraction.

2. In the engineering process we will see a number of baselines, each in turn will more concrete and closer to the products that will be realised. Each of these baselines will evolve over the life of a system.
3. For any system under consideration we need to consider a number of end user products and a number of support systems (as defined in EIA-632), Figure 1. These systems are interrelated, but not necessary interconnected.

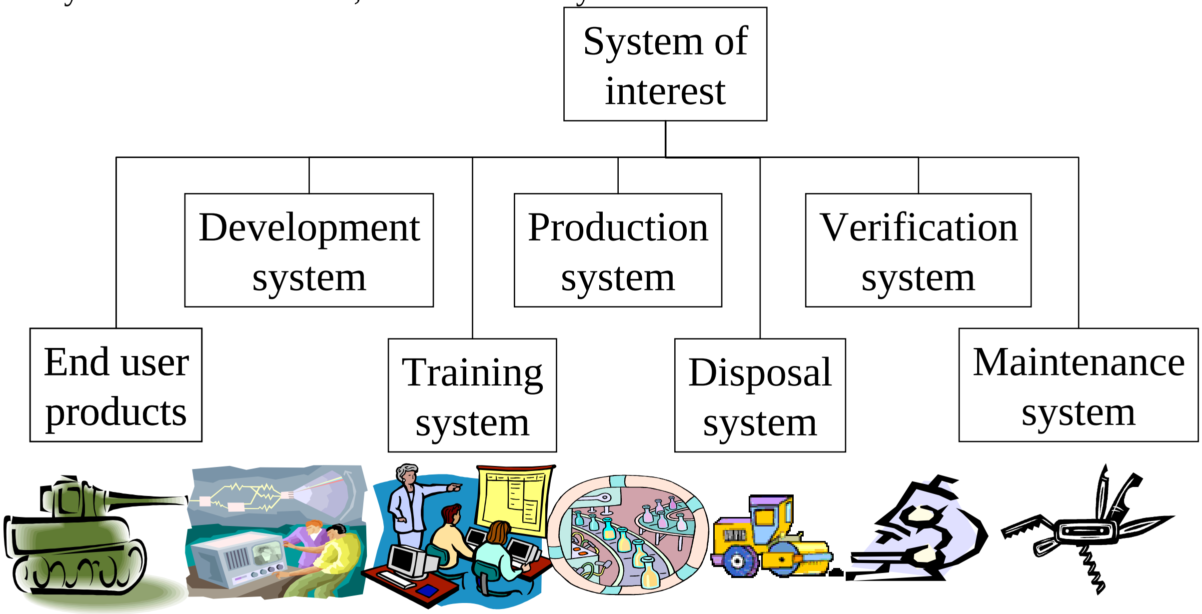

Figure 1, End user systems and the required support systems

AP-233 WD5 definitions

The scope of AP-233 WD5 is presented in UML notation in Figure 2. The figure presented the information model at a high level of abstraction – each UML class but represent a large number of Express entities. The Requirements, Functional, Physical and system architecture boxes all contain elements that are subtypes of the STEP product, product_version and product_view_definition entities[2].
The significant elements for the purpose of this document are the system architecture and the physical architecture boxes.

[2] These are the namnes used in the PDM modules. The Part 41 entities are named differently, if my memory serves me correctly (but not correct enough to state the actual part 41 names).

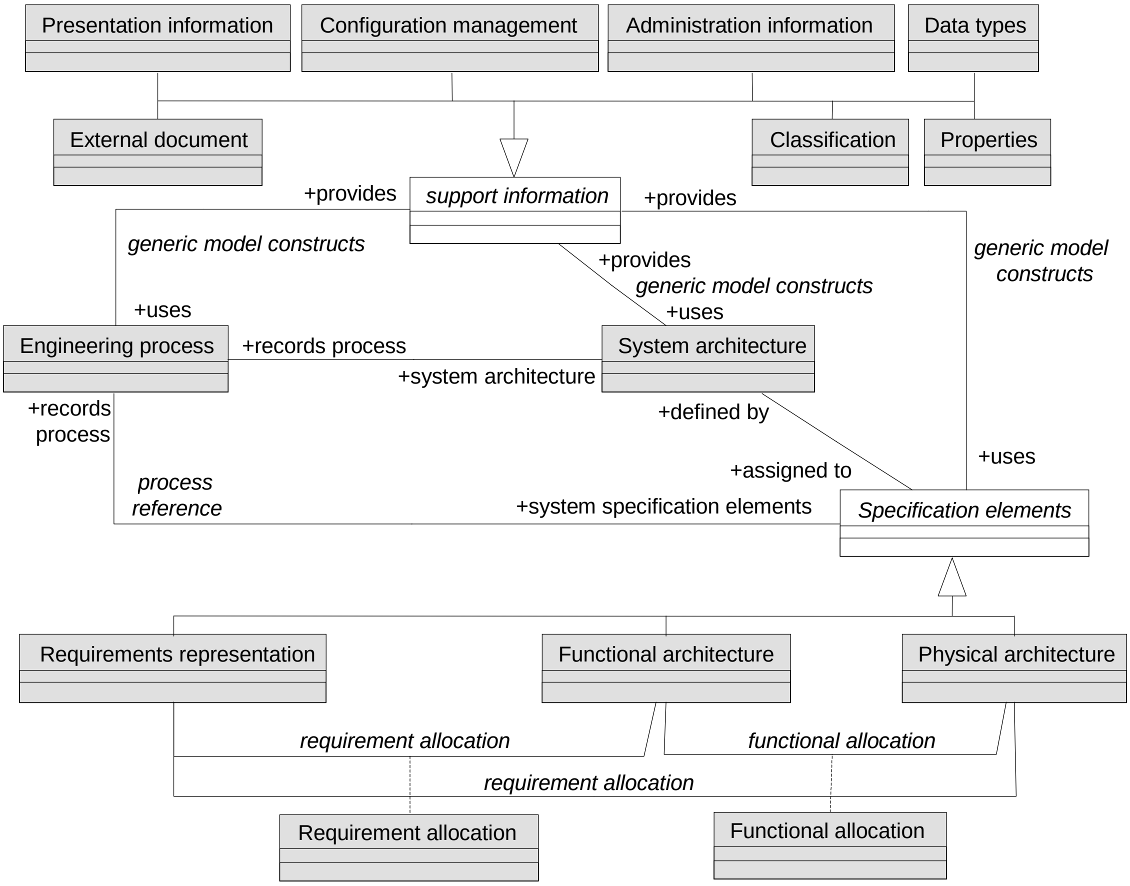

Figure 2 AP-233 WD 5 scope

The following high level definitions were made in AP-233 WD5:

- The system architecture elements allow for the capture of the complete system under specification, including the required support systems. Another use is to related end user products and support systems. These are different systems, but they are related. The system architecture may also represent the system breakdown structure
  The system architecture does not mandate structure any particular realisation structure, but is merely a placeholder for collecting relevant technical and administrative information.
- The physical architecture represent logical or physical realisation elements at a high level of abstraction. The physical architecture model support the definition of physical interfaces and definition of system interaction, i.e., a network of elements (systems does not).

Example

The example in this section is created to illustrate the difference between the physical architecture and system architecture. It is not complete, and not 100% accurate to the AP 233 WD5 EXPRESS models. Also note that the example include a set of ports representing interfaces to physical architecture elements. The ports are included to illustrate a property of a physical element not present for a system element.

Note that requirements, physical_components and system entities are subtypes of the STEP product entity. I.e., for each requirement, there is one requirement object, at least one requirement_version object and for each requirement_version object a requirement_view_definition object. These STEP structures are not highlighted in the example.

Figure 3 presents a simple instantiated example of a system specification. A number of requirements and 2 components are identified for a system. One requirement is allocated to each one of the components. The components are interconnected via ports to define the physical interfaces.
It is important to notice that AP233 WD5 allowed for two types of association of a function or component to a system (not shown in the figure).

- The association of a function/component as a system function, indicating that the function/component is internal to the system
- The association of a function/component as an external function, indicating that the function/component is of interest as it interacts with the system.

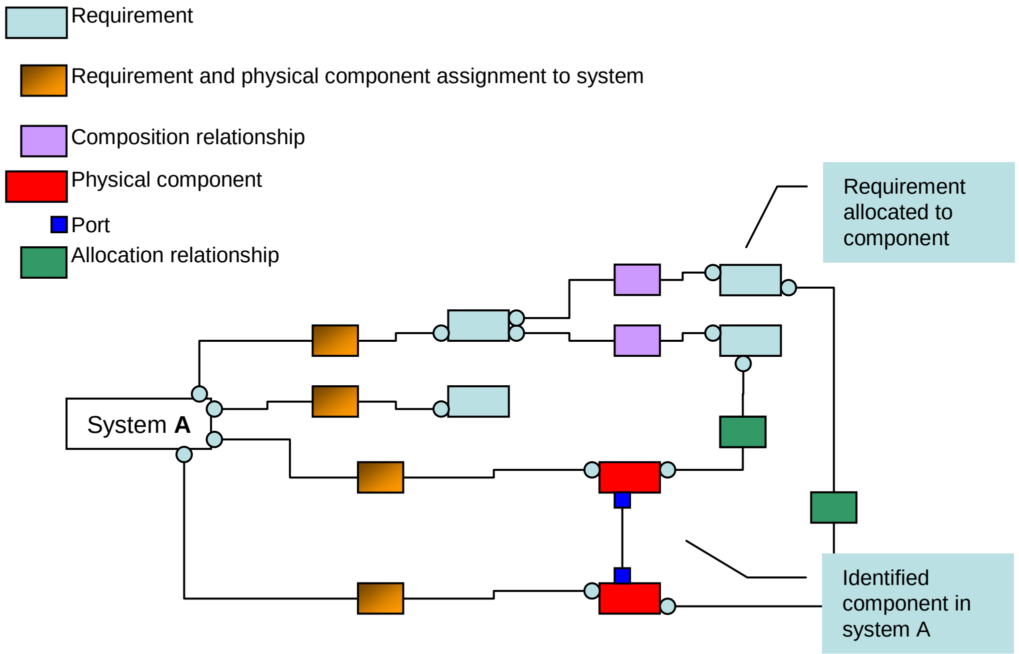

Figure 3: Simple system example

In Figure 3 the object "System A" is an element of the system architecture and the red and blue boxes represent physical components and ports from the physical architecture part of the model.

Figure 4 adds a subsystem specification to the elements in Figure 3. In the example the requirements for the subsystem (system B) are all derived from the requirement allocated to one of the components in (system A).

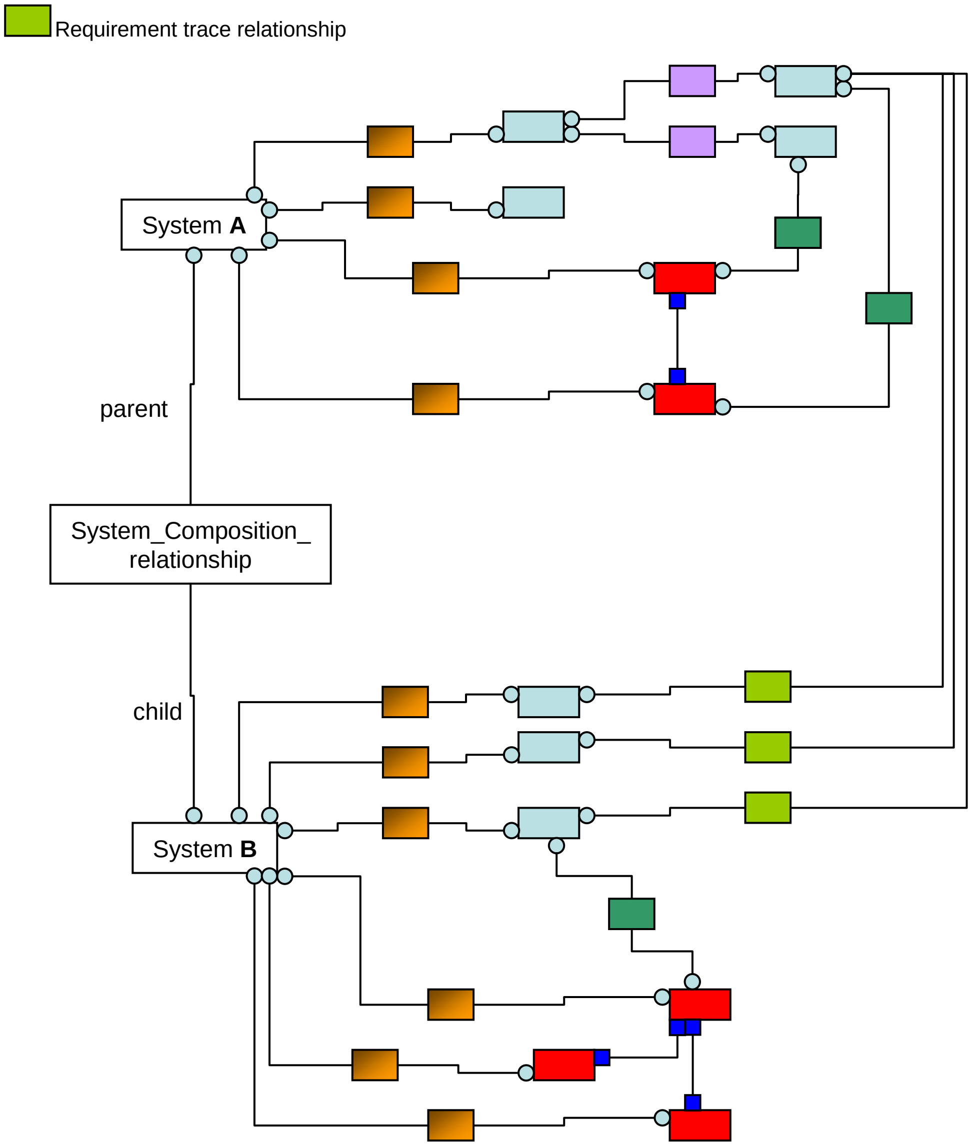

Figure 4 Subsystem layer added

Discussion

As stated previously, there is no algorithm for determining the correctness of a model.
In my view the separation of system and physical architecture allow

- Representation of requirements (individual statements and/or functions) without considering realisation structure (structure is a result of analysis or requirements). Also, some systems are virtual, they do not have any realisation elements, even at systems engineering level.
- The system architecture is convenient to capture related systems, in terms of
  - a system composed of other systems, and
  - support systems for an end user product, and
  - systems captured at abstraction levels, e.g., the system as seen by a stakeholder (stakeholder reuquirements) and the corresponding supplier system requirements.
- The system architecture view is tool independent. It is a view that does not imply any particular notation.

- System architecture properties may include cost and schedules that are unique for each system, whereas a physical component may be part of multiple systems.
- The physical architecture is a modelling tool view and does capture component interfaces and interaction, the system architecture does not.

Alternatively, it is equally correct to arrive to the position “So, what the nuances are minor and of no importance.