System Engineering, Plant Engineering and Functional Models

Dr. Martin Witte

martin.witte@siemens.com

Introduction

System Engineering, as defined in the IEEE 1220 [4], promotes a model-driven process for designing complex systems. This approach comprises three main activities: the requirement analysis, the definition of the functional architecture and the allocation to the design architecture; all of these together lead to the system architecture. The result is a system specification that defines subsystems, components and interfaces. Interfaces and components are finally linked to discipline specific specifications.

One of the central ideas in system engineering is the duality between the functional architecture and the design architecture [5]. These models serve as interdisciplinary description of "means" in the system design process; the functional model not only bridges the gap between operational requirements and solutions, but is also important for intent analysis and interface specifications.

This paper considers system engineering methods - and especially functional models - in two ways: Firstly, how a functional model must be defined to describe the intent of a system as postulated in [6], and secondly, how this can be used in plant engineering.

Semantic functional models

The idea behind functional models is to provide a structure which describes the functionality of a system as independently as possible from the specific component structure. The model consists of function objects and relationships between them. Typically, relationships are flows between different types of functions, such as energy, information or current. Unfortunately, the type of description of a function itself depends on the flows it receives or transmits: if a function transforms energy, differential equations are used; if it transforms information, state machines might be appropriate. It can be seen that the flow type makes a functional description discipline specific (in the sense of engineering disciplines); functions with multiple flow inputs and outputs are those that are independent of any discipline and consequently hard to describe. However, functions that are dependent on a discipline regarding their description, or hard to understand, cannot be used to define a system in an interdisciplinary way. We need a common language in order to be able to communicate.

The function of a device that transforms heat from a heat source into an analog signal proportional to the temperature would be called "measure temperature" by an electrical engineer – a software engineer may have no idea what the device is, but certainly understands the phrase "measure temperature". Formalizing such descriptive names (see [1], [2]):

> "A semantic functional description consists of an action and an object which is acted on."

A thermometer has the action "measure temperature"; the object it acts upon is the heat source.

We observe that we can classify both action and object in a hierarchic structure. The action "measure temperature" is a special case of "measure", which is a special case of "acting without changing the object". A special case of a heat source would be "liquid", which can be more specifically defined to be either "oil" or "water".

We can assign a value to each action and object pair, which indicates if it has a meaning or is meaningless. By being able to make conclusions on meaningful statements in a classification hierarchy, given a meaningful action and object pair, we define the classification hierarchies as being well defined relative to one another if the following applies:

1. The pair, built by generalizing the action with the given object, remains meaningful
2. The pair, built by specializing the object that is given an action, remains meaningful

It should be noted that similar rules can be formulated for meaningless action and object pairs. Also note that a partial ordering in the course of the generalization of actions and specialization of objects can be defined where meaningfulness is preserved.

Just as in a natural language, we can make the description more precise by adding adverbs and adjectives describing qualitative and quantitative parameters of functions. We can now formulate smarter searches for functionality in complex systems than by just making a simple string comparison.

The next step associates means to functions. As example, the "means" for "measure temperature" could be a thermometer. The triple of means, actions and objects defines a complete sentence. If the action and object pair are meaningful, we can assign boolean values to the sentence with the third value being undecided. Because means, just like objects, can also be hierarchically classified, it is possible to make assertions regarding generalizations and specializations of sentences with known truth value; this will not be discussed in any more detail here. This can be used when searching for solutions, where possible means are searched for having a specific function. As example, we can look for all means for measuring water temperature; every thermometer type for liquid is a possible solution; a thermometer for oil could be a solution, if it is undecided, because its functionality is "close" to the functionality of a thermometer for water: oil and water have a close common classification parent "liquid".

Plant engineering

Current plant engineering methods and their resulting entities do not match the standard system engineering process; this is especially the case as there is no explicit functional model used. Additionally, the two main production site types – manufacturing industry and process industry – are treated with different methodologies, even if the design results, namely physical construction, electrical diagrams, or automation and control software, are similar.

We can formulate a functional model with semantic description as a model between the production process descriptions and the resource allocation, i.e. the physical layout. The model describes the basic functionality of each component of the system by an action and object pair, but also the higher-level functionality that is obtained through the cooperation of such functions. Such functionality can be emergent or even unwanted, or it might be associated with software functions combining hardware functionality with higher level functions.

As example, consider a sequence of connected pipes, pumps and valves defining a pipe run in a P&ID diagram. Each element has the function "transport fluid". The function "pump fluid" of a pump is a specialization of "transport fluid". By using such definitions, each element of a P&ID has a standardized, basic functional description. The function "transport fluid from container A to container B" can be decomposed into such basic functions of connected elements. But not all possible paths are realized in a P&ID, because a path needs control software defining under what conditions the path is allowed and which pumps and valves have to be activated. By analyzing the semantics of a P&ID, we have detected missing automation functionality, and created a link between automation programming and equipment description. This is crucial for being able to efficiently qualify process plants with respect to legal standards.

Now let us assume that each piece of equipment in the pipe run must be electrically grounded, which is a functionality outside the piping discipline. Normally, this is automatically done because the pipes are made of metal. Formally, each pipe (and pump) has the second functionality "conduct current", which is associated with the grounding function. If a metal pipe is exchanged with a plastic pipe, we silently remove functionality from the system; the pipe run no longer has the function "conduct current", which means that the grounding functionality might be violated. A system knowing semantic functional models can automatically detect such problems, while models for specific disciplines will not be able to discover this easily.

In manufacturing, "functional engineering" denotes the association of a production step to a piece of equipment [3]. A "functional group" is another name for a working cell that is built for a specific manufacturing step. At the machine level, functions often denote machine movement capabilities or basic sensor functions. The close connection of functionality to hardware is the reason why functional engineering fails to be useful for software-dominated functionality. On the other hand, there are multiple standards available that define "action" hierarchies, which can be used for interdisciplinary semantic descriptions. Using an example of a conveyor system, we can show how a semantic functional model using such standardized hierarchies helps us to understand the system in a holistic way, when compared to an MCAD description.

Summary

This paper presents that semantic functional descriptions based on classifications are a useful method to capture intent in the multi-disciplinary environment of plant engineering. Using some examples, we attempt to show that such models can be used for analysis and to search for material handling functions in existing plants. Not only this, that they are useful for describing functional issues not directly linked to physical components, especially for automation and production control functionality.

References

[1] DIN 69910: Value analysis.

[2] VDI 2803: Function analysis – Fundamentals and method.

[3] DIN 8580: Manufacturing processes: Terms and definitions.

[4] IEEE 1220: Application and management of the systems engineering process.

[5] Department of Defense. Systems Engineering Fundamentals, Fort Belvoir, Virginia 2001.

[6] Nancy G. Leveson. Intent Specifications: An Approach to Building Human-Centered Specifications. IEEE Trans. Softw. Eng., 26(1):15-35, 2000.