Systems Engineering and the Construction Industry

by
Erik W. Aslaksen
Sinclair Knight Merz
St. Leonards, Australia

Introduction

The topic of my position paper is the application of systems engineering in the construction induxtry and, in particular, the application of the requirements development process. As this is not a formal paper, but a presentation meant to stimulate the following discussion, some of the observations are presented slightly tongue-in-cheek, but on the whole they are a reflection of my more than fifteen years' association with the construction industry.

The presentation consists of five short segments:

1. The industry context
2. Where does the engineering complexity and focus lie?
3. Where do the requirements come from?
4. The legal aspect
5. Implications for INCOSE

The Industry Context

Due to its genesis in the defence and areospace industry, it has become customary within INCOSE to consider the industrial landscape to be partitioned into defence and aerospace on the one hand and commercial industry on the other, although there is obviously not a sharp division between them. With this vew of things, the commercial industry can be further subdivided into some major industries (again, without any absolute and sharp divisions):

- the manufacturing industry;
- the service industry;
- the process industry; and
- the construction industry.

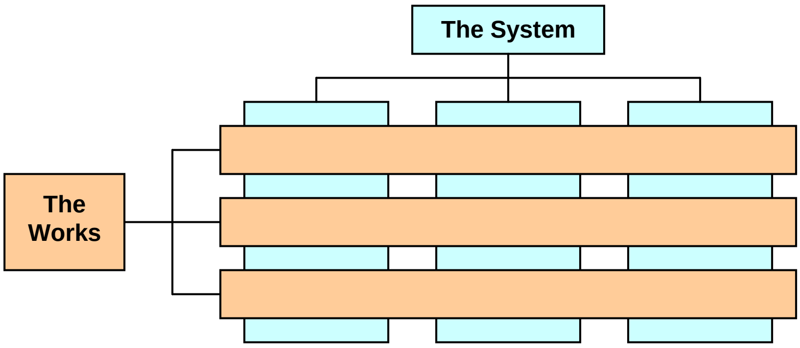
The degree to which (formal) systems engineering is employed in these industries is something like this:

The manufacturing industry already has a significant presence in INCOSE (e.g. through the car and software manufacturers), and the service industry is represented mostly thorugh the ICT side of the business. It is interesting to note that the process industry, in particular the chemical insustry, has a highly developed version of systems engineering, but does not have a significant representation in INCOSE.

Where does the Engineering Complexity and Focus lie in the Construction Industry?

Systems engineering is a methodology for handling complex engineering projects, and traditionally that complexity has been found in the design and development process. That is, in the process of converting the user requirements into specifications for the manufacturing, integration, and testing of the products, and that focus is reflected in the systems engineering literature. The first level of breakdown of such systems is into *functional* elements.

In the construction industry, this is quite different. The whole development process, determining the *what* is to be constructed, is only about 10 % of the engineering effort, and seen by most construction companies as a necessary evil; their whole focus is on the *how* to carry out the construction in the most cost-effective manner. That is where the engineering complexity lies, and the first level of breakdown to handle this complexity is into elements defined by *disciplines* - civil, structural, mechanical, electrical, etc. In the construction industry the system is called the Works, and a typical system element would be the earthworks. The difference can be illustrated as follows:



This different focus in the construction industry has a major impact on all aspects of the engineering process. For example, the concept on integration takes on a different meaning to that defined by the V-model; it is not the integration of modules into equipment, equipment into subsystems, and subsystems into the system, but the integration of the various parts of the construction process, as defined by the disciplines. And the requirements development is centered on the disciplines, on the *how*, not on functional elements or the *what*.

Where Do the Requirements Come From?

In traditional systems engineering, most (but by no means all) of the requirements come from the users of the systems, e.g. defence personnel in the case of military systems, and the focus is on the functional requirements. That is, on what the system must do for the user in order to give the user the required capability. In the construction industry, the functional requirements are relatively trivial. For example, the functional requirements on a highway are straight forward; a major part of the requirements are concerned with how the highway is constructed. The issue is never that one will not be able to drive on the highway when it is built, it is that there will be settlement in five years’ time, or that the concrete will develop cracks, or that embankments will become unstable; all issues that arise from how the highway was constructed.

Another significant portion of the requirements arise from community concerns. Just to get a feel for the relativity of things, consider that the whole engineering development process, from user requirements to for-construction documentation, of a high voltage transmission line takes something on the order of six months; the various community consultation and approval processes take five years. Correspondingly, environmental requirements, from construction noise to the preservation of endangered species, requirements regarding access over private land and reinstatement of property, requirements regarding relocation of services, and so on, can run to hundreds of pages.

The Legal Aspect

Driven by the two aspects of construction industry requirements discussed above (i.e. process orientation and community concerns), but also due to a lack of attention by engineers and a skillful exploitation of this by the legal profession, both the threat and reality of litigation play a major roles in the construction industry. Therefore, a good requirement is, above all, a requirement that is enforcable through the courts, and the influence of the legal profession in formulating requirements has reached a point where it is interfering with the requirements development process.

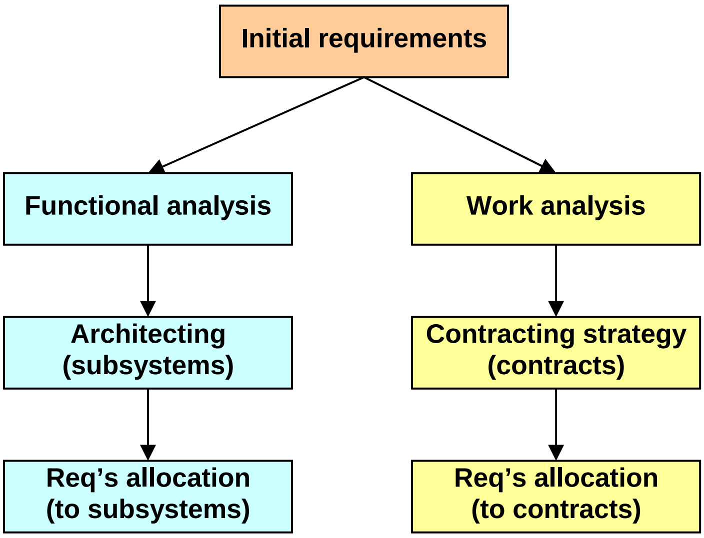
A measure of the state of affairs is the increasing popularity of Design and Construct contracts, rather than separate design and construct contracts. The rationale behind this is something like "let's get the project to the contractor as soon as possible, so that he can get started on developing the most cost-effective construction method; this will result in the lowest price. Any inadequacies in the specifications can be covered by smart legal clauses, so that if we don't like the result, we'll sue". That this has come to pass is in no small measure due to the design (consulting) engineers taking an ivory-tower approach and isolating the design process from the concerns of the construction contractor, and thereby reducing their relevance and influence in times of increasing cost consciousness.

Another illustration of the state of affairs is that after the Principal has spent considerable effort on developing a preliminary design, fully reviewed and checked, this is then given to the D&C contractor with the caveat "This design is provided for information only and will not be included in the Contract". Gets the Principal off the hook, but what is the contractor to do; redo the whole preliminary design, or take a chance and accept it? The cost-effectiveness of the design process is sacrificed to legal expediency.

The requirements development and management process within systems engineering provides a degree of rigour that might go a long way to reverse the current drift towards a legalistic approach to contracting.

Implications for INCOSE

This small and very rough overview of requirements in the construction industry has hopefully made you aware that systems engineering, and thereby INCOSE, has an important role to play in the construction industry, not least with regard to the requirements development process. In a somewhat simplistic view, all that is required is to think of the construction process as the system. Let us consider a simplified version of the requirements development process for two cases - a new military aircraft and a railway. This is illustrated in the following figure.



In both cases the starting point is an initial set of requirements, but in the first case these are functional requirements developed during the capability analysis process, in the second case they are requirements on the physical object, developed through a conglomerate of processes, including fiscal, social, and political ones. The first case then progresses through the well-known steps of functional analysis (to determine what functions the system must have in order to meet the requirements), architecting (to determine the best grouping of functions into subsystems and interactions between these subsystems), and requirements allocation (determining the best allocation of requirements to the subsystems in order to meet the performance targets). This process may then be repeated for each of the subsystems, and so on.

In the second case, the requirements are analysed to determine what work has to be carried out in order to construct the railway for the lowest possible cost. In a second step, this work is then structured into contracts, and in a third step, requirements are allocated to these contracts so that the overall result will meet the initial requirements. This process may then be repeated for each contract, *but not by the same organisational entity*. This is a significant difference between the two cases. In the first case (the "classical" INCOSE version of systems engineering) it is implicitly assumed that the whole systems development process, i.e. most of the descending arm of the V and the process of flow-down of requirements, is carried out by the same organisation. In the second case, there are contractual boundaries at each step in the process, i.e. between Principal and the Engineer doing the preliminary design, between the Engineer and the Contractor, and between the Contractor and the Subcontracor, and each of these interfaces the letting party tries to push the responsibility and liability down to the contracting party. The flow-down of requirements is across contractual boundaries, which alters the nature of the process.

The implications for INCOSE of this little contemplation of the construction industry is that if we want to embrace these different industries, we need to structure our products into two levels - the general methodology level and the industry-specific application level, perhaps with a transition section in between, as shown below. This structuring ought to come quite naturally to us as systems engineers, or not?