
Elements of Systemics: Systems Architecture

1.1. Introduction

This chapter is a summary that my colleague, Professor Daniel Krob, President of the French Center for Excellence in Architectural, Management and Economy of Systems (*Centre d'excellence sur l'architecture, le management et l'économie des systems*, CESAMES), allowed me to reproduce from his lectures and various publications, with some minor adjustments for this present edition. This synthesis contains eight fundamental definitions and a theorem on existence. It is in perfect harmony with my previous book, *System Architecture and Complexity: Contribution of Systems of Systems to Systems Thinking*, from the same collection.

The systems that humans design and make are at the heart of the modern world: it is indeed enough to think of social and political systems, economic and industrial systems, monetary and financial systems, energy production and distribution systems, transport systems, communication systems and more generally, the many organizational and technological systems that surround us, to realize that “engineered” systems are simply everywhere¹!

More and more, all these systems are characterized by both a *great intrinsic complexity* and ever-increasing interdependencies. Mastering these systems has

For a color version of all the figures in this chapter, see www.iste.co.uk/printz/pedagogy.zip.

¹ Living systems in particular will not be considered in this chapter (see the Conclusion, however), which is restricted to engineered systems, in other words, artificial systems (as opposed to natural systems) designed by man, which already covers a vast spectrum, given that technical systems as well as organizational systems can fall into this conceptual category.

become a high stake in a world going through profound transformations. They must be able to resist the constant and often contradictory pressures of citizens and consumers, markets and regulation, competition and technological innovation, to cite but a few examples of the forces to which they are subjected².

But mastering the complexity of a system fundamentally means being able to master its *integration processes*, in other words, its construction through interconnectivity and/or the networking of smaller systems. This process surely remains largely misunderstood, for it generates the mysterious phenomenon that is emergence: an integrated system will indeed always have new “emergent” properties that cannot be easily obtained or expressed through its building blocks.

Theorizing the integration of a heterogeneous complex system resulting from the interaction of several homogeneous systems consequently requires us to reason transversally, by constructing a new type of model that captures its emergence by coherently integrating the parts of each homogeneous model constituting the general systemic model. Now, we must acknowledge that nothing prepares us for this new paradigm of systemic reasoning, which obliges us to make connections between very different types of expertise and transcend the traditionally siloed categories of knowledge³.

The emergence of a true science of systems – systemics – capable of rigorously considering the numerous problems posed by the conception and management of the evolution of modern complex systems, is thus an urgent necessity if we are to be able to provide satisfactory responses to the many profoundly systemic challenges that humanity will have to face at the start of the third millennium.

This introduction to the subject thus seeks to be a modest yet resolute contribution to this new scientific discipline in proposing an integrated vision of systems architecture, that is, of the area of systemics that deals with the structure of systems and their process of conception.

2 Not to mention the intrinsic limits of the physical and environmental resources of our planet that are no longer impossible to ignore (see Meadows, D.H., Meadows, D.L., Randers, J., Berhens III, W.W. (1971). *The Limits to Growth*. Universe Books, more well known under the name “The Club of Rome Report”).

3 The property of emergence cannot be understood, *stricto sensu*, in the context of Aristotle’s logic since the whole is greater than the sum of its parts.

1.2. Systemic definition of a system

1.2.1. From real systems to formal systems

When we speak of a “system” in systemics, it is fundamental to understand that we are speaking from a logical point of view of modeling, which immediately leads us to the distinction between the concepts of the “real system” and the “formal system.”

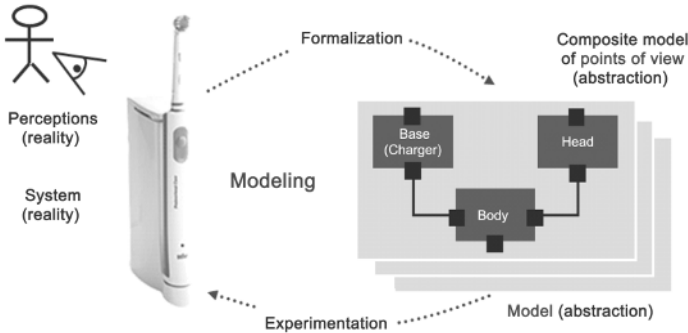


Figure 1.1. *The process of systemic modeling*

The process of *systemic modeling* can indeed be seen as a permanent cycle between a process of formalization that consists of constructing a *formal abstract model* from an object in the real world – using the tools of systemic analysis – and a process of experimentation that consists of verifying that the structure and behavior of the real object, given or predicted by the model, are indeed those observed in reality: the real object will therefore not be considered a (real) system except in that a modeling process analyzes it as a (formal) system within the context of systemic modeling⁴. In short, we naturally often conflate these two notions of a “system”, but it is important to never forget that “the map is not the territory” in order to maintain a clear idea of what we are conceptualizing⁵.

1.2.2. Definition of a system

With these preliminary considerations posed, we are now equipped to provide our first two key definitions, namely, that of a formal system and a real system.

⁴ We will use an electronic toothbrush as an elementary system to illustrate the systemic concepts in this chapter.

⁵ See Korzybski (1994). *Science and Sanity*. Institute of General Semantics, p. 58.

DEFINITION 1.1.– *Formal system: a formal system S is characterized on the one hand by the overall data on inputs X , outputs Y and internal states Q , and on the other hand, by the two following kinds of behavior that tie these systemic variables to each other over time:*

– *functional behavior, which produces an output $y(t) \in Y$ at any time t depending on the input $x(t) \in X$ and the ordinary internal state $q(t) \in Q$ of the system;*

– *internal behavior, which causes the internal state $q(t) \in Q$ of the system to evolve over the course of time t as a result of an input $x(t) \in X$ into the system.*

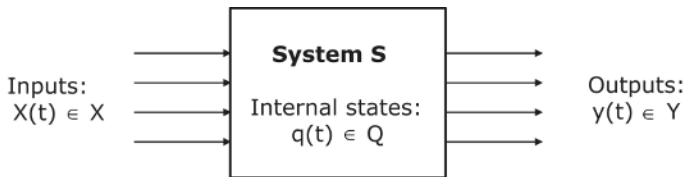


Figure 1.2. Classic understanding of a formal system

This definition is intentionally extremely flimsy, for it simply tells us, in essence, that a system is the data of input/output behavior and internal behavior⁶. This is not strange because the purpose is to capture the common points among *all* real systems, which are necessarily very limited given the diversity of objects which we must take into account. On the other hand, we now have a unified framework within which we will be able to reason in a consistent manner about every type of system, and thus theorize integration, which was the first of our objectives (see also, D. Krob, *Elements of Complex Systems Architecture*)⁷.

DEFINITION 1.2.– *Real system: an object from the real world is called a real system from the moment its structure and behavior can be described by a formal system (in the sense of Definition 1.1), which we then call a model of the real system.*

The near totality of human inventions – whether they be of an electro-mechanic, computing, or organizational nature – are thus analyzable as systems: the only change that occurs is in the nature of the laws – in this case physical, logical or

⁶ Modeled by a law of evolution of its internal variables, which we call “states” here.

⁷ In Appriou, A. (ed.) (2009). *Gestion de la complexité et de l’information dans les grands systèmes critiques*. CNRS Editions, 179–207. See also *Model-based Systems Architecting*, ISTE-Wiley, 2022.

sociological – which we use to define the functional and internal behaviors that make them into systems. The previous definitions also underscore that it is neither the nature, nor the size, nor the hierarchical position that make up a system, since almost every object can be seen as a system⁸. Considering an object as a system rests first and foremost on the choice in modeling: it is indeed the same as considering a particular object in the abstract to restrict the analysis to the systemic perspective, that is, within the framework given by Definition 1.1.

Let us point out that from here on, we will only provide definitions for formal systems, bearing in mind that they transfer directly onto real systems, given the modeling perspective that we have adopted here.

1.2.3. Integration of systems

We are now equipped to introduce the notion of system integration, which is the fundamental mechanism that allows, in practice, for the construction of a new system using other, smaller systems (materials, software, humans) by organizing them in such a way that it can execute – within a given environment – its assigned tasks.

DEFINITION 1.3.– *Integration: let S_1, \dots, S_N be a set of N (formal) systems. We can thus say that a (formal) system S is the result of the integration of these systems if there exists, on the one hand, a (formal) system C obtained through the composition of systems S_1, \dots, S_N , and on the other hand, dual abstraction and concretization mechanisms that let us express:*

- system S as an abstraction of system C ;
- system C as a concretization of system S ⁹.

This somewhat technical definition has the fundamental objective of positing integration as a mechanism that is distinct from the simple composition of models, in order to try to capture the emergence in our approach, structurally.

⁸ A notable common error is to assume that a “sub-system” of a system (see section 1.2.3 for more details) is not a system, which is naturally not the case (and which lets us apply the principles of systemic analysis to this level too).

⁹ We will not specify the notions of abstraction and concretization here, which should be taken in the sense of the theory of abstract interpretation (see, for example: Cousot, P., Cousot, R. (1996). Abstract interpretation. *Symposium on Models of Programming Languages and Computation, ACM Computing Surveys*, 28(2), 324–328, for more details on this subject).

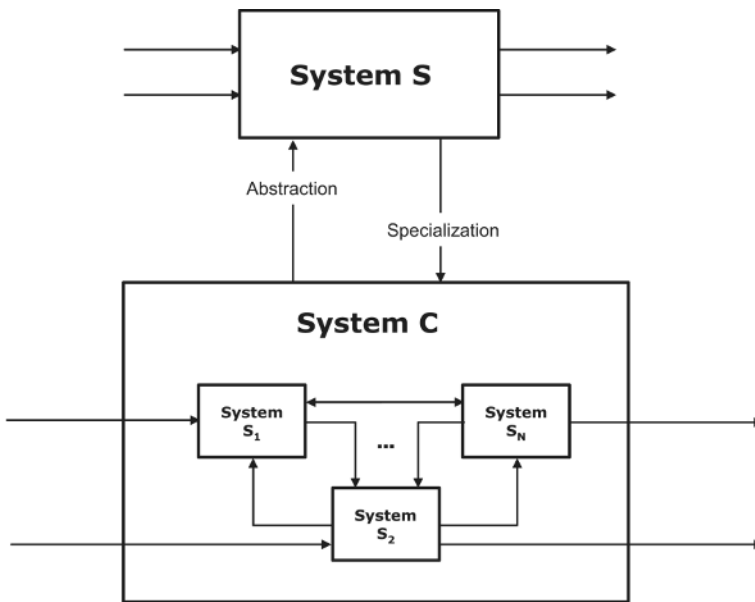


Figure 1.3. *Formal integration of (formal) systems*

Our definition indeed clearly presents the fact that the description of a high-level system obtained through the interconnection of other systems requires us to have:

- models of each of the systems that constitute the high-level system;
- a new, high-level model that is specific to this system.

In other words, simply knowing the models of the constitutive parts of a system is never enough to model this system! This modeling postulate can be seen as the direct translation of the universal phenomenon that is emergence. It can moreover be verified within all the elementary systems of everyday life.

Let us consider, for example, a wall made only of bricks (without mortar to hold them together), to simplify things. A simplified systemic model of a brick can thus be characterized by:

- functional behavior consisting of the absorption of rays of light;
- internal behavior specified by the constant states of “length, width, height.”

In making a model of bricks, it is thus difficult to obtain anything other than the functional behavior of absorbing rays of light, which is clearly not the usual behavior of a wall since we want to be able to reflect the holes that exist in walls

(windows), which themselves actually let the light through! We will also note that it is difficult – and surely in vain – to try to express the form of a wall (which is one of these typical internal states) in terms of the lengths, widths and heights of the bricks that make it up. All of this is therefore an argument for the fact that we must clearly create a dedicated, more abstract, systemic model in order to model the wall.

These observations perhaps appear to be merely common sense, if not naïve, but we unfortunately see that their consequences are more often than not misunderstood, which is the direct source of many of the quality problems observed in the great engineered systems of modern times. In fact, most people in the industry continue to think that mastering the components of an integrated system is sufficient for mastering the system as a whole. But the very nature of the mechanism of integration requires us to go beyond the satisfaction of managing components, when what we should be doing is appreciating it as an integrated system: it is just as necessary to have someone specifically managing the integrative model of the high-level system, that is, a systems architect, who is in fact usually missing from industrial structures, however strange that may appear¹⁰!

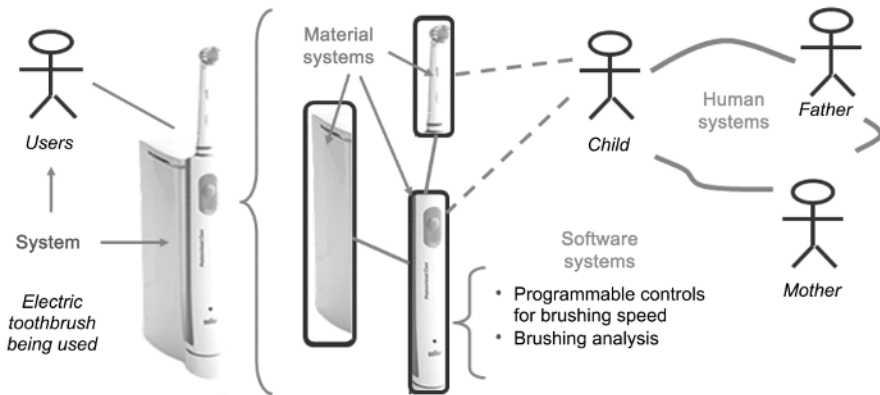


Figure 1.4. Concrete integration of (real) systems

Finally, we should point out that the integration mechanism naturally creates hierarchies in systems, while simultaneously giving the systemic analysis a recursive dimension. Engineered systems are indeed made through successive system integrations: any integrated system therefore creates an integration hierarchy, which is also a hierarchy in abstraction given the nature of integration, and is called the

¹⁰ The high-level models of integrated systems also often being consequently almost inexistent, in practice.

systemic hierarchy of the initial system, which allows us to speak of the levels, sub-systems, sub-sub-systems, etc. of an integrated system.

1.3. Organization of a systemic model

1.3.1. Architectural visions of a system

The systemic approach consists of analyzing real systems as formal systems, which are therefore models that need to be identified through a systemic lens, in which we only think in terms of the interacting systems. The recursive character of systemic analysis thus automatically leads us to introduce the notion of the system environment, which denotes a closed super-system (i.e. without inputs or outputs) containing the system, therefore acting as the natural starting point for the process of recursive system analysis (see the following section).

DEFINITION 1.4.– *Environment of a system: let S be a (formal) system. We can therefore say that a (formal) system $E(S)$ is an environment for S if it is a (formal) closed system¹¹ that results in the integration of S and another (formal) system $Ext(S)$, which we call the exterior of S in the environment $E(S)$.*

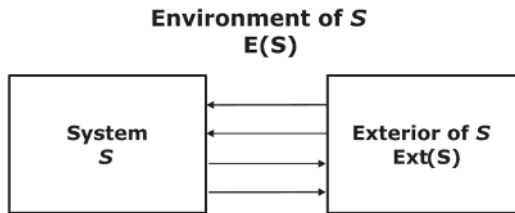


Figure 1.5. *Formal environment of a (formal) system*

A real system of course has several (real) “environment” systems in the sense of Definition 1.4 (the physical world as a whole is the typical environment common to all real systems). The pragmatic constraints of the process of systemically conceiving of a given (real) system S lead us however to introduce the reference environment for S (which is often simply called the environment of S , in short), corresponding to the smallest “useful” environment of S . This is in fact none other than the system resulting from the integration of system S with all the

¹¹ That is, a system whose sets of inputs and outputs are empty, which is the same as saying that there are no inputs or outputs.

(real) systems external to S that have an influence on its conception, consequently ignoring all the other (real) systems external to S when they are considered to have no interdependence (and thus no functional interaction) with the system being envisaged¹².

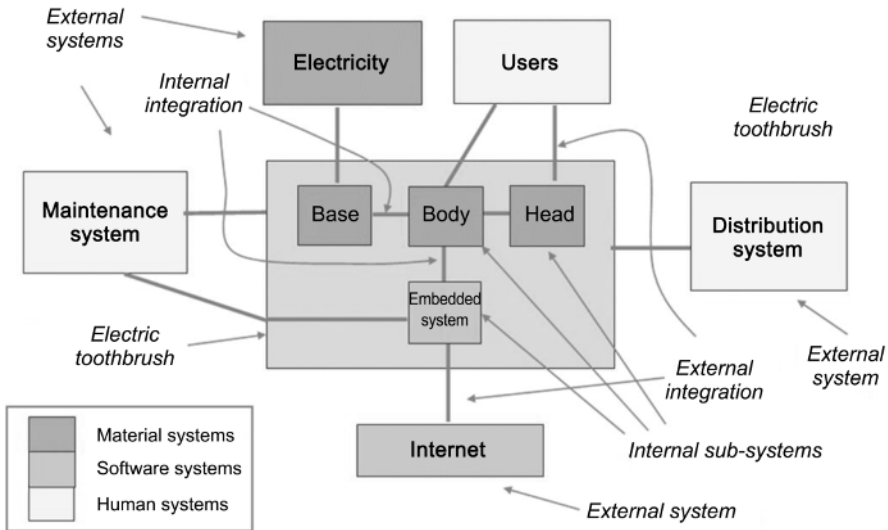


Figure 1.6. Example of the reference environment of a real system

We should point out that from this point on, everything that follows will be relative to a chosen environment for a system that will therefore always be considered as fixed a priori, which will let us speak without misnomers about the “environment” of a system. We can thus naturally adopt the following definitions, which model the fact that the environment of a system S automatically defines a boundary between the interior and the exterior of S .

DEFINITION 1.5.— *Interior and exterior:* let S be a (formal) system, $E(S)$ its (formal) environment, and $Ext(S)$ its (formal) exterior in this environment, in the sense given in Definition 1.4. We can thus say that a (formal) system I (with respect to E) is in the interior (with respect to the exterior) of S if system S (with respect to system $Ext(S)$) can be obtained by integrating I (with respect to E) and another (formal)

¹² We can in fact reason by considering that Proxima Centauri has no influence on most earthbound engineered systems.

system U (with respect to V), necessarily in the interior (with respect to the exterior) of S .

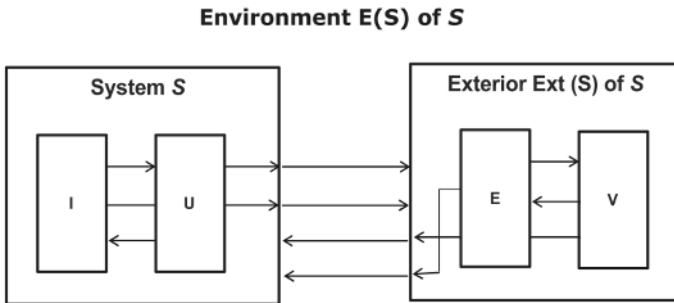


Figure 1.7. Generic examples of systems in the interior of (I) and the exterior (E) of a (formal) system S

In practice, placing this boundary between the exterior and the interior of a (real) system is in fact the first key act in modeling since it allows us to clearly define the system, something that is generally always difficult to do, in reality. We can also note that this distinction is at the heart of the categorization of architectural visions used in systemic analysis (see below), and notably the definition of the first architectural vision of a (real) system, called operational, that synthesizes the interactions of the system's exterior with the system itself.

We are now equipped to introduce the notion of the architectural visions of a system. Keep in mind, these are concepts about modeling that therefore only refer to a real system, contrary to the previous definitions, which all had a common meaning that works for both formal and real systems.

DEFINITION 1.6.— *Architectural visions: let S be a real system made by integrating a set $C(S)$ of smaller, concrete systems. We can therefore introduce the three overall categories of models forming its architectural visions:*

- an operational model of S (in short) is the part of a formal system constituting a model of the exterior of S ;

- a functional model of S is the part of a formal system that is a non-organic model of S (see below);

- an organic model of S is the part of a formal system modeling S , obtained through the composition of formal systems modeling the concrete systems of $C(S)$.

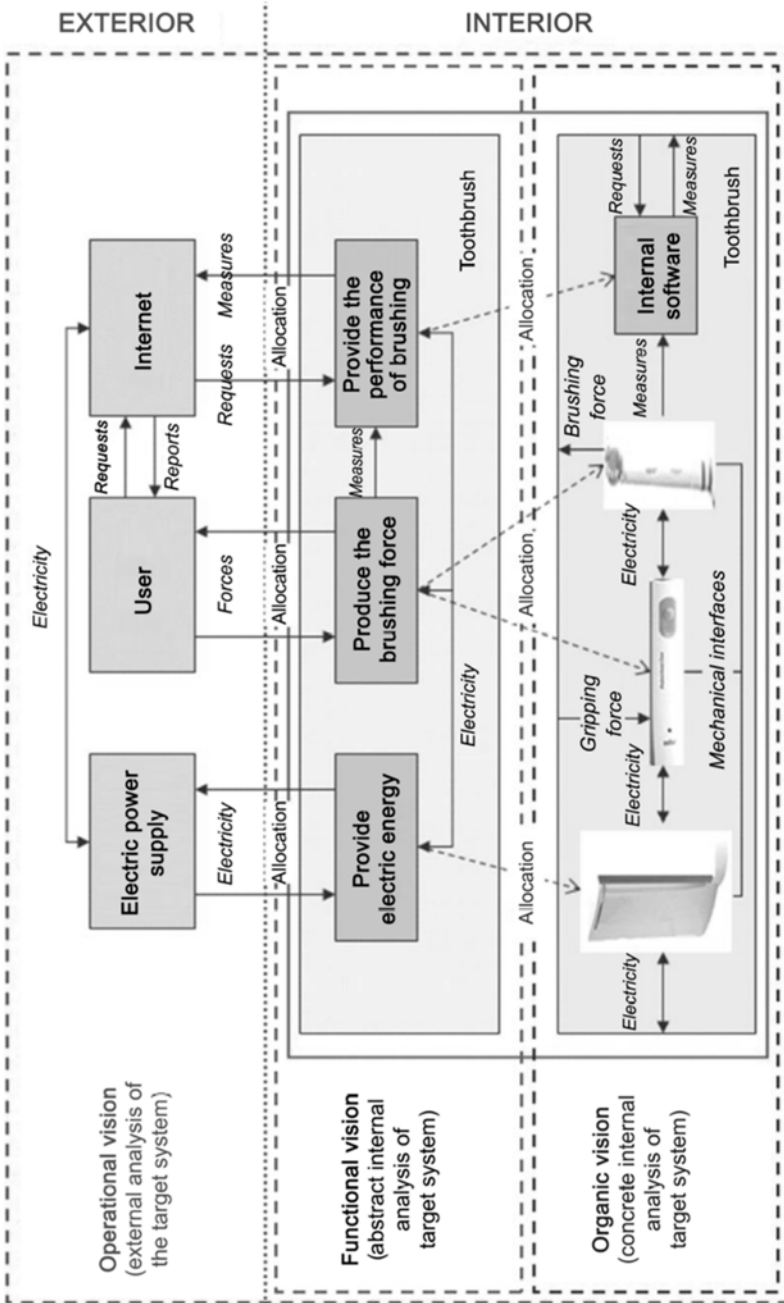


Figure 1.8. Example of architectural visions for a real system

We can first note that Definition 1.6 is indeed consistent, since the existence of non-organic functional models is directly tied to the postulate of emergence (see section 1.2.3), which, let us recall, confirms that the mere knowledge of the concrete components of a system and their laws of interaction is never enough to describe the behavior of the integrated system resulting from their interconnection. This explains why there are still purely functional models of systems whose role is to clearly express the emergent functions that we would not be able to see at the organic level¹³.

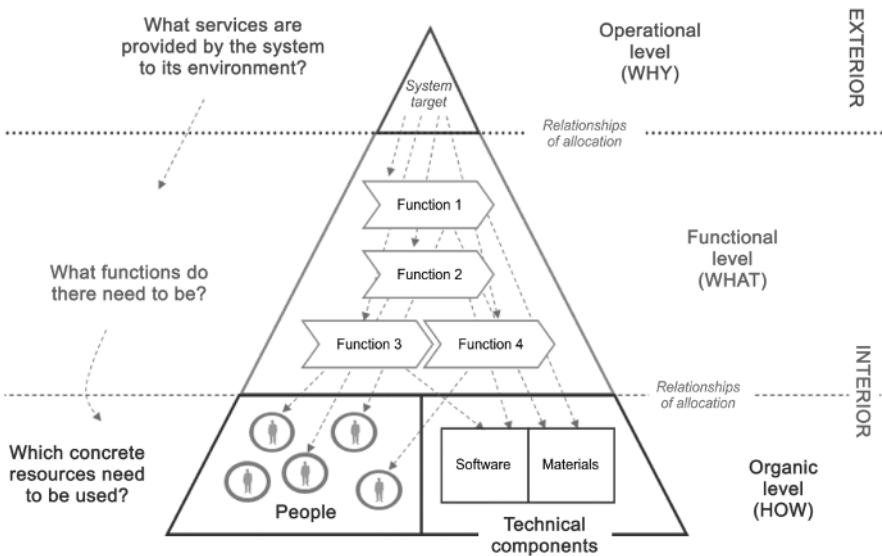


Figure 1.9. A universal framework for the architectural analysis of a real system

¹³ In order to understand this phenomenon well, let us take the example of a car whose (high-level) systemic constitutive components are the undercarriage, the motor, the passenger compartment, the contact with the ground and the built-in electronics. The interaction of these different components is typically sufficient for the performance of functions such as the detection of obstacles, which requires the cooperation of a radar (placed in the undercarriage), an accelerometer (part of the built-in electronics), an illuminated indicator (placed in the passenger compartment) and even the contact with the ground or the motor if we want to act upon the brakes and/or reduce the engine torque when an obstacle is too close to a car. Such a function – called transversal – would clearly be difficult to capture in a purely organic model of the car, where we see exchanged flows between the different components of the vehicle without being able to explain their overall logical structure. Only a functional model of the whole car can therefore capture the semantics of such a transversal function.

We now also understand why it is necessary to have three types of models to model a real system in practice: the operational vision captures the external point of view, while the functional and organic perspectives capture the internal point of view by respectively modeling the emergent behaviors on the one hand, and, on the other, the concrete constitution of the system being considered.

With this now elucidated, it is useful to underscore that the architectural visions play a key role in systems architecture: the first job of a systems architect will as matter of fact consist of classifying the modeling elements at hand, which most often, in practice, include various conceptual levels, according to the classification from Definition 1.6, in order to obtain homogeneous models, each capturing a very limited perspective.

1.3.2. *Properties of a system*

The first way of systemically analyzing a system consists of describing the properties that must be satisfied, which gives rise to the logical models describing the systems that these requirements constitute (which are typically used, in practice, as specification tools for writing requirement specifications in engineering).

DEFINITION 1.7.– *Requirement: let S be a (formal) system. Any logical property (temporal or not) that can be expressed in the language of S , that is, by way of its inputs, outputs and states, can thus be called a requirement of system S .*

To say that an airplane is safe, for example, is in essence the same as imposing a safety requirement upon the “airplane” system, affirming that there is no moment in time in which this system is in a state of “catastrophe” (which must naturally be defined precisely and integrated into the model of the airplane in advance in order to be able to express the previous requirement of safety).

We can now identify how this generic notion of a requirement overlaps with all the architectural perspectives defined in the previous section, which brings us to Definition 1.8.

DEFINITION 1.8.– *Need, functional requirement and organic requirement: let S be a real system. We can thus say that a logical property is:*

- a need if it is a requirement of an operational model of S ;
- a functional requirement if it is a requirement of a functional model of S ;
- an organic requirement if it is a requirement of an organic model of S .

These three concepts are fundamental since the entire architectural reasoning consists of starting from needs, that is, the properties expected by the exterior of the system (what the “clients” of the system want), in order to translate them into functional requirements (what the system must do), and then into organic requirements (how the system must concretely be made).

In practice, organizing the properties that appear empirically throughout the engineering process according to the categories in Definition 1.8 presents major difficulty. It is indeed very easy to identify properties that merrily blend the three architectural perspectives: from the point of view of the constructor of military vehicles, the property “soldiers must move around in a transport vehicle for troops that is painted in green” is thus not a need, nor a requirement of the “transport vehicle” system as it combines external elements (soldiers) and internal elements (the color of the vehicle). Architectural reasoning demands us to rewrite this property in the form of an organic requirement: “the vehicle must be the color green,” which then immediately raises the question of what the (unspoken) need of the soldiers was. We can easily understand with this example that a functional requirement for the vehicle is missing, namely, “the vehicle must be invisible,” which itself is derived directly from a very simple need that the soldiers have: “the soldiers must not die during operations!” By proceeding with this reasoning, we thus make explicit the role of the field of operations in the environment of the target transport system, which in turn lets us realize that the choice of green paint is perhaps well adapted to the wooded terrains of Europe, but surely much less so in the sands of the Gulf, where an ochre color or camouflage would be much more efficient in covering the primitive need of the soldiers, now made clearly explicit.

1.3.3. *Descriptions of a system*

The second way of analyzing a system systemically consists of explicitly describing its behavior in the form of symbolic descriptions using a (formal) ad hoc syntax or, in other words, making its “blueprint.”

The architectural perspectives introduced in Definition 1.5 thus let us posit the following definitions.

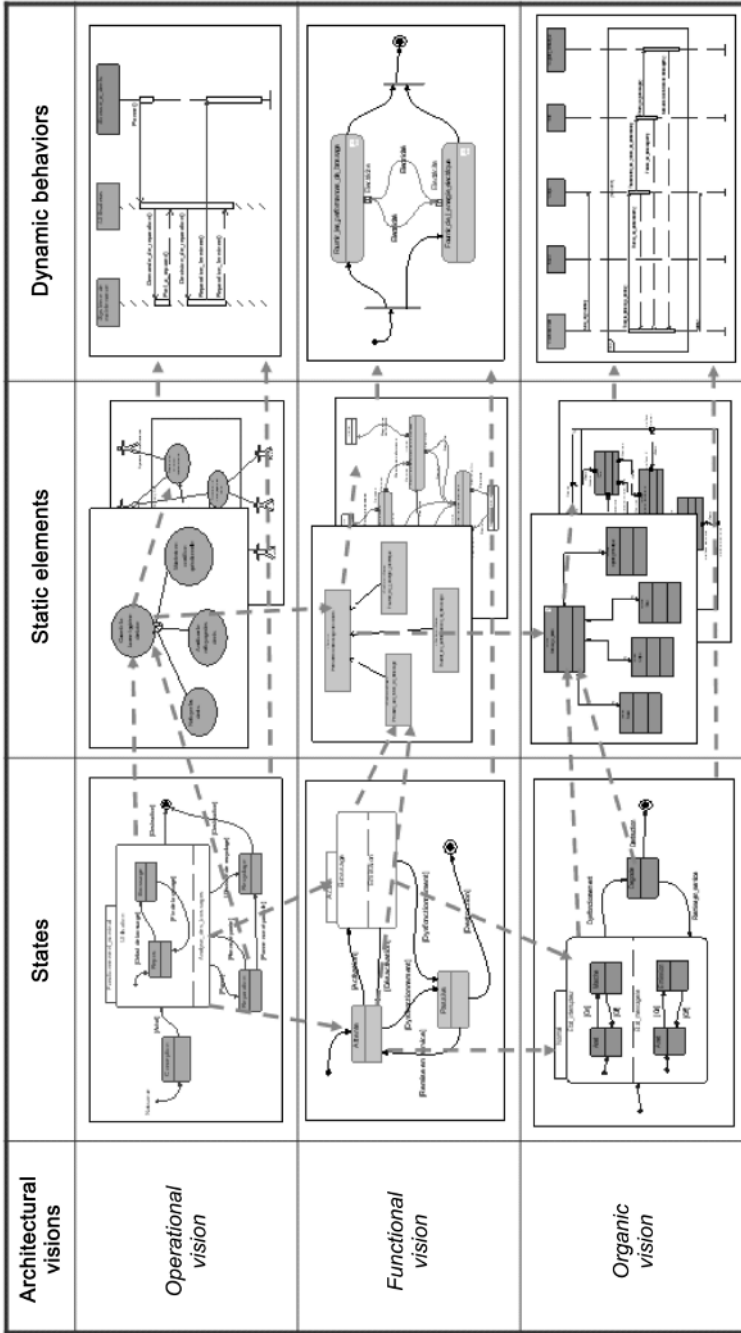


Figure 1.10. Sample organization of real system descriptions

DEFINITION 1.9.— *Operational, functional and organic descriptions: let S be a real system. We can thus say that a symbolic representation of this system is:*

- *an operational description if it is part of an operational model of S ;*
- *a functional description if it is part of a functional model of S ;*
- *an organic description if it is part of an organic model of S .*

We should take note that the very definition of a (formal) system naturally allows us to organize each description of a system according to the following axes, which convey all of the possible components of such a (formal) model of the system:

- 1) a (static) representation listing states;
- 2) a (static) representation listing inputs/outputs;
- 3) a representation of the dynamics of the evolution of states;
- 4) a representation of the dynamics of the behavior of inputs/outputs.

For practical reasons, we usually group the representations in categories (1) and (3) into just one representation – in the form of a finite-state machine – focused on states, giving both their list and their evolution dynamic, and we break representation (4) into two representations, the first of which simply lists the components necessary for the second in a static manner, focused on modeling the behavior dynamic of the system. Figure 1.10 illustrates an example – by abstracting the representation of inputs/outputs – of this way of organizing the descriptions of a system.

1.3.4. Reference framework for system analysis

In summary, we have thus clearly shown that a real system can be analyzed according to the three angles given by its architectural perspectives (operational, functional and organic), its properties and its descriptions (see Figure 1.11).

If we also integrate the fact that a system additionally has a systemic hierarchy (its system levels, sub-systems, sub-sub-systems, etc.), we understand that a system is a fundamentally multidimensional object that is structurally difficult to analyze since we must understand not only each of its architectural components, but also the entirety of their relationships!

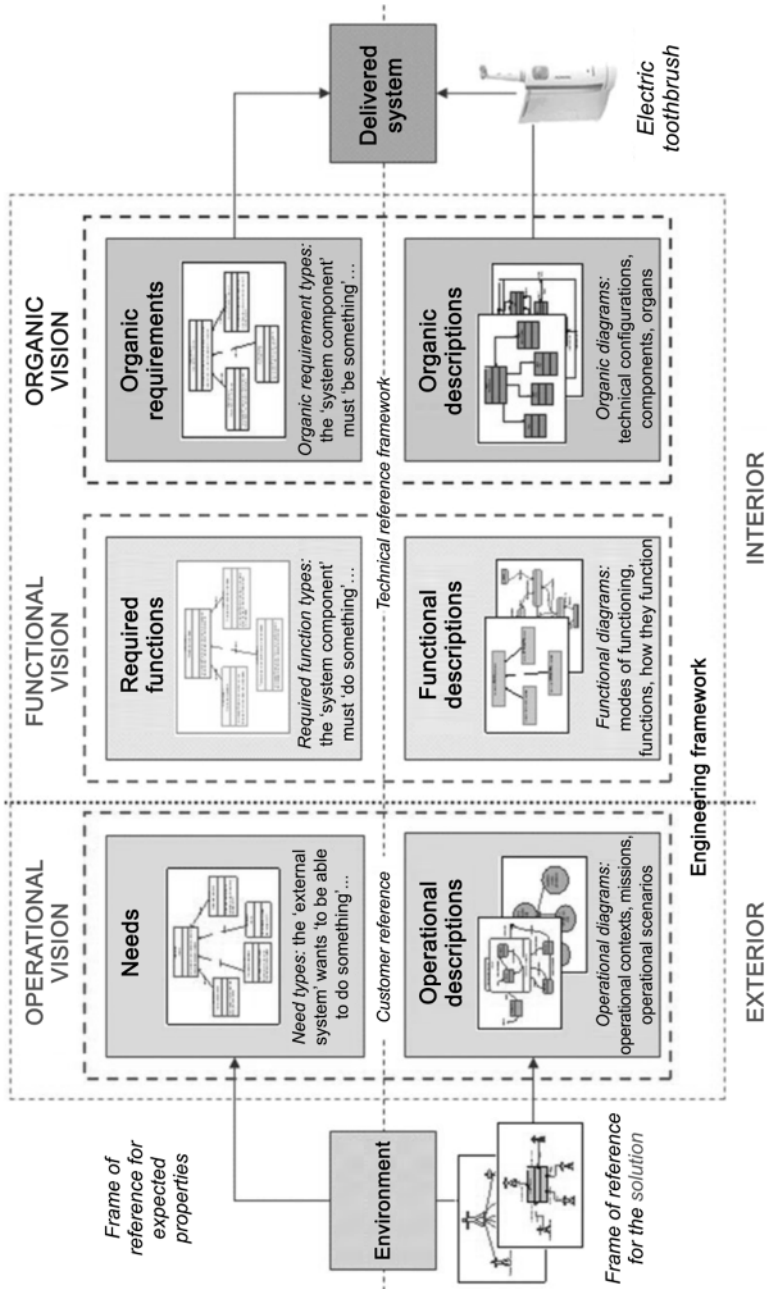


Figure 1.11. The angles of analysis of a (real) system

Finally, we should note that all our considerations can be summarized in the architectural analysis cube presented in Figure 1.12, which provides a generic organization of a *systemic model* of a real system and therefore necessarily contains the *reference framework of the systemic analysis* (which consists of analyzing a real system according to the different axes of the cube).

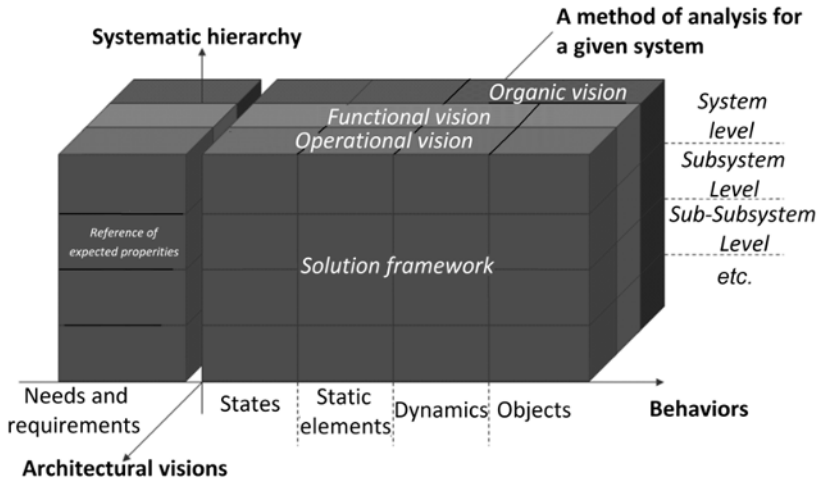


Figure 1.12. The cube of architectural analysis of (real) systems

1.4. Architecture of a system

1.4.1. Systemic perspective of the systems architecture process

In the previous section, we presented the universal structures that exist in all systems which, in particular, led us to the mode of universal organization for systemic models. We can likewise note that the problems in systemic conception also have a common, universal structure since they “simply” consist of making constructive proofs of theorems that (theoretically) all have the following form¹⁴.

THEOREM 1.1.– *There exists a system S that satisfies a set B of needs.*

Equation 1 – Universal structure of a statement of systemic conception.

¹⁴ That is, by concretely building a system that verifies the theorem that needs to be proven.

We thus understand more easily why the existence of a universal identification and construction algorithm for systemic models, which we can call the systems architecture process, is not at all abnormal. It is this latter process that we will now quickly present in section 1.4.2.

Let us first note that the systems architecture process is the same as the functional perspective of the organization in charge of this process, in other words, the functional trace of an architectural conception system that can therefore be systemically analyzed (see Figure 1.13). This “strange cycle” – very characteristic of the reflexivity of systemics which can make use, like mathematics, of its own tools for studying its internal functioning – lets us use the reference framework of architectural analysis, introduced in the previous section, to describe the systems architecture process, as we will now do¹⁵.

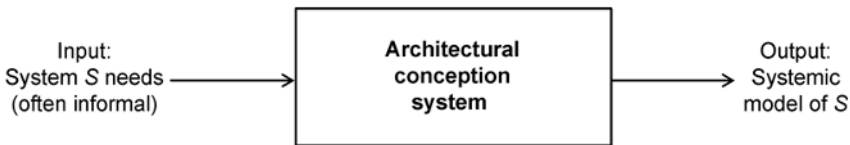


Figure 1.13. *Systemic perspective of the architectural conception system*

1.4.2. Architectural perspectives of the systems architecture process

As we underscored above, any method of systemic analysis first consists of identifying the environment of a system. In the case of an architectural conception system, it is none other than the environment of the engineering process itself, of which the systems architecture process is a component: the latter is typically formed of the company in which the process is deployed, its competitors and clients, the users, operators and suppliers it interacts with, and the regulations and technologies it must account for.

Performing an operational analysis of the architectural conception system thus means asking ourselves what the service that the systems architecture process provides its environment with is; recall simply that systems architecture seeks to master the conception and evolution projects of systems, respecting their cost, deadline, quality and performance constraints. Seen from the outside, the

¹⁵ See the book Hofstadter, D.R. (1979). *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, which also exists in a French translation.

architectural conception system therefore has the principal objective of helping the actors in a project making or transforming a system to always act in a synchronized manner, the absolute condition for succeeding in any integration, by leaning on universal, exhaustive, coherent and shared models.

At the *functional* level, the activities of the systems architecture process are very simple to describe since they automatically derive from the reference framework of the systemic analysis, introduced in the previous section. These can indeed be separated naturally into the following activities:

- the architecture of a (real) system’s needs and requirements, with the objective of formalizing its functional and organic needs and requirements;
- the operational, functional and organic architecture of a (real) system, with the objective of building its operational, functional and organic models;
- verification and validation, which consist of moving up the cycle in order to ensure the real system actually satisfies its requirements and needs.

The key point that should be underscored here is that we can associate a generic process (that would of course be too long to describe here) to each of these architectural activities, which simultaneously allows for the universal method of system conception that we spoke of earlier to emerge, along with a “toolkit” of systems architecture¹⁶. It is also interesting to give the overall dynamic that exists between these different activities (see Figure 1.14). Finally, let us finish with the organic dimension of the architectural conception system. It is made up of two primary components:

- *humans* including, notably, the key character of the systems architect, who has a double role in managing the integration of the technical systems and human systems in their field of charge: building coherent interfaces within a technical system always in fact requires bringing the interested parties of these interfaces around to a unified vision of their system;
- the *tools* of the systems architecture process, which on the one hand include software tools for the engineering requirements and behavior modeling, based on description languages such as BPMN (Business Process Modeling Notation), UML (Unified Modeling Language) or SysML (System Modeling Language), and on the other hand, the conceptual tools introduced in section 1.3.3, not to mention common

16 See Printz (2019), which exists in an English translation.

sense and heuristic rules, of course, as well as the good practices that come from experience¹⁷.

1.5. In conclusion

We hope that this summary has allowed the reader to better understand the stakes and issues involved in systemics. It naturally does not have the pretension of covering the entirety of this discipline and intentionally omits numerous very important subjects: systems dynamics, systems sizing and optimization, systems evolution trajectories, the conception of homogeneous families of systems, agile and collaborative approaches to systems development, systems governance, etc., which are covered in the previously mentioned book, *Systems Architecture and Complexity*, and the others mentioned throughout the following chapters and in the references. The world of systems is indeed a “continent” that forms a real scientific field for a science of systems¹⁸.

Systemics still however needs to be developed, necessarily through a profound interaction between theory and practice, despite realizations that are, even now, old – and unfortunately almost without follow-up – by the great fathers of the field such as Simon¹⁹ or von Bertalanffy²⁰. Let us therefore hope that the systemic challenges that the world is beginning to face, with the financial crisis of 2008, then the much more profound one caused by the emergence of the Covid-19 virus, will finally allow the field to fully emerge²¹.

17 On BPMN, see White, S.A. and Miers, D. (2008). *BPMN Modeling and Reference Guide, Understanding and Using BPMN*. Future Strategies, as well as the general bibliography. On UML, see Booch, G., Jacobson, I., Rumbaugh, J. (2004). *The Unified Modeling Language Reference Manual*, 2nd edition. Addison-Wesley, as well as the general bibliography. On SysML, see Friedenthal, S., Moore, A.C., Steiner, R. (2012). *A Practical Guide to SysML: the Systems Modeling Language*. Morgan Kaufmann OMG Press.

18 To use the expression that Marcel Paul Schützenberger surely would not have rejected (founding “father” of the French school of theoretical computer science, with Maurice Nivat).

19 See Simon, H. (1962). The architecture of complexity. *Proceedings of the American Philosophical Association*, 106(6), 467–482, as well as the general bibliography.

20 See von Bertalanffy, K.L. (1968–1976). *General System Theory: Foundations, Development, Applications*, George Braziller, a French translation has been published with Dunod, *Théorie générale des systèmes*, 1973. For a detailed study of the historical aspects, see the thesis Pouvreau, D. (2013). “Une histoire de la ‘systémologie générale’ de Ludwig Bertalanffy – Généalogie, genèse, actualisation et postérité d’un projet herméneutique”, EHESS, March.

21 See De Weck, O., Krob, D., Lefei, L., Lui Pao, C., Rauzy, A., Zhang, X. (2020). Handling the COVID-19 crisis: Towards an agile model-based systems approach. *Systems Engineering*, 23(5), 656–670.

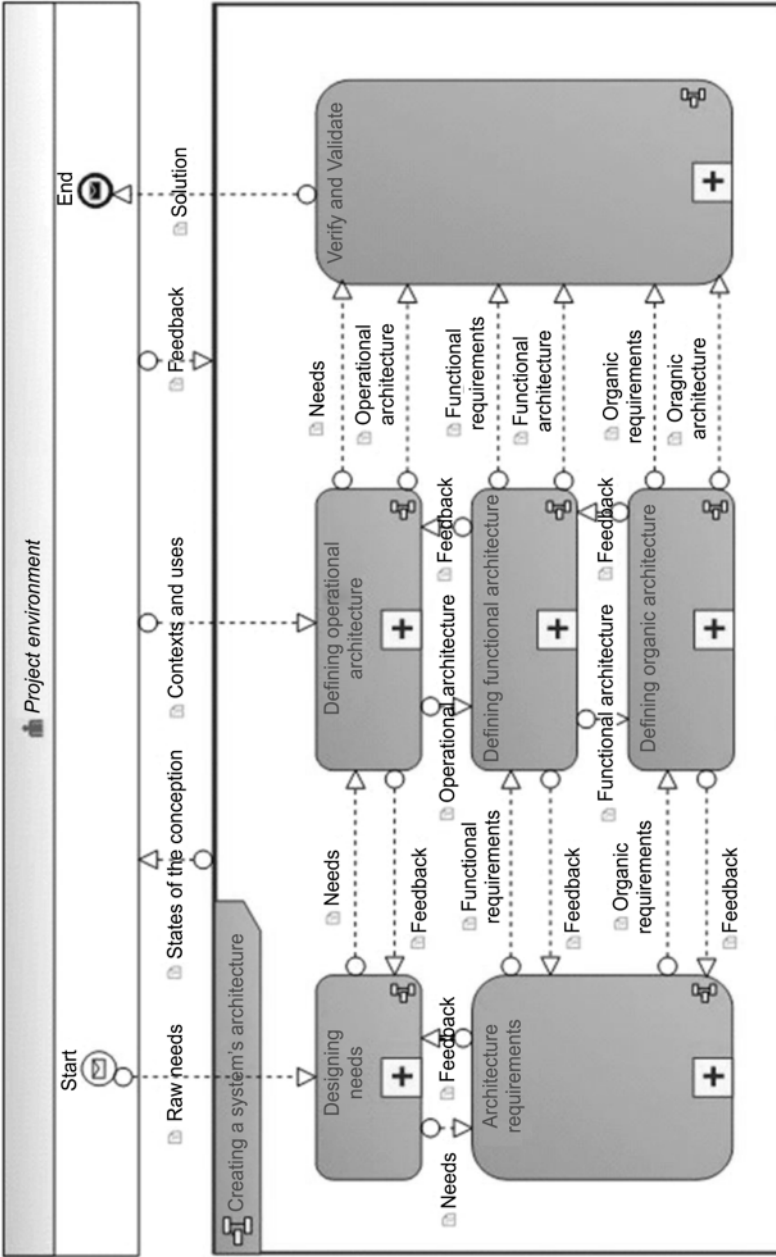


Figure 1.14. Description of the functional dynamic of the architecture process of systems

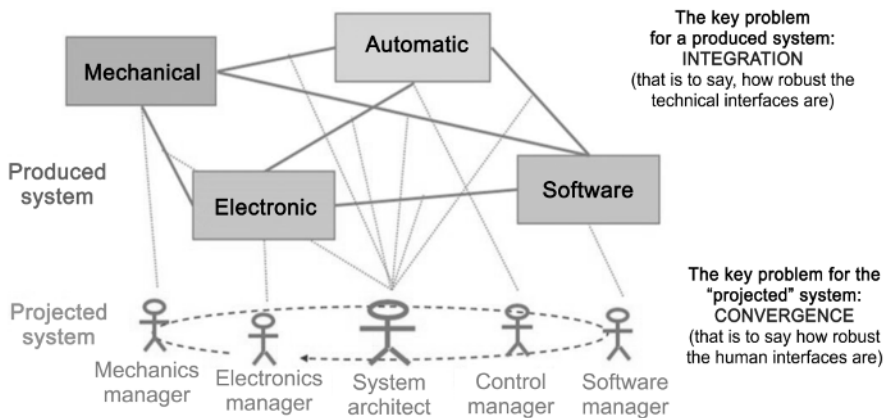


Figure 1.15. *The role of a systems architect*

