On the Notion of a System

Hans-Gert Gräbe

October 18, 2021

1 Once more about the Goal of the Seminar

Systematic innovation methodologies such as TRIZ are essentially based on a better understanding of the development dynamics of corresponding (technical and non-technical) *systems*. The results are rooted in engineering experience from structured processes of planning, implementation and operation of technical systems. Increasingly, cooperative interdisciplinary collaboration matters rather than the one brilliant mind that commands thousand hands. The *socio-technical character* of contradictions is thereby intensified and opens up new dimensions of contradiction management.

Today, managers face similar challenges when it comes to placing decision-making processes on a systematic basis, aligning the processes under control with long term goals, and also achieving the targeted goal corridors. It turns out that many engineering experiences on structured procedures in contradictory requirement situations can be transferred to this area, which has been investigated within the topic "TRIZ and Business" for 20 years.

Nevertheless, experiences and approaches to theories of systemic concepts are based more broadly and also have much longer historical traditions. *In the seminar*, we want to study such concepts more closely, with special emphasis on cooperative approaches in interdisciplinary contexts.

2 What is a System?

Operation and use of technical systems is a central element of today world changing human practices. For this purpose planned and coordinated action along a division of labour is necessary, because exploiting the benefit of a system requires its operation. Conversely, it makes little sense to operate a system that is not being used. Closely related to this distinction between definition and call of a function, well known from computer science, is the distinction between design time and runtime, that is even more important in the real-world use of technical systems – during design time, the principal cooperative interaction is planned, during the runtime the plan is executed. For technical systems one has to distinguish the description, interpersonally communicated as justified expectations, and the results of operation, interpersonally communicated as practical experience.

Most definitions grasp the term *system* as a delimited set of interacting components, whereby the interaction of the components gives rise to a unified whole, which realises an emergent function and is thus more than the sum of its parts.

> A *system* (lat. greek "system", "composed", a whole consisting of parts; connection) is a set of elements that are interconnected and interact with each other, forming a unified whole that possesses properties that are not already contained in the constituent elements considered individually. [2]
>
> A *system* is a set of elements that are in relationship and connection with each other and that constitute a well defined unity, an integrity. The necessity of the use of the term "system" occurs when it is required to emphasize that something is large, complex, immediately not wholly comprehensible, but at the same time a unified whole. Unlike the notions "set" or "aggregate", the concept of a system emphasizes the ordering, the integrity, the regularity of construction, functioning and development. [6]
>
> Systems Engineering "is an interdisciplinary field of engineering and engineering management that focuses on how to design, integrate, and manage complex systems over their life cycles. At its core, systems engineering utilizes systems thinking principles to organize this body of knowledge. The individual outcome of such efforts, an engineered system, can be defined as a combination of components that work in synergy to collectively perform a useful function." (English Wikipedia).

The second definition also points to the purpose of systemic delimitations – it is about making complex relationships accessible to description by reducing them to essentials.

In all these definitions, the *structuredness* and thus *decomposability* of the system in the analytic dimension is emphasised on the one hand, and the *interdependence* and thus *indecomposability* in the execution dimension on the other. This corresponds to the practical experience of the engineer when she assembles a system from individual components – the system is only viable when it is assembled. In the assembled system in addition to the components, the *connecting elements* also play an important role. They mediate the *flow of energy, material and information* that is required for the operation of each component. In component software [5], with *deployment*, *installation* and *configuration* three stages of preparing components for their operation in a systemic context are distinguished, and this preparation for operation is often considered as a separate state – for example, as *maintenance mode* different from the *operation mode.*

The aspect of operating a system did not play a role in the first two definitions. Only here, however, the dialectical interrelationship between decomposability and indecomposability comes to light: Viable components deliver processual services in *guaranteed* quantity and quality during operation, if the *external operational conditions* are guaranteed. These processual services of the components in combination form the emergent function of the overall system. The self-similarity of the concept is obvious: components themselves have an inner life that can be described systemically, but which is largely abstracted from at the level of the overall system. The component enters the description of the dynamics of the overall system only as Black Box with a precise specification. This specification is divided into input and output interfaces. The former describe the necessary operating conditions, the latter the performance parameters of the respective component.

In [1] the system concept is identified as descriptional focusing to make real-world phenomena accessible for a description by reduction to the essentials. Such a reduction focuses on the following three dimensions:

(1) Outer demarcation of the system against an *environment*, reduction of these relationships to input/output relationships and guaranteed throughput.
(2) Inner demarcation of the system by combining subareas to *components*, whose functioning is reduced to “behavioural control” via input/output relations.
(3) Reduction of the relations in the system itself to “causally essential” relationships.

Further, it is stated that such a reductive description (explicitly or implicitly) exploits output from prior life:

(1) An at least vague idea about the (working) input/output services of the environment.
(2) A clear idea of the inner workings of the components (beyond the pure specification).
(3) An at least vague idea about causalities in the system itself, that precedes the detailed modelling.

3 Systems, Components and Reuse

One important aspect, especially of technical systems, has not yet been taken into account in the considerations so far: the aspect of reuse. Reuse plays a central role in computer science – copy/paste of code, outsourcing of repetitive pieces of code in function definitions, grouping of related function definitions in pre-compiled libraries, etc. This in no way exhausts possible forms of reuse, not to mention higher forms of reuse such as design, patterns or frameworks. Szyperski discusses in [5, ch. 8] aspects of the relationship between goals and forms of reuse.

Hence in addition to the description and operation, for technical systems the aspect of reuse plays an important role. However, this applies, at least on the artifact level, not to larger technical systems – these are unique specimen, even though assembled using standardised components. Also the majority of computer scientists is concerned with the creation of such unique specimens, because the IT systems that control such plants are also unique.

Computer science has long struggled with a form of reuse that is widespread in developed engineering sciences and ultimately turned the manufacture of tools and products from an art first to a craft and later to an industrial process – the *use of components produced by third parties* (components off the shelf).

Thus, after the analytical and operational dimension of systems and components, the *production by independent third parties* and thus the technical-economic interrelationships of an industrial mode of production based on the division of labour move into the focus of attention.

In such a context, the concept of a technical system is fourfold overloaded. A technical systems can be considered

1. as a real-world unique specimen (e.g. as a product or a service),
2. as a description of this real-world unique specimen (e.g. in the form of a special product configuration)

and for components produced in larger quantities also

3. as description of the design of the system template (product design) and
4. as description and operation of the delivery and operating structures of the real-world unique specimens of this system produced according to this template (as production, quality assurance, delivery, operational and maintenance plans).

The concept of a technical system thus has also in this context a clearly epistemic function of (functional) “reduction to the essential”. To Einstein the recommendation is attributed “to make things as simple as possible but not simpler”. The TRIZ *law of completeness of a system* expresses exactly this thought, however, not as a *law*, but as an engineering *modeling directive*. The apparent “law” of the observed dynamics therefore essentially addresses *reasonable human action.*

In an approach of “reduction to the essential” and “guaranteed specification-compliant operation” human practices are inherently built in, since only in such a context the terms “essential”, “guarantee” and “operation” can be filled with sense in a meaningful way. These essential terms from the socially determined practical relationship of people are deeply rooted in the concept generation processes of descriptions of special technical systems and find their “natural” continuation in the special social settings of a legally constituted societal system.

4 Socio-technical Systems

The last considerations already embed the concept of system in social practices of cooperatively acting people. This embedding is also present in TRIZ system definitions, when the emergent function realised by the system is considered as *main useful function* MUF and linked to a *purpose*, why this (technical) system exists or was designed or redesigned in this way. This *aspect of purposefulness* (Zweckmäßigkeit) plays only a subordinate role for ”natural” systems, namely for socio-ecological systems, since in this context in most cases the ”purposefulness” comes up against hard limits or causes massive problems or has even already caused them. Nevertheless, this *orientation on purposes* is another throughput parameter (e.g. as monetary throughput) from a social environment relevant for the inner dynamics of a system. It can ultimately be subsumed under the *throughput of information* if a sufficiently viable concept of information is taken as a basis.

This purposefulness transforms the mass of technical systems into an interconnected *world of techical systems* full of preconditions and conditionalities, which opens up a fourth dimension of the concept of system, to secure stable operating conditions of the systems themselves.

The self-similarity of the systems concept provides a solution for this challenge – consider systems as components and the relations of purposefulness as interdependencies, delineate larger socio-technical systemic units, develop appropriate forms of description and operation. The transformation towards a sustainable mode of production and living that is on the agenda just requires a big step forward in this direction. This is one of the objectives of management and hence in the primary focus of our seminar. However, socio-technical systems are, in addition to technical restrictions, charged with the contradictory expectations and interests of concrete people and groups of people.

Ian Sommerville [4] elaborates a number of challenges in this regard. He also starts with the concept of a goal-centered system.

> A *system* is a meaningful set of interconnected components that work together to achieve a specific goal. [4]

Right after he develops a distinction between technical and socio-technical systems:

Technical computer-based systems are systems that contain hardware and software components, but not procedures and processes. ... Individuals and organisations use technical systems for specific purposes, but knowledge of that purpose is not part of the system. For example, the word processor I use does not know that I am using it to write a book.

Socio-technical systems contain one or more technical systems, but beyond that – and this is crucial – the knowledge of how the system should be used to achieve a broader purpose. This means that these systems have *defined work processes*, *human operators* as integral part of the system, are *governed by organisational policies* and are *affected by external constraints* such as national laws and regulations.

Essential characteristics of socio-technical systems:

1. They have special properties that affect the system as a whole, and are not related to individual parts of the system. These special properties depend on the system components and the relationships between them. Because of this complexity, the system-specific properties can only be evaluated when the system is composed.
2. They are often not deterministic. The behaviour of the system depends on the human operators and on other people who do not always react in the same way. Also, the operation of the system can change the system itself.
3. The extent to which the system supports organisational goals depends not only on the system itself. It also depends on the *stability of the goals*, the relationships and *conflicts between organisational goals*, and how people in the organisation *interpret those goals.*

In this context, there is a clear shift on the scale of controllability from direct control by external human operators to indirect control and movement according to intrinsic laws, which is even more prevalent in **socio-economic systems** with a large number of stakeholders or even **socio-ecological systems**.

5 Philosophical Consequences. Shchedrovitsky on Systems Analysis

The system concept thus serves to delimit a part of the complex, all-connected world (hereafter *reality*) in order to make this part accessible for description. However, this human activity, which Shchedrovitsky refers to as *mental activity* [3, p. 47], is itself part of that reality and thus also of practical relevance. Real-world processes are thereby charged with description forms. Thus in systems these two dimensions – description and operation – must therefore be distinguished. Charging a system with a description form is what Engels' calls, in reference to Kant's *thing in itself*, the transformation of the *thing in itself* into a *thing for us.*

Shchedrovitsky [3, p. 80 ff.] conceptualises this process in two different concepts of system [3, pp. 89 and 98] as process of breaking down the system into parts (components), charging the components with description forms and then reassembling the components thus charged into a whole. The result is a *new* system in the sense that it is the old one but charged with a description form. In this way, the *structural organisation* of a system can be grasped.

The real world and thus also systems develop and change over time. In order to understand the *development of a system*, its *processual organisation* must be examined. Shchedrovitsky emphasises that the development of a system can *never* be described in its disassembled form, since disassembly destroys the systemic coherence. An aeroplane disassembled into its individual parts cannot fly, only an assembled one. We are dealing here with a fundamental epistemic contradiction.

For details we refer to [3].

6 Theory of Dynamical Systems

6.1 The Approach

The Theory of Dynamical Systems as a branch of mathematics investigates the dynamics of such structurally defined and modelled systems. Attributes essential for the description of the system are combined into a *phase space* and the changes in the attribute values are described as *equations of motion* by differential equations. If only temporal changes are considered, this leads to systems of *Ordinary Differential Equations* (ODE), complex spatio-temporal changes lead to Partial Differential Equations. We restrict ourselves to the first case, i.e. purely temporal structural changes.

In the simplest case, such as the pendulum or the movement of two bodies in a homogeneous gravitaional field, a *trajectory* can be calculated from the equations of motion.

Examples:

- Pendulum: `https://en.wikipedia.org/wiki/Pendulum_(mechanics)`
- Two body problem: `https://en.wikipedia.org/wiki/Two-body_problem`

6.2 Model and Reality

However, the solution of this equations only describes the motion $m(t)$ in the model. Good modelling is characterised by the fact that the real movement $f(t)$ and the movement $m(t)$ according to the model differ only insignificantly $r(t) = f(t) - m(t)$ in practically relevant parameter ranges (the *context of observation*). This can only be verified empirically through experiments that are to be planned more or less precisely, since reality is only accessible empirically.

Particularly interesting are modellings in which the residual $r(t)$ decreases "by itself". Such systems strive towards an equilibrium, which structure can be derived from the model.

6.3 How Chaotic can Trajectories be?

Examples:

- Double Pendulum, `https://en.wikipedia.org/wiki/Double_pendulum`

- Magnetic pendulum with three attracting magnets,
- 3-body model: `https://en.wikipedia.org/wiki/Three-body_problem`

We see that there is apparent stability for a long time, but in phase space there are certain *areas of instability* in which (exactly calculable!) trajectories passing through points in phase space that are close to each other strongly diverge. Such locations are called *bifurcations*. Often there is a single phase parameter that makes this bifurcation particularly clear. Such a bifurcation on a one-dimensional scale is also called a *tipping point*.

Not everything that looks like chaos has to be chaotic:
`https://i.redd.it/zr7tet9mdfl01.gif`

6.4 Attractors

How complicated can an equilibrium position be?

Examples:

- Pendulum,
- pendulum with three attracting magnets,
- pendulum with one repelling magnet.

Limit cycles: `https://en.wikipedia.org/wiki/Limit_cycle`

When the body is on the limit cycle, it remains there, i.e. the limit cycle is a *stable solution* of the equations of motion of the system, called **steady-state equilibrium**.

In many cases the real movement $f(t)$ in time is *attracted* by that limit cycle, i.e. $f(t)$ can be decomposed into $f(t) = l(t) + r(t)$ with $l(t)$ the projection on the limit cycle and $r(t)$ a (small) orthogonal deviation. In this way, it is often possible to simplify complicated models.

An attractor is a specific steady-state equilibrium with just this attracting property.

More precisely: Let $f(t, a)$ be a function which specifies the dynamics of the system with starting point $f(0, a) = a$. An **attractor** is a subset A of the phase space characterized by the following three conditions:

- A is forward invariant under f: if a is an element of A then so is $f(t, a)$, for all $t > 0$.
- There exists a neighborhood of A, called the basin of attraction for A and denoted $B(A)$, which consists of all points b that "enter A in the limit $t \to \infty$".
- There is no proper (non-empty) subset of A having the first two properties.

Attractor as stable solution of the corresponding system of ODE
`https://en.wikipedia.org/wiki/Attractor`

On the importance of "stable" cyclical processes in nature. We are able to perceive such *approximately* repeating patterns in natural processes (i.e. attractors), i.e. perform such a reduction also independently of mathematical abilities.

For given (deterministic) equations of motion one can compute the geometry of such an attractor as *global deterministic* invariant of the equations of motion.

How Complicated can an Attractor be?

- https://en.wikipedia.org/wiki/Attractor
- https://en.wikipedia.org/wiki/Lorenz_system
- https://de.wikipedia.org/wiki/Lorenz-Attraktor
- Attention, with the numerical methods used there for visualisation it is difficult to distinguish whether they are calculating a chaotic trajectory or really the attractor, which is a *global* artefact.
- "Almost all initial points will tend to an invariant set – the Lorenz attractor – a strange attractor, a fractal, and a self-excited attractor" (Wikipedia)

6.5 Dissipative Systems

Importance of a (stable) throughput of energy, matter and information for the inner structure formation in systems. Previous investigations were directed towards the inner dynamics of an autonomous, i.e. closed system.

- Self-organisation in dissipative structures
  - https://en.wikipedia.org/wiki/Rayleigh-Bnard_convection
  - https://en.wikipedia.org/wiki/Belousov-Zhabotinsky_reaction
- Dissipative systems https://en.wikipedia.org/wiki/Dissipative_system
- Life on Earth as a dissipative system.

References

[1] Hans-Gert Gräbe (2020). Men and their technical systems (in German). LIFIS Online, 19 May 2020. https://doi.org/10.14625/graebe_20200519.

A shorter English version is available at https://hg-graebe.de/EigeneTexte/sys-20-en.pdf

[2] Vladimir Petrov (2020). Laws and patterns of systems development (in Russian). ISBN 978-5-0051-5728-7.

[3] Georgy P. Shchedrovitsky. Selected Works. Part I in Viktor B. Khristenko, Andrei G. Reus, Alexander P. Zinchenko et al. (2014). Methodological School of Management. Bloomsbury Publishing. ISBN 978-1-4729-1029-5.

[4] Ian Sommerville (2015). Software Engineering. Chapter 19 „Systems Engineering“.

Slide stack available at Sildeshare https://www.slideshare.net/software-engineering-book/ch19-systems-engineering

[5] Clemens Szyperski (2002). Component Software. Pearson Education. 2. Auflage. ISBN 0201745720.

[6] The TRIZ Ontology Project. https://wumm-project.github.io/Ontology.html.