Title

Towards a Theory of General Autonomous Systems

ASLab Seminars on Autonomous Systems I

Author Ignacio López and Ricardo Sanz

Reference R-2006-009
Release 0.4 Draft
Date November 11-12, 2006

Address
Autonomous Systems Laboratory
UPM - ETS Ingenieros Industriales
José Gutierrez Abascal 2
28006 Madrid
SPAIN

Towards a Theory of General Autonomous Systems

ASLab R-2006-009 v 0.4 Draft of November 11-12, 2006

Abstract

One of the objectives of the ASLab Research Group is the establishment of a solid scientific foundation on autonomy and autonomous systems. This document addressees this task from the perspective of General Systems Theory; a bold, all-encompassing, perhaps too weak vision and strategy but, difficulties notwithstanding, an effort in the line of *unified* science and technology.

Keywords

Autonomy, autonomous systems, general systems theory, minds and bodies, perception and action.

Acknowledgements

We acknowledge the support of the Spanish Ministry of Education and Science through grant *SCC: Semántica y Control Consciente* and the European Commission thorough Grant *ICEA: Integrating Cognition, Emotion and Autonomy*.

Table of Contents

Chapter 1

Seminar Description

1.1 Seminar Motivation

The ASLab long term project ASys (http://www.aslab.org/public/projects/ASYS/) is focused in the development of universal technology for the construction of high autonomy systems.

As part of this effort, we must do a solid scientific grounding of the concept of *system*. To do this we are trying the conceptual scaffolding provided by Bertalanffy's *General Systems Theory* (**?**).

As a first step, it is necessary the evaluation of General Systems Theory for the domains and purposes we are interested in.

The purpose of this seminar is the introduction and starting of this GTS view of cognitive control systems under the real-world constraints for embedded system technology.

1.2 Seminar Structure

The purpose of the seminar is the introduction, evaluation and profiling of GST concepts in the context of autonomous systems engineering.

Hence the necessary steps to follow are:

1. Acquisition of knowledge about GST
2. Use of this knowledge in autonomous systems
3. Evaluation of the applicability/effectiveness
4. Profiling and modification of GST ontology

For the phase 1, information will be disseminated among the participants. For the phase 2 each participant will apply the concepts to a particular system of election. For phase 3, a discussion will be held and for phase 4 some GST profiling proposals will be distilled.

1.3 Seminar contents

Some of the topics that may be addressed include:

General Systems Theory: Systems theory is a transdisciplinary/multiperspectual theory that studies structure and properties of systems in terms of relationships from which new properties of wholes emerge.

Information Theory: Information theory is a discipline in applied mathematics involving the quantification of data with the goal of enabling as much data as possible to be reliably stored on a medium and/or communicated over a channel.

Epistemology: Epistemology or theory of knowledge is the branch of Western philosophy that studies the nature and scope of knowledge.

Technical systems and Informational systems: the AD/DA frontier

Cognitive "Systems"

Control theory: In engineering and mathematics, control theory deals with the behavior of dynamical systems. The desired output of a system is called the *reference*. When one or more output variables of a system need to follow a certain reference over time, a controller manipulates the system inputs to obtain the desired effect on the system output.

1.4 Agenda

9:30 Meeting to travel

12:00 Start. Organisation of the seminar and agenda definition. Setting of objectives for the seminar.

13:30 Lunch

15:30 Afternoon session

Ignacio: GST

Jaime: Epistemology

All: Exemplar systems to consider

19:00 End

9:30 Start

open time

13:30 Lunch

15:30 Travel back

1.5 Venue

Residencia Forestal Lucas Olazábal

Universidad Politécnica de Madrid

http://www.upm.es/laupm/servicios/residencias/lucas_olazabal.html

Chapter 2

Control and the Mind–Body Problem

2.1 On minds and bodies

The question of what control systems engineering is about is quite simple: we build minds for machines. In a sense, we build minds for bodies giving them some special character that can only be addressed in the realm of the mental. Is in this context that we are continuously addressing the age-old mind-body problem.

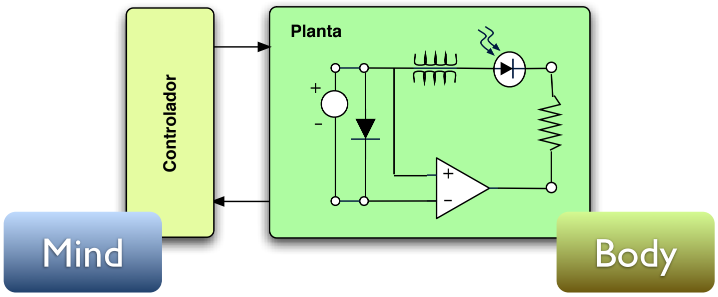

Figure 2.1: Control engineers are continuously addressing the age-old, much discussed, mind-body problem.

2.2 What is the Mind–Body Problem?

In the words of Wikipedia:

> *"The mind-body problem is essentially the problem of explaining the relationship between minds, or mental processes, and bodily states or processes (**?**). Our perceptual experiences depend on stimuli which arrive at our various sensory organs from the external world and these stimuli cause changes in the states of our brain, ultimately causing us to feel a sensation which may be pleasant or unpleasant. Someone's desire for a slice of pizza will tend to cause that person to move their body in*

a certain manner in a certain direction in an effort to obtain what they want. But how is it possible that conscious experiences can arise out of an inert lump of gray matter endowed with electrochemical properties? How does someone's desire cause that individual's neurons to fire and his muscles to contract in exactly the right manner? These are some of the essential puzzles that have confronted epistemologists and philosophers of mind at least from the time of René Descartes."

Obviously this question of experience is yet to be solved but the issues regarding how it is possible that sensed data can affect what we do are well known in control engineering: AD/DA conversion and some processing inside. The experiential, consciousness issues would find an answer after the brain structure is clarified. It may be the case that computers cannot feel due to their digital nature or seen the other way, that continuous domain machines can (**?**).

Things get mixed here as we tend to think that we use our brains to think when indeed we only use them to act (in some cases to learn to act). In the words of Powers:

> *I have often said that we learn control systems, not acts or responses. What does this mean? It means that when we learn to control some variable, we don't learn to produce any particular action, either as a reaction to events in the world or as outputs that are planned and then executed.*
>
> *William T. Powers*

The analysis of inteligent systems has many approaches and focus points. From the quite humanistic psychology to the purely technical control engineering, disciplines abound: cognitive science, philosophy, artificial intelligence, robotics, etc.

An interesting —but somewhat confusing— effort that we have recently discovered in the line of the good, old *cybernetics*, is the Global TOGA Meta-Theory proposed by Gadomski in his Top-down Object-based Goal-oriented Approach (TOGA) to intelligent socio-technical, decision systems construction [1].

In Gadomski words:

> *The first TOGA ontological axiom is:*
>
> *In every real-world problem exists the couple: an intelligent abstract entity and its environment.*
>
> *This entity is arbitrarily called "agent" or "intelligent agent".*
>
> *Therefore, TOGA identifies every such couple as its own domain of interest, and it is identified/specified from the perspective of this embedded*

[1] http://erg4146.casaccia.enea.it/wwwerg26701/Gad-toga.htm

abstract intelligent agent/entity.

In the above sense, the problem of the existence of the"absolute" reality does not exist in TOGA. We should recall that it is goal-oriented, it means, it has to be useful, not "true", and goals are the attributes of only intelligent being/entity/agent.

The first TOGA epistemological axiom is:

Every intelligent abstract entity , its environment and their interaction, are decomposable in parallel, top-down, goal-oriented and using an object-based conceptualization framework.

Gadomski, TOGA

2.3 The question of knowledge

Action generation in an intelilgent system seems to be produced by means of knowledge exploitation. However, some questions appear beyond the core philosophical issues of the vary possibility of knowledge:

What does an agent need to know to achieve its objectives?

How is this knowledge acquired?

How is it stored?

How is it used?

In artificial systems, is it the same knowledge as its engineer's?

For sure there are many types of knowledge and even there is a lot of confusion in the use of some words: information, data, knowledge, etc.

According to Russell Ackoff, a systems theorist and professor of organizational change, the content of the human mind can be classified into five categories:

Data: symbols

Information: data that are processed to be useful; provides answers to "who", "what", "where", and "when" questions

Knowledge: application of data and information; answers "how" questions

Understanding: appreciation of "why"

Wisdom: evaluated understanding.

2.4 The question of models

From a systems perspective, the best way of handling knowledge about any reality is to create a *model* of it. This is well known in the software commu-

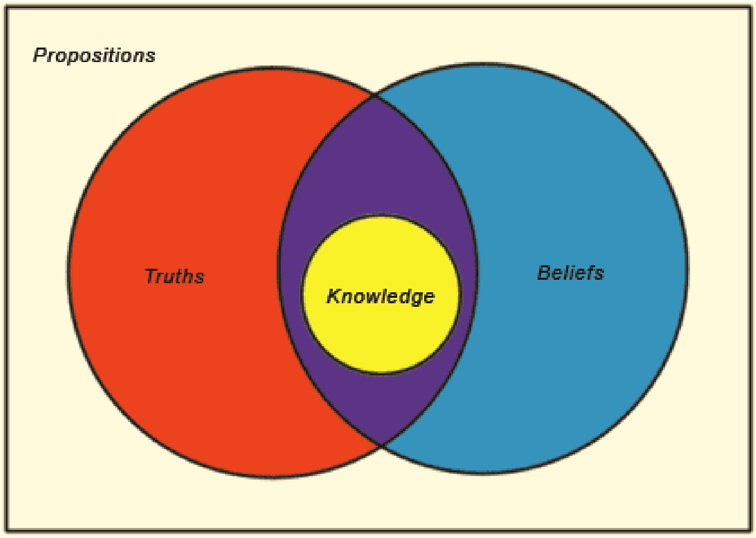

Figure 2.2: A classical definition says that Knowledge is what is both true and believed, though not all that is both true and believed counts as knowledge.

nity where models are becoming the touchstone of complex software systems engineering.

Model, like so many words in the English language, has a multitude of meanings depending on the context in which it is used. For example,in a social systems context Bellinger[2] has come to understand model to mean:

"A simplification of reality intended to promote understanding.´´

Is this just a transposition to "meaning" ?

2.5 Mental organisation

Last but not least, is the question of how an agent mind is organised to get profit from what it knows. The mind organisation is usually layered to provide a hierarchy of abstraction and spatiotemporal resolution. This is also the case of biological minds; mammal brains have a triple layered organisation (autonomic-reptilian, emotional-limbic and cognitive-cortex). The same can be said from a purely psychological perspective (consider for example Maslow's hierarchy of needs).

[2]http://www.systems-thinking.org/simulation/model.htm

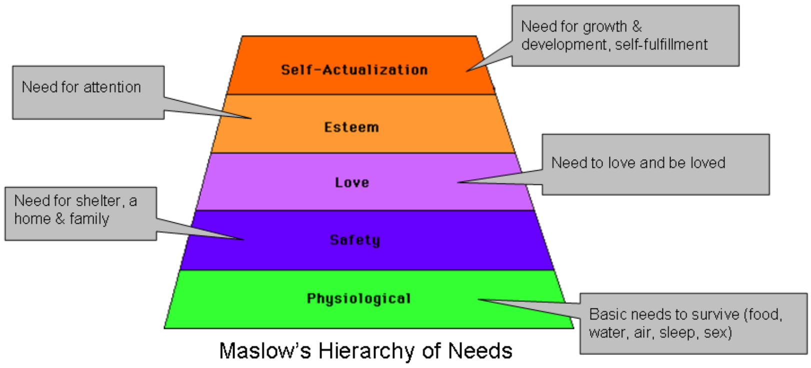

Figure 2.3: Maslow hierarchy of needs.

Chapter 3

General Systems Theory

3.1 Basic Notions

Let us think about what we understand by system, by considering it in relation to what surrounds it. If all possible entities form the *universe*, a *system* can be regarded as a part of it, which is considered isolated from the rest for its investigation. All which is not system is called *environment*. The different disciplines of science share this general understanding in particular ways, usually differentiated from each other in the criteria for separating the system from the universe.

The observer selects a system according to a set of main features which we shall call *traits*. They will be characterized by the observer through the values of a set of *quantities*. Sometimes, these values may be measured, being the quantities *physical*, such as length or mass. Other times quantities are *abstract*, and they cannot be measured, and their values are observed. The instants of time and the locations in space where quantities are observed constitute the *space-time resolution level*. The values of the quantities over a period of time constitutes the *activity* of the system.

In general, analyzing a system one may find that observed quantities are not sufficient to explain its behaviour. There must exist other quantities, which we shall call *internal*, which play a mediatory part. The observed quantities of the system will be called *external*. We shall call the set formed by all the values of the system quantities at a certain instant the *state* of the system, distinguishing between *internal state* and *external state*.

The main task of the observer is to explain the activity of a system. This will be accomplished by identifying patterns in the activity of the system. The quantities of the system may satisfy *time–invariant* relations, by which the values of some quantities may be expressed as function of others. The set of all time–invariant relations is the formal notion of *behaviour* of the system.

We may realize that the behaviour is due to the *properties* of the system. In other words, a system with different properties would exhibit a different behaviour. The set of all properties will be called the *organization* of the system.

3.2 Kinds of Behaviour and Organization

If we consider a particular system during a particular activity, we may observe that some of the time–invariant relations between its quantities may hold for a certain interval but eventually change. We shall say that these relations correspond to the *local* scope. Observing the same system during a different activity, we may observe that some of the time–invariant relations hold. If we again observe the system during a third activity, we could find that some of these relations would have changed. We would say they are of *relatively permanent*, for they hold for only some of the activities of the system. If we were to observe the system during an infinitely large number of activities, we would find that a particular set of relations would always hold between its quantities. They would be *permanent*. Accordingly, we can distinguish three kinds of behaviour (**?**, p.43):

- Permanent behaviour.
- Relatively permanent behaviour.
- Temporary behaviour.

The first may also be called *real behaviour*. The second, *known behaviour*. Temporary behaviour refers to the local scope, for it holds only for sections within a particular activity.

We may observe that permanent and relatively permanent behaviour may not be clearly distinguished from each other when analyzing systems. This is due to the impossibility to test the temporal persistence of relations beyond a restricted range of activities.

Let us return to the organization of the system. We may realize that the different behaviours derive from different kinds of properties. We may distinguish two main kinds, which we shall call *program* and *structure*. The temporary behaviour of a system derives from its program, which is the set of properties of local scope. Permanent and relatively permanent behaviours derive from the structure of the system, which we may in turn classify in *real structure* and *hypothetic structure*, (**?**, p.44), so that the causal relations are as follows:

$$\text{organization} \longrightarrow \text{behaviour}$$

$$\begin{aligned}
\text{real structure} &\longrightarrow \text{permanent behaviour} \\
\text{hypothetic structure} &\longrightarrow \text{relatively permanent behaviour} \\
\text{program} &\longrightarrow \text{temporary behaviour}
\end{aligned}$$

3.3 Defining Systems

In this section, we are going to present fundamental concepts of systems from two points of view. First, by considering its constant parts. Then, by considering the system from the point of view of its evolution in time. Finally, we shall enumerate the requirements for defining a system.

The study of a system as a whole may result difficult due to complexity or to non-observability of some parts. In order to analyze complex systems, the set of quantities is divided into groups, and each studied separately from the rest, as if it were a system on its own. Generically, each of these groups will be called *subsystem*. A subsystem is also called *element* of the system, to indicate that it is considered a component of it. There may be elements which share a group of quantities. This group is called *coupling* between the elements.

If we conceive the system in terms of its elements, we realize that it is formed by a set of elements, which we shall call *universe of discourse*, and a set of couplings. Elements and couplings are structured following a particular topology which we shall call *structure of universe of discourse and couplings* of the system, and abbreviate by *UC-structure*.

However, the system is not perfectly determined by its UC-structure, for the dynamic aspects of the system are unspecified. In order to complement the description of a system given by its UC-structure, it is necessary to analyze the evolution of the values of its quantities.

If we imagine a system at a certain point of its activity, we will find its quantities at certain values, forming its state. At the next instant of observation, the system will have evolved to a different state. We shall call this evolution a *state transition*. We may assume that, given the system at a certain state, not any transition is possible, or, in other words, that only a set of other states is reachable from the original one.

We may understand that each state is associated to a set of possible transitions. The set of all possible states of the system and their respective transitions form the *state–transition structure* of the system, abbreviated by *SC-structure*.

The necessary information for perfectly defining a system consists of its *primary traits* (**?**, p.52):

1. The set of external quantities together with the resolution level.
2. A given activity.
3. Permanent behaviour.
4. Real UC–structure.
5. Real ST–structure.

If a definition contains only some of the five primary traits, it results in a partial definition, that leaves aspects undetermined. In this case, we consider it defines a *class of systems* instead of a system in particular.

Example 3.1 (Quantities, Environment, UC and ST-structures)
Let us imagine we design a simple mechanical oscilator as the one in figure 3.1. When excited, the mass will describe harmonic motion at a frequency of $2\pi\sqrt{\frac{k}{m}}$. This frequency is fixed for constant values of the spring constant, k, and the mass, m, and it can therefore be used as a time reference for a larger system. This principle is used in mechanical watches and clocks.

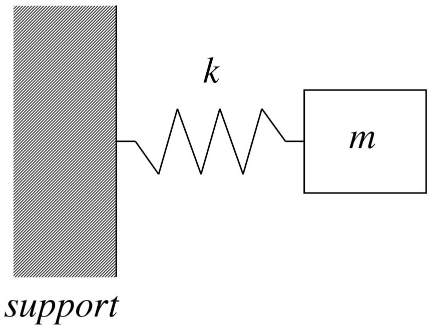

Figure 3.1: Mechanical Oscillator. A mass m, coupled to a spring of rigidity constant k, coupled to a fixed support.

UC-structure

We may distinguish three elements in the system, which define the *universe of discourse*. They are: mass, spring and support. The couplings between them are as follows: the mass transmits a force F to the spring. The spring, in turn, fixes the position of the mass, x, relative to the spring's equilibrium point. The spring transmits the force to the support, which returns an equal and opposed reaction force F_R to the spring. On the other hand, the support transmits force F to the environment, which returns a reaction force F_R.

The three elements and their couplings define the *structure of universe of discourse and couplings* of the system (*UC-structure*) shown in figure 3.2.

There is one coupling between system and environment which, for clarity, has not been shown. It is the action of the operator or device (part of the environment) that sets the initial conditions for the system.

ST-structure

In order to analyze the state–transition structure of the system, let us divide

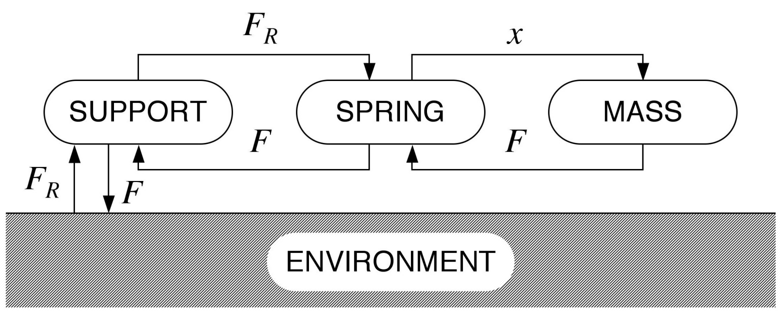

Figure 3.2: Oscillator UC-structure.

operation of the system in three regions, as shown in figure 3.3.

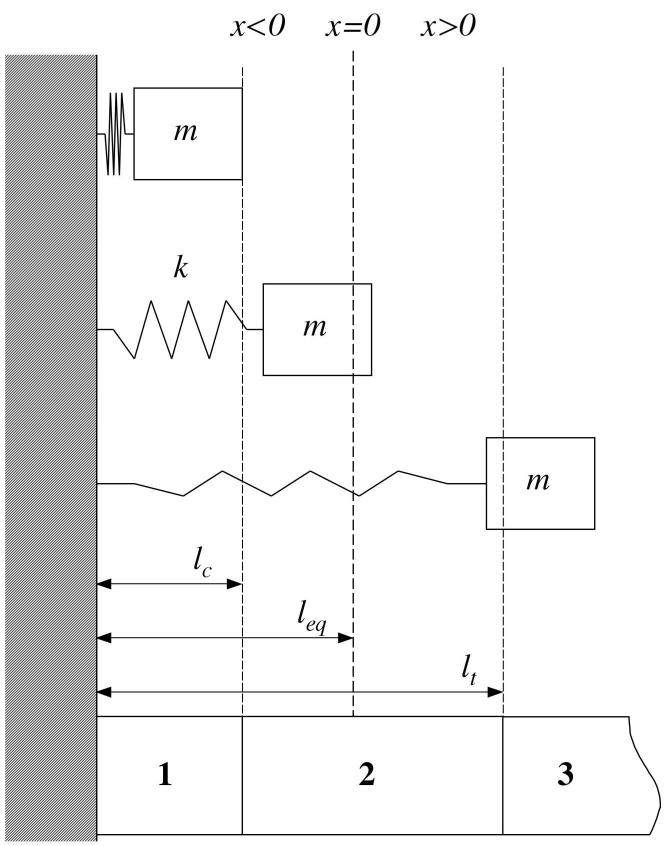

Figure 3.3: Regions of Operation of Oscillator. l_c– length at maximum compression, when the spires of the spring are adjacent to each other. l_{eq}– length at the equilibrium point of the spring, $x = 0$. l_t– length at the limit of elasticity of the spring.

In region 1, the spring admits no further compression, imposing the constraint $x = x_c$. In region 2, the spring follows Hooke's law, and therefore its force is proportional to the displacement from the equilibrium point, x. In region 3, the spring is over its limit of elasticity (at $x = x_t$) and can be assumed as a rigid rod, therefore imposing $x = 0$ and $\ddot{x} = 0$. Although it is not represented in the figure, if $x >> x_t$, the spring would break (region 4.)

These constraints define the states and transitions of the system in regions 1 and 3. Region 2 can be determined by state–space analysis. In this region, the system is described by:

$$m \cdot \ddot{x} + k \cdot x = 0$$

The dynamics of the system is given by this equation and a set of initial conditions. We can consider two state variables, x_1 and x_2, so that[1]:

$$\begin{aligned} x_1 &= x \\ x_2 &= \dot{x_1} \end{aligned}$$

The equation of the system can then be expressed in the classical form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$, where $\mathbf{x}$ is the state vector, A and B are matrices and $\mathbf{u}$ represents the input to the system:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We observe that the system is autonomous, ie: it has no B matrix and no inputs (u).

This system is represented in the phase plane by concentric ellipses (circles if suitable values of k and m are chosen) as shown in figure 3.4.[2] If the mass is set loose at a certain initial position, x_0, the state variables will follow the ellipse containing $x_1 = x_0$.

The frequency in which a trajectory is repeated is $f = 2\pi\sqrt{\frac{k}{m}}$, for the solution of the system equation is:

$$x = x_0 \cdot \sin\sqrt{\frac{k}{m}} \cdot t$$

[1]We might realize that the choosing of state variables is arbitrary. A different x_1 and x_2 could have been chosen leading to a different, but equivalent, analysis. These correspond to the classical analysis of this system.

[2]We realize that building phase plane representations (also called phase portrait) of systems might not be straightforward. Tools such as Matlab provide means for this. By hand, two methods are described in (**?**, pp.23-29).

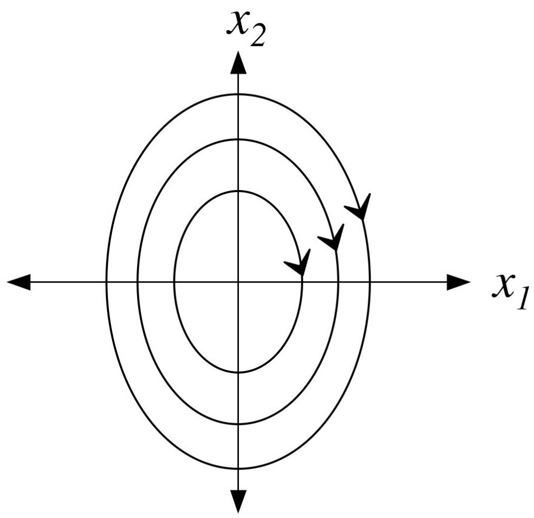

Figure 3.4: Oscillator Phase Portrait in Region 2.

However, this only holds for region 2. Globally, we may understand the phase portrait of the system will be as shown in figure 3.5. The system cannot exist in coloured regions.

To the left of x_c, the spring can be compressed no further. We shall assume that the support will absorb the energy that would push the mass further to the left, to a hypothetical position x_{fc}:[3]

$$\int_{x_c}^{x_{fc}} kx \cdot dx$$

To the right of x_t, the spring is a rigid rod. Any initial conditions x_0, such as points d, are equilibrium points.[4]

In region 2, between x_c and $-x_c$, the system follows Hooke's law and the trajectories are elliptical, as explained above. For initial conditions in $(-x_c, x_t)$, such as points a, b and c, the system follows the corresponding ellipse until the spring can be compressed no further. It then evolves toward the ellipse passing through x_t. This ellipse is, therefore, a *limit cycle*.

[3] This is an ideal case. In reality, the energy absorbed by the support, the environment or both would be between 0 and this value. It would be determined by the elasticity of the materials involved.

[4] We have simplified the problem in this region for clarity, by assuming a sudden pass from a spring constant k to a rigid rod. An intermediate region would exist in reality, in which plastic deformations of the spring would occur, by which the system would not recover its position at equilibrium, x_0 (ellipses would progressively shift to the right.) As a result, the dynamics of the system would grow more complex and the phase portrait would show phenomena out of the scope of this text.

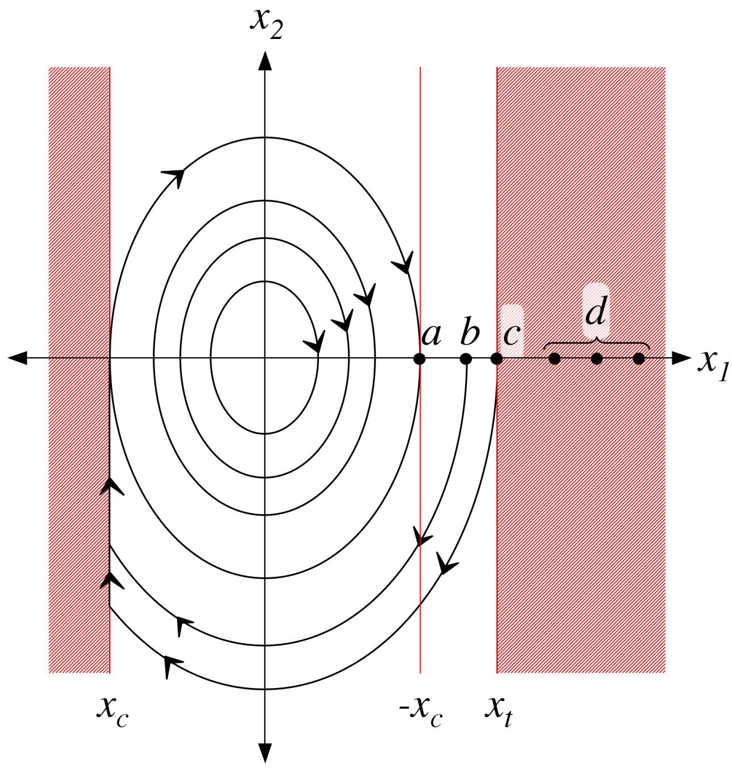

Figure 3.5: Oscillator Phase Portrait.

Let us consider a set of typical states within the continuum of the figure, as indicated in figure 3.6. The structure of states and transitions for this set is represented in figure 3.7.

As we have mentioned previously, the definition of a particular oscillator is completed by a set of initial conditions. The system portrayed in figures 3.2, 3.6 and 3.7, which stands for many possible initial conditions, stands, therefore, for many particular systems. We can say that these figures define a *class of systems*. In other words, they define a general system, which can exist in multiple, different forms.

In order to use our oscillator in a real mechanical device, we must define a starting point for its oscillation, in other words, a set of initial conditions.

These are the initial values for x_1 and x_2. Physically, initial position and speed of the mass. In figures 3.6 and 3.7, we have portrayed the system under different initial conditions assuming $x_2 = 0$. This is not necessary. For non–zero x_2, the system would follow the corresponding ellipse through (x_{01}, x_{02}). Mechanically, it is more complicated to build such device, and therefore we shall continue assuming $x_2 = 0$.

Let us now consider a particular oscillator, under specific initial conditions,

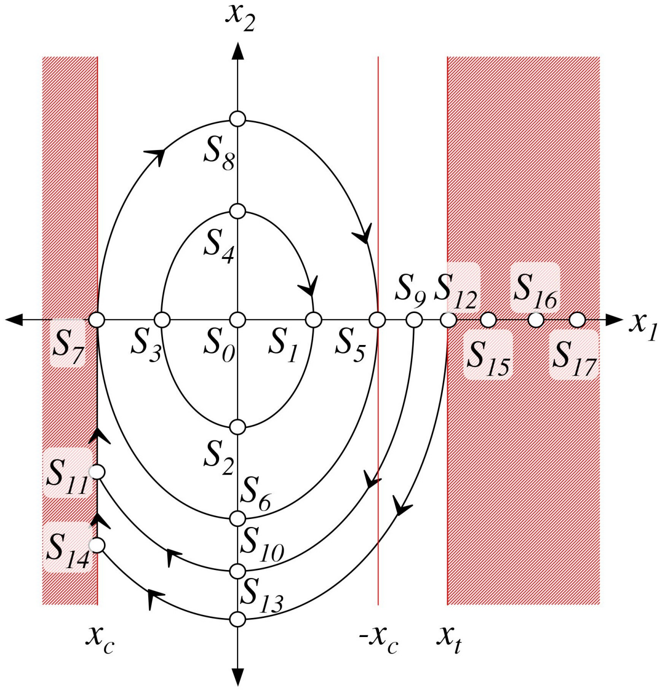

Figure 3.6: Oscillator Typical States.

$(x_0, 0)$ so that $x_0 \in (-x_c, x_t)$. Its phase portrait and ST–structure, subsets of figures 3.6 and 3.7, are shown in figure 3.8.

Quantities, State

In order to analyze the ST–structure of the system, we have used two state variables, x_1 and x_2, which have proved advantageous, allowing us to apply powerful methods of system modelling to provide a state–space description of the system. However, we might realize that our definition of *state*, in section 3.1, does not correspond to these chosen state variables. In fact, in our diagram of the structure of universe and couplings, figure **??**, they do not even appear. Let us see how both views, the (x_1, x_2) on one side, and the (x, F) on the other, come together.

Instead of adopting the point of view of the designer, we shall imagine that we are to analyze an oscillator which is already constructed and working. We are going to imagine that we chose to observe quantity x only (*external quantity*.)

The relation between x and the state variable is straightforward: $x_1 = x$. The external state of the system is therefore equal to x_1.[5]

[5]We also consider the quantities k, and m, although we shall not mention them explicitly for clarity, understood their values remain constant.

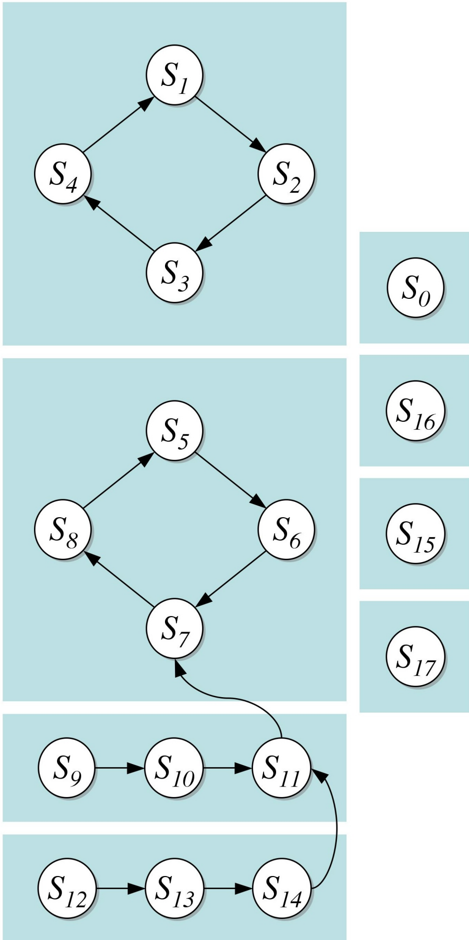

Figure 3.7: Oscillator ST–structure.

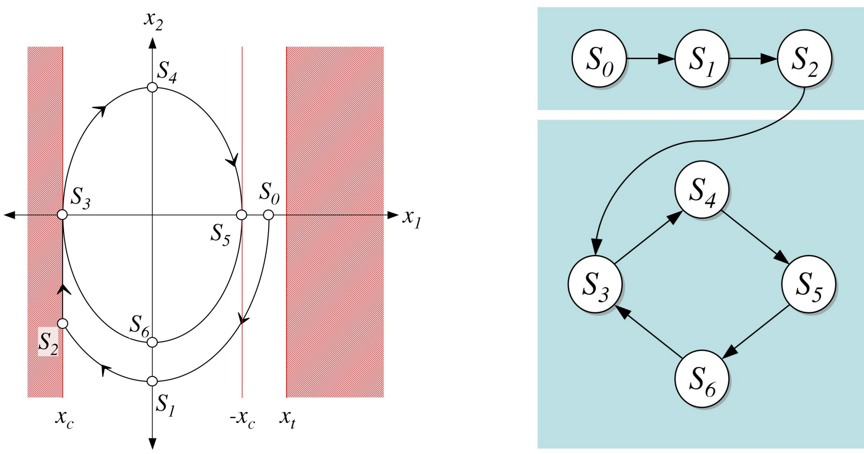

Figure 3.8: ST–structure of a Particular Oscillation.

We should find, however, that the external quantity x would not explain all the aspects of the system. Experimenting with the system, we would find that the part played by k and m would be undetermined. If we stroke the mass during its motion, we would not be able to explain the following values of x.

We could deduce from this that there would exist internal aspects of the system which would remain hidden from out observation. They would disappear if we would consider an *internal quantity* which would reflect in some way the inertia of the mass or its *momentum*. We could well consider the speed of the movement, $\dot{x}$, or its acceleration, $\ddot{x}$. We could then arrive to a set of *time–invariant relations* between its quantities, which would hold in the region of operation of the oscillator:

$$\begin{aligned} m \cdot \ddot{x} + k \cdot x &= 0 \\ x_c < \quad x \quad &< -x_c \end{aligned}$$

In conclusion, the state of the system would be given by (x_1, x_2'), where x_2' would stand for our chosen internal variable. Continuing the analysis from this point, we would arrive to a ST–structure which would be analogous to the above, in terms of x_2. In fact, there would always exist a transformation allowing to represent the system in terms of (x_1, x_2') or (x_1, x_2) indistinctively. $\square$

3.4 Classification of Systems

The concepts of quantity and structure introduced in the previous sections may lead to a classification of systems. We shall consider the short classification of systems illustrated in figure 3.4. The full classification is offered in figure 3.4, taken from (**?**, p.73).

Let us briefly explain the categories of systems. We have seen that quantities whose values are measurable are physical quantities, and the rest are abstract. Accordingly, systems formed by physical quantities are physical and the rest are abstract. If we focus on physical systems, we may distinguish two kinds. If quantities really exist, the system is real. If the quantities are only assumed, as in the case of systems which are modelled or imagined, the system is conceptual.

As to the number of quantities and structure a system has, we may distinguish two cases. First, that the system has a finite number of quantities and a finite structure. In this case, it would be a *bounded system*. Otherwise it would be an *unbounded system*. We may see that real physical systems are always bounded, while conceptual or abstract systems may be unbounded.

Finally, if we analyze the quantities of a system, we may find that they can be of two kinds. First, they can adopt values independently from the sys-

- Systems
  - physical
    - real → (∗)
    - conceptual
      - bounded → (∗)
      - unbounded → (∗)
  - abstract
    - bounded → (∗)
    - unbounded → (∗)
- (∗)
  - controlled
  - neutral

Figure 3.9: Short classification of systems, adapted from (**?**, p.73).

tem, given by the environment. In this case, they are *independent quantities*. Second, their values might depend on the values of other system quantities, and they are called *dependent quantities*. When analyzing real systems, discriminating between dependent and independent quantities is frequently impossible in practice. However, if dependent and independent quantities are known to the observer, the system is a *controlled system*. Otherwise it is a *neutral system*.

3.5 This Approach and GST

In this approach we shall be analyzing autonomous systems and perception from the background of GST introduced in the previous sections. This summary will be enhanced in the aspects required by each topic, introducing further concepts.

The analysis of autonomous system in this text, is written according to several assumptions that define its point of view. Specifically, we shall be considering a bounded, controlled, sequential system as the background. There are several reasons for adopting this perspective.

As to a bounded system, it has been considered because it is the most common case in engineering, and general in cognitive systems. Introducing concepts from the point of view of a controlled system permits explaining them unambiguously. It must be understood that in practice not all quantities and time–invariant relations of a system will be actually known, fact which makes impossible, among other aspects, separating dependent from independent quantities. In many cases, such as normally happens in engineering, the chosen quantities for designing or analyzing a system are few and sufficient to model a system, in which case we may assume the system is controlled. In other cases, the concepts introduced in this text will have to be understood

- Systems
  - physical
    - real → (1)
    - conceptual
      - bounded → (1)
      - unbounded → (1)
  - abstract
    - bounded → (1)
    - unbounded → (1)

- (1)
  - continuous → (2)
  - discrete → (2)
  - pulse → (2)
  - hybrid → (2)

- (2)
  - unique
    - controlled → (3)
    - neutral
  - repeated
    - controlled → (3)
    - neutral

- (3)
  - deterministic
    - combinational (memoryless)
    - sequential → (4)
  - probabilistic (stochastic)
    - simple
    - complex → (4)

- (4)
  - anticipatory (teleological)
  - nonanticipatory (physically realizable)

Figure 3.10: Classification of systems taken from (**?**, p.73).

associated to a probability distribution.

Finally, the point of view of a sequential system has been adopted for two reasons. First, because it is regarded as a more general case than a memoryless system. Second, because it is understood that highly autonomous systems are necessarily sequential.

Let us explain the difference between both kinds of system. A memoryless system produces a response which corresponds to the instantaneous stimulus. In some disciplines these systems are called combinational, because past history of either the system or the inputs do not influence its output.

However, sequential systems are those which use past values of their state and/or inputs to generate a specific output. This implies the existence of a certain memory element to store past values. We understand that a memoryless system is a special case of a sequential system, in which the capacity of the memory element tends to null. Thus, a memoryless system could be analyzed in terms of the concepts of this text by particularizing them for this context.

Systems may only exhibit highly autonomous behaviour if they can react appropriately to uncertainty in their environment. The only means to achieve this is by basing their operation on knowledge. The type and amount of knowledge on one side, and the way in which the system uses it on the other, determine the degree of autonomy of the system.

We would like to add, however, that some aspects of highly developed cognitive systems transcend the point of view stated above. For example, the knowledge of a system may be considered from two perspectives. First, with respect to the resources from which it is formed. Second, relatively to the information represented in it.

In the first case, it can be analyzed as a subsystem, formed by a finite set of quantities more or less related between themselves. These quantities could be, for example, the states (on/off) of the transistors in a RAM memory array. In this case, we can clearly see how the number of quantities is finite and equal to the number of transistors in the memory module.

In the second case, we might realize that the information expressed by the state of the resources (finite, as we have seen,) depends on the way it is interpreted. Returning to the previous example, we could see that the same state of transistors could eventually be interpreted as alphabetical characters, pixel luminance, pixel colours, etc. In general, we may assume that there exists an infinite information, knowledge being, in this sense, an unbounded system. We shall develop these considerations throughout the text.

Chapter 4

System Principles for Autonomy

We may return to the practical senses of autonomy introduced previously, and summarize them in two:

1. Minimum dependence of the system from its environment.

2. System cohesion.

We may assume, without loss of generality, that achieving system objectives is implicit in these two senses.

In the light of the previous sections, we may conclude that the effective autonomy of a system relies on the robustness of is purposive and structural directiveness. This means that the mechanisms of structural and purposive directiveness generate adequate organizations in the system. As we have seen, system organization results in a state–transition structure. The functional structure of the system, result of its mechanisms of directiveness, consists in subprograms which drive the system to its objectives. We have seen that different factors may lead to anomalous behaviour which are out of the specifications of the functional structure. This mismatch between functions (organization) and actual behaviour of the system stand for loss of efficiency in convergence, and eventually for divergence or system instability (understood as loss of system cohesion, see section 4.2.)

Intuitively, the degree of autonomy of a system stands for the scope of intensive and qualitative uncertainty in which the system is capable of preserving convergence and cohesion.

In the previous sections, we have analyzed different aspects of systems related to autonomy separately. We have discussed finality, objectives, directiveness and organization, and identified several points of relation with knowledge. In this section we shall try to build a unified vision, assuming knowledge and reconfigurability as a basis for system autonomy.

4.1 The Cognitive–Grounded System Model

The cognitive–grounded system model is a structure of concepts which serves as a background for explaining global aspects of the operations of autonomous cognitive systems. It will be abbreviated by CGSM.

The basic idea of the model is to conceive an autonomous system as a duality of a *cognitive system*, CS, and a *grounded system*, GS. As we have seen in the introduction to GST in chapter **??**, real quantities refer to those which actually exist, while conceptual quantities refer to those which are assumed. We may observe that both types of quantities are involved in the operation of an autonomous system based on knowledge. It is useful to separate both types of quantities. The conceptual operations and knowledge (quantities) form CS. The physical quantities and their dynamics constitute GS, see figure 4.1.

We may assume that there exists certain independence of operation between CS and GS, so that operations in CS may not necessarily cause a change in the state of GS. This, from the point of view of autonomy, provides the system with degrees of freedom of action, which stand for its capacity for reacting to environmental uncertainty. As we have seen throughout the text, knowledge is a factor in several aspects of directiveness and functions.

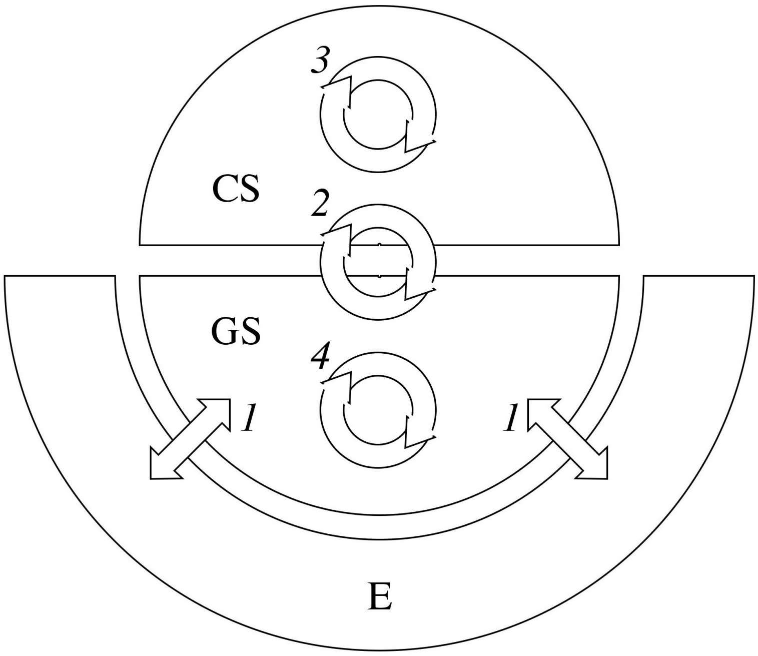

Figure 4.1: The Cognitive–Grounded System Model. CS–Cognitive System, GS–Grounded System, SE–System Environment. 1–interaction with the environment, 2–perception and grounding, 3–cognitive operation, 4–grounded operation.

4.1.1 Operation of CS and GS

Let us consider system operation following structural directiveness, in which there exists a defined functional structure. In these circumstances, the parts of the afferent, efferent, deliberative and integrative elements of the system functions involving conceptual quantities form part of CS.

The processes of purposive directiveness are intrinsically part of CS operation, as they involve symbolic representations of system, environment, objectives, and conceptual processes. The result is a conceptual representation, which is then grounded into GS.

As we have advanced, CS may operate separately from GS, although the separation might not be perfect. Some processes and knowledge tokens may take place in CS regardless the environment and the current state of GS. One example of this kind of operation in humans is abstract detached thought, or learning from experience.

Let us build a global notion of the operation of GS and CS. We may observe that the notions of finality, directiveness, objective structure and functional structure, commented in previous sections, explain the dynamics of the whole system (including GS and CS.) This stands for GS, and CS considered from the point of view of the physical quantities that serve for its *substrate*. However, CS must be analyzed separately in its conceptual dimension, as it constitutes a critical factor for system autonomy.

We may assume that CS consists in a set of quantities, which we shall generically call *cognitive quantities*. Some of the cognitive quantities may refer to the current system and its environment. We shall call them *instantiated quantities*. Other quantities may become instantiated quantities in the future, because they refer to possible scenarios of operation of the system. We shall call them *potentially instantiated quantities*. There may exist abstract quantities not referred either to system or environment, of value for cognitive operations, which we shall call *intrinsically cognitive quantities*.

As any system, CS admits analysis of its organization in terms of temporal scope, as it has been done in the previous sections, derived from the analysis of GST (chapter **??**.) However, it could be more illuminating to consider the properties of CS in terms of the conceptual value of its quantities. We may understand CS as a superorganization of more elementary organizations. We shall call *instantiated organization* to the properties associated to instantiated quantities. It represents and corresponds to the actual system, and therefore we may understand that this organization displays a conceptual image of the actual GS and its behaviour. It is a self–model. We shall therefore call *cognitive model of the system* to designate instantiated quantities and their organization.

We shall call *general knowledge* to potentially instantiated quantities and their organization. The organization of general knowledge represents the actual knowledge of the system. The organization of intrinsically cognitive quantities defines cognitive processes, buffer quantities, configurable registers, etc.

4.1.2 Cognitive–Grounded Coupling

We may understand that the dynamics of CS and GS are determined by a combination of explicit and implicit factors. The dynamics of the organization of the system as a whole, common to GS and the substrate of CS, is partially affected by the environment implicitly. On the other hand, the explicit output of perception are symbolic representations, which determine the explicit operation of the system.

We have already discussed functions in previous sections, which answer to the dynamics of GS. We may call the dynamics of CS from the conceptual point of view *cognitive operation*. We may distinguish between *instantiated operation*, which corresponds to the dynamics of the cognitive model of the system, and *non–instantiated operation*. This stands for the rest of CS, formed by potentially instantiated quantities and intrinsically cognitive quantities.

Cognitive operation is input by *perception*,[1] and its output is *grounding*. Perception produces informational content. Grounding, which we have already mentioned, stands for the realization of conceptual quantities. As we have mentioned above, the dynamics of CS is partially determined by implicit factors, which affect cognitive operation.

4.2 Autonomous Operation

Let us return to the two senses of system autonomy mentioned previously, adding finality explicitly in order to clarify the exposition:

1. Independence from the environment.
2. System cohesion.
3. Finality (directiveness towards objectives.)

Let us analyze the process by which the system loses cohesion and finality. We may assume that the uncertainty of the environment appears as perturbances to the system. Perturbances are represented as block–arrows of type 1 in figure 4.2. The program of the system, P in the figure, has certain capacity for compensating perturbances which we shall call *performance*. This stands for the actual efficacy of the local behaviour of the system. As we have mentioned, performance may eventually prove insufficient for compensating certain perturbances leading to what we shall call *program failure*.

The consequences of program failure may affect the hypothetical structure, HS in the figure. At this level, directiveness mechanisms can operate in order to reconfigure HS for correcting operation. This implies modifying algorithms or entire parts of the functional structure. We shall call this *adaptivity*.

[1]Perception will be analyzed in detail in part **??**.

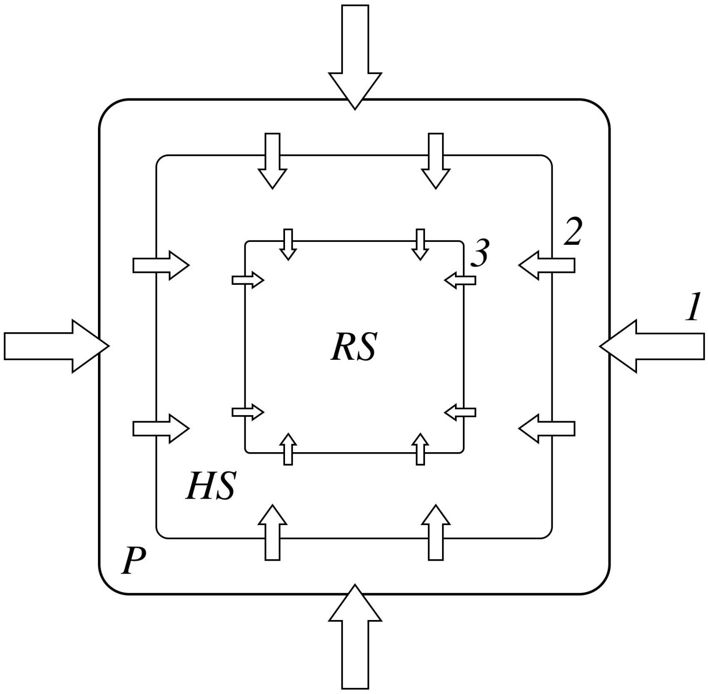

Figure 4.2: Propagation of Perturbances in an Autonomous System. The square represents an autonomous system. Intermediate squares represent the parts of the organization. RS–real structure, HS–hypothetical structure, P–program. 1–Perturbances to the system, 2–program failure, 3–structural failure.

We may realize from the previous sections on directiveness and objectives, that adaptivity may be *structural* in the case that it follows a function of the current structure, or *purposive*, in the case it is designed dynamically (we may assume that this implies symbolic operation.) In the event that adaptivity was not sufficient to recover convergence, this would lead to *structural failure*.

Structural failure would propagate to the real structure of the system, as represented by block–arrows of type 3 in the figure. Structural failure stands for loss of system cohesion. The real structure of the system, as we have mentioned previously, refers to the main, constitutive properties of the system. They represent intrinsic aspects of the system regarding its elements, its topology, and its finality. Structural failure leads to alterations of these properties, and therefore affect the system itself. The effect of structural failure on RS will be called *degradation*.

4.2.1 Performance

We may see that system autonomy is equal to performance and adaptivity. *Performance* is a notion of the capacity to maintain convergent local behaviour against perturbances. Returning to the notion of function as subprogram, developed in section **??**, we may realize that the performance of a function depends mainly on three aspects:

- Accuracy of the state specifications.
- Feasibility of the transition specifications.
- Completeness of the specification.

As we mentioned, the specification of a function may be based on undetermined state specifications, or define excessively demanding transitions. This could lead to anomalous behaviour, and eventually, to program failure.

In the example of figure **??** in page **??**, we may observe that states S5-S10 were not included in the specification, therefore leaving their corresponding transitions undetermined by the function. In the case that the system would leave the specified succession, the return would be at random. A more complete specification would specify transitions so that the system would tend to the main succession of states in case of anomalous behaviour, as the transitions S2-S7-S5-S3.

4.2.2 Adaptivity

We may see that *adaptivity* involves essentially two kinds processes: *objective configuration* and *functional decomposition*. As we have mentioned previously, functional decomposition is not uniquely defined, and different decompositions could be carried out for the same objective structure. We can consider that functional decomposition consists of the following phases:

- [Algorithm generation.]
- Algorithm selection.
- Grounding.

Algorithm generation is represented in brackets to express that it may take place independently from the other two. We could consider that adaptivity by functional decomposition may take place in four levels:

1. By maintaining a same level and re-grounding it, in order to improve its implementation for the actual scenario of operation.
2. By selecting a different algorithm from system knowledge, and then grounding it.
3. By generating a new algorithm dynamically and grounding it.

4. Finally, any of these alternatives, due to unfeasibility, may lead to redefinition of part of the functional structure. Major changes may imply changes in the objective structure. We shall call this *backpropagation of adaptivity*.

Adaptivity by objective generation may be explained similarly to functional decomposition, in terms of three phases:

- [Objective generation.]
- Objective selection.
- Functional decomposition.

As in the case of algorithm generation, objective generation can take place independently from the rest, as well as triggered by events in the other two.

We may conclude from this insight into autonomous behaviour, performance and adaptivity that system autonomy is defined by: performance, standing for efficient algorithms, capacity for grounding, capacity for algorithm generation and capacity for objective generation. We may observe that they refer to two aspects of the system: the substrate on which it is implemented, and the knowledge and abstract processes it carries out, which we have conceptualized as LS.

4.3 Principles of Autonomy

Returing to autonomy in the system as a whole, we may conclude that there exist a short collection of factors for autonomy which enable high degrees of adaptivity. We shall call them *principles for autonomy*, in order to emphasize that they constitute principles of design of artificial systems. We may distil them as follows:

Minimal Structure: The organization of the system may be divided in two parts, program and structure. According to the principle of minimal structure, the structure of the system must be minimized for higher autonomy, which stands for maximizing its program. This equals, firstly, to maximize system performance. Secondly, within the structure, its stands for minimizing the real and maximizing the hypothetical structure. This equals to providing maximum adaptivity.

Ideally, maximizing performance equals to minimizing program failure.[2] This consists in increasing the accuracy of the state and time specifications of functions, and their completeness. Increasing performance

[2]This may not be so in real systems, especially in the artificial case. Greater program equals to higher probability of errors, as well as need for greater magement resources.

means on one side that algorithm specifications are adapted to the system resources. On the other side, that system resources provide the required characteristics.

Maximizing performance equals to minimizing the cases of anomalous local behaviour. This means that the program of the system is capable of compensating for most cases of intensive and qualitative uncertainty.

Within the structure of the system, minimizing the real structure is equal to preserving system cohesion. Maximizing the hypothetical structure equals to increasing reconfigurability, a factor for system adaptivity.

Encapsulation: This principle stands for two main aspects. First, minimization of the couplings between elements. Second, for the construction of interfaces, in order to *encapsulate* heterogeneous elements.

We may realize that encapsulation contributes to autonomy in several ways. First, minimization of couplings is a factor for minimization of structure. Second, encapsulation favours reconfigurability. Third, encapsulation favours the accuracy of algorithms and knowledge.

Homogeneity: The principle of homogeneity is best understood if explained referring to the elements and couplings of the system. Homogeneity stands for similarity between system elements and couplings.

The UC–structure stands basically for the real and hypothetical structures, and therefore we understand that part of it is constant, and that the rest is reconfigurable. We have seen that the principle of minimal structure requires the constant part to be minimum. As to the remaining elements and couplings, similarity is a factor for interchangeability.

Similarity may not be possible. In this case, homogeneity may be increased by intermediate elements which enable indirect coupling of heterogeneous elements. These intermediate elements are generically called *interfaces*.

From the point of view of the LGSM, homogeneity may be considered in two other senses. First, as knowledge constituting a common resource for all the system. This means accessible to all elements of the system. Second, as to the elements of LS, for them having a common structure, as in the case of the elements of the UC–structure. In fact, reconfigurability in system elements stands for connectivity between elements of knowledge.

We may realize that homogeneity represents increasing system efficiency, in the sense of optimizing the use of its resources. Homogeneity of its elements, implying interchangeability, maximizes the available resources for grounding functions, as well as the possible ways of reconfigurating. Similarly, homogeneity of knowledge maximizes its potential scope of use and power of representation.

Isotropy of knowledge: stands for the quality of presenting coherent meanings under different contexts of interpretation. We may realize that system knowledge is generated within a particular scenario. This equals specific functional and objective structures which partially define the knowledge acquired. We shall call *biasing* to this partial definition. We may understand that different conditions of operation produce different biasing. Perfect isotropy means that the content of knowledge is independent of biasing; lower degrees of isotropy stand for reusability of knowledge in different contexts from the one in which it was created.

4.3.1 Ideally Autonomous Systems

The notion of an *ideally autonomous system*, *IAS*, will stand for a conceptualization of a system in which the principles of autonomy are optimally realized.

We might realize that a system which is *absolutely* autonomous is impossible, if we understand it as capable of achieving its objectives in any circumstances, under total uncertainty. We may realize that this could mean impossible reaction speeds, instantaneous characterization of uncertainty, decision taking, reconfiguration and action. These requirements could only be achieved by a system having infinite resources and infinite knowledge. In this case any perturbance could be anticipated, characterized and compensated for.

Let us analyze the case of a non–infinite ideal system in which performance of resources and knowledge are maximal, which we shall call *ideally autonomous system*.

We must remark that the principles of autonomy stand for system optimization for autonomy. Nevertheless, there exist two additional criteria, in relation to the paradigm of absolute autonomy. It follows that system knowledge and resources constitute a factor for autonomy per se. Therefore, increasing both aspects in accordance to the principles contributes to autonomy.

As we have seen, there exist two kinds of uncertainty: qualitative and intensive. We may understand, grossly, that intensive uncertainty is compensated by the performance of the system, and qualitative uncertainty by adaptivity. We may regard this as an intuitive, general understanding. In this sense, increasing resources would mainly contribute to performance, while increased knowledge would mainly contribute to adaptivity.

We might realize that the qualitative uncertainty may only be compensated for by mechanisms based on general knowledge. Qualitative uncertainty stands for the occurrence of unexpected events, or unknown events which we shall call *qualitative events*. This type of uncertainty requires a dynamic response of the system. More knowledge implies availability of a broader variety of models for explaining qualitative events, and therefore, increased efficacy in adaptivity.

Chapter 5

Exemplar Autonomous Systems

This chapter needs completion by all the participants and more.

5.1 Electric stairs

Electric stairs that can be found in most subway stations, airports, or malls are a very simple example of system, but they present a behaviour, based on responding to external stimulus, quite appropriate for our area of autonomous systems. There are many types of electric stairs according to their behaviour. Let us suppose an electrical stair that works as follows:

- Moves upstairs.
- Starts working and start a timer when someone arrives.
- Resets timer when working and someone arrives.
- Stops working when timer reaches *time-out*.

Figure 5.1: default

5.1.1 UC-structure

We are going to analyse the system according to TGS taking the broader scope to simplify: that one referring to functionality. According to this, the universe of discourse is composed out of three elements:

- sensor
- timer
- mechanism

The quantity considered in the sensor is the electronic value of its electronic output signal s that can be '0' if there is no one at the entrance of the stairs and '1' if there is. We consider two quantities for the timer, the time it indicates t and the electronic signal of time-out t_{out}. The mechanism can be reduced to a quantity named *movement* (m) whose possible values 1, 0 indicate if the stairs are moving or not respectively.

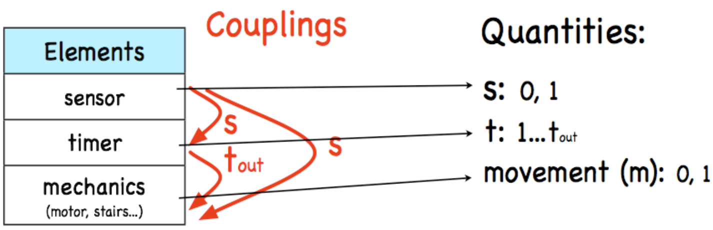

Figure 5.2: default

The couplings between these three elements are as follows: the sensor sends a signal s to the timer and to the electronics controlling the mechanism when detects the presence of someone entering the stairs because of his/her weight. This signal causes the mechanism to start moving and the timer to start or restart if it is already on. The timer sends a signal tout to the mechanism when it reaches a time-out value. The mechanism and the timer then stop. We suppose that the mechanism has no couplings with the environment, obviating friction, user's weight etc. The figure below illustrates the uc-structure described.

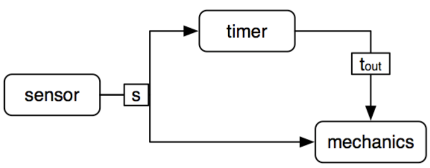

Figure 5.3: default

5.1.2 ST-structure

Thanks to the discrete nature of the system in the scope selected for its study, the state-transition structure can be explained with the following diagram:

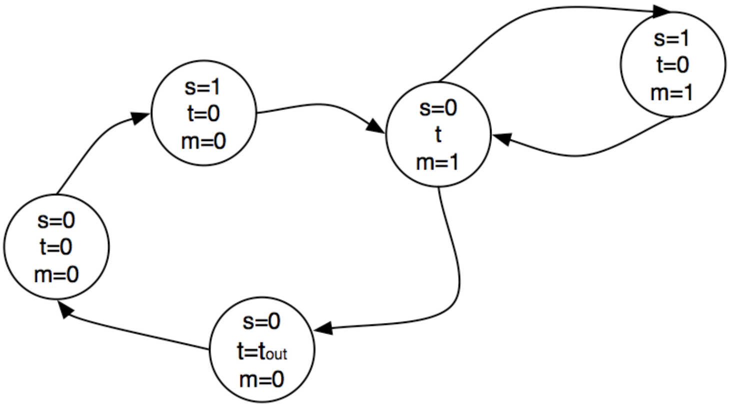

Figure 5.4: default

5.2 Discrete automation

Pedro Conejo analyzed a parking barrier control system.

Adolfo Hernando analyzed a lift.

5.3 Mobile robots

Juan Escasany proposed analyzing a two-whisker robot for moving on surfaces without falling.

5.4 Chess player

Adolfo Yela proposed analyzing an 'intelligent system' such as a chess-playing system.

5.5 Escherichia Coli

A *model organism* is a species that is extensively studied to understand particular biological phenomena, with the expectation that discoveries made in the organism model will provide insight into the workings of other organisms. In particular, model organisms are widely used to explore potential causes and treatments for human disease when experimentation on humans would be unfeasible or unethical. This strategy is made possible by the common descent of all living organisms, and the conservation of metabolic and developmental pathways and genetic material over the course of evolution.

Some well known Examples are the **lambda phage** (*Enterobacteria phage* λ), the **Escherichia coli**, the **fruit fly** (*Drosophila Melanogaster*), the **aplysia slug**[1] or the **rat** (*Rattus norvegicus*).

Figure 5.5: A nice exemplar of Aplysia Californica (a sea slug). They are particularly valuable in neuroscience because they have a relatively simple nervous system and 'brain' with extremely large nerve cells which can be individually mapped.

Escherichia coli , usually abbreviated to E. coli, (coli is Latin for "of the colon") discovered by Theodor Escherich, a German pediatrician and bacteriologist, is one of the main species of bacteria that live in the lower intestines of mammals. They are collectively known as gut flora and are necessary for the well being of the mammal (but in some cases, special strains may produce illness).

John Brown's 'What the Heck is an E. coli?'

5.6 Robot Control Testbed (RCT)

5.7 Process Control Testbed (RCT)

David García proposed analyzing a continuous plant control system.

[1] Very well known in neuroscience.

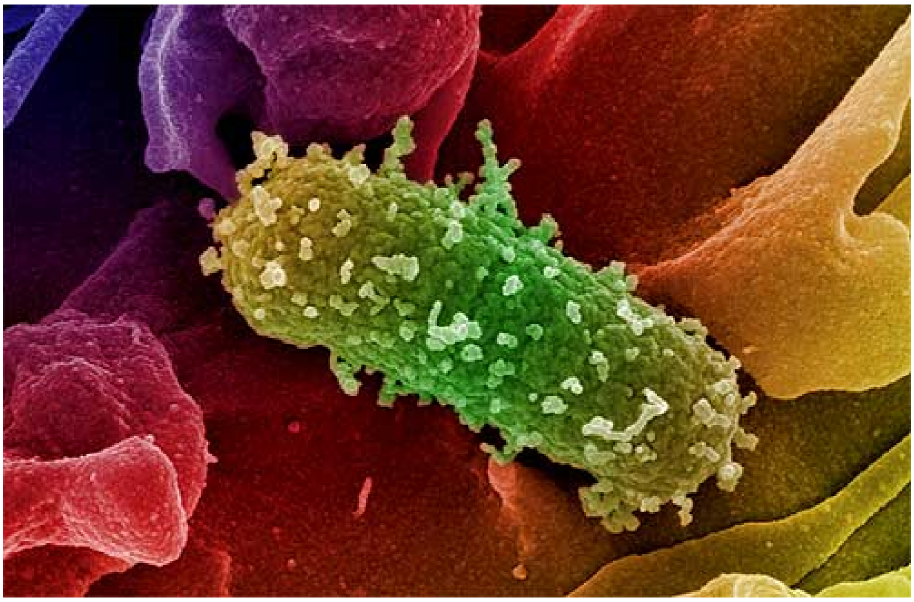

Figure 5.6: A nice exemplar of autonomy.

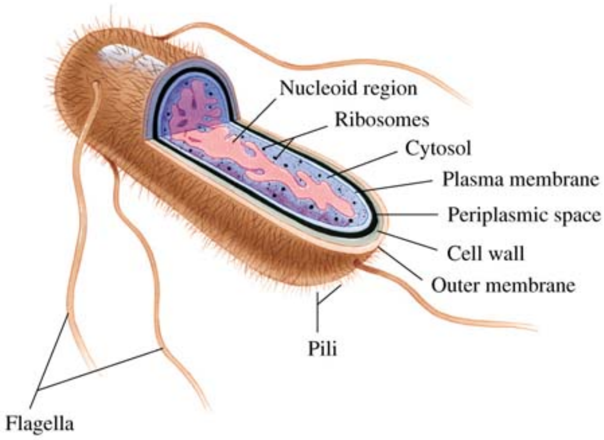

Figure 5.7: *E.Coli* inner structure.

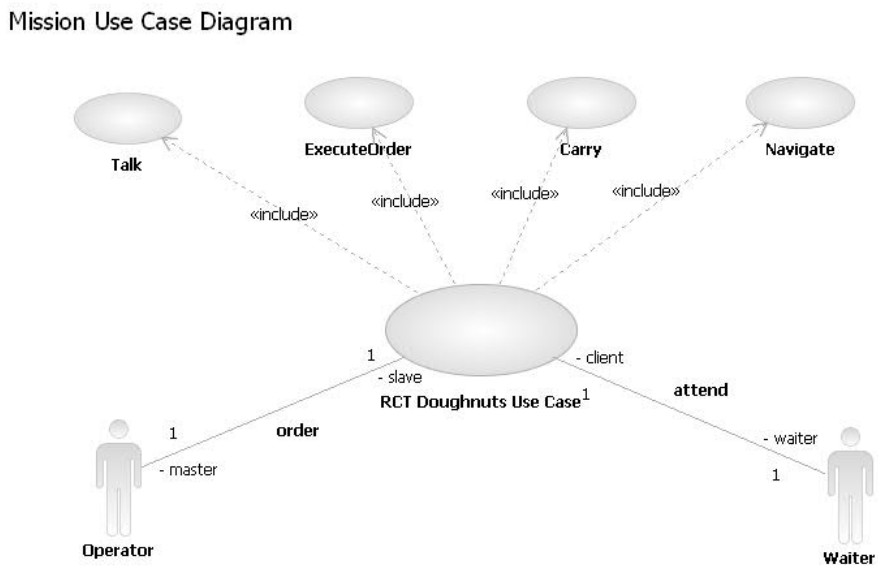

Figure 5.8: An use case of the RCT (Robot Control Testbed).

5.8 Personal Digital Assistant (PDA)

5.9 Automatic Driver Assistant (ADA)

5.10 Electrical Generation/Distribution Network

Chapter 6

Minutes of the seminar

6.1 Seminar execution

The seminar started Saturday, November 11th at noon. The attendants were:

- Pedro Conejo
- Juan Escasany
- Carlos García joined the group after lunch until before dinner on Sat. 11th.
- David García
- Jaime Gómez
- Carlos Hernández
- Adolfo Hernando
- Ignacio López (lecturer)
- Ricardo Sanz
- Adolfo Yela

After a lunch break, it was resumed until 18:15. During this first day, the main concepts of the formulation of the General Systems Theory proposed by George J. Klir were introduced. The most relevant are listed following:

- system, environment
- quantities
- time–invariant relations
- activity

- properties
- behaviour
- organization
- program
- structure
- ...

On Sunday 12th the session started at 11h[1]. After a brief summary of the concepts introduced the previous day, new content was presented. First, it was shown how the autonomy of systems can be related to their organizational aspects: program, hypothetic and real structures. Then, three alternative (possibly complementary) approaches were pointed out as ways to formalize the theory:

- fractals,
- gnomonic growth and
- category theory[2].

The session was followed by a brief exercise by each participant, applying the concepts of GST. Carlos Hernández analyzed an electric stair lift, Juan Escasany proposed analyzing a two-whisker robot for moving on surfaces without falling. Pedro Conejo analyzed a parking barrier control system. David García proposed analyzing a continuous plant control system. Adolfo Hernando analyzed a lift. Adolfo Yela proposed analyzing an 'intelligent system' such as a chess-playing system.

Ricardo didn't propose a system because he had not analised it properly the GST way. This system is *Escherichia Coli* (as is decribed in the previous chapter).

This final exposition gave rise to debate referring to the concept of intelligence. It also led to introducing the notions of *real quantities* and *conceptual quantities* of GST, which had not been mentioned the day before. This was a slight mention to the topic of mind–body problem, which was scheduled as part of the Seminar, but finally dropped from explicit, dedicated discussion.

6.2 Recollection of Technical Comments

Abstract/Physical Quantities: Unclear distinction between abstract and physical quantities. The frontier is arguable: measurable/non–measurable.

[1] Had to go far for a breakfast !
[2] Some references for these topics would be interesting.

Properties/Quantities: Unclear meaning of 'property.' After some debate with some of the attendants it was partially cleared: properties are the cause of behaviour (as Klir says in p. 43: "If the system exhibits a particular behavior, it must possess, as mentioned previously, certain properties producing the behaviour. These properties will be called the *organization* of the system.") The first understanding led to a more or less diffuse meaning, but evolved towards a more refined conception. Some properties can be expressed by quantities (eg. mass.)

Scalability: Doubts were expressed during the seminar and a priori (esp. Manolo Rodríguez, who did not attend) about the suitableness of the GST framework for analyzing complex systems. Ignacio did not comment on the topic, only indicated that Klir offered simplification mechanisms in the book. Ignacio deliberately avoided further exploring this topic, considering it was better to consolidate concepts prior to further discussion.

Applicability: The topics explained during the morning of the second day seemed to clear some of the doubts about the potential of GST for designing and analyzing autonomous systems. Nevertheless, everybody was conscious that it required formalization. Ignacio suggested gnomonic growth, category theory and fractals as possible ways for achieving this and explained why, helped by Ricardo.

6.3 Recollection of Organizational/Group Comments

Attendants must study course documentation in advance: This will make the seminar much more productive. While initial examination seems not very adequate, we compromise ourselves to do it better the next time.

Possibility of seminars every 6 months: Spring and autumn.

Possible technical seminar on fractals by Juan Escasany: When?

6.4 Conclusions and Future Work

The main conclusion from the seminar concerning the focus on GST is that the level of precision and detail that this method provides is extremely interesting from the point of view of systems formalisation.

However this same carefulness in the details produces the impresssion of being a daunting —and possible usseless[3]— task when applied to a complex system.

However, the formalisation road is the road to follow if we look back at the history of science. If GST is not suitable enough for modelling autonomous

[3] Due to the difficulty of doing it properly.

systems we must strive to find an extension or replacement for it. But the clarity about where are the roots of the separation system/world should be now well planted in our minds.

Other references

Gadomski, Adam Maria, Information, Preferences and Knowledge, An Interesting Evolution in Thought

Sharma, Nikhil, The Origin of the Data Information Knowledge Wisdom Hierarchy

Mpitsos G.J., S. Soinila. ”In search of a unifying theory of biological organization: What does the motor system of a sea slug tell us about human motor integration?” In: Variability and Motor Control, J.M. Newell and D. Corcos (Eds), Human Kinetics, Champaign (1993) pp 225-290.

Links

http://www.seaslugforum.net/general.cfm

http://www.systems-thinking.org/dikw/dikw.htm

Title: Towards a Theory of General Autonomous Systems
Subtitle: ASLab Seminars on Autonomous Systems I
Author: Ignacio López and Ricardo Sanz

Date: November 11-12, 2006
Reference: R-2006-009 v 0.4 Draft

URL: http://www.aslab.org/public/documents/R-2006-009.pdf

Autonomous Systems Laboratory

UNIVERSIDAD POLITÉCNICA DE MADRID
C/JOSÉ GUTIÉRREZ ABASCAL, 2
MADRID 28006 (SPAIN)