Semiotic Synthesis and Semionic Networks

Ricardo R. Gudwin
DCA-FEEC-UNICAMP
gudwin@dca.fee.unicamp.br

ABSTRACT: In this paper we discuss a set of conceptual issues regarding the synthesis of semiotic processes within artificial systems, and its technological offspring, the semionic networks. We start by putting in context what is this thing - semiotic synthesis, and how and why it differs from semiotic analysis. We follow proposing the concept of a "semion", as both an elementary entity with semiotic processing abilities, and a container of knowledge, explaining its structure and dynamics. At the end, we address the interaction of semions within semionic networks, and its use with the purpose of building different architectures for artificial minds.

1 Introduction

The human mind fascinated many generations of philosophers, mathematicians and scientists during centuries. A whole branch of philosophy was dedicated to this main theme - the philosophy of mind - on which great philosophers, like Aristotle, Plato, Locke and Kant have contributed with many different theories trying to explain the mechanisms supporting the behavior of the human mind. Partially drinking from these sources, a totally independent area of study was born during the 19^{th} century - ***Semiotics*** - the study of signs - the study of the processes of signification and representation both in man and in nature (NOTH 1995). There is an obvious connection between semiotics and the philosophy of mind, in the sense that the use of signs are the basis for the processes of cognition and communication, which are among the main thought processes occurring in the mind. To be precise, the study of sign processes is documented in literature since the works of Plato and Aristotle (NOTH 1998), but Semiotics, as an independent area of research was organized and structured only with the work of Charles S. Peirce, an american philosopher, during the middle of the 19^{th} century (PEIRCE 1960). Many others have contributed since then to the development of semiotics, like Saussure, Hjelmslev, Jakobson, Greimas and Morris - more recently Eco, Sebeok, Merrell and others (NOTH 1996; MORRIS 1947; MORRIS 1964; MORRIS 1971; SEBEOK 1997).

Until recently, the study of semiotics was restricted to semiotic analysis, i.e., the use of the theory of semiotics to the analysis of sign processes in different activities of living beings. With the development of computer sciences - specifically the areas of artificial intelligence (or as we prefer - intelligent systems), there was open an opportunity of experimenting the opposite way - that is - the way of synthesis - the way of artificially creating semiotic processes - semiotic beings. For the first time, we were not mere expectants of the miracles of nature, but now we were able to fully synthesize devices where semiotic processes do occur. This was the beginning of "Computational Semiotics" - an interdisciplinary area of research dedicated to the creation of intelligent systems supported by the theory of semiotics. In other words - instead of merely analyzing natural devices (where by natural devices we mean since biological organisms up to human beings) and speculating about the sign processes happening there, our aim is to fully design these sign processes and further create instance devices where those sign processes do appear. Those sign processes will be the texture on which we expect to build thought processes, thoughts and intelligent behavior.

So, Semiotic Synthesis refers to the attempt of emulating the semiosis cycle within a digital computer. Among other things, this is done aiming for the construction of autonomous intelligent systems able to perform intelligent behavior, what includes perception, world modeling, value judgment and behavior generation. There is an implicit claim here that most part of intelligent behavior should be due to semiotic processing within autonomous systems, in the sense that an intelligent system should be comparable to a semiotic system.

Since Computational Semiotics efforts started to appear, the mathematical modeling of such semiotic systems has being the target for a group of researchers studying the interactions encountered between semiotics and intelligent systems. The key issue on this study is the discovery of elementary, or minimum units of intelligence, and their relation to semiotics. Some attempts have been made aiming for the determination of such elementary units of intelligence, i.e., a minimum set of operators that would be responsible for building intelligent behavior within intelligent systems. These attempts include Albus' outline for a theory of intelligence (ALBUS 1991) and Meystel's GFACS algorithm (MEYSTEL 1995). In our contribution to the Computational Semiotics field, we try to depict the basic elements composing an intelligent system, in terms of its semiotic understanding. We do this by proposing the concept of a "semion", or a basic conceptual entity able to carry on semiotic processing, and at the same time able to store knowledge. We derive a whole taxonomy of semions, each one adequate to represent a different kind of knowledge. Semions from different types and behaviors, are mathematically described, and used as atomic components for an intelligent system. They are at the same time, containers of information and active agents on the processing of such information.

Semiotic Synthesis and Semionic Networks are though a correlated pair of a theory and a practice, both enrolling each other into the target of being a kernel for the design and construction of artificial minds. They are our contribution in order to foster the computational semiotics field of research.

We will see that modeling semiotic synthesis is not an easy task. In such an endeavor, many questions are still to be answered. Some of them very basic questions like: Does the computer (or, more formally, the Touring machine) have sufficient power to be an instance of a semiotic process ? We still don't have definitive answers to this question. Some researchers claim that living organisms embed in themselves some kinds of characteristics that make them complex systems, not able to being modeled by means of machines. An example of such a claim can be found on the work of Rosen (ROSEN, 1985; ROSEN, 1991). This is why we do not call our systems "semiotic machines" but "semiotic devices". Our basic claim here is that even though a Turing machine is not enough to build up a semiotic device by itself, we expect that a Turing machine (a computer), with the right software, if not able to fully "emulate" a semiotic process, will be able to "simulate" it up to an acceptable level of quality. But up to this point, this is just a hypothesis which, if proofed wrong, may lead to the creation of a new kind of device - the semiotic device. If it is the case, this semiotic device will be superior in power than today's computer. As a first bet, though, we expect that, using the right software, we will be able to synthesize the semiotic device using a standard computer.

There are other difficult points to be analyzed. For example, the notion of purposive action (Aristotle's final cause), or thirdness (as in Peirce), when embedded into a machine (or semiotic device). Is it possible ? Can a machine have a purpose by itself ? A purpose that it uses to guide its actions ? Or it only works as a sequence of mechanical behaviors ? These questions connect directly to the notion of autopoiesis (MATURANA & VARELA 1980). Is it possible to create an artificial system that performs autopoiesis ?

We still don't have final answers to those questions. But while those answers do not appear, other developments are appearing, making computational semiotics alive and, we hope, paving the way to the construction of a new generation of intelligent systems.

2 Semiotic Synthesis : The Basic Foundations

In this session, we set up the basic foundations in order to describe what is this thing - semiotic synthesis. Our goal is just to build up a generic scenario, in which semiotic synthesis is going to be discussed. We do this, in the hope to get clues on how semiotic processes really happen in a natural interpreter. This is absolutely necessary if we want to implement artificial systems performing semiotic behavior. Our framework will need to allow the implementation of a computational version of semiotic processes.

The terminology we are going to use is very much related to standard semiotic terminology, basically Peircean semiotics terminology, but also including elements from non-Peircean semiotics. So, we are going to talk about signs, objects and interpretants, but also interpreters, signals and other terms that, besides sometimes used by Peirce, are not exactly used in its Peircean meaning (NOTH 1995). We would also be making extrapolations in order to include other terms, like icons, indexes, symbols, tones, tokens, types, rheme, dicent, arguments, firstness, secondness, thirdness, etc (PEIRCE 1960). Sometimes, though, we will have to sacrifice the original generality of those terms. So, the Peircean purists don't blame us if I am not using these terms with their original acception, here. Our aim is to set up a scenario to discuss semiotic synthesis. As a first trial, we will need to let some considerations aside. We expect to address them in the future, making the necessary correction that would emerge from the critics of our model. We are trying to create a computational understanding here of what are signs, objects, interpretants and interpreters. Sometimes, this understanding is a little bit disappointing. Some people will not agree with our claims. The point is ... we can not create so generic definitions that would encapsulate mechanisms that we still don't know how they work. So, our definitions will appear a little bit limited, and sometimes naive. But this is only our first exercise, so, ... , we don't need to be so concerned with this limitation. Let's just see if the whole thing make sense to you !

2.1 Interpreters and Representation Spaces

Our framework description starts with the notion of an Interpreter and its Representation Space. These notions can be illustrated in figure 1.

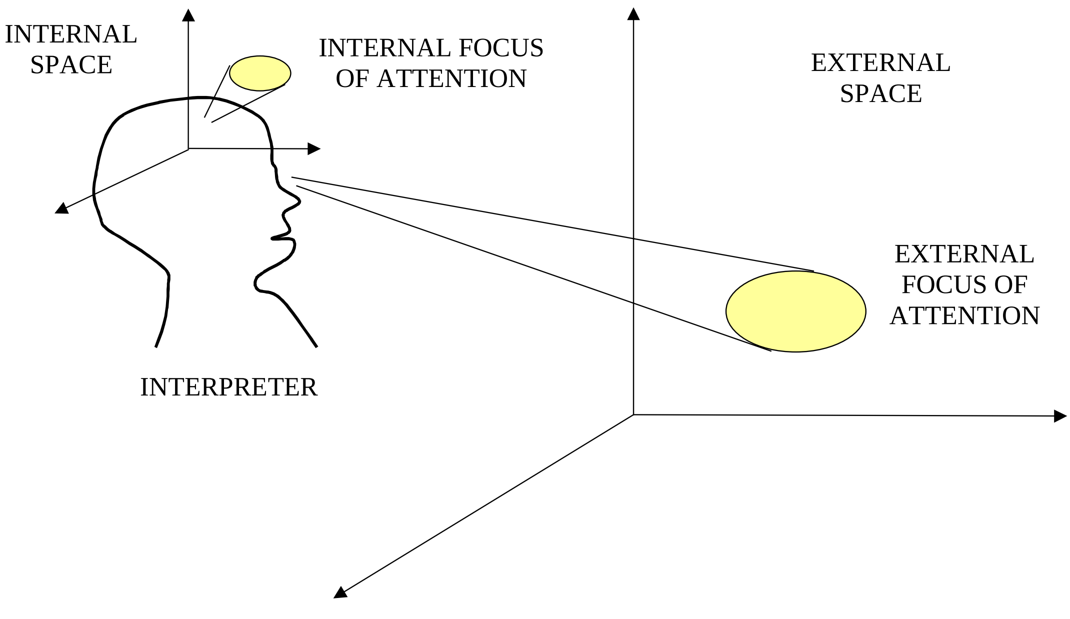

Figure 1 - Representation Spaces

The interpreter is our semiotic being - the system that will be holding the sign-processes. In a pure relational understanding of the semiosis process, if we were followers of a pure Peircean tradition, maybe the notion of an interpreter is disposable, being surpassed by the notion of interpretant. But in our case, given that it is exactly what we are interested in synthesize, it will be absolutely a part of our core concepts. So, the interpreter is our main object of study and our first concept being introduced.

The interpreter is immersed into an environment, where he is able to get information by means of sensors, and actuate by means of actuators. The interpreter is also able to actuate internally to itself, changing its internal states and configuration, if necessary. The external world surrounding the interpreter takes place in what we call an External Space. The external space is not equal to the environment, but only the space in which the environment happens. For now, let's just forget the real world, and cope only with this notion of external space. The standard mechanism by which the interpreter gets information from environment is through its external focus of attention, which selects a region of the external space.

Besides the external space, there is also an internal space, localized within the interpreter, with his own internal focus of attention too, that is a region of internal space that is selected for some purpose.

With those two concepts, we have some points to play with. First, like we show in figure 2, external spaces can be shared by multiple interpreters, but internal spaces are exclusive to an interpreter in itself. Also, an interpreter is able to control both its internal and external focus of attention.

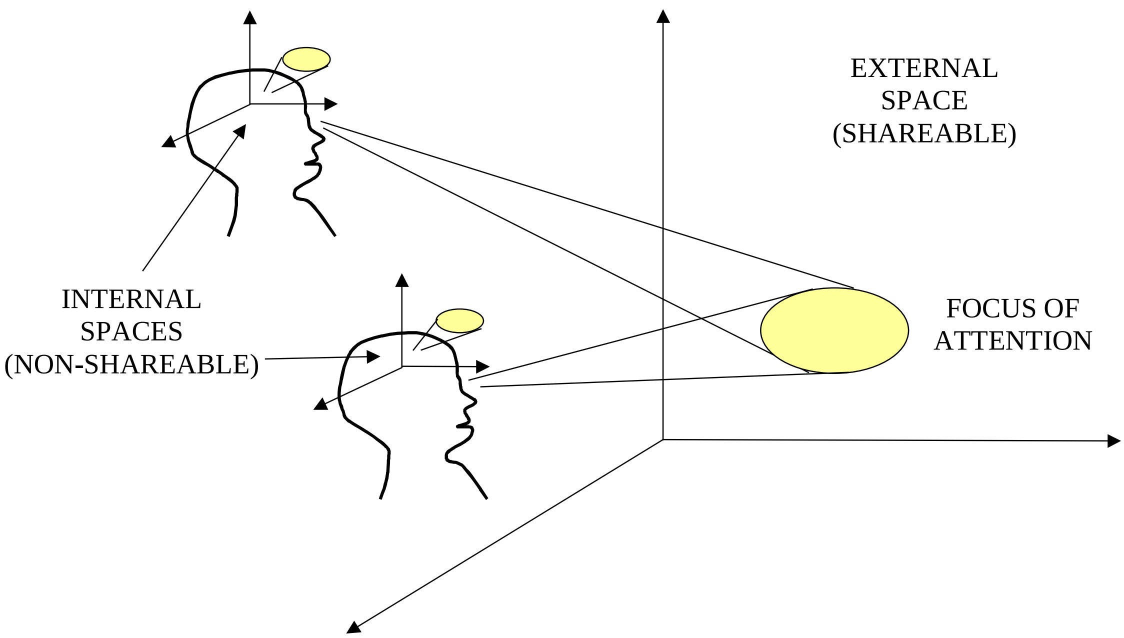

Figure 2 - Shareable and Non-Shareable Spaces

But all of this is meaningless if there is nothing at the representation spaces (internal and external). In order to make these concepts useful we introduce, then, the notion of **signal field**. This notion is derived from field theory, and basically is a function $\psi(x,y,z,t)$ that to each point in space and time, attach a unique informational value. The notion of a signal field is currently an abstraction and a model for things that may exist in the real world, in a non-objective and measurable interpretation of this real world (which says that the world is not crowded of objects, as preached by objectivist theories, but only that we can collect measures with sensors - signals - from this same world). The signal field relates to the union of all possible measurements able to be collected from this real world. For example, if the space and time hosting the signal field is our real world (or whatever we feel it is - just to avoid some philosophical debate around it), we may consider the space to be 3-dimensional, and the value of the signal field to be a state associated to a point (x,y,z), at time t. This state may be related to an energy function, or a wave function, or any kind of physical model we choose for describing reality. If, instead using the real world to host the signal field, we choose to use another abstraction, e.g. the notion of a computer memory, then the space will be the computer memory, the location will be the address and the value of the signal field will be the informational bytes stored at each address, for each moment of time. The main point behind the notion of a signal field is the notion that its value conveys some sort of information, which will be used in order to build the notion of a sign. Our conception of a sign is build up on the notion of a signal that by itself is build up over the notion of a signal field.

To each representation space, we assume that there exists a corresponding signal field. So, we have at least an external signal field and an internal signal field. We do not necessarily consider that these signal fields are known. Sometimes they are really not known. In most of the times they are partially known. But we only consider that these signal fields do really exist. These notions are illustrated in figure 3.

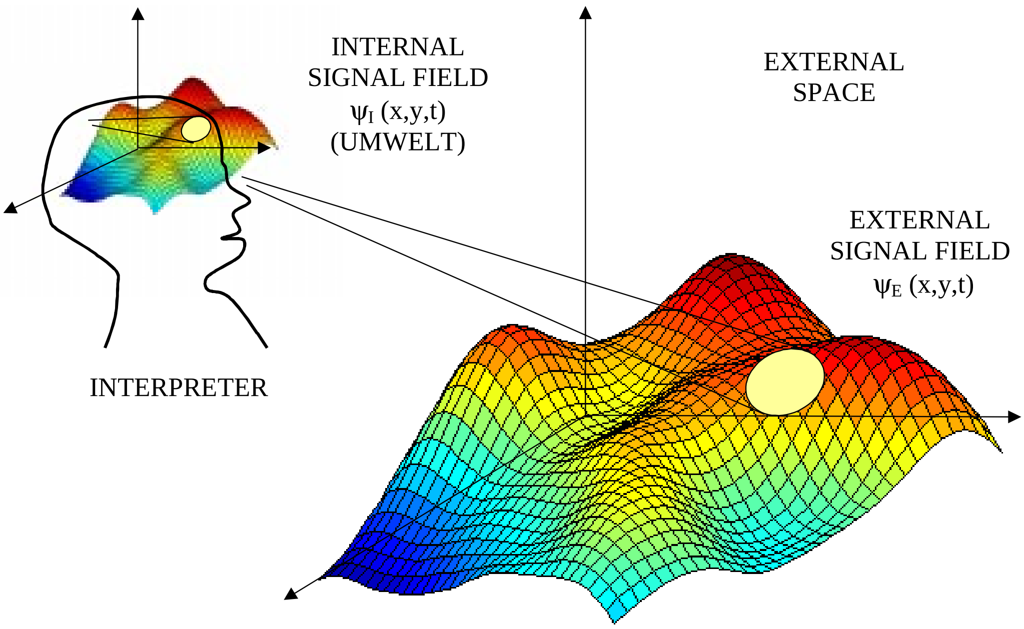

Figure 3 - Signal Fields

Now, for the sake of trying to configure a semiotic model, let us consider some specific types of representation spaces and signal fields. First, we will consider the real world (i.e. whatever really exists at the physical space) to be an external signal field. For our purposes, we may assume that the external space is the 3 dimensional space we humans have the capacity to perceive. There are physical theories that try to detect more dimensions (KAKU 1994), but we will stay with the three we usually perceive, plus time. One important difference here is that we are not modeling our world as a world filled of things like in the objectivist tradition. For us, the real world is a giant and continuous wave function embracing all the 3-dimensional space. The signal field that models the world, though, is not just any function, but a function with a set of certain kind of particularities. These particularities are those that gives us the illusion that the world is constituted of things. There seems to be some rules on how points with lower energy (points where there is only emptiness) and points with higher energy (points where we detect the presence of things - being it solid, liquid or gas) and how those regions can move in time. It is obvious from this definition, that the external signal field is unknowable in its entirety. Interpreters are only able to know part of the external signal field, where, at each time, its focus of attention is set up.

The internal space will accommodate a model of external signal field, by means of an internal signal field. Ideally, there should be multiple internal spaces (and signal fields), not just one (figure 4).

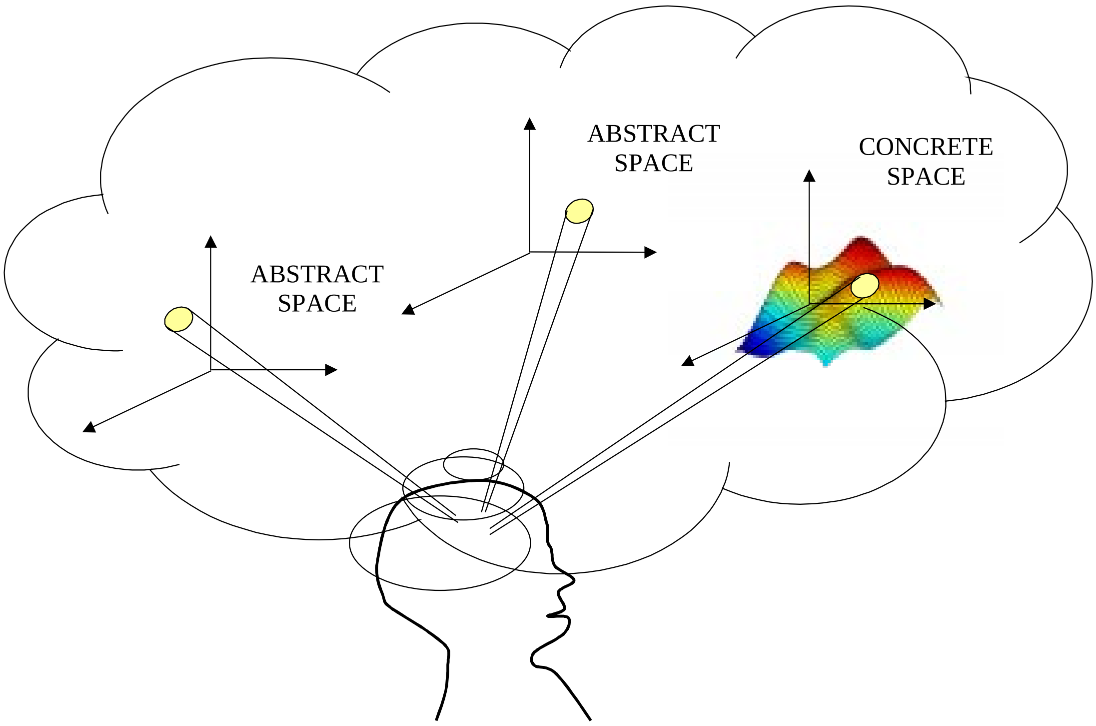

Figure 4 - Multiple Internal Spaces and Signal fields

There should be one internal space that should hold in its signal field our best model of the external signal field. This will be called the concrete space, and will be an image of our Umwelt (SEBEOK 1997) or, in other words, our sensible environment. The other spaces will be called generically abstract spaces. They will be used to store abstract concepts necessary in order to reconstitute the concrete signal field.

In a general way, both internal spaces and internal signal fields will depend on the type of semiotic synthesis we are trying to model.

There are some important things we should always remember regarding those issues.

First, the external signal field is infinite, continuous and probably takes values on continuous sets. So, they can not be known as a whole. But, it can be known in parts, with approximations.

Second, the only way we are able to know the external signal field is due to sensors. Our sensors are established by our focus of attention. To put our focus of attention on some region of external space is equivalent to locate our sensors on this region.

We will call the signal field related to the region under the focus of attention (internal or external) generically as a "signal". So, a signal is by definition a multi-dimensional value, possibly structured in space and time.

Having that in mind, we are ready for the definitions to come.

3 Modeling Semiosis

Given the backward conceptual framework, we are now prepared to build up our model of semiosis, or sign process that - we know - is not a complete one, but is specially interesting for the purposes we are concerned with. We will start with a triadic interpretation of the sign that is, in some sense, similar to Morris' one, but we hope, will evolve to a more Peircean like in the future. Our aim is to reach a Peircean definition, that we conclude to be more complete. In this sense, our definition of semiosis will require the definition of three main partners - sign, object and interpretant, which play a relational (and with some of them this relation is a causal relation) role where signs causally generate new signs and so over. So, there come our limited definitions for signs, objects and interpretants.

A **sign** (in the sense of the interpreter) is understood as everything under the interpreter's focus of attention (internal OR external) that would cause an interpreter action. In other words, a sign is a signal in either the internal or external representation space, able to cause an interpreter action. The possible interpreter actions are the change in the focuses of attention (internal and/or external) and/or the determination, for the next future time $t_{new} = t_{current}+1$ (or $t_{new} = t_{current} + \varepsilon$, $\varepsilon \to 0$, if considering a continuous timeframe), of a new value for any signal field (internal or external), at a point (x,y,z) covered by the focus of attention in that space.

The **object**, just like the sign, is a signal existing at the external or internal spaces, but **not** currently under the interpreter's focus of attention that, **if** under an interpreter's focus of attention, would cause the exactly same interpreter action actually generated by the presence of the sign. In this sense, we may say that the sign refers to the object, in the sense that it causes within the interpreter, the exactly same reaction that would be caused due to the presence of the object.

The **interpretant** (in the sense of the interpreter) of this sign will be:

- any interpreter action caused due to the presence of the sign under the focus of attention, AND/OR
- any change in internal and external signal fields for a future time-step $t_{new} = t_{current}+1$, caused by an interpreter action due to the effect of the sign.

These notions may be compared (but loosely compared) to the Peircean notions of energetic and logic interpretant (PEIRCE 1960), despite this comparison is not exact neither fair.

With these notions in hand, we are ready to define what would be an **external semiosis**. We call an external semiosis to the generation of an interpretant, due to the presence of a sign, where this interpretant happens at the external space. In other words, the **effect of the sign** is made to **happen on the external space**. Either by changing the external focus of attention or by changing the values (for the next time-step t+1) of external signal fields at some point (x,y,z) within the external focus of attention.

A change in external signal field corresponds to a change in environment. This change would be of course shareable with other interpreters that are focusing attention in the same region of external space. This change can act as a sign for the same interpreter or to other interpreters, generating a chain of interpretations.

External semiosis happens mainly on interpreters that do not have an internal space. Examples of such kinds of semiosis would be e.g. on molecules and chemical reactions. The presence of a given molecule will affect other molecule, making them to enter a reaction that is a kind of mutual semiosis. This type of semiosis happens too on very simple biological organisms that have a purely reactive behavior, didn't storing any kind of information in an internal memory.

A special case of external semiosis happens when the sign is not at the external space too, but it is in some of the internal spaces. Usually, this is a final result of a chain of internal semiosis, resulting in an external behavior.

We call an **internal semiosis**, a process in which the **interpretant of a sign happens within any of the internal spaces of the interpreter**. If the sign is at the external space, then we call this process a semiotic transduction, because an external sign cause an internal interpretant. The process in which an internal sign causes an external interpretant (which is an external semiosis) can also be called a semiotic transduction, because the cause and effect are in different spaces.

In this sense, a typical semiosis chain starts with an external sign, which generates a set of internal interpretants. Those internal interpretants become then internal signs, generating new internal interpretants. This cycle continues, until some internal interpretant become an internal sign that generates an external interpretant, resulting in a change on the environment.

4 The Role of Information within Signals and Signs

From the notions given in last session, we need to point out some important relations and differences regarding the role of information in signs and signals. This role of information is fundamental for us to clearly make distinctions regarding signals and signs. The signal is a support for the sign, but it is just a part of it. For a signal to become a sign, it must be able to generate another signal, which will be the interpretant of it. More than this, this signal must generate the same exact reaction that a second signal - its object - would generate in that interpreter. So, there is a clear connection between the notion of a signal and the notion of data, in terms of data processing. The signal, by itself is just a vehicle for conveying information, nothing else. As soon as this information is used to change a signal field, this signal turns into a sign, and starts its meaningful activity. Some semioticians do not like this vision, and prefer to say that the elementary entity is the sign, and that the signal is just a kind of degenerate kind of sign. We prefer to define a signal as an elementary entity and build up the notion of sign over it. Engineers and computer scientists are already used to the notion of signal (in terms of signal processing, control, etc…), and to grow the notion of sign using the notion of signal may be easier to understand, regarding this community of engineers and computer scientists. The information conveyed by signs (signals that do perform actions) are associated to the actions they are able to cause. When that happens (signals becoming signs), then the information is called knowledge.

We make then a strong correlation relating to the role of information that is conveyed. Signals convey just data, and signs convey knowledge.

5 Knowledge Units

Now, if we want to turn this model of semiosis into a functional (computational) model, we need to make some simplifications of it. The first simplification will be to turn from this generic, potentially continuous model into a discrete one. From our definition of a sign, we see that it can be related to continuous signals happening at the environment. Despite this is a very abstract vision of what is a sign, for computational purposes, we will need a more convenient way of dealing with those signs. We will call a discrete sign (that is, the region under a focus of attention of some discrete space, that is a sign or a set of signals, depending if causing or not a behavior at the interpreter) a **knowledge unit**, because even though it is not used like that, it is able to be used as knowledge.

The basic behavior of a computational interpreter is then to select knowledge units both from the external and internal spaces, generating new knowledge units at the external or internal spaces. Knowledge units and interpreters will be viewed as the elementary pieces of sign processes within a computational semiotics device. Let's see how they integrate each other in order to build what we mean by a semiotic device. Observe that the process hereon is built on the notions of semiotic synthesis given in section 2.

First of all, we consider the existence of an **environment,** or **real world**, which is defined as a set of dynamic continuous phenomena running in parallel (it's our external signal field). We assume we are not able to know this environment in its whole. The part of environment we are able to know, in a process that goes through our sensors, is called our Umwelt (SEBEOK 1997). The Umwelt, also called our **sensible environment**, is our best possible comprehension of reality. It is very important to stress, though, that Umwelt is not reality. It comprises only our best understanding of reality. In this sense, our sensors are the primary source of information that flows into our mind. These sensors do provide a continuous and partial information about phenomena occurring in Umwelt. From this continuous source of information, we extract what we call singularities (in the sense of (MEYSTEL 1996)), i.e., clusters of information that can be aggregated under a single concept. These singularities are discrete entities that model, in a specific level of resolution, the phenomena occurring in the world. We can also view these singularities as an intensional definition for what we are calling here knowledge units (figure 5).

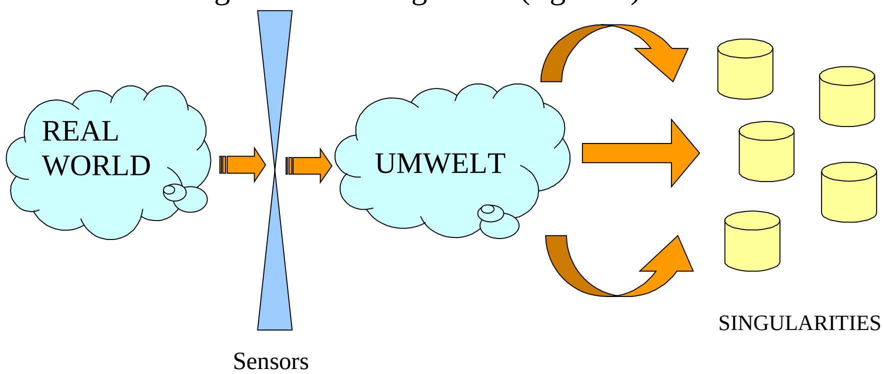

Figure 5- Singularities Extraction

Once those granules of information (singularities) are identified in the Umwelt, they need to be encoded to become a knowledge unit. This codification needs a **representation space** and an **embodiment vehicle** (structure) that is placed within the representation space. These structures may be abstracted to mathematical structures (figure 6), i.e., (a) numbers, (b) lists, (c) trees and (d) graphs.

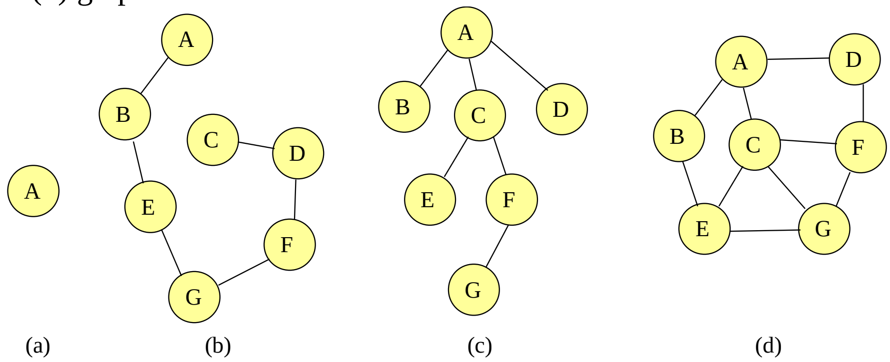

Figure 6 – Mathematical Structures

Each structure has a place at the representation space (figure 7).

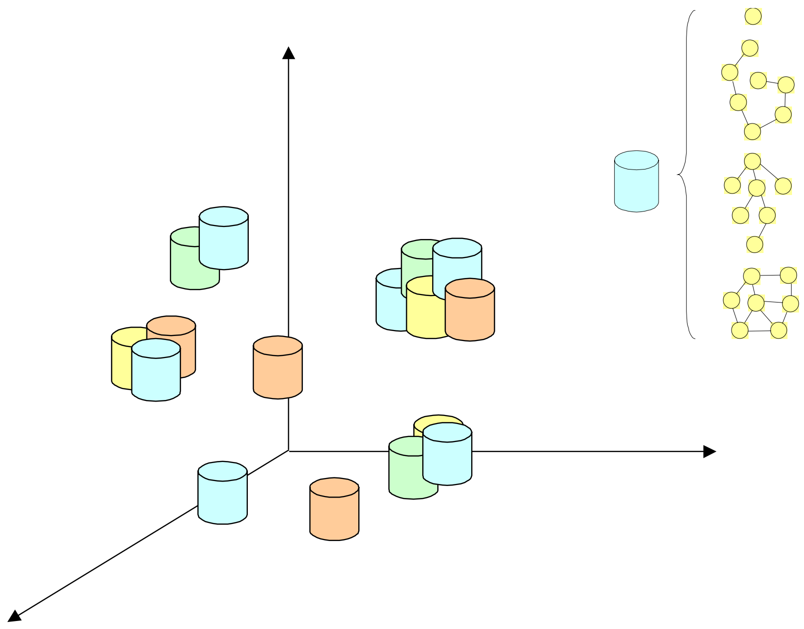

Figure 7 – Representation Space

The view shown in figure 7 is though, our view of representation space **after** an interpretation. Before interpretation, the representation space is more like in figure 8: a set of values occupying a place in space. To build a knowledge unit, then, we need what we called the "focus of attention" mechanism, which selects a closed region of representation space, that is our primary field of interpretation.

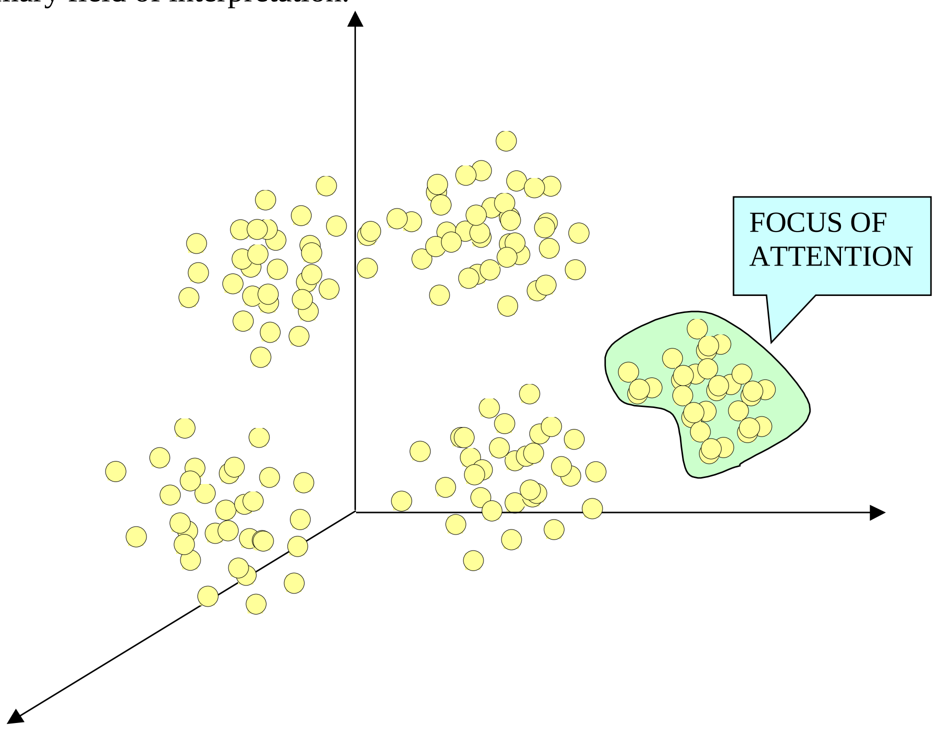

Figure 8 – Focus of Attention and Structures Identification

Then comes what we call the first interpretation problem, (illustrated in figure 9). How a set of values embraced by the focus of attention is going to be interpreted ? This is called the structural identification problem.

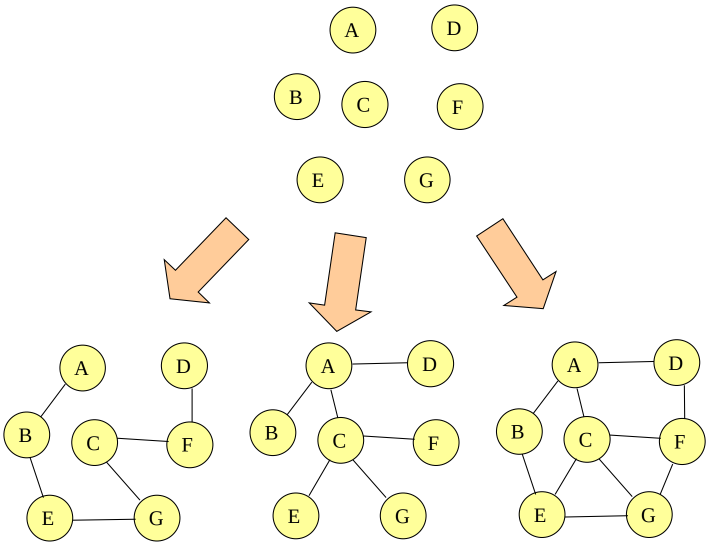

Figure 9 – Interpretation Problem

A second interpretation problem, that comes once we identified the structure within our focus of attention, is related with the semantic identification of information within the structure. If the data represented by the structure respects to a direct modeling of an environment phenomenon, this knowledge unit is called an **icon**. If it gives the localization within the representation space of another structure, it is called an **index**. And, if it is a key in a conversion table, it is a **symbol**. In this case, we will need to use a conversion table (that should be another structure in the representation space), in order to locate the icon representing the phenomenon we want to refer to.

Elementary knowledge units (usually icons) are formed due to these singularity extraction mechanisms. More elaborate knowledge units, though, are formed by the application of knowledge processing operators, illustrated in figure 10.

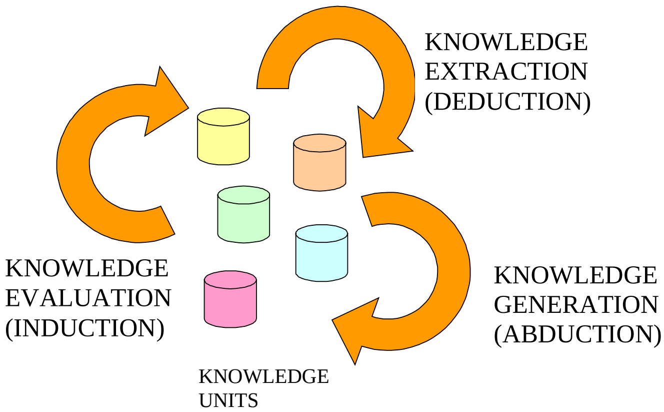

Figure 10 – Knowledge Processing Operators

These knowledge processing operators are from 3 basic types, that we are going to call here generalized deduction, generalized induction and generalized abduction. We are going to address them in the following sections.

6 A Taxonomy for Knowledge Units

We know from our daily life that our understanding of phenomena surrounding our environment is built within different categories of concepts, which are mirrored on the different notions we aggregate to words in a natural language. So, we know that the ideas brought by a substantive are categorically different from the ideas brought by an adjective, which are different from the ideas brought by a verb, and the same with adverbs, prepositions, articles, etc. They are not just different, as say, an *elephant* is different from an *ant*, but they are categorically different. We cannot compare, let's say, the understanding of what is "blue", with what is "ball", with what is "move". The many facets of *understanding* were depicted by Locke, in his seminal philosophical work -"An Essay Concerning Human Understanding" from 1689 (see a reprint of this work in (LOCKE 1997)), where he distinguishes simple ideas from complex ideas from knowledge.

Following this same intuitive understanding, we classified knowledge units according to a taxonomy of categories of knowledge (GUDWIN 1996; GUDWIN & GOMIDE 1997A; GUDWIN & GOMIDE 1997B; GUDWIN & GOMIDE 1997C). This taxonomy is inspired on the classification of different types of signs, given by Peirce (PEIRCE 1960), and the different dimensions for an interpretant, by Morris (MORRIS 1947; MORRIS 1964; MORRIS 1971). Peirce's semiotics introduced a signical taxonomy, where different kinds of signs (e.g. rhemes, dicents, arguments, icons, indexes, symbols, qualisigns, sinsigns, legisigns) were proposed, addressing different characteristics of its structure and signic function. Morris identified 3 possible dimensions for an interpretant (designative, appraisive and prescriptive). The ideas from Peirce and Morris were unified in order to generate this taxonomy for knowledge units.

Basically, each type of knowledge is associated with a different category of concept (or idea), i.e., the semantic that is intrinsic to a given knowledge type. One first distinction we made was the distinction between what we mean by passive (or static) knowledge and active (or dynamic) knowledge. A passive piece of knowledge is simply a container of organized information, which needs to be manipulated by interpreters in order to generate other pieces of knowledge, generating a kind of *thought flow*. Active pieces of knowledge, on the other hand, at the same time that can contain organized information, may be used by interpreters in order to generate full actions of manipulation of other pieces of knowledge. So, when attached to interpreters (or micro-interpreters - very rudimentary and simple interpreters, as we will propose soon), these active pieces of knowledge may work as engines for the creation of newer pieces of knowledge. We divide passive pieces of knowledge into two categories of knowledge units, rhematic and dicent, and the active knowledge units comprises a category of knowledge we call argumentative knowledge. The types referred as rhematic and dicent (GUDWIN 1996; GUDWIN & GOMIDE 1997A; GUDWIN & GOMIDE 1997B; GUDWIN & GOMIDE 1997C) are passive, in the sense that they only exist as data. The knowledge types known as arguments are active, in the sense that they do not only exist as data but, if used by an interpreter, they may also perform transformations in the system. A direct analogy that allows us to better understand the difference between passive and active types is the classification of information within a computer memory as data and code. Passive types are just like data in a computer memory. Active types are like code in a computer memory. They can be seen as data or code, depending on the context being analyzed. Active knowledge units are the primary source of activity in a semiotic system. They are responsible for the extraction of singularities and also for the further discovery and manipulation of new knowledge units within the semiotic system.

Originally, two taxonomies were developed. The first one, concerning the nature of knowledge, and the second one, concerning its pragmatical use in the construction of an intelligent system. They are presented in figures 11 and 12 below.

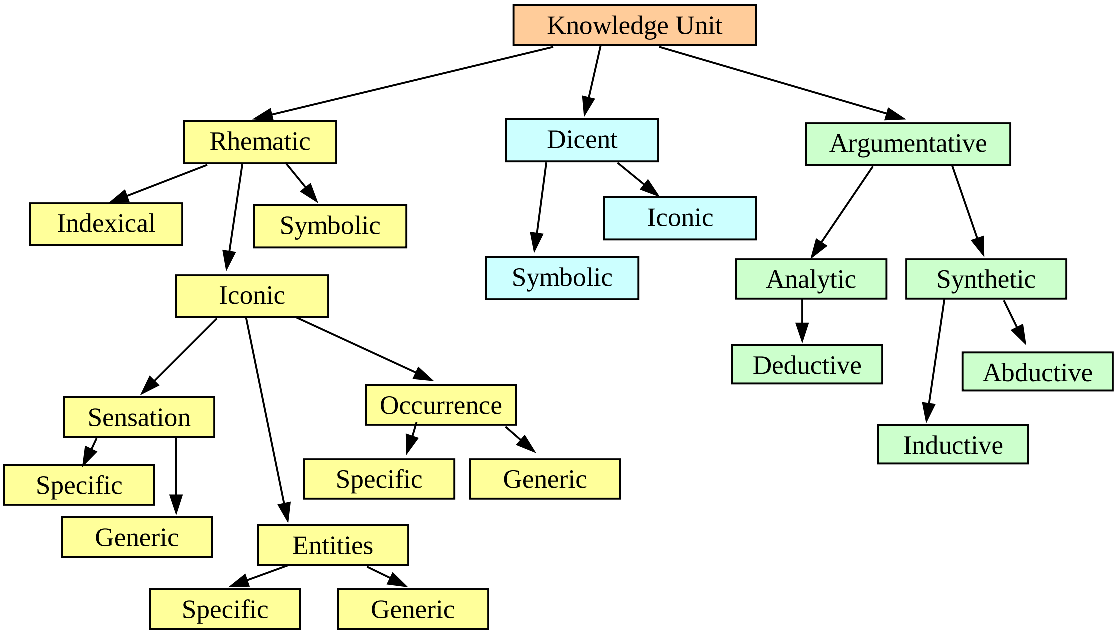

Figure 11 - Knowledge Units classified by its Nature

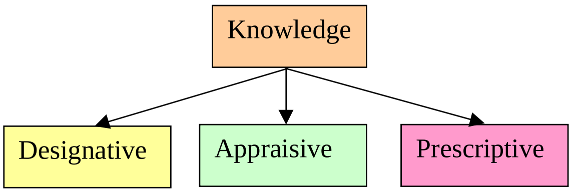

Figure 12 - Knowledge Units classified by its use in an Intelligent System

These taxonomies were further unified and summarized, as in figure 13. In this figure, we adopt the following convention:

R means *Rhematic*;
D means *Dicent*;
Ic means *iconic*;
Ob means *object*;
Sp means *specific*;
G means *generic*;
Sy means *symbolic*;
In means *indexical*;
Se means *sensorial*;
Oc means *occurrence*.

In figure 13, all arrows between categories refer to argumentative knowledge (both analytic and synthetic). The notation between brackets is used to specify the different kinds of knowledge. For example, *{RIcSeG}* means a rhematic iconic sensorial generic piece of knowledge. Figure 13, besides representing the categories by themselves, also shows the conceived types of transformation of knowledge units, in order to better compact information. In this sense, sensors will generate *{RIcSeSp}* knowledge units that will be further converted to *{RIcSeG}* knowledge units, in order to save memory space. They will be used further to generate *{RIcObSp}* or *{RIcObG}* knowledge units, and so over.

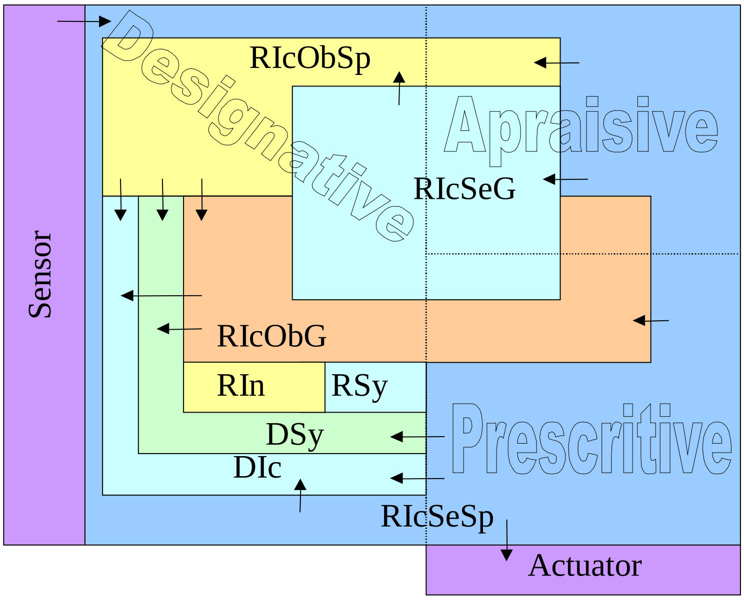

Figure 13 - A Summary for Knowledge Units

In this summary view, a passive knowledge unit is both classified according to its functionality (designative, appraisive, prescriptive) and to its structure (rhematic, dicent). Argumentative knowledge units are a special case, given its active characteristic, being classified both as functional and structural.

6.1 Rhematic Knowledge Units

Rhematic ({*R*}) knowledge units refer to ideas that can be assigned to isolated words in a natural language. They concern what we call the semantic memory of an intelligent system. Usually, they are used in the representation of environmental phenomena like sensorial experiences, objects and events. Sensorial experiences can be assigned as the semantics of e.g. adjectives, the existence of physical or logical objects may relate to substantives and occurrences and events may be assigned as the semantics of verbs. In a last analysis, though, all of them should evolve from perceptual data. Rhematic Knowledge Units may be still divided into icons (Rhematic Iconic - {*RIc*}), symbols (or names) (Rhematic Symbolic - {*RSy*}), or indexes (Rhematic Indexical - {*RIn*}). A {*RIc*} knowledge unit refers to a direct model for the phenomenon it represents. A {*RSy*} knowledge unit is a name that refers the phenomenon. A {*RIn*} knowledge unit is an indirect reference to the phenomenon.

The {*RIc*} knowledge unit may still be divided into three different classes:

Sensorial - {*RIcSe*}, usually models of adjectives;
Object - {*RIcOb*}, usually models of substantives;
Occurrence - {*RIcOc*}, usually models of verbs.

Those three categories of knowledge units can all be still divided into two classes: specific (*Sp*) and generic (*G*). For example, a {*RIcSe*} knowledge unit refers to a sensorial information like an image or a temperature sensor measuring. A {*RIcSeSp*} knowledge unit refers to a particular instance of a sensorial pattern, e.g. a specific image, or the temperature in a given time. A {*RIcSeG*} knowledge unit refers to a generic knowledge of all time occurrences of some sensorial input. An example of a {*RIcSeSp*} would be a measuring from a temperature sensor, which assigns to a specific region of space and time a temperature of 28 degrees Celsius. An example of a {*RIcSeG*} knowledge unit will be a fuzzy set comprising all the temperatures that may be considered as a high temperature.

The {*RIcOb*} knowledge units are used to represent real world objects (which may or may not exist in such a real world). As before, a {*RIcObSp*} knowledge unit refers to a specific object in time and space. This object may be an instance of a class of objects, which may be represented then by a {*RIcObG*} knowledge unit.

The {*RIcOc*} knowledge units are representations for the semantic of verbs. Usually, they set-up attribute values to an object, or represent changes in one or more attribute values in one or many objects through time. Examples of {*RIcOc*} comprise the assignment of state properties to an object e.g. the assignment of a given temperature to an attribute called temperature, etc. They may refer also to the creation or destruction of an object, and so on. The {*RIcOcSp*} knowledge units are related to specific occurrences in time and space, e.g. representing the changing of traffic light, in a given time, from red to green. The {*RIcOcG*} knowledge units represent generic occurrence, e.g., the changing of a traffic light from red to green, without specifying a particular instance of time.

6.2 Dicent Knowledge Units

The dicent ({*D*}) knowledge units are representations for the meaning of phrases in a given language. They may be propositions, but also commands or questions. They are used to compose what we call an episodic memory within an intelligent system. The most studied kind of dicent knowledge is the proposition, usually studied within the context of predicate logic calculus. The difference between a proposition and a term is fundamentally that the proposition has a truth-value associated with it. Usually, this truth-value represents the belief the intelligent system have on the proposition and it may vary from false to true using a multivalued logic or not (e.g. fuzzy logic). Compared to a rhematic knowledge unit, we may say that if a rhematic knowledge unit represents the meaning of a single word in a natural language, dicent knowledge units do represent the semantic of a combination of words forming expressions and phrases in such a natural language, and referring to a situation happening in a real or logic world.

We represent a dicent knowledge unit by associating a term (usually an expression of *{R}* knowledge units) to a truth-value. Dicent knowledge units may also be formed by the association of other dicent knowledge units, linked by logical connectives. For examples:

the knowledge that "A" is true;
the knowledge that "A $\wedge$ B" is false;
the knowledge that "IF A $\wedge$ B THEN C" is true.

Dicent knowledge units may be iconic ({*DIc*}) or symbolic ({*DSy*}). The {*DSy*} knowledge units use a *{RSy}* knowledge unit to refer to the meaning of a whole phrase, to which we also attach a truth-value. An example of such knowledge unit would be, e.g., a label for a first order logic sentence. The {*DIc*} knowledge units are explicit compositions of *{RIc}* knowledge units, plus a measure of the belief that this composition really describes something happening on the real world.

6.3 Designative Knowledge Units

Designative knowledge units are used to model the world in which the intelligent system is immersed. For this purpose they may be Rhematic and/r Dicent knowledge units, either specific or generic. Designative knowledge units may also be viewed as descriptions of facts or things on the real world. An intelligent system initially has just a few, or eventually no designative knowledge units "a priori". Usually designative knowledge units emerge from the interaction between the system and world.

6.4 Appraisive Knowledge Units

Appraisive knowledge units are used to represent an evaluation (a judgment, a criteria) regarding the possible success in achieving goals. In natural systems, appraisive knowledge units may be related to the essential goals of a being, like reproduction, survival of the individual, survival of the specie and increasing the system's knowledge about the world, for example. Depending on the goal they are related to, appraisive knowledge units may assume special forms like: desire, repulse, fear, anger, hate, love, pleasure, pain, comfort, discomfort, etc. Essentially, appraisive knowledge units evaluate if a given sensation, object, or occurrence is good or not, as far as goal achievement is concerned. An appraisive knowledge unit is usually also a *{RIc}* knowledge unit.

Appraisive knowledge units are used to build what we call the "Value System" of an intelligent system, being related to the notion of an "emotion", as observed in living beings. So, we may say we use appraisive knowledge units to model "emotions" within intelligent systems, and our claim here is that this intelligent system not only is able to simulate emotions, but that they actually really are able to feel those emotions, making a kind of emulation of such an emotion. This is a very polemic claim, which we already discussed in (GONÇALVES & GUDWIN 1999), but we have confidence that this claim can be justified, depending on the semantic we attribute to the word "emotion".

6.5 Prescriptive Knowledge Units

Prescriptive knowledge units are used in order to let an intelligent system act on the real world. Basically, prescriptive knowledge units are used to establish and to implement plans through actuators. However, prescriptive knowledge units will not necessarily end up into an action, but remain just a plan of action. Many prescriptive knowledge units may be used to perform a prediction or a plan, but only one of them will be selected to generate an action.

Strictly speaking, prescriptive knowledge units are commands, used to plan, predict and actuate in the real world through actuators. Just like appraisive knowledge units, prescriptive knowledge units are usually classified as rhematic iconic.

6.6 Argumentative Knowledge Units

Argumentative knowledge units are used to perform the processing and transformation of other knowledge units, by means of some algorithm, specially tailored to the particular categories of knowledge units they are able to process. In this sense, the information encoded in its structure, besides carrying some data content, is also used by an interpreter in order to promote transformation actions on itself and in other knowledge units. So, its data carries also a sort of *program code* beyond its data ability. We say that knowledge units of this kind are functional because they have an explicit functional role. A good metaphor for understanding the dual nature of argumentative knowledge units is to look to machine instructions within a computer memory. Those instructions can be seen both as data (a sequence of bytes) and code (processor instructions). In the same way, argumentative knowledge units are both data (structure) and code (instructions on how to process). The argumentative knowledge units ({*Ar*}) can be synthetic ({*ArSt*}) or analytic ({*ArAn*}). These qualifications refer to an ability of knowledge units of, depending on their type, storing many other knowledge units (potentially infinite), within single knowledge units of a higher level type. An analytic knowledge unit is able to extract from a higher level knowledge unit, one or more of its components, stored in a reduced form in the higher level knowledge unit. The synthetic knowledge units are able to create those higher-level knowledge units, based on lower level ones. They are able, also, to modify in some sense a given higher level knowledge unit, in order to let it adapt to new knowledge units inputs. In other words, they are able to synthesize new knowledge units. There are two kinds of {*ArSt*} knowledge units: inductive knowledge units ({*ArStId*}) and abductive knowledge units ({*ArStAb*}). The {*ArStId*} knowledge units perform modifications into parameters of their input knowledge units, making adaptations of them. The most useful example of an {*ArStId*} knowledge unit is the generalization operator, which assigns new truth-values (or belief measures), based on the evidence of input knowledge units, to a higher-level knowledge unit which encodes a belief measure inside it. Other examples of inductive arguments are the learning rules used within neural networks. Supposing that we have *{RIcObSp}* knowledge units corresponding to neurons, the encoding of a learning rule that modify these *{RIcObSp}* knowledge units, changing their internal parameters are an example of an {*ArStId*} knowledge unit. The {*ArStAb*} or abductive knowledge units are responsible for the creation and generation of totally newer knowledge units, usually based on fragments of other knowledge units, but always with some random insertion. Examples of {*ArStAb*} or abductive knowledge units are the mutation and crossover operators used within genetic algorithms. Supposing that two chromossomes are encoded into two *{RIcObSp}* knowledge units, an {*ArStAb*} is able to generate a new pair of *{RIcObSp}* knowledge units, generated by means of a crossover of the previous two.

7 Knowledge Units Manipulation

Active knowledge units, as pointed above, are special types classified into the knowledge hierarchy within a particular branch, involving the family of argumentative knowledge units. Opposed to passive knowledge units, they carry the property of being able, when used by an interpreter, to process other knowledge units. Basically, an argumentative knowledge unit is a piece of knowledge whose semantic is the understanding of knowledge manipulation. In other words, it indicates how to produce new pieces of knowledge, taking as input a set of knowledge units.

In this section, we will analyze more deeply the nature of argumentative knowledge units. Their behavior is based on a set of three basic operators, namely knowledge generation, knowledge extraction, and knowledge evaluation, which should be understood as generalizations for the abduction, deduction and induction reasoning methods respectively. In this sense, knowledge generation is viewed as an abstraction for abduction, knowledge extraction is an abstraction for deduction and knowledge evaluation is an abstraction for induction, leading us to universal operators of generalized abduction, deduction and induction.

Despite this seems to be a clear and concise understanding, we will see that it is not. The notions of abduction, deduction and induction are not a commonsense in literature. Peirce himself has developed an elaborated study of Aristotle's syllogisms in order to grow up an understanding of what those terms really mean. During his early life, he (as many other men of science) associated induction with generalization and abduction with hypothesis making. These were the two main ways of generating new knowledge, as recognized by science during the time of Peirce. During the early ages of philosophy of mind, deduction was the main and unique knowledge operator recognized as valid. By the time of Peirce, another knowledge operator had appeared - induction, viewed as the capacity of making generalizations. By this time, the concept of thought was reduced to cycles of induction and deduction, in which induction was responsible for creating concepts and deduction for using them. Peirce noticed that besides induction (generalization), there was another knowledge operator - hypothesis making, that was able to create new concepts, but was not related to any kind of generalization. Peirce then introduced hypothesis making (sometimes referred as retroduction) as a third possible knowledge operator. But this was just the beginning. Peirce had many theoretical problems in order to differentiate some strange kinds of induction and hypothesis making, which were very close in behavior. During most part of his life, Peirce struggled with this problem, and was able to solve it only on his late life. Fann, in a very enlightening book studying the chronological writings of Peirce (FANN 1970), showed the evolution of Peirce's thoughts regarding abduction, deduction and induction, and how Peirce could redefine those concepts in a sound framework. And the results were surprising. Peirce had to break the idea that induction was generalization and also break the process of hypothesis making in order to reach his current understanding of what is induction and what is (now) abduction. He realized that both of them (generalization and hypothesis making) have phases of structural constructions and a further evaluation of results. In the case of generalization, there was first the construction of a structure representing the generalized concept, and further an evaluation that gives belief to this construction. In the case of hypothesis making, there was a first phase in which the hypothesis was constructed (its structure) and then a second phase in which different hypotheses were evaluated and the best hypothesis was selected. So, in both cases there is a phase of construction and a phase of evaluation. Peirce then associated the idea of construction to what he called "abduction" and the phase of evaluation of what was now the new understanding of the word "induction". Making this distinction, he could see that both generalization and hypothesis making have within them elements of abduction and induction. Unfortunately, for people that do not know Peirce (and for many that know him, but are not aware of this change in meaning for induction along his chronological writings), induction is still associated to generalization, and this is the source of very much noise in literature regarding knowledge processing. For our understanding, we will keep the late Peircean understanding of those terms, and to avoid misunderstandings, instead of using abduction, deduction and induction, we will refer to knowledge generation, knowledge extraction and knowledge evaluation.

But, before going on to detail these three knowledge operators, we have to first discuss the issues of abstraction, and the intensional and extensional modes of representation.

7.1 Abstractions

The Webster dictionary defines abstraction, or abstract idea as an idea that is disassociated from any specific instance, or an idea that is expressing a quality apart from an object. Without loosing this understanding, we will define an abstraction as a way of representing potentially infinite sets of elements, by creating a law of pertinence to the set. We know that there are two ways of describing a set. We may list all its elements, or we may create a law and say that any element that respects this law is a member of the set. The first representation of a set is called an **extensional representation** of the set. The second representation of the set is called an **intensional representation** of the set. If we now take a finite representation of the law governing set membership, we will have an **abstraction** of the set. In other words, an abstraction is an intensional representation of a set, given in the form of a sign.

Knowledge units from some types of knowledge can be compared to each other by means of an "abstraction" partial order relation ($\prec$). In this sense, if some knowledge units a and b are related by $a \prec b$, then we say that b is an abstraction of a. Or, in other words, that b is a generalization of a, and a is an specialization of b. These concepts are fundamental for our definitions of knowledge extraction, knowledge generation and knowledge selection. The key issue for understanding the abstraction relation is to remember the two possible ways of defining a set. We may define a set using an *extensional definition*, where we explicitly list all elements within the set. This way of definition is fine for finite sets only. There is also the *intensional definition* of a set, where we define a set as the collection of all possible points satisfying a condition. Using an *intensional definition*, we may represent a whole infinite set with only a finite number of parameters. This implies in an encoding able to convert from the intensional representation for elements in the extensional representation. For example, let a set S be defined as $S = \{(x,y) \in R^2 \mid y = 2x^3+7x+1 \}$. This is an intensional definition for set S. We may represent set S, as the tuple $(2,0,7,1)$, that encodes all the information necessary to reconstruct the points (x,y) belonging to S. Suppose now a knowledge unit $a = (1,10)$ and a knowledge unit $b = (2,0,7,1)$. If we interpret a as being a pair in R^2, and b as being the parameters representing the infinite set S, we may say that $a \prec b$, because knowledge unit b comprises not only a, but a whole set of pairs obeying the same relationship. Notice that we may have also a knowledge unit $c = (0,1,1,10,2,31)$, that should be decoded as the set $T = \{(0,1),(1,10),(2,31)\}$, and we would also have $c \prec b$, and $a \prec c \prec b$. This is only a clue for understanding the nature of abstraction operator. The way in which we decided to encode set S in tuple $(2,0,7,1)$ is not trivial. We may view this operation as a kind of data compression operator. Each knowledge unit b that can be expanded to other knowledge units a_i through some particular interpretation is said to be a generalization of them. And the a_i's are said to be specializations of b.

7.2 *The Elementary Knowledge Operators*

We propose a minimum set of operators as a conceptual basis for the construction of intelligent systems. These are the "knowledge extraction" operator, "knowledge generation" operator and "knowledge selection" operator. In fact, they are not exactly operators, but classes of operators.

7.2.1 Knowledge Extraction

Suppose a knowledge unit b and a knowledge unit a, such that $a \prec b$. Then, a function f_{ke} that maps b (remembering that b is a structure, e.g. a structured number) onto a: $a = f_{ke}(b)$, is called a knowledge extraction operator.

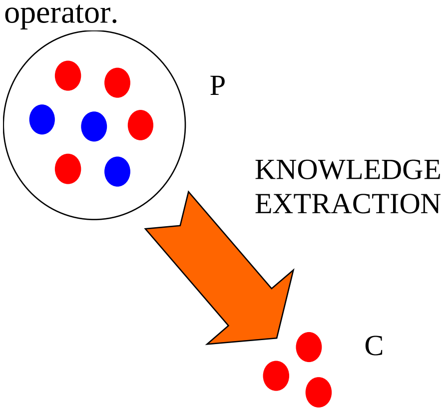

Figure 14 – Knowledge Extraction

In figure 14 we have an example of knowledge extraction. From a set P of knowledge units, called the "premise", the operator extracts a set C of knowledge units, called the "conclusion". We call this operation *knowledge extraction*, because the extensional definition of the knowledge units in C is a subset of the extensional definition of knowledge units in P. So, it "extracts" from P only part of its semantic content.

7.2.2 Knowledge Generation

Suppose now the same a and b above, and also a function f_{kg} that maps a onto b, i.e.,$b = f_{kg}(a)$. Then f_{kg} is called a knowledge generation operator. Usually, this kind of operator is not single input/single output, but comprises a set of input knowledge units and a corresponding set of output knowledge units. Then, we have e.g. that $(b_1,...,b_m) = f_{kg}(a_1,a_2,...,a_n)$.

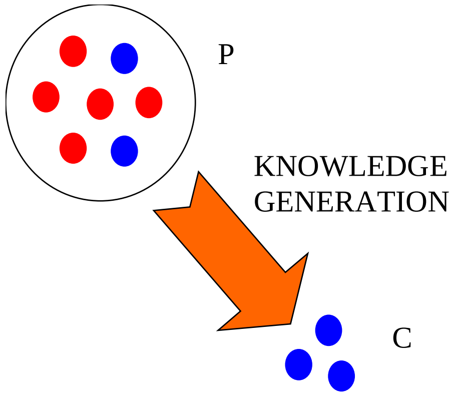

Figure 15 – Knowledge Generation

In figure 15, the premise P is the collection of knowledge units a_i and the conclusion C is the collection of knowledge units b_i. One of the particularities of this operator is that the extensional definition of the knowledge units in C necessarily contains elements that are not originally in the extensional definition of knowledge units in P. They have been added during the process of knowledge generation. This is what characterizes the *knowledge generation* operation. This process can be done in many different ways, including **combination** of knowledge units, **fusion** of knowledge units, **transformation** of knowledge units (including insertion of noise), **interpolation, fitting and topologic expansion** of knowledge units or any hybridism of these techniques. A lot of examples may be set here. For example, the interpolation of functions adds all the points surrounding the original samples. The fitting of functions neither requires the inclusion of sample points. Topologic expansion from a number to a fuzzy number, adds all the points in the vicinity of it. The learning algorithm of a neural network transforms a set of weights describing a nonlinear classifying function into another, by adjusting it in order to include new sample points.

7.2.3 Knowledge Evaluation

Suppose now, that we have a set of input knowledge units, $\{a_1,a_2,...,a_n\}$, that must be judged or evaluated regarding a purpose or objective p. Suppose now a function f_{ke}, that performs the evaluation of the input knowledge units, regarding the purpose p, generating an appraisive knowledge unit as output, or changing internal parameters of input knowledge units: $b = f_{ke}(a_1,a_2,...,a_n, p)$, in the sense that b is either a new knowledge unit with an appraisive content or any of the a_i's with their parameters modified. Then, f_{ke} is called a knowledge evaluation operator

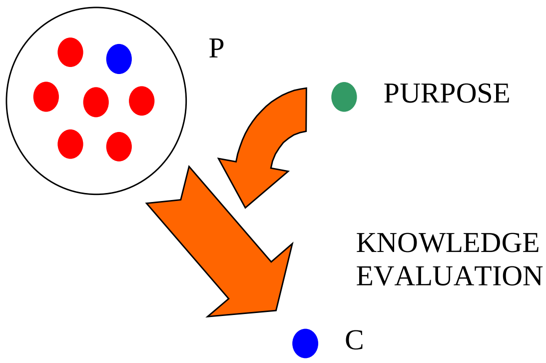

Figure 16 – Knowledge Evaluation

In figure 16, the knowledge units a_i are within set P (Premise) and b is equivalent to the conclusion C.

8 Semions

In order to build a computable model of such knowledge unit processing, we derived a related concept, which aggregates both a knowledge unit and what we call a micro-interpreter into an elementary component we called a **semion.** Basically, semions are the basic entities of semiosis. The idea of a semion is to embed in a same component, both a knowledge unit and a micro-interpreter. So, a semion is neither just a signal, especially encoded in order to store different types of knowledge units, nor just a micro-interpreter, able to transform argumentative knowledge units into inferential behavior. A semion is both things. Whenever there is no argumentative knowledge to be interpreted, the semion degenerate simply to a knowledge unit. A model of a semion can be viewed in figure 17.

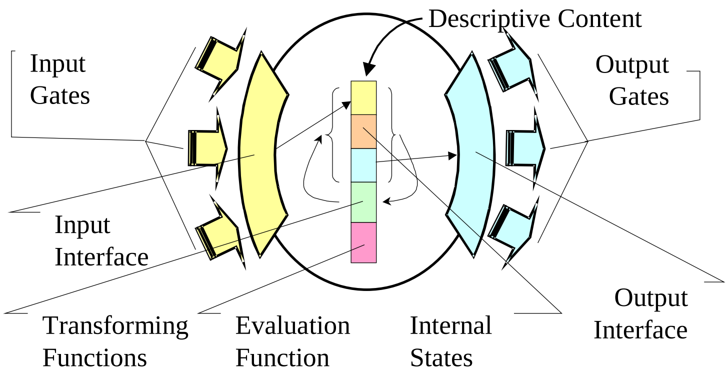

Figure 17 - The Model of a Generic Semion

The structural part of a semion (as seen in figure 17) is its descriptive content, a kind of a genetic string for the semion, which holds the signals referring to the knowledge unit personality of a semion. This descriptive content can be divided into five sectors, regarding the use the micro-interpreter personality of the semion is going to make of each sector. It is divided into three regions which store states (the input, internal and output states) and two regions that store argumentative functions (the transforming functions and the evaluation function). Semions can be classified into classes, which depends basically on the structure of their descriptive contents, and the information that each part of it encodes. A semion also have a set of input gates that feed an input interface, and a set of output gates, that are fed by the output interface. Now, to understand the meaning of each of these sections, we need to make the semion interact with other semions in a semionic system.

The interaction of a semion with other semions, in a semionic system is illustrated in figure 18. In this figure, we have semions s_1, s_2 and s_3 which will perform the role of signs, s_6

which is going to perform the role of a micro-interpreter and s_1, s_4 and s_5 which will perform the role of interpretants.

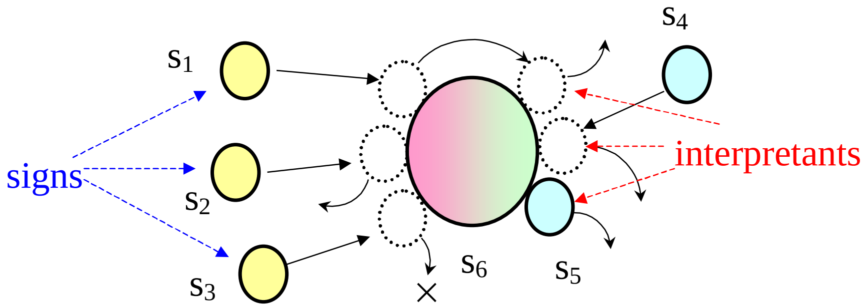

Figure 18 - Interaction among Semions

How does the semion work ? The behavior of the semion is divided into five different phases, which happen in sequence:

- The Evaluation Phase
- The Assignment Phase
- The Assimilation Phase
- The Transformation Phase
- The Consummation Phase

The first phase of interaction, the **evaluation phase**, starts when a given semion (in our case, semion s_6) chooses which other semions it wants to interact to. For this phase, he must evaluate each available semion for interaction, and decide what it intends to do with this semion after the interaction. For this purpose, for each transformation function available within the semion, a set of interaction semions of the correct type is searched for. The semion tests all possible combinations of other semions matching the inputs types of his transformation functions. Each possible combination is called then an enabling scope for this particular transformation function, being basically a list of a potential set of other semions for interaction. Each enabling scope is given an evaluation through the evaluation function, which also sets if the interacting semions will be modified, simply released back to their places or destroyed. This phase ends when the semion evaluate all alternative enabling scopes and gives for each one an evaluation grade and an intended access mode. The intended access mode sets up the intentions of this semion regarding its interacting semions. It must inform if the semion allow sharing the interacting semion with other semions (for interactions of this same type), and if the semion intends to destroy the interaction semion after the interaction. In a second phase, the **assignment phase**, a central supervisory algorithm collects the desires of interaction of all active semions in a semionic systems, and assigns to each active semion an enabling scope, with the restriction that this assignment must avoid any sort of conflict in the semions desires. Many different algorithms can be used in this phase. We developed for our tests, an algorithm we called BMSA (Best Matching Search Algorithm), which assigns the interacting semions to the active semions which better evaluated them, respected the access mode intentions of all semions. In a third phase, called the **assimilation phase**, the active semions assimilate the descriptive contents of his assigned interacting semions, making a copy of them through his input gates to his input interface. After that, the input interface is copied back again to the input states within his own descriptive content. Next to that, the interacting semions are released back to environment, destroyed or transferred to a semion output gate, depending on the access mode prescribed. In figure 18, semion s_1 was transferred to an output gate, semion s_2 was released back to environment and semion s_3 was destroyed.

On a fourth phase, the **transformation phase**, the assigned transformation function is invoked, determining new values for the semion internal states and output states. In a fifth phase, the **consummation phase**, the output states of the descriptive content is copied into the output interface, in order to feed output interacting semions, which bind the output gates. These output interacting semions may be input interacting semions which were trasferred to an output gate, an external interacting semion specially designed to interact or a totally new semion, created in the exact time of this phase. The descriptive content of these semions is changed to the newer values stored on the output interface, and after this inoculation, the output interacting semions are released back to the environment. This same process happens on all active semions on a semionic system.

If we make an analysis of this behavior of a semionic system, we will see that in a big system, with lots of semions interacting among them, there will be a combinatory explosion during the evaluation phase if all semions are able to interact with all other semions. In order to make this workable, a possible solution is to segregate semions to places, connecting places with arcs and allowing semions interact only with semions in places which are connected to the place where the wishful semion is located. This idea defines then a special kind of semionic system we call a semionic network.

9 Semionic Networks and Semiotic Synthesis

Our definition of a semionic network, must now be matched with our ambitions of semiotic synthesis, given in section 2. We remember that in section 3, we simplified our semiotic synthesis scenario, transforming continuous regions of the representation space into discrete knowledge units, as shown in figure 19 below.

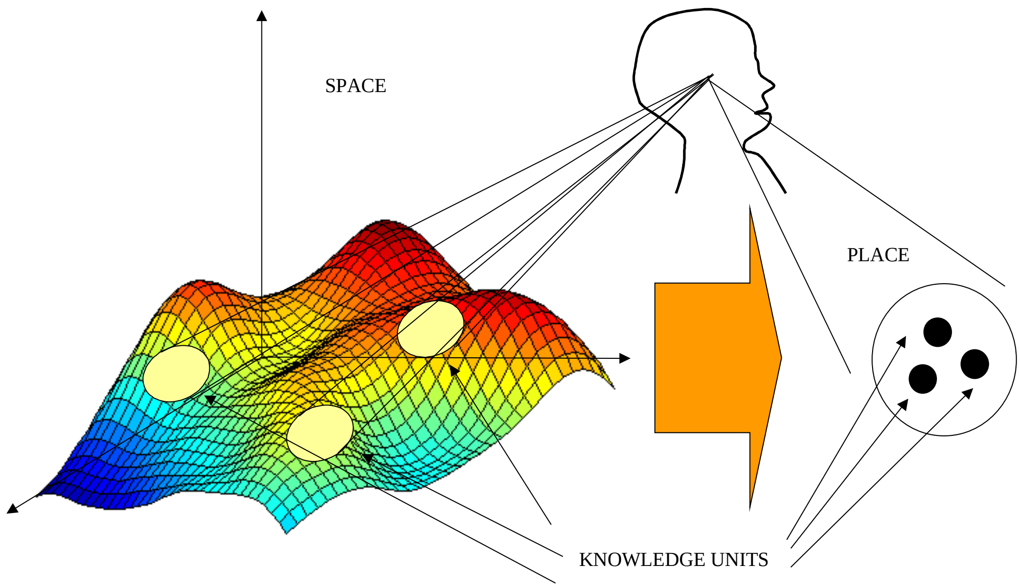

Figure 19 - Simplification of Semiotic Synthesis

If we now embed the interpreter into an active semion, and knowledge units into passive semions, we reach our definition of a semionic network, as is shown in figure 20.

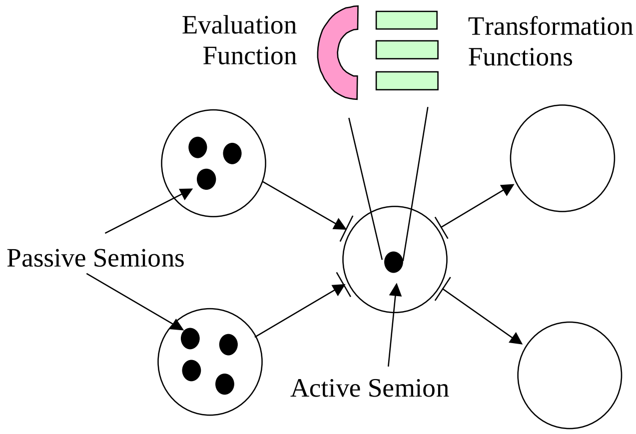

Figure 20 - A Simple Semionic Network

Our aim with semionic networks is to **design artificial minds**, which we want to use as a brain for intelligent systems. This semionic network can then be simulated or emulated in terms of computer software, and be used to control intelligent systems.

In order to test this idea, we have created a software implementation - called SNToolkit - Semionic Network Toolkit - that is a general tool for building, running and testing semionic networks. This application was written in Java language, and has its details described in (GUERRERO ET. AL., 1999). A screenshot of the application is showed in figure 21 below. It has been used to solve some small toy problems (like e.g. the traveling salesman problem, the eating philosophers problem, etc), simulate some kinds of neural networks, genetic algorithms and fuzzy systems. We also built a semionic network to control an autonomous guided vehicle that runs on a simulator. The semionic network showed in figure 21 is an initial sketch of such a network.

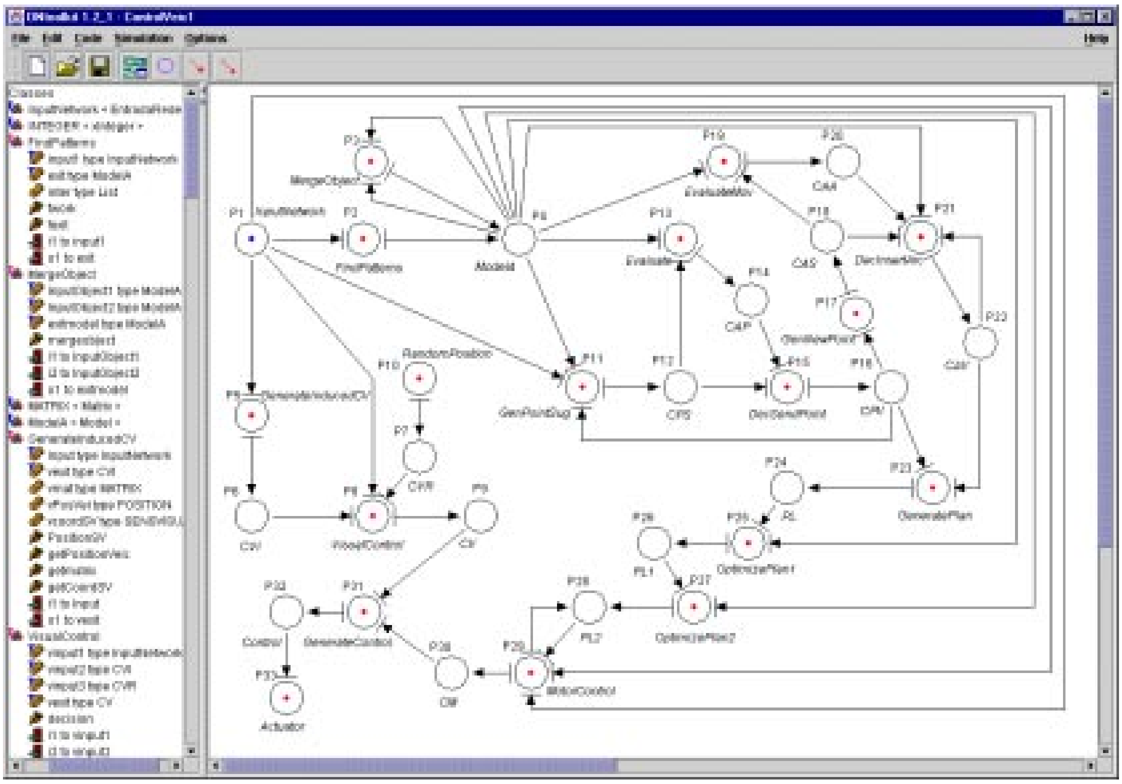

Figure 21 - The Semionic Network Toolkit

10 Conclusions

We envision many different applications for semionic networks, some of then under construction by now, some of them still for the future. Our main purpose is to use semionic networks in order to develop artificial minds for intelligent agents. It's interesting to notice that using semionic networks, we are able to depict and design the sign-processing schemes we intend to happen within this artificial mind. Depending on the types of knowledge units the passive semions encode, we will be able to process higher level types of signs, which means higher levels of knowledge and knowledge processing. With semionic networks, we are able to build up different mind architectures, since the already known BDI (Belief, Desire, Intention) Agent Architecture (SINGH ET.AL. 1999, WOOLDRIDGE 1999), NIST RCS Architecture (ALBUS 1999), the AIS Architecture (HAYES-ROTH 1995) and others. Semionic networks are able to implement fuzzy systems, neural networks and evolutionary computation algorithms, and all sort of hybridisms among them (GUDWIN & GOMIDE 1997D), so we can use these theoretical developments as parts and components of our artificial mind. We aim also trying to implement architectures of mind as proposed by great philosophers of the past, like Kant, Heidegger, Wittgenstein, etc. Despite this idea may seem to be a little bit venturous (and maybe naive), some proposals in this sense already succeeded in producing interesting kinds of intelligent systems (BETTONI 1998; BETTONI 2000).

Despite these more ambitious pretensions, we also noticed that our semionic networks are also very good for modeling organization and sign processing within organizations. A very new area of research, called organisational semiotics (ALDERSON ET.AL. 1999; LIU ET. AL.2000) studies the interrelationships between individuals and groups, and between humans and technology, functioning in organisations and society. We believe that the semionic networks will be possibly used as a tool in order to model general organizations within the organisational semiotics field of application. It may be used to model business companies, commercial interactions (including e-business interactions), and not only able to model, but to simulate such organizations, measuring results and eventually allowing us to test alternative configurations for these organizations.

Other potential field of application is under manufacturing context. Flexible manufacturing systems are systems that need to configurate themselves in order to attend rapid changes in production and product lines. Despite very good models (CASSANDRAS 1993) do already exist to model manufacturing systems (including Petri Nets, Automata Theory and others), these models were designed for systems that do not change during time. The semionic networks provide a framework allowing learning and adaptation within a discrete event system framework, that would be very interesting for more complex manufacturing problems.

If we make a more deep analysis of the behavior of a semionic network system, we will see that it gives us a very powerful tool, that is both mathematically and computationally addressed, being very convenient for a multitude of different applications, in engineering, in philosophy and in cognitive sciences. Now, it's just a matter of exploring these potentialities and turning potentiality into facts.

11 References

(ALBUS 1991) J.Albus, "Outline for a Theory of Intelligence", IEEE Transactions on Systems, Man and Cybernetics, vol. 21, n.3, May/June 1991.

(ALBUS 1999) J.S. Albus, "The Engineering of Mind", - Information Sciences, Volume: 117, Issue: 1-2, July, 1999, pp. 1-18.

(ALDERSON ET. AL. 1999) A. Alderson, J. Yap, K. Liu and H.U. Shah, (1999), Relating Organisational Semiotics, Process Modelling and Stakeholder Viewpoints to

Elucidate and Record Requirements, Proceedings of Systems Modelling For Business Process Improvement March 1999, pp. 168- 220

(BETTONI 1998) M.C. Bettoni, "Constructivist Foundations of Modeling - A Kantian Perspective", International Journal of Intelligent Systems, Vol 12, Number 8, August 1998, p. 577-595.

(BETTONI 2000) M.C. Bettoni, "Learning from Kant How To Construct Robots - Suggestions for an Engineering Approach to the Analytic of Concepts", IX. Internat. Kant-Kongress, Berlin, 26.3 - 31.3.2000 - Section 18 - Kant und die Folgen.

(CASSANDRAS 1993) C.G. Cassandras, "Discrete Event Systems: Modeling and Performance Analysis", Aksen Associates Incorporate Publishers, 1993.

(DOEBEN-HENISCH 1996) G. Doeben-Henisch, "Semiotic Machines - an Introduction" - Workshop on Semiotic Machines, Theory, Implementation, Semiotic Relevance, at the 8th International Conference on Semiotics organized by the German and the Netherlands Semiotic Societies, Amsterdam, Aug 5-9 1996 - available on-line at http://www.inm.de/kip/SEMIOTIC/DOEBEN_INTRO/semiotic_cover.HTML

(FANN 1970) K.T. Fann, "Peirce's Theory of Abduction" - Martinus Nijhoff / The Hague, 1970.

(GOLDFARB & NIGAM 1994) L.Goldfarb and. S.Nigam, "The Unified Learning Paradigm : A Foundation for AI", in Artificial Intelligence and Neural Networks: Steps toward Principled Integration, eds. V.Honavar and L.Uhr, Academic Press, Boston, 1994.

(GOLDFARB 1997) L.Goldfarb, "Inductive Theory of Semiotic Systems", Proc. of the ISAS'97 – International Conference on Intelligent Systems and Semiotics, Gaithersburg, MD, USA, Sept.1997, pp. 111-116.

(GONÇALVES & GUDWIN 1999) R. Gonçalves, R.R. Gudwin - "Emotions: A Computational Semiotics Perspective " . Proceedings of 1999 IEEE International Symposium on Intelligent Control, Intelligent Systems and Semiotics - ISIC/ISAS 99, Cambridge, MA, USA.

(GUDWIN 1996) R.R.Gudwin - "Contributions to the Mathematical Study of Intelligent Systems" - Ph.D. Thesis - DCA-FEEC-UNICAMP - May, 1996. (in portuguese) available on-line at http://www.dca.fee.unicamp.br/~gudwin/Publications/thesis.html

(GUDWIN & GOMIDE 1997A) R.R.Gudwin and F.A.C. Gomide, "Computational Semiotics : An Approach for the Study of Intelligent Systems - Part I : Foundations" – Technical Report RT-DCA09 - DCA-FEEC-UNICAMP, 1997. available on-line at http://www.dca.fee.unicamp.br/~gudwin/ftp/publications/rep1_97.pdf

(GUDWIN & GOMIDE 1997B) R.R.Gudwin and F.A.C. Gomide, "Computational Semiotics : An Approach for the Study of Intelligent Systems - Part II : Theory and Applications" - Technical Report RT-DCA09 - DCA-FEEC-UNICAMP, 1997. available on-line at http://www.dca.fee.unicamp.br/~gudwin/ftp/publications/rep2_97.pdf

(GUDWIN & GOMIDE 1997C) R.R.Gudwin and F.A.C.Gomide,"An Approach to Computational Semiotics"- Proceedings of the 1997 ISAS Conference, Gaithersburg, 1997, pp.467-470.

(GUDWIN & GOMIDE 1997D) R.Gudwin and F.Gomide, "A Computational Semiotics Approach for Soft Computing", Proc. of the SMC'97, IEEE International Conference on Systems, Man and Cybernetics, Orlando, FL,USA, Oct. 1997, pp. 3981-3986

(GUDWIN & GOMIDE 1998) R. Gudwin and F. Gomide, "Object Networks – A Modeling Tool" – Proceedings of FUZZ-IEEE98, WCCI'98 - IEEE World Congress on Computational Intelligence, 4-9 May 1998, Anchorage, Alaska, USA, pp. 77-82

(GUERRERO ET.AL. 1999) J.A.S Guerrero, A.S.R. Gomes, R.R. Gudwin, "A Computational Tool to Model Intelligent Systems" - Anais do 4o SBAI - Simpósio Brasileiro de Automação Inteligente, 8-10 September, 1999, São Paulo, Brasil , pp. 227-232

(HAYES-ROTH 1995) B. Hayes-Roth, "An Architecture for Adaptive Intelligent Systems", Artificial Intelligence, volume 72, n. 1-2, 1995, pp 329--365.

(KAKU 1994) M. Kaku - "Hyperspace : A Scientific Odyssey Through Parallel Universes, Time Warps and the Tenth Dimension", Oxford University Press, 1994.

(LIU ET.AL 2000) K. Liu, R.J. Clarke, P. B. Andersen, R.K. Stamper - "Information, Organisation and Technology - Studies in Organisational Semiotics" (Information and Organization Design Series, Volume 1), Kluwer Academic Publishers, Boston, 2000.

(LUBBE & EGBERS) J.C.A. van der Lubbe and E.Egbers, "An Integrated Approach to Fuzzy Learning and Reasoning in Hierarchical Knowledge Structures", Proc. of the ISAS'97 – International Conference on Intelligent Systems and Semiotics, Gaithersburg, MD, USA, Sept. 1997, pp.37-43

(MATURANA & VARELA 1980) H.R. Maturana and F.J. Varela, "Autopoiesis and Cognition: The Realization of the Living", Boston Studies in the Philosophy of Science [Cohen, Robert S., and Marx W. Wartofsky (eds.)], Vol. 42, Dordecht (Holland): D. Reidel Publishing Co., 1980.

(MEYSTEL 1995) A.Meystel, "Semiotic Modeling and Situation Analysis : An Introduction", AdRem Inc., 1995.

(MEYSTEL 1996) A.Meystel, "Intelligent Systems: A Semiotic Perspective" - International Journal of Intelligent Control and Systems, Vol 1, n. 1 (1996), pp. 31-57, World Scientific Publishing Company.

(MEYSTEL & ALBUS 1996) A.M. Meystel, J.Albus, "Intelligent Systems : A Semiotic Perspective" - Report NIST, Intelligent Systems Division, Bldb. 220 Room B124, Gaithersburg, MD 20899, USA, 1996.

(MEYSTEL 1996) - A.M. Meystel, "Intelligent System as an Object of Semiotic Research" - 1996 Biennial Conference of the North American Fuzzy Information Processing Society - NAFIPS - New Frontiers in Fuzzy Logic and Soft Computing - Berkeley, California, USA, June 1996.

(RAM & LEAKE 1991) A.Ram and D. Leake, "Evaluation of Explanatory Hypotheses", Proc. of the 13th Annual Conference of the Cognitive Science Society, Chicago, IL, August, 1991, pp. 867-871.

(LOCKE 1997) J. Locke - "An Essay Concerning Human Understanding" - Penguin Books, 1997

(MORRIS 1947) C.W. Morris, "Signs, Language and Behavior" - New York : Prentice Hall, 1947

(MORRIS 1964) C.W. Morris, "Significant and Significance" – New York – Prentice Hall, 1964.

(MORRIS 1971) C.W. Morris, "Foundation for a Theory of Signs" - in Writings on the General Theory of Signs - The Hague : Mouton, 1971

(NOTH 1995) W. Noth, "Handbook of Semiotics" - Indiana University Press - Bloomington & Indianapolis, 1995.

(NOTH 1996) W. Noth, "A Semiótica no Século XX" - (Semiotics in the 20th Century - in portuguese), Annablume Editora, São Paulo, Brasil, 1996.

(NOTH 1998) W. Noth, "Panorama da Semiótica: De Platão a Peirce" - (A Semiotic Panorama : From Plato to Peirce - in portuguese) Annablume Editora - São Paulo, Brasil, 1998 2d edition.

(PEIRCE 1960) C.S. Peirce - "Collected Papers of Charles Sanders Peirce" - vol I - Principles of Philosophy; vol II - Elements of Logic; vol III - Exact Logic; vol IV - The Simplest Mathematics; vol V - Pragmatism and Pragmaticism; vol. VI - Scientific Metaphysics - edited by Charles Hartshorne and Paul Weiss - Belknap Press of Harvard University Press - Cambridge, Massachussets, 2nd printing, 1960.

(Pospelov 1991) D. Pospelov, "Situational Control: Theory and Practice" - (unpublished translation of the book with the same tittle in russian, from Nauka Publishers, Moscow, 1986)

(Prueitt 1999) P. Prueitt, "Foundations of Knowledge Management for the 21st Century" - BCN Group - (available online at: http://www.bcngroup.org/area3/pprueitt/book.htm)

(Rosen 1985) R. Rosen - "Anticipatory Systems - Philosophical, Mathematical & Methodological Foundations" - Pergamon Press, 1985.

(Rosen 1991) R. Rosen - "Life Itself: A Comprehensive Inquiry Into the Nature, Origin and Fabrication of Life" - Columbia University Press - New York, 1991.

(Sebeok 1997) T.A. Sebeok, "The Evolution of Semiosis", Tutorials 1 – Semiotics - ISAS Conference, Gaithersburg, 1997.

(Singh et.al. 1999) M.P. Singh A.S. Rao and M.P. Georgeff, "Formal Methods in DAI: Logic-Based Representation and Reasoning" in Gerhard Weiss, "Multiagent Systems", MIT Press, 1999, pp. 331-376.

(Von Neumann 1958) J. Von Neumann, "The Computer and the Brain", Yale, January 1958.

(Von Neumann 1966) J. Von Neumann, "Theory of Self-Reproducing Automata" - (edited and completed by Arthur W. Burks) - Urbana, University of Illinois Press, 1966.

(Wooldridge 1999) M. Wooldridge, "Intelligent Agents", in Gerhard Weiss, "Multiagent Systems", MIT Press, 1999, pp. 27-77.

(Zadeh 1994) L. Zadeh, “Soft Computing and Fuzzy Logic”, IEEE Software, vol 11, n. 6, pp. 48-56, 1994.

(Zurada et.al. 1994) J. Zurada, R.J.Marks II, C.J Robinson, "Computational Intelligence - Imitating Life" - IEEE Press, USA, 1994.