DERRIDA'S MACHINES

by Rudolf Kaehr

PART ONE

Cloning the Natural–and other Fragments

Parallelism and Panalogy

PART TWO

Dynamic Semantic Web

Final-Report 27. 07. 2004

(DRAFT-Version)

The TransComputation Institute

ThinkArt Lab Glasgow

rkaehr@btinternet.com

DERRIDA'S MACHINES

PART ONE

CLONING THE NATURAL, PROEMIALITY, PANANALOGY, PARALLELISM

Cloning the Natural–and other Fragments

Some applications

Exploiting Parallelism in PCL-Systems

Minsky´s new Machine

Comparatistics of Models and Metaphors of Machines

PART TWO

DYNAMIC SEMANTIC WEB

Wozu Dynamic Semantic Web?*

Towards a Dynamic Semantic Web

Cybernetic Ontology and Web Semantics

Dynamic Semantic Web

DERRIDA‘S MACHINES PART ONE

Cloning the Natural
– and other Fragments

Cloning the Natural
– and other Fragments

There is no safety in numbers, or in anything else. Thurber

1 A fundamental theory of the natural

If there is anything left in this world we live which is still untouched and natural then it is the naturalness of the natural numbers–and nothing else.

"´Natural´ because they are given at the outset, taken for granted as a founding, unanalizable intuition, outside any critique that might demand an account of how they come or came–potentially or actually–to ´be´." Brian Rotman

„0.5 Le nombre règle les représentations culturelles.
„0.6. Le nombre, évidement, regle l´économie, et sans doute est-ce la ce que Luis Althusser aurait appelé la „détermination en dernière instance" de sa suprématie. L´idéologie des sociétés parlemantaire moderne, s´il y a une, n´est pas l´humanisme, le Droit du Sujet. C´est le nombre, le comptable, la comptabilité."
Alain Badiou, Le Nombre et les nombres. Seuil 1990

And why not Leopold Kronecker?
"God made the integers, all the rest is the work of Man."

As Natural as 0,1,2
Philip Wadler. Evans and Sutherland Distinguished Lecture, University of Utah, 20 November 2002.
"Whether a visitor comes from another place, another planet, or another plane of being we can be sure that he, she, or it will count just as we do: though their symbols vary, the numbers are universal*. The history of logic and computing suggests a programming language that is equally* natural*. The language, called lambda calculus, is in exact correspondence with a formulation of the laws of reason, called* natural deduction*. Lambda calculus and natural deduction were devised, independently of each other, around 1930, just before the development of the first stored program computer. Yet the correspondence between them was not recognized until decades later, and not published until 1980. Today, languages based on lambda calculus have a few thousand users. Tomorrow, reliable use of the Internet may depend on languages with logical foundations. "*
http://homepages.inf.ed.ac.uk/wadler/topics/history.html#drdobbs

Our whole economy of living and thinking would immediately crash, if the slightest change in the nature of natural numbers would occur.

If someone wants to achieve to be nominated as the most cranky mind, the crackpot par excellence, he / she / it should try to prove a paradox or even a defect in the very nature of the natural numbers.

Also such efforts are not unknown, they didn't have any impact on the nature of our natural numbers. It seems to be much more accepted to invent new and deviant logic systems than to change anything in arithmetics.

In my study, I will not touch these tabus. In contrary, I will accept them in all their principality, I even will celebrate them in disseminating them in their whole sacrality.

In doing so, the exclusive nature of the natural numbers will boil down to a very mundane activity in our cultural, that is, artificial world.

The naturality of the natural number system, as we know it, will be entangled in an activity of increasing artificiality of multitudes of natural number systems.

Also there is no culture without numbers, numbers are not cultural, but natural. They are the very nature in / of our culture. To transform this situation will change radically what we will understand by culture. The most advanced development of this classical arithmetical trance of naturality is still the global movement of digitalism and its technology.

In other words, my old question is still virulent: *What´s after digitalism?* (ISEA ´98)

1.1 Natural number series

1.1.1 Natural numbers as models of fundamental abstract systems

"A first attempt at a theory to describe numbers begins with a fundamental abstract type called nat0 as follows:

nat0 =
sorts
nat
opns
zero: —> nat
suc : nat —> nat

Any theory that consists of a signature without any equations is said to be fundamental because it generates all possible strings of symbols without defining any equivalences between the strings. In this particular case the signature contains an arity-zero operation called *zero* and an arity-one operation called *suc*. These operations generate the following infinite series of expressions:

zero, suc(zero), suc(suc(zero)), suc(suc(suc(zero))), ...

in their Herbrand universe of the type.

The only *well-formed* applications of these operators are the constant *zero* itself or succesive applications of the *suc* function beginning with *zero*.

Since there are no equations in this theory, every element is *distinct* and we obtain an *infinite* number of one-element equivalence classes.

One very obvious *interpretation* for the possible elements of the *abstract type* nat0 is the series of denary numbers {1, 2, 3, ...}, setting *zero* equal to 0, *suc(zero)* equal to 1, and so forth, ..." Michael Downward, Logic, p. 181

This is well known, well established and usefull and for some strange reasons it is called *word algebra*. And it offers a stable fundament for the natural number series and all other types of linearly ordered series, too. At least there are enough people who strongly believe in that.

As we see, and will see in the following, natural numbers, despite on being natural, are not naturally accessible in mathematics. They need all sorts of sophisticated notational systems and interpreting mechanisms.

Technically, the natural numbers are accessible and representable only *„up to isomorphism“* and not to some real-world concrete inscriptions. Insofar, there is a structural gap between our intuition of natural numbers and the formalization of this intuition.

Strategy of extensive citations

Because most of the stuff about natural numbers and their formalization is standard I will make extensive use of citations of this material. The way a bring these citations together will give us new insights about the different strategies of formalizing the intuition of natural numbers. Mostly we are reading one of these ways of presentation and not much comparison is done. My approach is not only confronting the different approaches but also putting them together in a new light of mutual contrast and explanations.

The *Stroke Calculus approach* emphasis the aspect of step-wise construction by rules applied to an initial object. This shows us more the internal structure of the type of construction.

The *Set Theory* approach develops an understanding of natural numbers out of a special set theoretical operation, bracket-operation for sets, based on a logical definition of the empty set which in itself is not very self-evident.

In contrast, the *Category Theory* approach emphasis on the external relationships of the constructors and gives us an explication of the intuition of natural numbers up to isomorphism.

All these approaches make it quite clear that the naturality of natural numbers is not as easy captured as it is suggest to be for a natural intuition of a natural object.

Some preliminary questions

What are natural numbers?

„Are „the" intuitive natural numbers categorical? That is, is the description of natural number as clear and definitive as we usually take it to be?

This was no idle question for Frege who in the Foundation of Arithmetics attempted to achieve an absolute and clear description of the natural numbers. Any denial of categoricity has importand consequences.

Whenever we define a class of mathematical objects via inductive definition and then proceed to establish results about objects in that class we make tacit use of properties of certain functions." Isle p. 111

What is the importance of the natural numbers?

As the citations show, natural numbers are of importance on the very base of our culture and technology.

Natural numbers deliver the prototype of any constructivist theory, even of any theory of construction.

How are natural numbers notated?

Numbers, numerals, marks, ciphers

What is natural to natural numbers?

What would be unnatural for natural numbers?

gaps, multitudes, obstacles, neighbors

1.1.2 Natural numbers in a constructivist Stroke Calculus

In a stroke calculus the representations of natural numbers are produced by three rules.

R1) Write down a stroke |;
R2) Given a set of strokes (call it X) write down X|.
R3) Now apply R1 once and then apply R2 again and again.

Interpretation
Set | as 1
Set || as 2
Set ||| as 3, and so on.

"An understanding of the "structure of the natural numbers" thus consists in an understanding of these rules. But what has actually been presented here? Rules R1 and R2 are fairly unambiguous, in fact, one could easily use them to write down a few numerals.

But rule R3 is in a different category. It does not determine a unique method of proceeding because that determination is contained in the words "apply R2 again and again".

But these words make use of the very conception of natural number and indefinite repetition whose explanation is being attempted: in other words, this description is circular." Isle, p. 133

Even if we accept this criticism of the rules, we have to accept, that rule R2 demands some preconditions, at least, we have to add the new stroke in line with the other strokes, and not somewhere else, e.g. behind the blackboard. If we use the quite misleading terminology of sets in rule R2, the new stroke has to be written in the domain of the set and not outside of it.

But why should we accept that, if it is not explicitly asked? Therefore, the game is not so clear as it should be. The presupositions of the stroke calculus is linearity of repetition and atomizity of its strokes, short: semiotic identity.

All these presupositions may not be very natural, they are not pre-given, we simply have to learn them, that is, to internalize them by education.

We may interpret the stroke calculus as an example which starts the numerals with the initial object 1and has two rules as constructors to construct the object, that is any natural number.

1.1.3 Dialogical foundation of the natural numbers

This constructive approach of the *Stroke Calculus* can be made much more explicit and more adequate in formalizing the intuition of natural numbers in the framework of the dialogical approach. Especially the 3 rule has a more advanced treatment in the dialogical setting explained by Lorenzen.

This type of construction is more a type of reconstruction then a construction ab ovo of the natural numbers. With this distinction we have a more explicit idea of the process of formalizing an intuitive idea of the natural numbers because the intuitive knowledge of natural numbers is directly confronted with the formalism.

To ask if 10^{10} is a number means two things, first I have an intuitive knowledge about 10^{10} to be a number and second, I have a formalism to answer the question in applying rule 3 as long as I need to construct 10^{10} or to get an agreement with my proponent in the dialog about the construction of the intuitive number 10^{10}. Intuitive means in this case that I have a notational system to write my supposed number but I don´t have a procedure to produce this number, therefore it is the task of the proponent to use his formalism.

This setting also tries to escape the circularity of the situation, to ask if 10^{10} is a numbers presuppose that this object is a number. To ask if 10^{10} is a number means to ask for a procedure to produce step-wise without violating the intuition of counting the desired number 10^{10}.

To put this situation into a dialog between opponent and proponent seems to be an explication as an interlocking game between intuition as pre-understanding and formalism as construction which escapes the purely formalist thematization which results in circularity.

But all that doesn’t mean that the game between intuition and formalisms has stopped and we are now in possession of an ultimate formalism which corresponds to the very intuition of natural numbers. The results are limitations, as we can learn from Kurt Gödel, there is no strict formalism which would be able to formalize in full the idea of natural numbers, what means, the idea of infinite induction which is postulated in the "and so on" of rule 3 of the stroke calculus.

Look at: http://www.ltg.ed.ac.uk/~zinn/Colosseum/MetaDL.html
also: http://plato.stanford.edu/entries/logic-games/

Diagramm 1 **Lorenzen´s dialogs**

1.1.4 Semiotics of natural numbers

1.1.5 Natural numbers in set theory

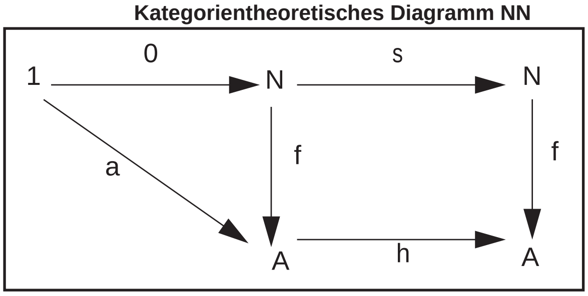
We may interpret the set theoretical definition of the numbers as an example which starts the numerals with the initial object nil, the *empty set*. That is, with a negation and with non-existence (in set theory). Thus, the empty set is defined internally, and not from an external point of view, by its attribute to be the set of all objects which are not identical with themselves. The presupposition clearly is, that there is no such object in the universe.

Again, there is no escape, circularity is at the very beginning.

1.1.6 Natural numbers in Category Theory

We may interpret the category theoretical definition of the numbers as an example which starts the numerals with an initial object without giving any information of its internal structure, but only about its external relations to other objects.

Diagramm 2 **Kategorientheoretisches Diagramm NN**



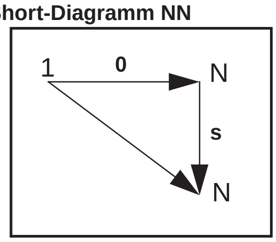
A natural number object consists of an object and two morphisms

0: 1 —> N

s: N —> N

such that for all objects A and all morphisms a: 1 —> A, h: A —> A there exist a unique morphism f: N —> A making commute the diagramm NN.

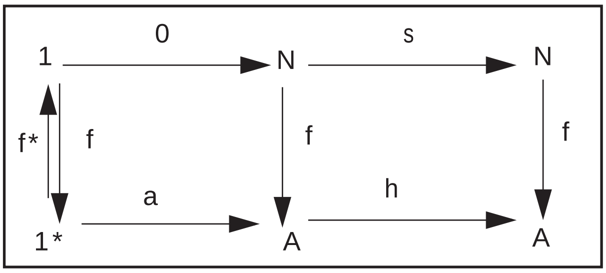
Diagramm 3 **Short-Diagramm NN**



Terminal Objects

An important fact is that any two terminal objects (as well as any two initial objects) in a category are uniquely isomorphic. In other words, if T and T‘ are two terminal objects, then there is a unique isomorphism between the two. Because of this, it is customary, to collapse all terminal objects into a representative and talk about the terminal object.

Diagramm 4



As we see, 1 and 1* are isomorphic in respect to f and f*.

up to isomorphism

„The categorical approach to characterize objects and morphisms in terms of their relation to other objects and morphisms has the particular consequence that universal properties specify objects only „up to isomorphism“*.*

Definition: Objects A and B are isomorphic if there exists morphisms f: A —> B, f*: B —> A such that f*.f=iA and f.f*=iB

1.1.7 Natural numbers as numerals in an arithmetical game

Goodstein
Bishop

1.1.8 Natural numbers as visitors of different modi of thematizations

1.2 Natural numbers and computability

Church´s hypothesis as a possible natural law

“We offer this conclusion at the present moment a a *working hypothesis*. And to our mind such is Church´s identification of effective calculability and recursiveness.
The success to the above program would, for us, change this hypothesis not so much to a definition or an axiom but to a *natural law*.
Only so, it seems to the writer, can Gödel´s theorem concerning the incompleteness of symbolic logics of a certain general type and Church´s result on the recursive unsolvability of certain problems be transformed into conclusions concerning all symbolic logics and all methods of solvability.”

“Actually the work already done by Church and others carries this identification considerably beyond working hypothesis stage. But to mask this identification under a definition hides the fact that a fundamental discovery in the limitations of the mathematicizing power of Homo Sapiens has been made and blinds us to the need of its continual verification.”
Emil Post, 1936
(binds?)

2 Cloning naturality

Today it seems that there is no reason to not to clone and replicate the naturality of the natural numbers with their ultimate Herbrand universe.

What is pre-given and natural, should be replicated to loose its magics.

To be modest I start with the replication of the Herbrand universe into only 3 clones. Each clone has its own word algebra with all its distinctions. But the distinction between the original system and its clones disappears in the sameness of the systems. These clones are not models of the abstract system, they are abstract in themselves. Insofar they are all the same, and we have lost the original system with which we started. That is, the original is only original for its role as a starting point of the process of replication. As products all these systems are the same.

The metaphors *clone* and *cloning* will help us to surpass the dictatorship of identity on all levels of our thinking and writing. The idea of *sameness* as logically different from identity and diversity and also not rooted in them, will lead our thoughts of distributing and mediating systems to realize a construction of cloning the naturality of the natural numbers. The poly-contextural approach to the new category of sameness, with its ontological, logical, semiotical and arithmetical consequences, goes far beyond such concepts like „multi-sets" which are commonly used for describing replication in biological systems.

The construct $nat0^{(3)}$ denotes the 3-fold replication of the abstract type nat0. Because these clones of the Herbrand universe are living together we call there spared space ultra-Herbrand multi-verse or simply their *multi-verse*. Also the replicands are all the same but not identical they can be distinguished and are therefore countable, and here we have 3. For separating the replicas and for bringing them together in their multi-verse I introduce the operation of dissemination DISS. In other words, the operator of dissemination DISS produces a distribution and mediation of the systems under consideration.

Desedimenting artificiality

The complementary movement of the process of cloning the naturality of natural numbers is given by the idea, that our natural numbers are not so natural as we have learned to think or believe. It is equally reasonable to think that the unizity of our natural numbers is the result of a powerful squashing and squeezing together the multi-linearity of numbers by force to the uni-linearity as we know it. Therefore we have a chance for a desedimentation and deliberation of the numbers from the terrorism of linearity to a free play of writing opening up not only a multi-linearity of numbers but a „living tissue".

This idea is easily supported by Aristotle´s condemnation and fight against Platonist and Pythagorean ideas of numbers.

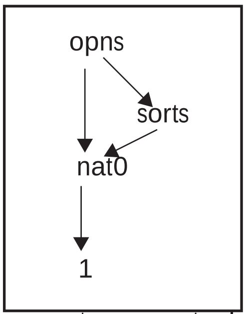
But also by the historical movement of creating a monetary equivalence between different money systems.

2.1 The conceptual graph of the abstract object nat0

To explain, in a first step, the concept of distribution and mediation, i.e. the concept of dissemination of formal systems, I introduce the notion of a *conceptual graph*. A conceptual graph shows the conceptual dependency structure, e.g. the dependency structure of the notions of a system or an abstract object.

It will offer us a practical tool to construct the chiastic structure of disseminated objects in general.

Diagramm 5 **graph of nat0**



The arrows in this diagram represents conceptual dependencies in the notion of nat0. The notation

opns —> sorts

for example, means that:

the concept of opns varies as the concept of sorts varies.

In particular, it means that the concept of opns, the one that we have in mind, cannot be independent of the concept of sorts and neither can a particular opn be independent of its particular sort.

The notation

sorts —> nat0

means that the concept of sorts varies as the concept of nat0 varies.

Therefore the notion of opns varies as the notion of nat0 varies:

opns —> nat0.

In a conceptual diagram, 1 represents the absolute. The notion

nat0 —>1

expresses that the nat0 notion is absolute, for it tells us that the nat0 notion varies as the absolute varies – which is not at all.

Normally the notion of the absolute is not included in the definition of an abstract object like nat0, simply because we presuppose that there is anyway one and only one such concept of an abstract object. But for the purpose of disseminating abstract objects it is exactly this part of the definition of abstract objects which has to be deconstructed, i.e to be distributed and mediated. Abstract objects have not to be confused with the multitude of models of abstract objects as concretizations of the abstraction into the concrete world of formal and not formal objects and entities.

In other words an abstract object, in this sense, is an institution.

3 Unicity, Intuition and Explication

3.1 Aspects of the interplay between intuition and formalism

Intuition is deeper than formalism

"Hower much we would like to ´mathematize´the definition of computability, we can never get completely rid of the semantic aspect of this concept. The process of computation is a linguistic notion (presupposing that our notion of language is sufficiently general); what we have to do is to delimit a class of those functions (considered as abstract mathematical objects) for whichexists a corresponding linguistic object (a process of computation)." Mostowski, Thirty Years of Foundational Studies, 1966, p. 33

„Truth is invariant under change of notation." (Goguen

Formalisms are more powerfull than intuition

"I think it is fair to say that most mathematicians no longer believe in the heroic ideal of a single generally accepted foundations for mathematics, and that many no longer believe in the possibility of finding "unshakable certainities" upon which to found all of mathematics." Goguen

Writing beyond intuition and formalism

What´s the base of intuition?
Egological foundation of intuition (Husserl, Brower)

4 Dissemination: Introducing the proemial relationship

There are many ways of combining abstract objects or institutions. *„For example, given two institutions INS1 and INS2 which, intuitively, are independent we can form their product. This product institution has all pairs of signatures from INS1 and INS2, respectively, as models, and sentences which are either sentences from INS1 or from INS2 with the obvious satisfaction relation."* Cat., p. 357
It is shown, that *the category of institutions is complete*.

The idea of dissemination tries to explicate and formalize a quite different intuition of combining institutions which is not producing diversity and multiplicity by combining a basic system as a product or sum or whatever construction but introduces multiple differences in the very concept of the basic system itself. After this construction a polylogical or polycontextural system can be combined in many ways. This idea of multitudes of basic differences in the notion of formality, taken seriously, is in fundamental contrast to the existing concepts of formality in mathematics. Obviosly, these multitudes are more fundamental than all types of many-sorted theories, typed logics or pluralities of regional ontologies, domains and contexts.

4.1 The idea of proemiality

A very first step in this direction was made by the philosopher Gotthard Gunther with his idea of a *„proemial relationship"* introduced in his paper „Cognition and Volition" (1970) about a Cybernetic Theory of Subjectivity.

„In order to obtain a general formula for the connection between cognition and volition we will have to ask a final question. It is: How could the distinction between form and content be reflected in any sort of logical algorithm if the classic tradition of logic insists that in all logical relations that are used in abstract calculi the division between form and content is absolute? The answer is: we have to introduce an operator (not admissible in classic logic) which exchanges form and content. In order to do so we have to distinguish clearly between three basic concepts. We must not confuse

a relation

a relationship (the relator)

the relatum.

The relata are the entities which are connected by a relationship, the relator, and the total of a relationship and the relata forms a relation. The latter consequently includes both, a relator and the relata.

„However, if we let the relator assume the place of a relatum the exchange is not mutual. The relator may become a relatum, not in the relation for which it formerly established the relationship, but only relative to a relationship of higher order. And vice versa the relatum may become a relator, not within the relation in which it has figured as a relational member or relatum but only relative to relata of lower order.

If:

R_{i+1}(xi, yi) *is given and the relaturn (x or y) becomes a relator, we obtain*

R_i *(xi-1, yi-1)* *where* $Ri = x_i$ *or* y_i. *But if the relator becomes a relatum, we obtain*

R_{i+2}*(xi+1, yi+1)* *where* $R_{i+1} = x_{i+1}$ *or* y_{i+1}. *The subscript i signifies higher or lower logical orders.*

We shall call this connection between relator and relatum the 'proemial' relationship, *for it 'pre-faces' the symmetrical exchange relation and the ordered relation and forms, as we shall see, their common basis."*

„Neither exchange nor ordered relation would be conceivable to us unless our subjectivity could establish a relationship between a relator in general and an individual relatum. Thus the proemial relationship provides a deeper foundation of logic as an abstract potential from which the classic relations of symmetrical exchange and proportioned order emerge.

It does so, because the proemial relationship constitutes relation as such; it defines the difference between relation and unity - or, which is the same - between a distinction and what is distinguished, which is again the same as the difference between subject and object.

It should be clear from what has been said that the proemial relationship crosses the distinction between form and matter, it relativizes their difference; what is matter (content) may become form, and what is form may be reduced to the status of mere „materiality"."

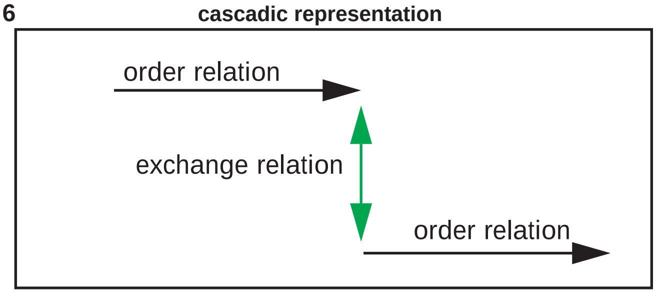
„We stated that the proemial relationship presents itself as an interlocking mechanism of exchange and order. This gave us the opportunity to look at it in a double way. We can either say that proemiality is an exchange founded on order; but since the order is only constituted by the fact that the exchange either transports a relator (as relatum) to a context of higher logical complexities or demotes a relatum to a lower level, we can also define proemiality as an ordered relation on the base of an exchange. If we apply that to the relation which a system of subjectivity has with its environment we may say that cognition and volition are for a subject exchangeable attitudes to establish contact but also keep distance from the world into which it is born. But the exchange is not a direct one.

If we switch in the summer from our snow skis to water skis and in the next winter back to snow skis, this is a direct exchange. But the switch in the proemial relationship always involves not two relata but four!" Gunther

4.2 Some explanations of the idea of proemiality

The proemial relationship is therefore at first an interlocking mechanism of the two concepts of exchange and order or symmetry and asymmetry.

Diagramm 6 **cascadic representation**

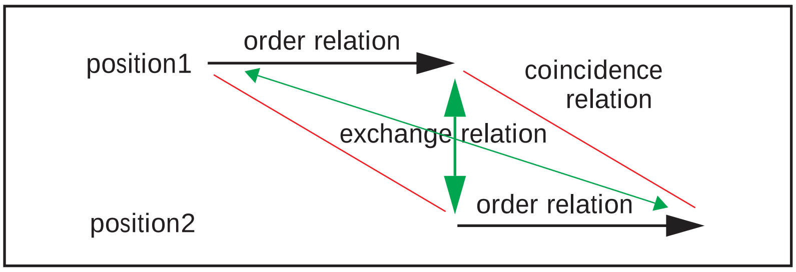


A further explication of the intuition of proemiality is achieved if we consider the fact that the objects, the relator and the relata of the relations, have to fit together in a categorical sense. There is a similarity of the relators of different levels as well as for the relata of different levels in the sense that the different relators are relators and not something else. And the relata on each level are relata and not relators. For that I introduce the *coincidence relation*, which designates categorical sameness (likeness, similtude).

To finish the picture I introduce the exchange relation between the „first" and the „last" element of the interlocking mechanism of order and exchange relations. As a last step I mention the position, the *logical locus*, of the order relations according to the *„higher or lower logical orders".*

PrObj = (Obj; Ord, Exch, Coin, Pos)

Diagramm 7



But this explanation still excludes the third term of the definition of a relation, the relation itself. Remember: *We must not confuse a relation, a relationship (the relator), the relatum.*

And finally I consider the fact that there is one and only one concept of relation and relationality under consideration. therefore the concept of relation is based on *unizity*, represented by 1. This is surely not a harmless statement, it suppose something like a common intuition of relationality or operativity which finds itself explained and formalized in some mathematical constructions which are accepted by the scientific community. Therefore, Gunthers chain "a *relation, a relationship (the relator), the relatum"* has to be completed by the very concept of relation, that is, *relationality* based in *unicity* (uniqueness, sigularity).

The full-fledged explanation, without the arrow "relation—>relationality", of the proemial relation over two loci is given by its conceptual graph. The scenario is the same for the distribution and mediation of other concepts, like operations, functions, categories, institutions etc.

A further concretization of the theory of proemiality would be achieved with the study of the structure between the different contextures, that is the structure of the distribution of the different loci, symbolizing singularity. We would have to deal with the distribution of the singularities over the kenogrammatical systems (grids) of proto-, deutero- and trito-structure. This would allow to introduce kenogrammatical differences between the disseminated contexturalities. Insofar the contexturalities are studied in their neutrality characterised by their singularity.

Thus the definition has to be expanded to:

PrObj = (Obj; Ord, Exch, Coin, Pos)

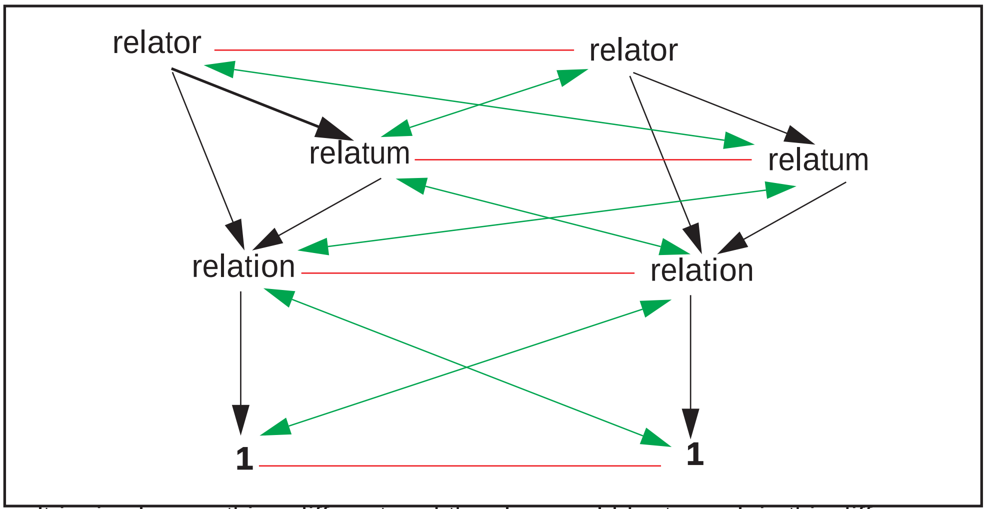
with Obj = { relator, relatum, relation, relationality, unizity}

In this context it is not my task to defend this construction against the many attempts to reduce it to something else. To go further in the game I make the option that it will be useful for developing some new mechanisms of combining abstract objects like institutions, logics, arithmetics, category theories and more. In exercising this game the new intuition will shape itself into a more academic form.

After having introduced the idea of proemiality it would be possible to formalize it further and to develop a preliminary theory of proemiality, also sometimes called *chiastics* or *theory of mediation*.

The main thesis, therefore, is that proemiality offers a mechanism of combining institutions which doesn't belong to the universe of combining categories.

This mechanism of combining institutions, e.g. distribution and mediation, is fundamentally different from the classical ones. Despite of this difference this strategy is in no contradiction or opposition to the known principles of combining systems of logics.



It is simply something different and the clou would be to explain this difference in full.

Don´t confuse the exchange of relator and relatum of a relation in the mechanism of the proemial relationship with the superposition of relator and relation in relational logics. There is no problem to apply a relator, or a operator or a functor to the result of a relation or operation or function as e.g. in recursion theory or in meta-level hierarchies.

Metaphor

If we proemialize the linguistic subject-object-relation of a sentence we shouldn't hesitate to be strictly structural.

The example is borrowed from Heinz von Foerster.

"The horse is gallopping" (Das Pferd gallopiert), the interchanged sentence can only be "The gallop is horsing" (Der Gallop pferdet).

Nobody supposed that we are doing analytic philosophy.

4.3 Proemiality and Architectonics

4.3.1 About the as-category in proemiality

What I have developed so far is only the half of the story. Also it might be obvious that the wording of e.g. “the operator (of one system) becomes an operand (of another system)” is in strong conflict with the identity of its terms therefore this situation needs a more precise explication. It should be clear that a term which is in one system an operator and simultaneously an operand in another systems is split in its own identity. It is at once itself and something else. This term has at once two functions, to be an operator and to be an operand. Therefore, from the point of view of identity and its logic, this term is in itself neither an operator nor an operand.

What then is it? How can we define it more accurate? It is part of a chiastic interplay and we have to be more explicit with our wording. Instead of speaking of an “operator” or of an “operand”, we should use the *as-category* and use the wording “an object X *as* an object Y *is* an object Z”. Thus, an operator as an operator is simultaneously an operand.

An operator as an operand is an operand (of another operator)

Metaphors

I as myself and I as another.
The other as itself and the other as another (e.g. myself).

All this stuff I have developed in extenso in my german texts (...).

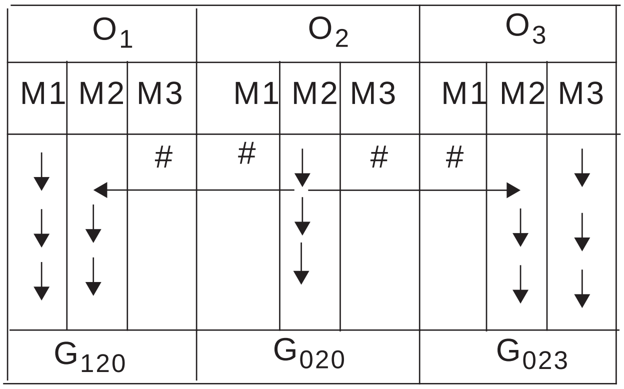
4.3.2 About the architectonics of the as-category

To make this wording more precise I introduce a diagram which is well known from the tableaux method of formalized polycontextural logic.

This type of diagrams was first introduced to deal in a proper way with the tableaux method in polycontextural logic. Especially to understand the functioning, and this gives probably also some light on its meaning, of the so called *transjunctions*, I introduced this tabulation of the step-wise decomposition of signed formulas in tableaux proofs.

The term *transjunction* has reached in different scientific and artistic areas some degree of acceptance and is widely used as an important mechanism of subversive thinking and modeling. Also the number of occurrence of this term in literature is quite impressive there is not much scientific understanding to find.

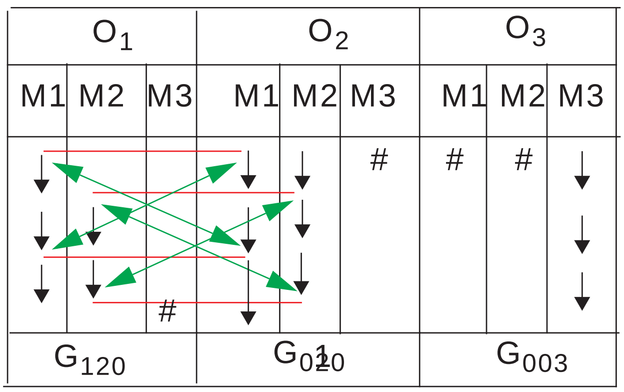
Transjunctions are logical functions or operators which are involved in some sorts of



bifurcations and are split into different parts belonging at once to different logical systems. They are therefore composed of partial functions in contrast to the total functions of classical logical junctions like conjunction, disjunction, implication and so on.

This change of logical system by bifurcation which presuppose the difference of an inside and an outside of a logical system is ruled by the proemial relation between the parts of the transjunction and the different logical systems involved. To the step-wise decomposition of a transjunctional formula corresponds an order relation, to the bifucation to other systems the exchange relation because of its inside / outside difference, and to the components and the steps of decomposition of the transjunctional formula as a whole the relation of coincidence. Therefore, the operation of transjunction can be understood as a proemial object.

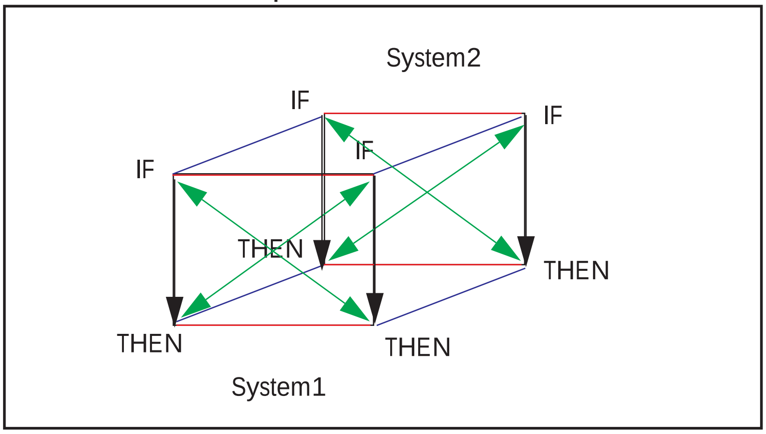
This diagram which gives some first steps in the design of polycontextural architectonics can now be used for further explications of the mechanism of proemiality.



The exchange between operator and operand has to be described simultaneously from both positions. That is why we have to realize a double description, a double gesture of inscribing the proemiality of the constellation. To visualize this prozedure we have to realize a double description of the diagram

The first diagrams are correct insofar as they describe the *structure* of proemiality. But at the same time they are abbreviations insofar as the *process* of reading them, that is to read them at once from both sides, is not inscribed. This process of reading has to be done by a reader. But we have to make it explicit and to visualize it. Therefore, even if it seems to be obvious, it has to be realized and not only be mentioned. The new diagram is focussing more the process of proemiality than on its general structure. To not to overload the scheme I reduced it to the distribution of the IF / THEN-relation. Maybe with all that in mind we are now reaching slowly the famous *proemial cube*.

Diagramm 8 **The proemial cube**



Again, the green double arrow represents the exchange relation, the red line the coincidence relation, the black arrow the order relation, and, new, the blue line represents the distribution of the two proemial relations in a common *architecture*.

I don´t comment the full combinatorics between all knots of the diagram. Also, I would like to leave the study of further dimensions of visualizations and their explanations as an interesting job to programmers. In this text DERRIDA´S MACHINES I will reduce my presentation to the graphically more simple case of the visualization of the structure of the concept of proemiality and its applications, that is, to the two-dimensional diamond diagram instead of the cube.

4.4 Proemiality and Heterarchy in a UML Framework

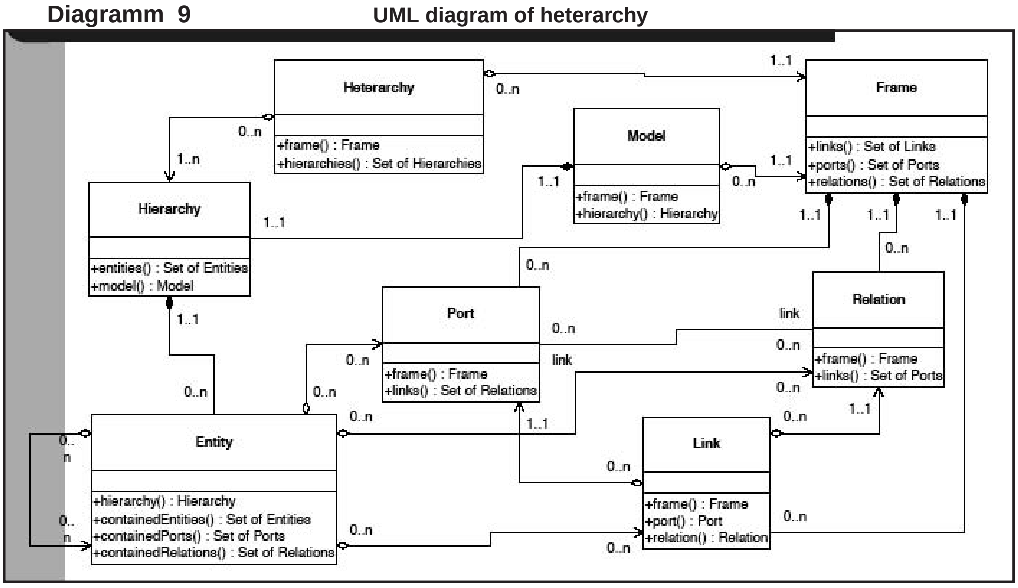
To give a more transparent modeling of the proemial relationship it maybe helpful to set the whole construction and wording into an UML diagram and to use the modeling of heterarchy worked out by Edward Lee as a helpful tool to explicate proemiality in terms of UML modeling.

Also the proemial relationship is not restricted to ontology and the distribution of hierarchical ontologies in a heterarchic framework and despite the fact that UML has no mechanisms of category change, metamorphosis and mediation it seems to be a helpful exercise to find a correspondence between the UML heterarchy diagram and the construction of proemiality which is more based on elementary terms of relationality. The heterarchy diagram is a class diagram which models the static structure of the system. Proemiality has, also it is fundamentally dynamic, its static aspects. It is this static aspect we can model with the help of the UML heterarchy diagram.

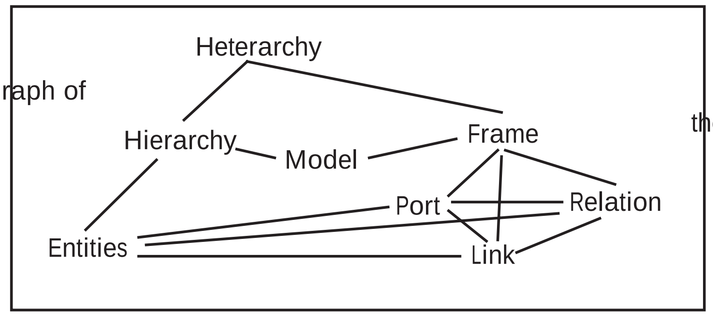
A further step of UML modeling of proemiality will have to involve more dynamic models like interaction and activity diagrams.

What is the difference in modelling between conceptual graphs and UML diagrams?

Diagramm 9 **UML diagram of heterarchy**



in: Edward A. Lee, Orthogonalizing the Issues, UC Berkeley



A conceptual graph of the UML heterarchy diagram.

Heterarchy
Hierarchy, Frame
Model
Port, Relation
Link
Entity

Polycontexturality
Mono-contexture, Proemiality
Type of Proemiality
Type of Metamorphosis, Relations: Order-, Exchange-, Coincidence
Transjunction
Objectionality

Port:: loci of mediation
Set of entities

To explain proemial metamorphosis between hierarchies I introduce the usual distinction of Urelements and sets as the two disjunct types of entities.

Another method would be to simply distinguish the levels of the hierarchy, especially the root and the rest.

Hierarchy

To each relation or contexture corresponds a hierarchy. Hierarchies are structured by order relations.

Frame

Port

The switches between hierarchies are realized by the *links* between *ports*. Ports are the meeting points of the links. (eg.Operatori <——> Operandj).

Relation

The way these switches are realized is defined by the *relation* between the ports. In the case of proemiality these relations consists of the exchange and the coincidence relations.

Model

Link

Entity

The entities of the proemial relationship between contextures are obviously the *terms* of the formal language in use (operator, operand, operation, uniqueness).

Concerned more with the content of the relations in a contexture the entities are the intra-contextural elements or terms of these releations., e.g. x, y in R(x,y).

4.5 An example: "Beyond Substance and process"

• 1 Metarules

"Metarule CAs introduce the required openness by postulating a hierarchy of CA rules. Each CA at a particular level in the hierarchy has a finite lattice, a finite number of states and a finite number of rules. However, there are two ways in which metarule CAs extend the standard CA definition:

(1) rules at level p are states at level p+1 in the hierarchy, where p>0.

(2) initial conditions are independently specifiable at each level p in the metarule hierarchy.

The scheme may be expressed formally as follows:

In our terminology, definition (1) has a simple translation: rules are operators and states are operands distributed over different levels. The second rule is of great importance, because it stipulates autonomous beginnings, "initial conditions", at each level. This is an important feature to distinguish this type of hierarchy from other well known, more type-theoretical hierarchies. This fits perfectly together with my dogma *"There are multitudes of beginnings and endings, without any origin."*

• 2 Beyond Substance and Process ...

"One possible objection to this scheme is that it is ontologically dualistic at the lowest level in the hierarchy (states and rules) and ontologically monistic at all other levels (rules and metarules). This problem may be overcome by extending the framework to a bidirectionally-infinite hierarchy in which states at level m are rules at level m-1 and rules at level m are states at level m+1 where -¥ m ©¯ +¥.

Such a framework replaces the dualistic ontology of state and rule, and their corresponding physical counterparts, substance and process, with a monistic ontology based on an instance of a more general kind."

• 3 Another Approach

Ali / Zimmer are using the set of rules from one level of the hierarchy of cellular automata to define the automata on a next level. This approach produces an exponentation of the quantitative complexity of the apparatus and accepts the basic rules of identity of the objects of the CA at each level.

Another, more holistic approach, is given, with the morphogrammatic abstraction applied to the set of the rules. The new level is then defined by morphograms which are beyond semiotical identity.

As a consequence of the morphogrammatic abstraction we will not have an endless hierarchy of levels. This endless iteration of levels is not excluded in our approach. For all morphogrammatic systems there are, from a secondary point of view, iterations in the modus of identity.

For example. In two-valued propositional logic we have a set of two truth-values and with two variables we have a set of 16 logical function, represented by 16 junctional operators, the junctors. Instead of taking this 16 junctors as the new set of "values" on a higher level for new operations between them, we take these 16 logical operators and reduce them to 8 by the morphogrammatical abstraction. We speak of 8 morphograms. And these morphograms don't preserve the principle of identity of the truth-values. To deal with them, which means, to transform them into each other, we introduce only a few new morphogrammatical operators, e.g. the reflector.

The proemial hierarchy of polycontextural logics

„However, if we let the relator assume the place of a relatum the exchange is not mutual. The relator may become a relatum, not in the relation for which it formerly

established the relationship, but only relative to a relationship of higher order. And vice versa the relatum may become a relator, not within the relation in which it has figured as a relational member or relatum but only relative to relata of lower order. If:

R_{i+1}(xi, yi)

is given and the relaturn (x or y) becomes a relator, we obtain

R_i *(xi-1, yi-1)*

where Ri = x_i *or* y_i*. But if the relator becomes a relatum, we obtain*

R_{i+2}*(xi+1, yi+1)*

where $R_{i+1} = x_{i+1}$ *or* y_{i+1}.

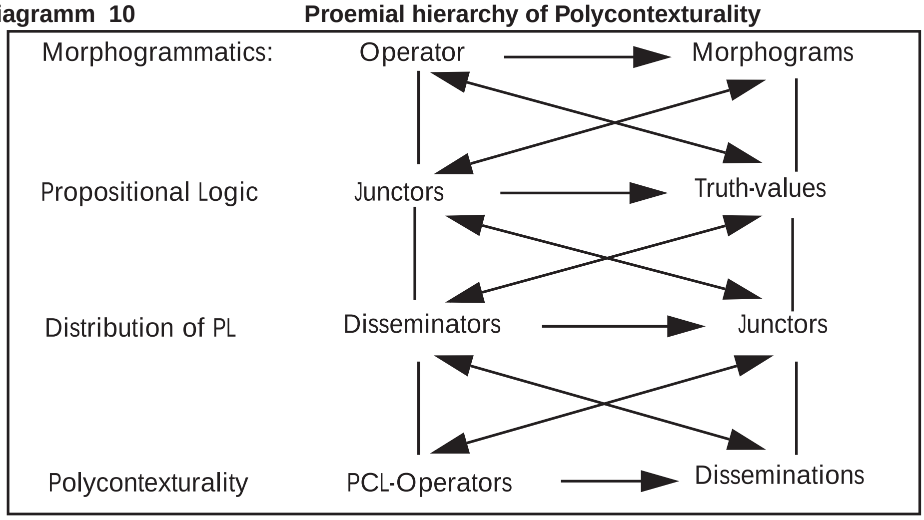
The subscript i signifies higher or lower logical orders.

We shall call this connection between relator and relatum the 'proemial' relationship, for it 'pre-faces' the symmetrical exchange relation and the ordered relation and forms, as we shall see, their common basis.“ Gotthard Gunther, Cognition and Volition

This hierarchy of logical orders are not to be confused with the hierarchy of operators and operations as in the theory of types or the meta-language concept. The exchange happens between operators and operands and not between operators and the operation as a result of the application of operators to their operands like in recursive number theory or recursive formulas.

Presupposing the terminology of operators and operands or any other dichotomic order, e.g. relator / relatum, rule / statement, the PCL framework can be put into a proemial order which can be seen as a new type of hierarchical order in the following sense.

Diagramm 10 **Proemial hierarchy of Polycontexturality**



The junctors of propositional logic like conjunction, disjunction, implication but also negation becomes operands of the dissemination process, ruled by the operators of dissemination, distribution and mediation. This defines the homogeneous polycontextural logics as multi-negational systems.

The PCL-rules of identity, permutation, reduction and bifurcation over the network of distributed junctors produces the full PCL system with its transjunctions. It is heterogeneous, multi-negational, transjunctional. The transjunctions of the full PCL system are embedded in the extended system of morphograms which includes the morphograms of the system of transjunctions.

Additionally we can understand the classical junctors as having their foundation in some morphograms of the system of morphogrammatics. Classical junctors as operators are embedded in the domain of (their) morphograms. The morphogrammatic system itself has its foundation in itself, because the morphograms are the (re)presention of their own operators. Here the distinction of operator and operand is in some sense obsolete.

From a proemial point of view there is no need for an infinitary approach of levels. There is also no need for a monistic ontology. The proemiality of operator and operand or of rules and states is neither an operand nor an operator but the foundation of both.

From multi-level to one-level and zero-level ontologies

„Thus the proemial relation represents a peculiar interlocking of exchange and order. If we write it down as a formal expression it should have the following form:

$$\square \; R^{pr} \; \square$$

where the two empty squares represent kenograms which can either be filled in such a way that the value occupancy represents a symmetrical exchange relation or in a way that the relation assumes the character of an order." Gunther, p. 227

Proemiality is an interlocking mechanism of typed and zero-typed languages. A zero-typed (keno-typed) language is not a non-typed or a one level typed language but a language beyond the distinction of operator and operand as the base of types and typed languages.

4.6 Complementarity of dissemination and togetherness

Complementary to the notion and procedure of dissemination, which is motivated by the necessity of constructing complex and polycontextural systems out of simple ones, that is, mono-contextural systems, we have to consider the poly-contexturality of the complex system as such. One first category we observe is the category of *togetherness* of the local systems in the complex and inter-acting wholeness.

Another category that emerges naturally out of the disseminated systems is the category of *wholeness* or more precise the category of *super-additivity* of disseminated systems.

In this sense, dissemination is a process of disseminating single systems and at the same time it is the wholeness, the togetherness of the disseminated systems. This is also included in the notion of dissemination as a process of distribution and mediation of systems. Dissemination is always both: multitude and wholeness.

5 Combinatorics of chiastic changes of categories

5.1 Conservative mappings or Category theoretic combinations

If the contextural differences between two objects are denied we can *model* the relationship between them in terms of morphisms in the category theoretic sense.These morphisms are the structure preserving mappings of names to names, sorts to sorts, operations to operations, equalities to equalities, and unity to unity., etc. of the abstract objects. But again, in this case we are neglecting the fact, that they belong to different logical contextures.

On the other hand, if we take their contextural differences into consideration, these mappings are preserving the tectonical structure of the systems, despite their logical incompatibility. In terms of proemiality these mappings are not of the sort of order relations, like morphisms, but of the sort of coincidence relation. In a category theoretical model they would be some identity morphisms or isomorphisms.

5.2 Metamorphosis or Proemial combinations in abstract objects

• 1 Chiasm of sorts and names: CHI (sorts, names)

This is similar to the chiasm of sorts and the universe (of sorts) in a many-sorted logic.

It seems not to be unnatural that a sort can change into a name of a new object and on the other side a name as being hierarchically superior to the sorts can change into a lower level object as a sort in another contexture.

But this seems to be an ordinary procedure for interacting systems. The conceptualizing process of different agents can differ exactly in the sense that for one agent the set of sorts or of one of the sorts of the other agent corresponds to the name, that is, the whole or contexture of his own system. In contrast, what is the whole scope of one agent can be a sort with many other sorts for another agent. There is nothing magic with that. And there is also no reason for unsolvable conflicts if both are aware about this situation and understand the mechanism of change between each other. This common understanding can be modelled or realized in a further system, without being forced to negate the differences between the two agents.

Sorts and names occurs on different levels of the conceptual hierarchy. The mechanism is generalization and reduction or specialization of concepts.

• 2 Chiasm of sorts and operations: CHI (sorts, opns)

• 3 Chiasm of operations and equations: CHI (opns, eqns)

• 4 Chiasm of names and operations: CHI (names, opns)

• **5 Chiasm of names and equations: CHI (names, eqns)**

• **6 Chiasm of unizity and names: CHI (unizity, names)**

Unizity can be understood as the *contexture* of the local abstract algebra. Classical theories have not to be concerned with their contexture and unizity because they are unique per se, that is they are mono-contextural. Because of their uniqueness there is no reason to notify it by a special term like 1.

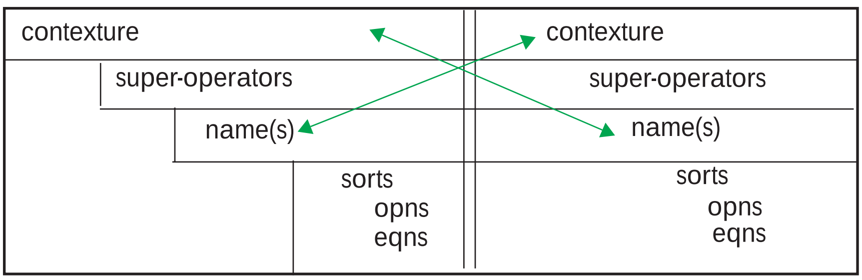
Because the unizity is absolute, every possible change of it has fundamental consequences for the whole framework of reasoning. The chiasm between the absolute unizity and the relativity of the names denies a simple mapping of the loci of the different systems onto the linearity of natural numbers. The chiasm between unizity and the other has no beginning and no end.

The chiasm is the mechanism of change. To connect the different unitizes with numbers we have to abandon the idea of an initial object, a starting point of the number series. Natural numbers, as we understand them, are constructed by algebras, induction and initiality. As a first step, we can try to model the chiastic situation in the context of co-algebras, co-inductivity and finality. This chiastic way of thinking is closer to the metaphors of *streams* and flows, and the lack of ultimate beginnings and endings as origins and telos.

More precisely, we should think of the chiastic paradigm as an interlocking play of algebraic and co-algebraic strategies and methods.

With this in mind, all attempts to formalize polycontextural systems, logics and arithmetics, with the methods of category theory alone have to be relativized. It is nevertheless of great importance to start the process of formalization of polycontexturality with the methods which are accessible. One very strong method, which is well accessible, is the method of fibering or indexing (Pfalzgraf, Gabbay). In other terms, the method of mapping local systems to an index set as a vehicle of distribution of formal systems. But this procedure involves the whole apparatus of the algebraic paradigm: equality, identity, linearity, initiality, inductivity, etc. Which, as I tried to make clear, is in strong conflict to the very idea of proemiality and its chiastic mechanisms.

The chiasm between names and contextures (unizity) is of great importance for a serious modeling of reflectional computation because it opens up the possibility of a distributed *self-referentiality* between systems as wholes. Furthermore, names in a contexture can be interpreted as the *reflectional mapping* of other contextures into the reflecting contexture.



5.3 Modularity and Metamorphosis

5.4 Chiasms, metamorphosis and super-operators

The super-operation CHI can be interpreted as the operator of changes of categorical perspectives, contexts or contextures and points of view.

These possibilities of changing the categorical terms is exactly what makes the difference between chiasms and category theoretic morphisms which are preserving the conceptual structures of the system in the process of mapping it into another system.

Proemiality incorporates both, category theoretical and chiastic morphisms.

Chiastic morphisms are not conservative in the sense that they are preserving the tectonical or conceptual structure of a system but more subversive in the sense quite analog to the catastrophes in Thom´s Catastrophe Theory that they are changing and not preserving the conceptual order. These morphisms are in a strict sense not only forgetful mappings but rules of metamorphosis.

Chaotic Logics

Chaotic logics are not the logics of chaos but the logics of change.

Change in chaotic systems is not a continuos process but the switch from one mode to another mode of a system by some changes of the states of the system.

Chaotic logics are the logics of interacting logical systems.

Changes in chaotic logics are modeled by transcontextural jumps from one system to another system and are defined in sharp contrast to the intracontextural steps of the expansion rule in a singular system. Transjunctional jumps don't exclude the possibility to stay in the primary system at the same time of the jump.

Cybernetic Ontology

Order from Noise.

####

As a consequence of these first insights, in this chiastic part of the proemial relationship, the category theoretic laws of identity and associativity are lost, or at least fundamentally transformed.

The possibility of metamorphosis is given by the interlocking mechanism of the chiasm. Also the super-operators had been introduced primarily to deal with contextures as such there is no reason to not to apply these operators to the internal structure of the contexture, that is here, to the internal structure of the abstract objects. Therefore the general operator of metamorphosis is composed, at first, by the super-operators {ID, PERM, RED, BIF).

This allows, that there may be an identity relation ID between to contextures and changes in their internal structure with e.g. sort1 in contexture1 becomes sort2 in contexture2 produced by the super-operator PERM. Or, the contextures and the sorts are stable, but the internal operations of the contextures may change.

It is not excluded in this chiastic concept of architectures of different systems, that for one system all the differences of the other system boils down to one notion. This would be a further step in mapping the architecture of one system into another system. Maybe that the interlocking mechanism between the systems would be reduced to a strong reduction produced by the extensive application of the super-operator RED to all categories of the system in consideration.

From the point of view of proemiality, metamorphosis is not a simple confusion of the categorical framework but a well ruled or at least rule guided change of categories in the process of change, emanation and evolution or other types of transformations. This type of metamorphosis is not wild in the sense of the absolute novum, because its scenario is founded on the known categories (names, sorts, operations, etc.) of the systems in transformation. If we would choose an other setting instead of algebras, we would have a similar scenario of change within the framework of the defining concepts. Another type of change could be thought for the case where the transformation changes to categories unknown before. For this case we would be forced to ad to our framework of proemial change between categories something like an empty box for the unknown. Why not?

Again, the process of transformation ruled by the proemial relationship has not to happen only between objects of the same architecture, like algebras to algebras. It also can happen between objects of different architectures. An interesting case could be the change between algebras and co-algebras. The same situation is to observe between distributed category systems. Morphism in one system can change to objects in another system of categories. Or even the very concept of category of one system can be transformed in a mere object of an other system. And so on.

Usual mathematical practice?

„Computer scientists have far more flexible view of formalism and sematics than traditional logicians. What is regarded as a semantic domain at one moment may later be regarded as a formalism in need of semantics.“

M.P. Fourman, Theories as Categories, in: Category Theory and Computerprogramming, Springer LNCS 240, p. 435, 1986

I don’t say that this is not the way mathematicians are anyway working. But it seems to be obvious that they are not reflecting or even formalizing this process, this use of terms and methods, that is their actual practice of doing creatively mathematics. Without ever mentioning what this means and how it is formalized, the „as“.

Maybe computer scientist have a more flexible use of formalisms than logicians. But logicians have not only produced most of these formalisms long before but also know very well that they are dealing with highly idealized situations governed by the principle of identity.

On the other side, philosophers and philosophical logicians have developed much work in explaining the *as-category* of thinking and being (analogy). But what is called, especially in European philosophy, hermeneutics, denies any possibility of formalization of the as-category. We also shouldn’t confuse the as-category with the more popular as-if-category of fictionalism (Hans Vaihinger) and constructivism.

It would be very interesting to start some case studies of this practice of computer scientists and mathematicians. A very interesting case would be the way or working with swinging types, that is the *switch* from algebras to coalgebras and back, in the sense of Peter Padawitz.

Or more traditional: In the summer term you get *Logics as algebras*, in the winter term they offer you *Algebras as logics*. And in-between you enjoy the summer holidays to forget any possible conflicts.

Translations, Goguens Semiotic Algebras

It turns out that correct translations are conservative metamorphosis.

Maybe the main problem of machine translation is just this decision, to start with conservative translations and to try to model common sense texts, which are full of games of violating this conservativity, with this restricted approach. In other words, conservative translations are based on disambiguated and context free semantics.

A case which is very artificial and doesn't match natural language at all.

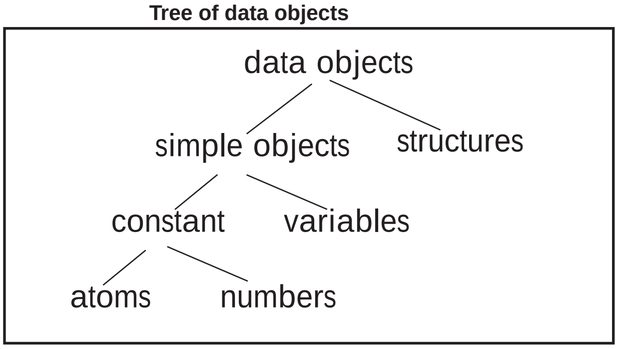
A conservative example: conflicts in the tree of data objects

All programming languages are based on very strict and stable conceptual structures. If the data objects are introduced as an ordered system like the „tree of data objects", this structure will never be changed in the process or execution of a program (Programmablauf). If something would be changed in this order it would automatically produce serious conflicts.

Because of the fact, that classical programs are essentially mono-logic, there is no space for conflicts in a positive sense. But real systems, that is interacting systems as today computing, are permanently confronted with conflicts. Why not introducing dialogs in the very structure of programming languages and systems? I'm not writing here about special interactive programs, e.g., but on the architecture and fundamental conceptuality or definition of programming languages as such and not of special applications of these languages. Like interactive proof systems or interactive games.

There is an easy way of producing conflicts in a dialogical system, if e.g. L1 declares A as a simple object and L2 declares simultaneously A as a complex object, that is as a structure. Obviously it is possible, in the polycontextural approach, to model this conflict and to resolve it in another logical system, say L3, this without producing a meta-system subordinating L1 and L2.

Diagramm 11 **Tree of data objects**



Furthermore, the conflict has a clear structure, it is a metamorphosis of the terms „simple object" in L1 and „structure" in L2. This metamorphosis is a simple permutation between sorts over two different contextures based on the chiastic structure of the mediation of the systems. But it respects the simultaneous correctness of both points of view in respect of being a „simple object" and being a „structure". In this sense it can be called a symmetrical metamorphosis.

Today computing is often characterized by its interactivity. But the programming languages have not changed to respond to this situation. They are still, in principle, mono-logic.

A further example of an interchange between programming languages would be the chiasm between data objects and control structures.

A very shy implementation of this interlocking mechanism, with far reaching consequences, is at the basis of all artificial intelligence attempts, the internal difference and possible ambiguity in LISP between data and programs ruled by the QUOTE/EVAL function.

These examples should not be confused with contradictions arising by a conflict in attributes between different informations. This implies a logical and linguistic level of communication and doesn’t touch the categorical framework of interaction.

After Wegner, interactions are paraconsistent, or at least belong to a paraconsitent type of logic. This maybe true on a linguistic-logical level, but it is not in correspondence with a more achitectonic and chiastic view of interactivity.

blind spots

Strategies of detecting the ontological, logical, computational, epistemological, reflectional, and what ever, blind spot of an interacting agent.

5.5 A simple typology of chiasms

To study some aspects of chiasms we can restrict ourself to the study of the interplay between relators and relata, neglecting the full-fledged exposition of the chiasm with its concept relation and unizity.

In practice it is easy to discover that many variants of realizations of chiasm are in the epistemological play. Mostly, chiasm are not fully designed, reductions are used and some times the use is over-determinated.

We can classify the single chiasms as balanced, under- and over-balanced. As distributed and embedded chiasms we can distinguish two modi of distribution, iteration and accretion and its combinations.

5.5.1 Iterations of chiasms

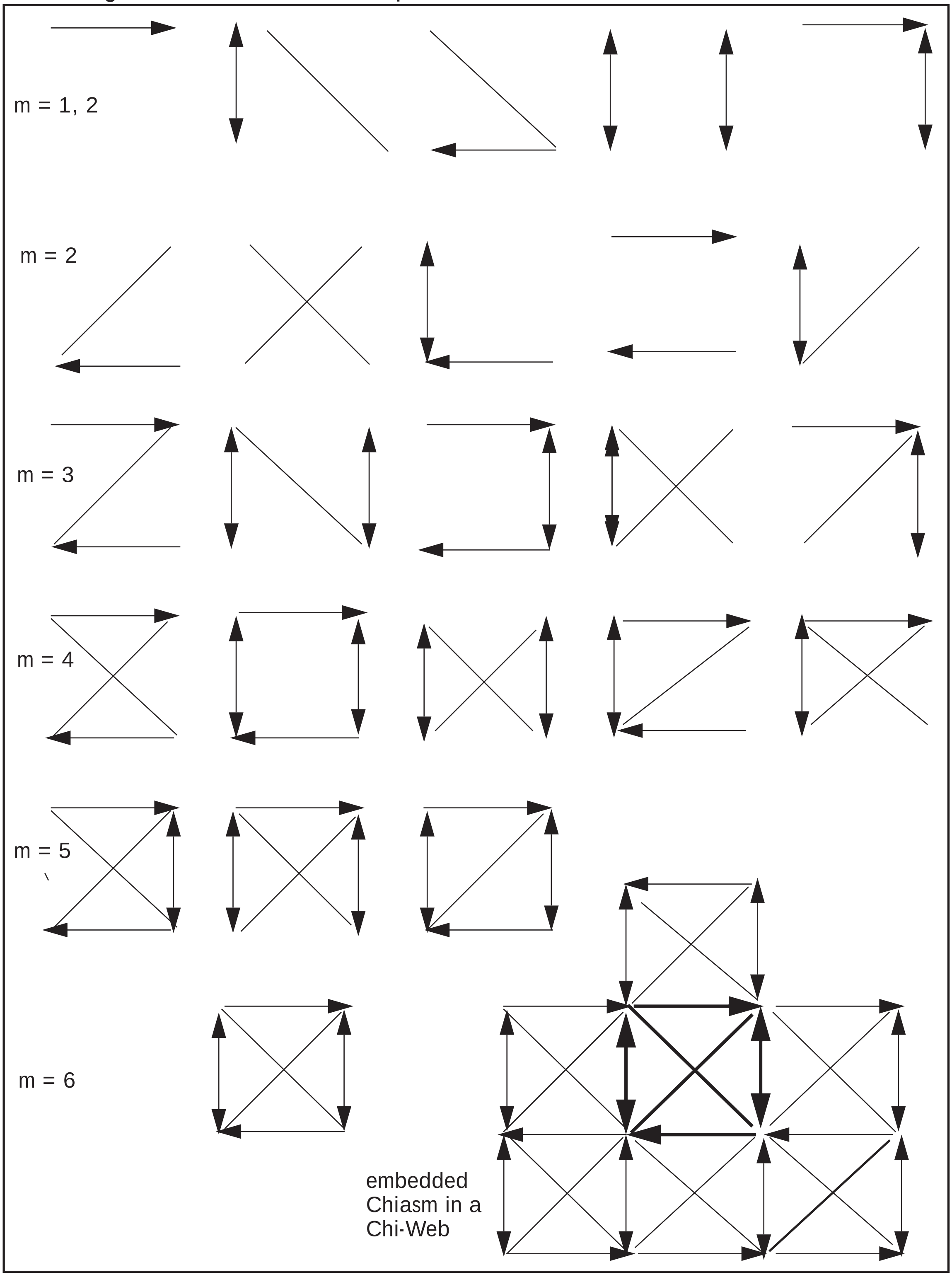
5.5.2 Accretions of chiasm

5.5.3 Mediation of iteration and accretion of chiasms

5.5.4 Over-determination of chiasms

5.5.5 Examples of under-balanced chiasms

Diagramm 12 Examples of chiasms



6 Proemiality between structural and processual understanding

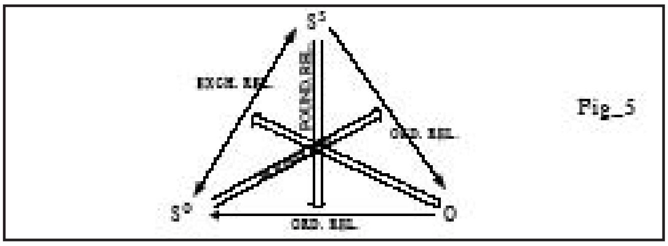
Gotthard Günther
F o r m a l L o g i c , To t a l i t y and T h e S u p e r - a d d i t i v e P r i n c i p l e
in: Beiträge zur Grundlegung einer operationsfähigen Dialektik, Band 1,
Meiner Verlag, Hamburg, 1976, p.329-351, first publ.: BCL Report, 1966

We have given the main reason above: if the relation between thought and its object is basically understood as a symmetric exchange relation the phenomenon of subjectivity disappears. But a "totality" in which everything is reduced to objectivity can never be total because something is missing.
A totality is, in Hegel´s terminology:
1) an iterated self-reflection of
2) a non-iterated self-reflection, and
3) a hetero-reflection.

If we permit, for the description of this structure, only logical operations which lead to reflection-symmetry then 1) is eliminated, and 2) and 3) turn out to be indistinguishable and logically identical … because 1) is nothing else but the capacity of keeping 2) and 3) apart.

(…)

However, if the concept of the universal subject, i.e. of ´Bewusstein überhaupt´(Kant), is eliminated the logical constraint to reduce everything to ultimate parity relations disappears. We will still have reflection-symmetry between SS and SO but not longer between S°—°—°— and O in general. In other words: it will turn out that the founding relation between subject and object or between Thought and Being is not a symmetrical exchange relation but something else. This is the point where the transition is made from formal classic logic of Aristotelian type to a theory of trans-classic, non-Aristotelian Rationality.
We begin by re-drawing Figure 1 omitting SU and having the phalanx of the SO replaced by a single S with the index O. We indicate the relations between SS , SO and O by arrows of four different shapes. According to the logical character of the relation an arrow will either be double-pointed or it will have one shaft or be double-shafted having either continuous or dotted lines. Figure 5 will then show the following configuration:



If SS designates a thinking subject and O its object in general (i.e. the Universe) the relation between SS and O is undoubtedly an ordered one because O must be considered the content of the reflective process of SS. On the other hand, seen from the viewpoint of SS any other subject (the Thou) is an observed subject and it is observed as having its place in the Universe. But if SS is (part of) the content of the Universe we obtain again an ordered relation, this time between O and SO. There remains the direct relation between SS and SO. This is obviously of a different type. SO is not only the passive object of the reflective process of SS. It is in its turn itself an active subject which may view the first subject (and everything else) from its vantage-point. In other words SO may assume the role of SS thus relegating the original subject, the Self, to the position of the Thou. And there is neither on earth nor in heaven the slightest indication that we should prefer one subjective vantage-point for viewing the Universe to another. In short, the relation between SS and SO is not an ordered relation. It is a completely symmetrical exchange relation, like "left" and "right". An ordered relation between different centers of subjective reflection comes into play only if we re-introduce the concept of a universal subject which contains all human "souls" as computing sub-centers. Of the two relations we have so far considered, the exchange relation is symmetrical and the ordered relation represents non-symmetry.

This investigation intends only to show that the concept of Totality or Ganzheit is closely linked to the problem of subjectivity and trans-classic logic and that it is based on three basic structural relations:

an exchange relation between logical positions
an ordered relation between logical positions
a founding relation which holds between the member
of a relation and a relation itself.

It may be said that the hierarchy of logical themes as indicated in table (II) represents an hierarchy of implicational power. All themes have in common that they are self-implications; they imply themselves. However the first theme (objective existence) implies only itself and nothing else. In this respect it differs from any succeeding theme which implies itself as well as all subordinated themes. For this reason it is proper to call the initial theme "irreflexive" and all the following "reflexive". Irreflexivity means that something we think of is only an implicate but not an implicand for something else. On the other hand if we refer logically to reflexivity we mean that our (pseudo-)object of thought is an implicand relative to a lower order and as well an implicate relative to a theme that follows it in the hierarchy of table (II).

We are now able to establish the fundamental law that governs the connections between exchange-, ordered- and founding-relation. We discover first in classic two-valued logic that affirmation and negation form an ordered relation. The positive value implies itself and only itself. The negative value implies itself and the positive. In other words: affirmation is never anything but implicate and negation is always implication. This is why we speak here of an ordered relation between the implicate and the implicand. The name of this relation in classic two-valued logic is - inference.

It is now necessary to remember that the possibility of coexistence of two independent subjects (I and Thou) in the Universe is based on an exchange relation between equipollent centers of reflection. Moreover, these subjects are all capable of being implicands. More objects do not operate inferentially. That means they do not imply anything else.

If we now consider the founding relation in which a subject constitutes itself as diametrically posed relative to all objects and the total objective concept of the Universe we will discover that this relation represents an interesting synthesis of an exchange relation between logical positions an ordered relation between logical positions a founding relation which holds between the member of a relation and a relation itself. 10
exchange and order. The founding relation is in itself an exchange relation in so far as the linking subject (SS) may assume the logical position of the other subject which is thought of (SO). SO may in its turn assume the rank of SS. Any two centers of subjective reflection of the same order mutually imply each other. But such an exchange does not operate between S°—°—°— and O. As we pointed out before: the bona fide object cannot infer the subject and by doing so usurp the role of a subject. If it could it would imply that subjects are irreflexive entities which for a subject is a contradictio in adjecto. It follows that the relation between implicate and implicand has two different aspects: between two subjects this relation assumes the role of a symmetrical exchange. Between subject and object it appears howev-

er as an ordered relation. The founding relation is therefore also an ordered relation. Or to put it differently: the founding relation is a combination of exchange and order. What is the implicand (SS) may become the implicate not relative to O but to our impartial observer S S S. We might say that the founding relation is a concatenation of sequences of exchange and sequences of ordered relations.

The diagram of Fig._6 will illustrate what we mean:

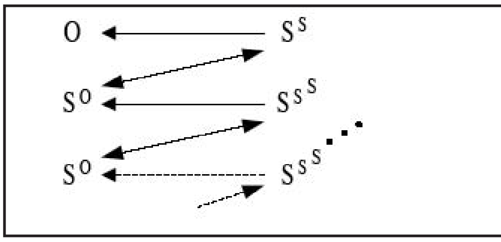


Fig_6

Fig._6 indicates a sequence of single-pointed and a second sequence of double-pointed arrows such that a single-pointed arrow always alternates with a double-pointed one.
A concrete example of what the figure illustrates is the father-son relation. This is first an ordered relation. But the son can also become a father. In this sense father-son is also an exchange relation. But the son does not acquire the status of father relative to his own father but relative to the grandson of his father.
In abstract terms: what is member (or argument) of the ordered relation O¬SS, namely SS, may become an argument of an exchange relation not relative to O but relative to S S S which implies this exchange SS«SO.

Thus we may say: *the founding-relation is an exchange-relation based on an ordered-relation. But since the exchange-relations can establish themselves only between ordered relations we might also say: the founding-relation is an ordered relation based on the succession of exchange-relations.*
When we stated that the founding-relation establishes subjectivity we referred to the fact that a self-reflecting system must always be: self-reflection of (self-and hetero-reflection). As Hegel pointed out in his dialectic logic one and a half centuries ago, the opposition of hetero- and self-reflection is not a parity relation because it requires an iteration of self-reflection in contrast to the non-iterative character of hetero-reflection. It follows as was pointed out above, that one value is sufficient to designate in hetero-reflection but two values are required - apart from the value S S
S O
O
S S S
S O S S S
Fig_6

for object-designation - to separate self-reflection from the object. This is confirmed by the character of the founding-relation. Table (VI) clearly shows that it requires a minimum of three values for its own establishment. But the introduction of a third value generates a new principle of superadditivity.

Irreflexivity as the ultimate beginning

In contrast to my working hypothesis “There is no origin, only a multitude of beginnings” irreflexivity in Gunther´s approach to the founding relation has the value of an ultimate beginning, which is the origin in its unizity. This origin is characterized as a self-implication.

It may be said that the hierarchy of logical themes as indicated in table (II) represents a hierarchy of implicational power. All themes have in common that they are self-implications; they imply themselves. However the first theme (objective existence) implies only itself and nothing else. In this respect it differs from any succeeding theme which implies itself as well as all subordinated themes. For this reason it is proper to call the initial theme “irreflexive” and all the following “reflexive”. Irreflexivity means that something we think of is only an implicate but not an implicand for something else.

To start proemiality (founding relation) with a beginning in the sense of an origin is not included in the general definition of the founding relation. It is an additional decision, based on special ontological interests.

Neither the abstract formulation nor the example given, father-son-relationship, involves an ultimative beginning. Otherwise the father-son-relationship connoted with an origin would force us to accept a “Ur-father”. Maybe God. But this is not philosophical thinking.

To interpret proemiality as having a beginning is guided by the principle of well-foundedness. This principle is necessary for an algebraic or constructivist approach. In contrast to this interpretation of the founding relation it is equally possible to understand this mechanism in a non-founded way of coalgebraic co-induction.

As an example we may think of a chain of alternating Xs and Ys without an origin nor an end:

...XYXYXYXYX...

Is it reasonable to take X or alternatively Y as the start element of the chain? Obviously not. It maybe, in some special situations, a reasonable decision to take Y as the start.

> We might say that the founding relation is a concatenation of sequences of exchange and sequences of ordered relations.

The same is true for the concatenation chain of order and exchange relations. But this decision is arbitrarily and not part of the mechanism of the founding relation.

To make these two interpretations more clare, I introduced in my *Materialien 1973-75* the distinction between *open* and *closed* proemial relationship.

Even if we accept that the environment of a living system has in contrast to its modeling of it an irreflexive character for the modeling system, it is important to see that this irreflexivity is of relative nature. Otherwise it will be very difficult for a cognitive system to have different interpretations of its environment and to change its ontology.

Many constructivists have introduced the distinction between reality and objectivity (Maturana) to deal with this difficulty. In their approach irreflexivity is pure reality, which as such escapes any knowledge. On the other side objectivity is a result of the process of interpretation. But since Kant we should know that this trick is not properly working.

7 Category Theory – and beyond?

7.1 Remembering categories

Categories. A category axiomatizes the abstract structural properties of sets and mappings between sets. Sets are considered as the objects and mappings are called the morphisms or arrows of the abstract category of sets. The language of category theory allows us to talk about arrows, their sources and targets and about their composition (o), of arrows, but not about the internal construction of sets and the nature of their elements. In particular, we cannot talk about the application "f (x)" of a map to an element of a set nor about the way f (x) is evaluated. One might say that sets and arrows are considered atomic particles of category theory and everything that is to be said about sets and mappings must be expressed solely in terms of the notion of composition, source and target.

To every object A, the existence of a particular identity arrow idA (sometimes written as 1_A) is postulated. Categorical language is too weak to axiomatize it using an equation such as e.g. "idA(x) = :x", for this refers to elements x inside the object A and to the application f (x) of f to x. In categorical language rather, idA must be characterized as an arrow satisfying:

- *source(idA) = target(idA) = A*
- *for all morphisms f with source (f) = A we have f o idA = f, and*
- *for all morphisms g with target (g) = A we have idA ° g= g.*

Note that composition is to be read from right to left - in accordance with traditional mathematical habit.

Definition 3.1. A category C consists of a class C_o of objects A, B, C, . .. and a class C_m of morphisms or arrows f,g,h,… between these objects together with the following operations:

- *dom: C_m —> C_o,*
- *codom: C_m —> C_o, and*
- *id:C_o —> C_m,*

associating with each arrow its source (domain), resp. its target (codomain), and with every object A its identity arrow idA. Moreover there is a partial operation (o) of composition of arrows. Composition of f and g is defined whenever codom(f) = dom(g). The result is a morphism go f with dom(g o f) = dom(f) and codom(g o f) = codom(g). The following lauls have to be satisfied wenever the composition is defined:

- *(h o g) o f =h o (g o f)*
- *idA o f = f and g = g o idA.*

3.1.1. Commutative Diagrams. Many notions have their origin in the standard example, the category of sets and mappings, so we borrow notions, symbols and graphical visualizations from there. For instance, we write f: A —> B, if f is a morphism with dom(f) = A and codom(f) = B. We use uppercase letters for objects *and lower case letters for arrows.*

It is convenient to draw objects as points and morphisms as arrows between these points. Such a representation is called a diagram. Often, compositions of arrows are not drawn - their presence is implied. A path of arrows represents the composition of the arrows involved. Whenever there are two different paths from an object A to an object B that enclose an area, it is often implied that their compositions are equal. One says that the diagram (or parts of it) commutes. To emphasize this, a circle is sometimes drawn inside the area whose bounding paths are assumed to commute.“

Basic Notions of Category Theory (H. Peter Gumm), p.13-14

Partial operation

In the case of sign systems as abstract objects, Goguen writes:

"A good (semiotic) morphism should preserve as much of the structure in its source (sign) system as possible. Certainly it should map sorts to sorts, subsorts to subsorts, data sorts to data sorts, constants to constants, constructures to constructors., etc.

But it turns out that in many real world example, not everything is preserved." Goguen, Algebraic Semiotics, p. 11

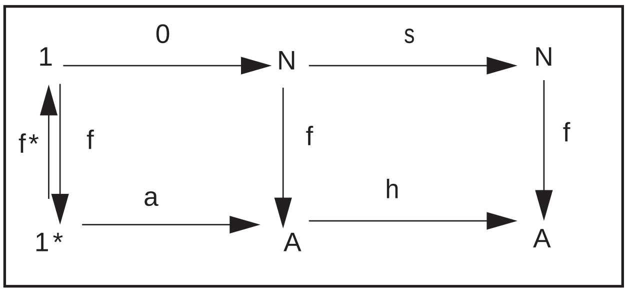
Duality principle

"After a second thought, it is perhaps not so surprising that the direction of morphisms can often be reversed in a categorical definition – we say that a definition is dualized – with the result that a reasonable concept emerges.

Probably, the phenomenon only reveals the predominance of symmetry in mathematical thinking. However, it appears that the generality of the duality principle has been first observed in category theory." Cat, p. 25

Terminal object

An important fact is that any two terminal objects (as well as any two initial objects) in a category are uniquely isomorphic. In other words, if T and T‘ are two terminal objects, then there is a unique isomorphism between the two. Because of this, it is customary, to collapse all terminal objects into a representative and talk about the terminal object.



Final object

up to isomorphism

„The categorical approach to characterize objects and morphisms in terms of their relation to other objects and morphisms has the particular consequence that universal properties specify objects only „up to isomorphism“.

Definition: Objects A and B are isomorphic if there exists morphisms f: A —> B, f*: B —> A such that f*.f=iA and f.f*=iB

Logical foundations of category theory by William S. Hatcher
http://www.rbjones.com/rbjpub/philos/bibliog/hatch82.htm

7.2 Motivations

Goguen´s Manifesto

Why do we need category theory?
Goguen: Manifesto

A big collection of translations vs. a common language

Uschold, Michael F
"If you want to call these correlations or mappings between languages as expressed in a common meta-language, well that's ok, but it really is in a common language.
If you say "oranges" in L1 are "apples" in L2 and are extensionally (or some other property) equivalent, then you are expressing something in a third language L3.
The power of category theory comes in because you can have the syntax of a logic be related to the semantics of a logic (via morphisms), and then have those related to other objects, such as e.g., another logic (with its syntax and semantics), but the mediating "language" here is category theory (or a corresponding categorical logic).

Leo
ps. Gabbay has also written on labelled deduction systems (probably generalized under fibred logics), which essentially are correlated logics expressed in parallel to work simultaneously but on different aspects of the data (e.g., one portion on the natural language syntax ala categorial grammar, another on the nl semantics ala categorial semantics).
> -----Original Message-----
> From: David Espinosa [mailto:davidespinosa@pacbell.net]
> Sent: Tuesday, August 21, 2001 10:12 PM
> To: Uschold, Michael F
> Subject: Re: languages and translations
>
> Heterogeneity vs homogeneity
> --------------------------
> However how we try to define standards, the world is heterogeneous.
> Sooner or later someone makes a new, incompatible variation. Homogeneity is pretty unstable. So we'll definitely need translations between systems.
> As for how to build them, sure, it's nice if you can find a common language, but that language probably won't capture every feature of every language. It's nice for translating a subset, but sooner or later you think of a nice way to map construct C1 of L1 into construct C2 of L2, and it won't go through the common language. So you have to revert back to individual translations, which are the general case.
> The real point is that we shouldn't get stuck on the idea of a common language-if it emerges naturally, great, but there's nothing wrong with a big collection of translations. The same goes for any collection of format or datatypes or whatever. Look at how we program these days:
> C, Lisp, Java, Perl, Python, a bunch of Unix tools, MS Excel, whatever gets the job done. (...)"

7.3 Diamond Strategies for Category Theory

Focussing on objects

Ein Ziel der Fokussierung sind die Objekte und ihre Strukturen:

„Für jede mathematische Theorie definiert man sich zunächst Objekte und dann zur Beschreibung dieser Objekte i.a. zulässige Abbildungen, die man Morphismen nennt. Dieses Vorgehen wird durch den Begriff der Kategorie exakt erfasst.“

Weiter:

„Definition: Eine Kategorie C besteht aus

(1) einer Klasse /C/ von Objekten, die mit A, B, C, … bezeichnet werden.“ Gerhard Preuss

Focussing on morphisms

„It is part of this guidline that in order to understand a structure, it is necessary to understand the morphisms that preserve it. Indeed, category theorists have argued that morphisms are more importand than objects, because they reveal what the structure really is. Moreover, the category concept can be defined using only morphisms. Perhaps the bias of modern Western languages and cultures towards objects rather than relationships accounts for this.“ Joseph Goguen

Focussing neither on objects nor on morphisms

Also the following citation of Gunther does not intent to gives a definitional clear explanation of a neither-nor situation it is useful as a hint in the right direction.

„Thus the proemial relation represents a peculiar interlocking of exchange and order. If we write it down as a formal expression it should have the following form:

$$\square \; R^{pr} \; \square$$

where the two empty squares represent kenograms which can either be filled in such a way that the value occupancy represents a symmetrical exchange relation or in a way that the relation assumes the character of an order.“ Gunther, p. 227

Obviously, the scheme or formula, represents neither an order nor an exchange relation.

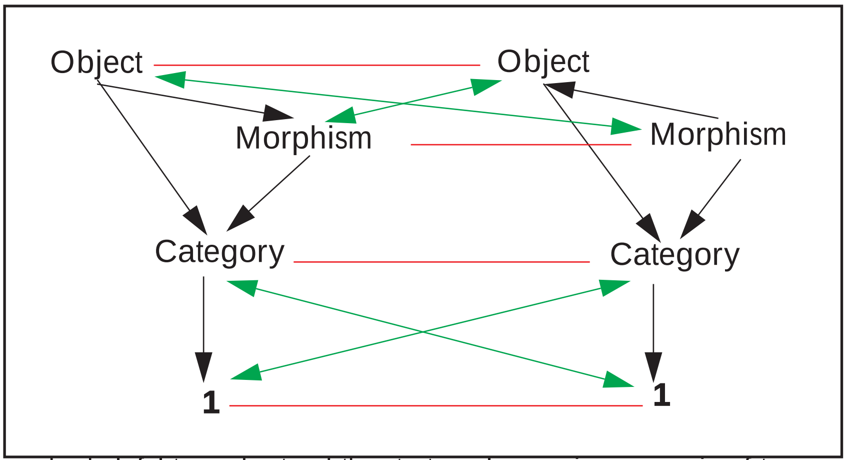
With this in mind, we can try to think the neither-nor of objects and morphisms of category theory as the inscription of the processuality of „categorization“ in itself into a scriptural domain beyond classical formal systems, that is into *kenogrammatics*.

We need this quite wild „anti-concept“ of kenogram and kenogrammatics to deal scientifically and technically with the structure of any change, the proemiality, which is not to catch by any construction based on semiotical identity.

Focussing on both at once, objects and morphisms: Disseminating Categories

Diagramm 13 **Conceptual Graph of Categories**

Apropos: Use and abuse of methods



It may be helpful to understand the strategy I am *using* as a mix of two movements: I am using scientific terms and methods to develop and inscribe, formalize, my ideas and at the same time, with the same gesture, I am *abusing* these methods to overpass the limitations of these scientific concepts. Amy criticism of my work should keep this double strategy in mind. After that, there is a lot of work to for all sorts of criticism.

It is not excluded, that all the abuse may be, step by step, filling some gaps, correcting unnecessary misuse, transformed into a more scientific use of concepts without abandoning the fundamental subversion of scientifity involved.

For example, in the diagram of the conceptual graph of the notion of category as composed by objects and morphisms, I am using arrows which are inscribing the notional dependency structure of the notion of category itself. But in the same sense it could be mentioned that the notional dependencies are a sort of morphisms so I am using arrows as morphisms to explain the dependency structure of objects and morphisms in the motion of category. Usually conceptual graph are applied to other domains than to itself as a categorial notion. The conceptual graphs are therefore used in a sort of self-application and the questionis open to what system the arrows of the reflexive use of the conceptual graphs belong.

The situation can easily be radicalized. We say a graph consists of the notions nodes and edges, thus the conceptual graph of the notion graph is a graph between nodes and edges. A graph is explained by the use of a graph.

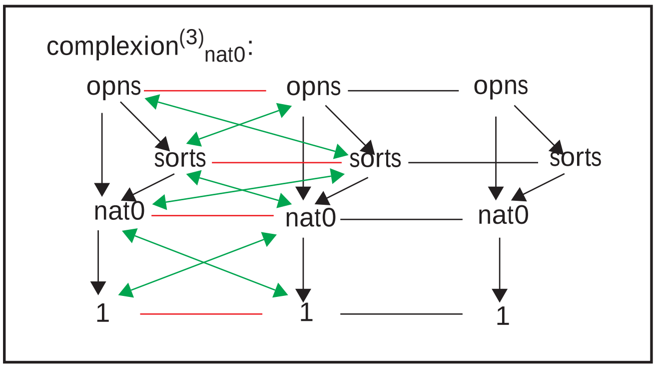
But we all learned that we should distinguish between use and mention, object and meta-language, notion and notation, and so on. Is this really always helpful? We can repeat this game of self-application on these terms too. And: Mention mention, use use, mention use and use mention, and neither-nor. Why not?

After having introduced all this ideas and hints to methods, we could start the real work of formalizing the whole stuff in the framework of category theory with the help of some strategies of rule-guided abuse, that is deconstruction. This will be done later in a different context.

8 Dissemination of natural objects

As an application of the idea of proemiality I introduce the dissemination of 3 abstract objects nat0. This is also a specification of the idea of cloning formal systems. Cloned formal systems as replicas are not represented by sets or by multisets which are commonly used in the formal theory of replication in mathematical biology but by the mechanism of dissemination, e.g. distribution and mediation.

(zero —> (nat —> nat)) —> nat0



$\text{dissemination}^{(3)}{}_{nat0}$

$DISS^{(3)}$ $(\text{object}_{\ nat0}) = \text{object1}_{nat0}$ § object2_{nat0} § object3_{nat0}

object1_{nat0} —> object1_{nat0}
object2_{nat0} —> object2_{nat0}
object3_{nat0} —> object3_{nat0}

Natural system:
nat0(3) = nat0(1) § nat0(2) § nat0(3)

Sorts:
sorts(3) = sorts(1) § sorts(2) §sorts(3)

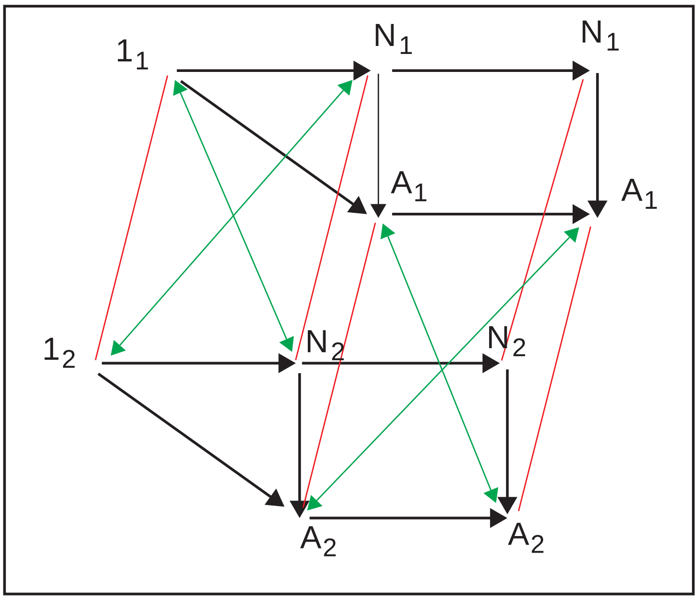
Operations:
opns(3) = opns(1) § opns(2) § opns(3)

zero(3) : —> nat(3) with
zero1: —> nat1
zero2: —> nat2
zero3: —> nat3

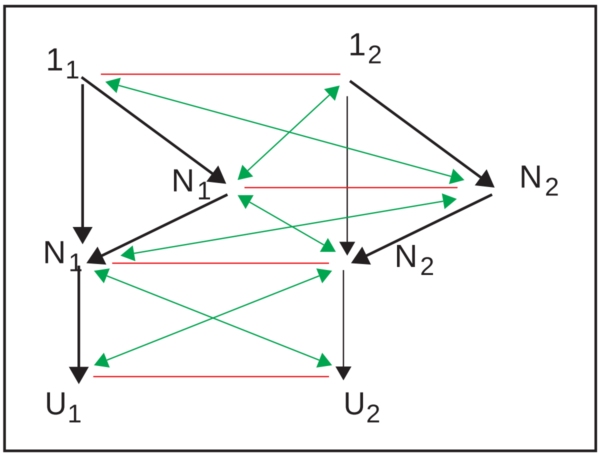
and

suc(3): nat(3) —> nat(3) with
nat1 —> nat1
nat2 —> nat2
nat3 —> nat3
This triple clone of the above natural system produces naturally infinite series of expressions of the following form if we apply the operations in a parallel way.
zero, suc(zero), suc(suc(zero)), suc(suc(suc(zero))), ...
zero, suc(zero), suc(suc(zero)), suc(suc(suc(zero))), ...
zero, suc(zero), suc(suc(zero)), suc(suc(suc(zero))), ...

In another notation this results in 3-tuples of terms.
(zero, zero, zero), (suc(zero), suc(zero), suc(zero)), ...



1 1
N 1
N 1
A 1
A 1
1 2
N 2
N 2
A 2
A 2



1 1
1 2
N 1
N 2
N 1
N 2
U 1
U 2

9 On interactivity between cloned systems

But we can surely suppose that these clones (or replications) will start to interact with each other and begin to produce slightly more interesting series of expressions than the purely isolated parallel ones which turn out to be a special case of the modi of interaction. Even more, the single fundamental system looks like a very special reduction of the interacting system, namely the case of the modus of interacting with itself.

Some evidence shines up that the interaction modi between systems are far more fundamental than the single systems in themself. Also I started with the idea of cloning formal systems, this start is ruled at first by the strategy of inhereting from the classical isolated system as much of its genotype, methods and constructions, as possible.

What should a multi-agent theory or formalism look like?

Multi-agent systems add another dimension to agent-oriented systems. Can single-agent formalisms be extended for multi-agent systems? What are the extra features of such systems that must be addressed, and how might this be done?
A formalism for a multi-agent system must also deal with
- the multiplicity of agents;
- group properties of agent systems, such as common knowledge and joint intention;
- interaction among agents, such as communication and cooperation.
Some single-agent formalisms are expandable in these directions to handle multiple agents. For example, multi-modal logics are a good tool for studying multiple agents which do not interact too much, and can deal with some group properties and the flow of time. Some other logics have also emerged which attempt to deal with several aspects of agency at the same time, or address several agents (e.g., (van Linder et al., 1996; Fagin et al., 1995; Kraus and Lehmann, 1988; Wooldridge, 1996)).
However, we are still some way off having a logic which addresses all these features, and there are some drawbacks, too. One key problem with such formalisms is that, if anything, they are even further from implementations than with single-agent formalisms. Also, the macro-level, emergent behaviour exhibited by a multi-agent system does not easily lend itself to formal analysis. For these reasons, it may be that the best one can do is reason about local interactions between agents.
Formalisms for Multi-Agent Systems?, Mark d'Inverno et al., First UK Workshop on Foundations of Multi-Agent Systems (Fo-MAS'96).

9.1 Reflectional architectures of interactivity

Systems as simple structured identities are not prepared for interaction. They may be fit for all sorts of exchanges of information, communication, as we know it from different programming languages and computer systems. But this is not what I try to introduce. Our cloned formal systems have to be able to interact with ofther systems, formal or others, too. Maybe that this is the point where the metaphor stops to work. These replications have to modify themselves to be able to interact together. They have to internalize, that is, to realize in themselves, the complex interactional structure of their environment. The new metaphor which is leading my studies of the construction of internalization (introspection, reflection and interaction), should be the metaphor of *togetherness*.

Togetherness may be realized for each single system by the complex reflectional structure of

auto-referentiality,
hetero-referentiality and
self-referentiality.

This structure is well known from Hegel and further explained by Gunther in his Cybernetic Ontology but as much it is used for sociological notions it is not formalized and implemented at all. A similar but more computer scientific approach can be found in the works of Computational Reflection (Smith, Maes, Sloma, Kennedy).

Therefore I postulate that each interacting system has to incorporate a complex reflectional architecture. Each system has to realize its referentiality to itself, that is its auto-referentiality, further it has to be able to give structural space in itself for the process of mapping its environment, that is its hetero-referentiality, and finally, at least for the moment, it has to have space in itself for the reflection or awareness of the processes of auto- and hetero-referentiality, that is its self-referentiality.

In more technical terms this means that our cloned systems are no more simple replicas of their original isolated single system but complex systems with an interesting architecture which is not to be confused with the hierarchical tectonics of classical formal systems. The structure of the reflectional architecture is of heterarchical nature. Which means that all reflectional domains of the system are on the same level of relevance. This does not exclude a complex interplay of temporary hierarchies of all sorts in the realm of heterarchy. Again, it is well acknowledged by natural sciences that nature is of highly complex hierarchical nature. Thus, our clones are realizing a quite unnatural behavior with their heterarchical architecture, they are therefore much more artificial than natural in any sense of the term.

This idea has to be considered in the process of inheriting methods and techniques from the classical formal constructions. In abandoning the superiority of alpha-nummerical notational systems with their atomizity and linearity the tabular notational texture of kenogrammatics and morphogrammatics is to be considered for the realization of interacting systems.

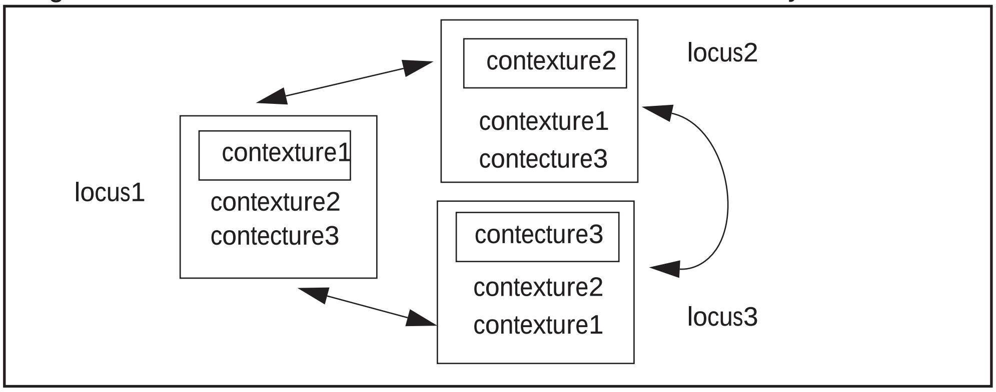
Now, our cloned systems, that is, the distributed and mediated systems, are to understand each as an intersection of a multitude of contextures at a logical locus. I mention shortly, that the theory of these "logical loci" can be found in the so called *kenogrammatics*. Each interacting logical system is realizing this complex reflectional architecture. In some sense we can say that this reflectional architecture of the systems is the *space of interaction*. Interaction is a interlocking mechanism of these reflectional architectures. In more metaphorical terms, architectonics is the home of togetherness. There is no flow of information, yet. It has to be mentioned, that even on this very basic level, all sorts of interactional conflicts are possible. For instance, if not all actors are realizing their reflectional architecture for the purpose of interaction. It is also clear that these logical systems are not permanently interacting. There are other modi of action too, which are realized by their own structures belonging to the framework of the full interaction architectonics.

As a consequence of the complex architectural structure of the system their objects are inheriting this structure from the very beginning. Atomizity of the single isolated system, we say the mono-contectural system, has to be replaced by complexity. Objects of poly-contextural systems are complex from the very beginning (ab ovo). This has a far reaching impact on the use of classical methods because these methods and techniques are all based on atomic concepts. But these classical methods are at the time the only accessible working concepts, therefore we have to use them even if this way of using them turns out to be more or less a well-organized abuse; sometimes called deconstruction.

"Wherever we extricate any two data from this world, we will find that they share in a common contexture and that their relations can be described by a two-valued logic. This test will never fail us. But since we pointed out that every ontological datum of the world must be considered an intersection of an infinite number of contextures, the fact that any two data we choose to describe in their common two-valued relations belong to one contexture does not exclude that the very same data may also–apart from the contexturality chosen for our description–belong separately to additional and different contexturalities. Our first datum may, e.g.,

be an intersection of the contexturalities a, b, g, l and the second may be intersected by the contextures b, d, k, p it. What we insist on, however, is that any two world data we choose to compare have at least one contexture in common. They may share in more but it is impossible that there is no contextural linkage between them at all. If that were the case then one of the two data would be "not of this world "." Gunther, Life

Diagramm 14 **Reflectional architecture of a 3-contextural system**



Architextonics: From architecture to architexture of computational systems

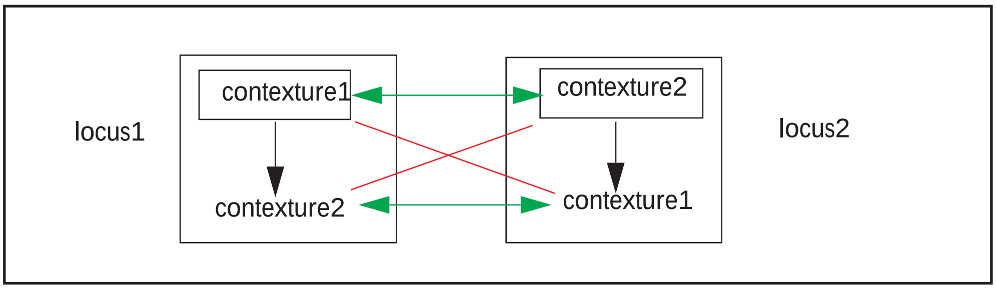
9.2 Proemiality and reflectional architectures

As far I have introduced two fundamental notions for poly-contextural systems: the notion of *proemiality* (chiastics) and the notion of *reflectional architecture*. How do this two concepts interact together?

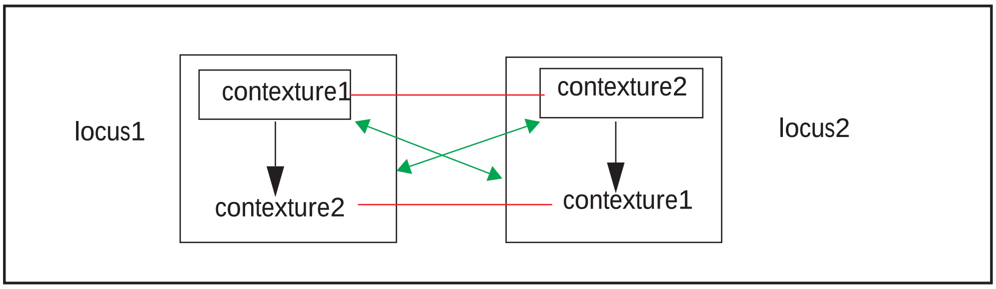
To answer this question I have to give an explication of the arrows in the diagram. The internal structure of the simple arrows is explained by the proemial structure, interlocking mechanism, between the primary and the mirrored contextures at their logical locus. Between contexture1 and contexture2 each in itsd functionality as a primary contextuer we have an exchange relation. The same holds for the mirrored case, contextures1 and contexture2. Both are in an exchange relation. Internally, for each compound contexture we observe an order relation between the original and the mirrored situation. And finally, between the same distributed contextures we have the case of categorical coincidence. All relations together are defining the situation of the proemial relation between four objects.

In more linguistic terms we could speak about the chiasm of I and Thou. Obviously this may sound similar to the "Mirror stage" as it is described by Jaques Lacan.

Proemiality and interactionality are together in a co-creative interplay. This constitutes the domain of togetherness.



Another thematization of the reflectional situation. The same in green. Here we are focusing more on the mutual asymmetry between the mirroring processes of the agents.



9.3 Towards architectures in computational reflection

"Although reflection is a popular term these days, the issues related to it are very complex and at the moment still poorly understood." Pattie Maes

The importance of architectures for artificial intelligence was early discovered by Sloman (1986) and Steels (1986). With my own interpretation of the concept of logical transjunctions in the early 80th I was on a similar path. What was hidden to the booming second-order cybernetics literature, especially in Germany, about reflection, i.g., self-referential systems and their circular logics, was exactly the notion of architecture for self-reflecting systems. Despite some nice drawings from Gordon Pask (H. von Foerster) and similar constructions by Levebvre there was no awareness of epistemic architectures at all. Probably as a result of rejecting any sort of psychologism and mentalism. The main interest was to construct all sorts of formalisms around the logical paradoxies involved in reflection and self-referential logical statements.

In hard contrast to this anti-internalism, Gotthard Gunther had proposed a complex ontology of epistemic “internalism” beyond the simple destinction of inside/outside, developing logical differences in the inside, and in the outside too. But neither the theories of reflection, introspection and representation of Smith and Maes, nor Gunther´s complex ontologies offered a hint how to implement these architectures in a mathematical and logical setting. The work of Sloman which I have read much later still remains in the same descriptive vagueness, despite of his different implementations.

In a work about industrial processes and the need of introducing CIM (Computer Integrated Manufacturing) I developed a description of complex "interior" modelling of environmental situations using Gunther´s polythematic ontologies and logical transjunctions. (Siemens Studie 1984) But only in the process of a much deeper formalization of poly-contextural logics, i.g., transjunctions, I developed an understanding of computational architecture in the sense of polycontexturality.

Gunther was emphasizing the importance of the rejectional aspect of transjunctions, complementary to this I was reading the acceptance aspect in the rejection process. That is the rejection has to be accepted by another system, otherwise it is blind. This dialogical function of rejection and acceptance for transjunctional operators I understood as “Einräumung”, giving space to the other, much in the sense of Ludwig Binswangers fundamental ontology of love. To give space doesn’t mean that I have to give up my own position, but that I open up space in my mental and logical system to another position, therefore producing internal environments at my own logical locus.

Because ideas about internal environments are still not well understood and only seldom mentioned I will give a helpful citation from the article of Perlis (1988).

> “Sloman (1986) and Steels (1986) have suggested that there is an important contextual phenomenon regarding *internal* symbolism in computational behavior. That is, there is an aspect to the execution of programs that bears on environmental events in a very direct way, namely the internal states of the executing hardware and software itself. This is a physical phenomenon and hence an environment even though it is not external to the computer. Still, when a value is read (copied) from a storage register or other (virtual) memory location, the resulting copy bears a physical relation (identity or equivalence) to the original and can be said to be about the original in ways that are significant for the ongoing functioning of the program execution. Thus what is a tenuous link in general between symbol and symboled, seems a firmer sort of thing for these internal cases.
> D. Perlis, Meta in Logic, in: P. Maes, D. Nardi, Meta-Level Architectures and Reflection, 1988, p. 44/45

9.4 Polycontextural logics and reflection

Polycontextural logics starts with the basic but simple idea that each rational agent has its own point of view to its world and that therefore each agent has its own logic.

To each agent corresponds a classical logic that determines the logification of its knowledge and experiences collected in a domain, called contexture. The next basic idea of the concept of polycontextural logic is given by the stipulation that these rational agents are not isolated from each other but are in a network of interaction. Polycontextural logic is describing the structural rules of interactions between rational agents.

The network of interaction of rational agents is modeled in the concept of Polycontexturality. This concept is independent of the notions of information and communication and other cybernetic and computer scientific terms. The proemial relationship or chiasm describes the general structure of interactivity between agents. The proemial relationship is introduced by the interlocking mechanism of four types of relations: order, exchange, coincidence and place.

Interactivity and co-operation between different rational agents is reflected in their logical connectives: junctions (plus negations) and transjunction. The junctional connectives are ruling the intra-contextural, the transjunctions the trans-contextural situations.

From a formal point of view what we are observing is a principle asymmetry between two communicational agents. An actor which has a model of an other actor may know that the other actor also has a internal model of him, but he will never be able to observe this model directly. He is only able to observe the actions of the other agent in his environment. He will never be able to observe content and structure of the inner model of his communicational partner agent. However he may make some corrections to his model in the process of observing the results of the interaction with the other agent. In other words, the world of interacting agents is cut twice, one cut is between agents and their world and an other is between agents and their models of the world. The simple cut between agent and world is the cartesian cut and its logic corresponds to the bivalent classical logic.

The fact of the double cut, the epistemical situation for interacting agents, forces to non-classical logics in which the process of cutting itself can be modeled.

introspection, self-reflection, action, Levebvre

It seems to be that this is the main difference between trans-classical and classical understanding, modeling, formalizing and constructing of artificial interacting systems.

In this sense it seems not to be sufficient enough to combine logics, to mix different logics and different methods to obtain a logical system for multi-agent robots (Pfalzgraf, Gabbay).

9.5 Pfalzgraf: Fibring logics and multi-agent robotics

Today´s concepts of fibering or combining logical systems does not include the reflectional aspects of interaction and togetherness. Basically its ontology remains mono-contextural. The problem of multitudes is shifted form the ontology or the universe of the underlying logic to its sorts in a many-sorted logic. Multitudes, plurality is therefore based in singularity and unizity as well explained in the theory of institutions. To model reflectional interacting systems we have to introduce a polycontextural ontology with all the consequences for logics, arithmetics, semiotics and so on.

cit.

10 Introducing the metaphor of a tissue of coloured logics

"When classical logic is applied to non-mathematical examples, the examples are first ´mathematicized´. In the real world we might argue about whether block B is behind block A or not – maybe it depends on one's point of view. But we can create an ideal world, a mathematical model, in which either B is behind A, or it isn't. Classical logic can be used to reason correctly about such a model. Whether the model accurately reflects the real world is a separate issue." Fitting, 1990, p. 1

Trans-classical logic is aimed to model the situation of rational reasoning between different agents where each agent has its own logic, that is, its own point of view in respect to his world. Therefore trans-classical logic reflects the world from a multitude of different logical points of view. Each locus has its own „mathematicized“ formal apparatus, its own mathematical formal logic. As a consequence, the monolitical or erratic concept of world disappears as a very special case of disambiguity in a dynamic multi-verse.

The mathematicization of a world including a multitude of different logical loci, points of view, is obviously different from the more abstract model of the classical mono-logical mathematicization of the world.

Maybe, the classical model reflects an ideal world or even investigates *„the principles of reasoning for perfect worlds“* (Fitting) trans-classical logic reflects the rational principle of a conflicting, interacting, co-operating world, where the participants of these interactions creates together their own worlds in a co-creating manner. In this sense reasoning and modeling are not structures but actions.

The new concept of trans-classical logic as a complex logic of a multitude of points of view does not introduce some „shades of grey“ between the strictness of the classical concepts. There is now fuzziness here. It introduces something different, each (classical) logic gets an index that indicates the point of view of the rational agent, which indicates the separation between the different agents. Trans-classical logic is not a logic of „white and black“ nor a logic with shades of grey, it is a logic of colors, a colored logic. The logics of the living tissue are colored ones.

Each color has its own formal and operative strength.

There is no ambiguity and fuzziness in this notion of colored logics.

But these colors are not only simply identical with themselves. Each colored system is able to reflect the other systems simultaneously in its own domain. This new ambiguity is produced by the complexity of the polycontectural logic as a whole with its interaction and reflection.

Another interesting feature of colored logical systems is given by the mechanism of change of systems. A system may change from one colour to an other colour. Or some systems may permute their colours.

In the example, the blue colored logical system has a picture of the green system, the blue system is able to reflect, to mirror, to model the other colored systems in its own domain. It has replications of the other systems.

This means, green is not simply green. Green as green is green, but green as blue is the blue of the green. Green is green and not blue but green can have aspects of blue. All this is possible only because the logical and ontological principle of identity is abandoned and transcended by the game of sameness.

As a result of the plurality of formal systems as differently colored logics, each logic and complexion of these logics is localized in a structural space. Every logic has its own locus. Each logical locus gives place for the replication of other logics which are located at other logical loci. The theory of these ontological or pre-logical loci is called *kenogrammatics*.

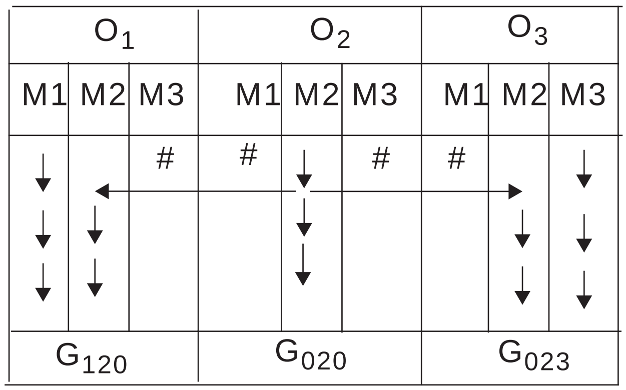
A more AI setting

In other words, we can say, that the mirroring of one contexture by another, is a belief function. One system beliefes something about another system. This more linguistic perspective opens up a connection to the work about beliefe systems, beliefe logics etc. in Artificial Intelligence (Konolige, van Harmelen).

But also to the *Algebra of Reflection* as it is proposed by Levebvre.

10.1 General poly-contextural diagram

As a result of the analysis of the structure of interaction I can introduce a general diagram or scheme of polycontextural logical and arithmetical reasoning and computation. Here I have restricted the complexitiy of interaction to the special case of 3 actors.



At each locus Oi which gives place for the logical system of the place as such the locus Oi also offers place for the modeling of the neighbor logical systems. That is, for the modeling the logics of the logics of the other interacting agents. In a classical setting, this situation is not modeled or as in computational reflection, the meta-level approach does not map on a structural level the complex logical constellations between different interacting agents. (This can be shown for Smith, Maes, Sloman, Kennedy)

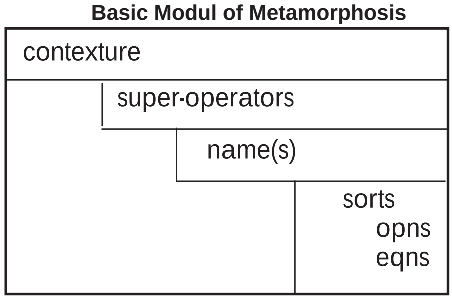
General scheme:

O1: (M1, M2, ..., Mn) § O2: (M1, M2, ..., Mn) § ... § On: (M1, M2, ..., Mn)

$n = 1/2m\ (m - 1)$

The difference between contextures as objects and contextures as morphisms has to be considered. In the very special case of m = 3, both concepts coincide

Diagramm 15 **Basic Modul of Metamorphosis**



11 Dissemination of deductive systems

11.1 Deductive system

11.2 Combining deductive systems

11.3 Disseminating deductive systems

12 Dissemination of a framework of Tableaux Logics

12.1 ASM specification of tableaux logics

Peter H. Schmitt, Egon Börger

12.2 Dissemination of ASM specification

Specifications after Börger and the problem of chiastification of their basic terms.

On the *first level* of specification Tableaus are lists of Branches. These Branches are connected with the Universe of Formulas. On this level of abstraction there is no root and there are no nodes specified.

These two universes, Branches and Formulas, are the basic concepts of the definition of tableaux-systems on this level of specification.

The question arise, how are they interconnected classically and how can they be involved in an interlocking mechanism of a chiasm in a next step?

The answer to the first part of the question is given by construction.

Universe: Fml

Universe: Branch = Fml*, list of formulas

Universe: Tableau = Branch*, list of branches

Also we are not yet speaking mathematically about trees but abaut tableaux, because our first level specification has not yet a root and nodes, I use the metaphor "tree", since all will end in some sorts of trees and forests of intermingled trees.

Positioning of one tree

(nxtbranch)

(nxtfml)

The second part of the question, dissemination and reflection, has to be developed.

Distribution and mediation of a multitude of single trees

(nxttree § nxttree § nxttree)

(nxtbranch § nxtbranch § nxtbranch)

(nxtfrml § nxtfrml § nxtfrml)

Interactivity: Trees reflecting trees

A third part has to deal with the experience we made before, we have to surpass the metaphor of (colored) trees to a more reflectional model. Trees are mirroring neighbor trees in their own domain. Trees in a forest of trees are not only strictly separated and parallel but also overlapped, interwoven and intermingled with other trees. Trees in trees is a refflectional and not a recursive notion. "Trees in trees" may serve as a metaphor.

(nxttree, nxttree,..., nxttree) § .... §(nxttree, nxttree, ..., nxttree)

(nxtbranch, nxtbranch, ..., nxtbranch)§ ..... §(nxtbranch, nxtbranch, ..., nxtbranch)

(nxtfrml, nxtfrml, ..., nxtfrml) § ....§(nxtfrml, nxtfrml, ..., nxtfrml)

Or: bundles of bundles of trees, and fibred bundles of trees.

On the *second level* of specification Trees and their Nodes are used as the basis or place were formulas are located. Quite naturally, all the architectonic definitions of distributed, mediated and reflected trees will be inherited.

Application

In a reflectional system the structure of the presupposed tableaux and later the trees of tableaux systems can be modelled as a mathematical theory.

Maybe there are new insights in the theory and techniques of graph theoretical systems like tableau and tree systems, new and faster algorithms or more abstract construtions and the introduction of new concepts. This knowledge can be given back to the first system which is based on graph theory.

As a consequence of the chiasm of Formulas and Branches in Tableau-Systems we can establish as an application an interlocking mechanism between hardware and software of a computer system even on the first level of specification. In the second system the structure of the hardware, e.g. the logic of the processor, is modelled. As an object of reflection it can be transformed and re-implemented in the former system. (Smart compilers, Evolving hardware)

12.2.1 Polycontextural tableaux logics

Despite of the true-value semantic approach in the first steps of introducing polycontextural logics, for instance, as a new interpretation for multi-valued logics by Gunther, recently by Meixl/Pfalzgraf, it should be insisted to the fact that the very idea of dissemination is strictly independent from the system being disseminated. This was pointed out at least in my paper (1981).

This remark is not only of scientific relevance, it has also a political significance. The reason is simple. In the 70th when I applied for a research project I got a denial because from the point of view of the constructivist Dialog Logic of Lorenzen, truth values are obsolete, at least secondary. And because the whole polycontextural, multi-negational, morpho- and kenogrammatic constructions of Gunther can be read as depending on truth-values, the whole proposal has to be rejecteted because it is simply nonsensical. This argument is based on a reference paper to a Gunther lecture at the Hegel Congress by Lorenzen (1962). In my 1981 paper I refuted this lack of thinking. Propaganda and thinking are obviosly two different activities. Nevertheless this attitude was not only influential but also killing. Also it is history it can easily be repeated, now from computerscience oriented logicians.

It comes without surprise, that Abramski insists on the reductability of multi-part interaction to two-person games.

Nevertheless, a lot can be learned, and is still to learn, by studying disseminated semantic based logics. (“Semantics of combined logics is hard”, Diss)

First it should be clear that the “mixing” of logics, neither in the project of combining logics nor in the polycontextural approach, is restricted at all to truth-value based logics. Mediated logics can be based on all known methods and the mix can “easily” be heterogeneous: semantic based logics mixed with constructivist systems with paraconsistent, etc.

12.2.2 Implementing PCL fragments in ML

```
tto(X,Y) :=   [T1:   T1: X && T1: Y  / / /  [] / / / ((F2: X && T2: Y) | | (T2: X && F2: Y))  / / / [] ]
              [F1:   F1: X && F1: Y / / /  []  / / / (T2: X && T2: Y)                        / / / [] ]
              [T2:   T2: X && T2: Y / / / (F1: X && F1: Y )                        / / / [] / / / [] ]
              [F2:   F2: X && F2: Y / / / ((F1: X && T1: Y) | | (T1: X && F1: Y))   / / / [] / / / [] ]
                [T3:   T3: X | | T3: Y   / / / (T1: X && T1: Y) / / / ((F2: X && T2: Y) | | (T2: X &&
F2: Y))  / / / [] ]
              [F3:   F3: X && F3: Y / / / ((F1: X && T1: Y) | | (T1: X && F1: Y)) / / / (F2: X && F2:
Y) / / / [] ]

tot(X,Y) :=  [T1:   T1: X && T1: Y / / / [] / / / [] / / / (T3: X && T3: Y)                           ]
            [F1:   F1: X && F1: Y / / / [] / / / [] / / / (F3: X && T3: Y) | | (T3: X && F3: Y)       ]
              [T2:   T2: X | | T2: Y   / / / F1: X && F1: Y  / / / [] / / / (F3: X && T3: Y) | | (T3: X &&
F3: Y) ]
           [F2:   F2: X && F2: Y / / / (F1: X && T1: Y) | | (T1: X && F1: Y)  / / / [] / / / F3: X && F3: Y ]
            [T3:   T3: X && T3: Y / / / T1: X && T1: Y                                 / / / [] / / / [] ]
           [F3:   F3: X && F3: Y / / / (F1: X && T1: Y) | | (T1: X && F1: Y)          / / / [] / / / [] ]
```

13 Classical and polycontextural logics

13.1 Tree farming of colored logics

To put all these more philosophical descriptions and ideas together in a more strict formal terminology and operative apparatus we connect these ideas of colored logics with the metaphor of the tree.

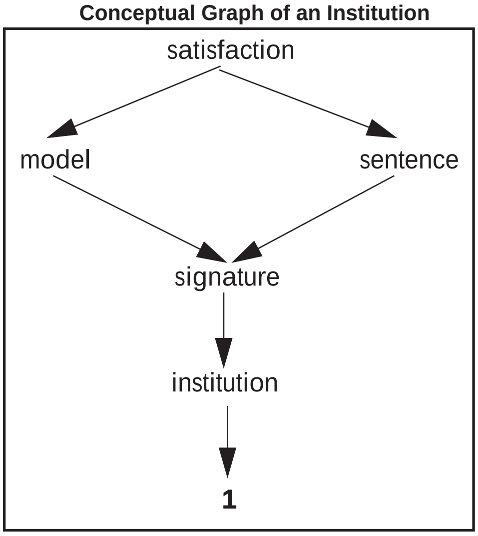
In classical logic trees figure as metaphor and as mathematical concepts on a very basic level.

We postulated that every colored logic is locally classical, e.g. each colored logic has the structure of a classic tree logic in its syntax, semantics and proof theory.

It is helpful to represent the structure of the body or the tectonics of a logical system by its conceptual graph.

13.1.1 Notation of an institution

Diagramm 16 **Conceptual Graph of an Institution**



„The arrows in this diagram represents conceptual dependencies. The notation

model —> signature

for example, means that:

the concept of model varies as signature varies.

In particular, it means that the concept of model, the one that we have in mind, cannot be independent of the concept of signature and neither can a particular model be independent of its particular signature.

In a conceptual diagram, 1 represents the absolute. The notion

institution—>1

expresses that the institution notion is absolute, for it tells us that the institution notion varies as the absolute varies – which is not at all.“ p. 488

absolute

The absolute 1 expresses that there is only one logic as such. There are many different logical systems but from a more philosophical and logical and not only mathematical point of view all these logical systems are isomorphic to one and only one logic. This is a (not provable Hypo) thesis.

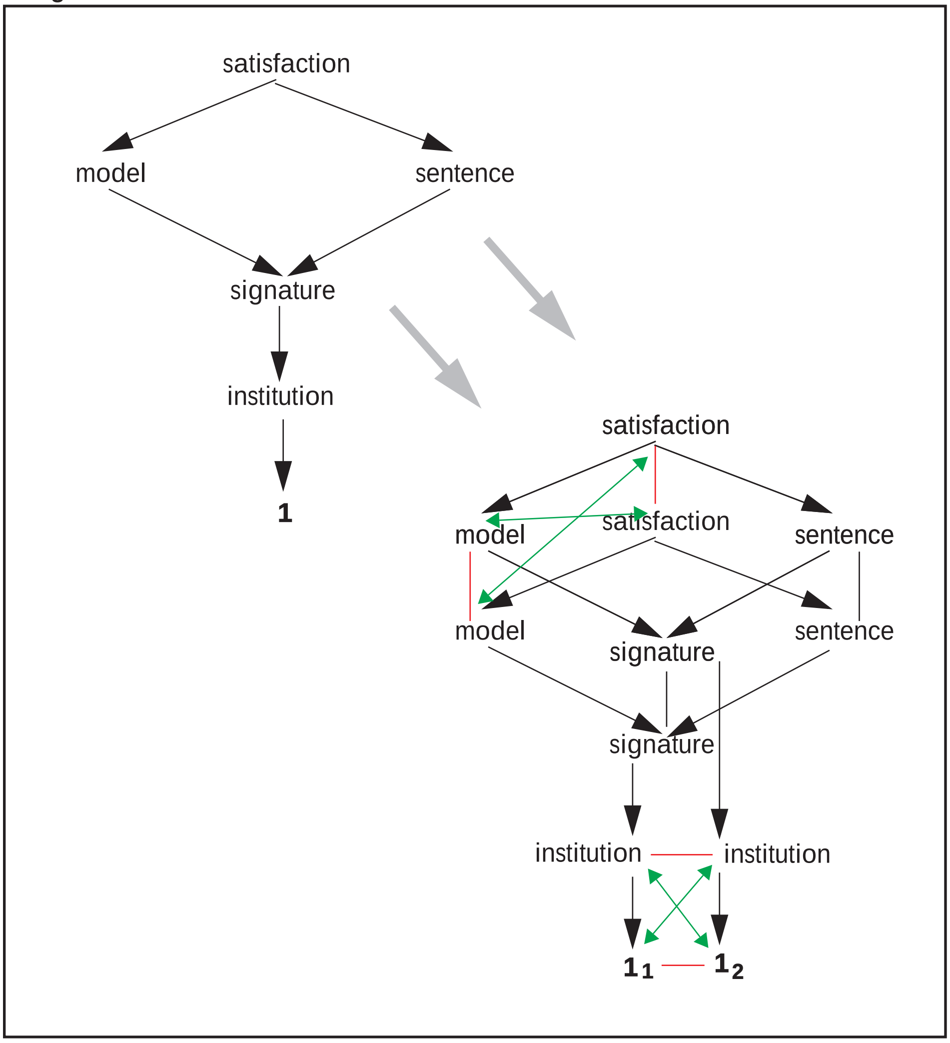
If we do not like this absolutism we should remember that the wording holds also in the more relativistic case. Considering a logical system working in it and with it means that we are working with this system and not at once or at the same time with another one. We can speak therefore of a relative absolute of the logic under consideration.

Even for mixed logics as in the project of Combining Logics there is a (relative) notion of the absolute of the system.

As we will see the situation will be totally different for polycontextural logics where a plurality of absolutes ordered in a heterarchical manner exists and the desire to have a mega-absolute for the whole complex system would turn out to be (simply) a new absolute within a plurality of other neighbored absolutes.

From the point of view of PCL the absolute means that the whole system is defined under the operation of identity ID. The system viewed as an object and as a morphism coincide.

Diagramm 17 **Monoforme Mediation of two Institutions**



institution

The institution represents the logical system under consideration as a whole, as a logic in its uniqueness.

signature

The signatures provides the vocabularies for the sentences of the logical system.

model

sentence

satisfaction

13.2 Preliminary Comments

13.2.1 Syntax

For each tree of the colored logical systems the ancestral property of its formulae holds.

In each tree there is a immediate predecessor relation which decompose the formula in its subformulas.

All that is ruled by the principle of induction over the formulas for each logical system. The induction principle is distributed over all logical systems.

Additionally to the concept of a predecessor for each system we have to introduce in the trans-classical context the operation or relation of the immediate neighbor. These makes possible some kind of permutation of logical systems over the range of the complex of distributed systems.

Further on we have to introduce the very special concept of the immediate bifurcation of a formula of a logical system into subformulas distributed over other logical systems and one part of the formula remains in its original system

From the point of view of functions and relations the operation of identical distribution, permutation and reduction are covered by the concept of total functions and relations.

Remember, all classical logical functions, e.g. of propositional logic, semantics and syntactics, are total functions.

The concept of distribution by bifurcation is possible only with the help of partial functions and relations. We call such logical functions transjunctions because they surpass the boundary of their own system and hold simultaneously in their own and in other systems. On a metalanguage level this way of speaking leads to the distinction of global and local concepts. PCL systems studies the interlocking mechanism of logical globality and locality. This mechanism guarantees that there is no hierarchical suprematy of the global over the local. Local and global concepts interlock together in a heterarchical manner.

13.2.2 Unification

On a meta-logical level, total functions are supporting symmetrical classifications and categorisations of logical particles.

This nice property of a logical symmetry is lost in trans-classical logics because of their transjunctions which are composed of partial functions. But with the help of the concept of partial functions we can introduce a new idea of a slightly more complex symmetry composed by partiality. The classical case of symmetry is then introduced as a regular composition of partial functions. This idea of a complex symmetry composed of asymmetrical functions needs additionally to the classical operator of conjugation a new operator of composition of partial functions.

13.2.3 Semantics

Truth-values in classical systems are connotated with the formal logical explication of truth and false. Formal truth and formal false do not involve ontological questions about truth and falsehood of sentence. This belongs to the level of examples for formal logical sentences.

In PCL systems truth values, if we are choosing a truth value semantics, have to realize two jobs, the first is more or less the same as in classical systems, they represent the formal logical concept of true and false of propositions of their logic. The second job is very different, they have to mark in which logical system the difference of true / false holds. Therefore they have an index of the system they belong or origin: {Ti, Fi}. As the splitting function of transjunction shows these truth values can occur in different systems at once. As a result, the whole semantics of propositions, sentences, phrases and truth-values has to be deconstructed.

As explained metaphorically earlier these logics are not isolated from each other but combined to a complex logical, or ultra-logical, system. Otherwise they would behave totally in parallel and it would be at least at first only an application of one classical logic at different epistemological places without any interaction or mediation.

A first, quite natural and elementary, connection is given by a (special) linear ordering of the systems and their truth-values.

To not to confuse this kind of order with other ordering systems I call it a chiastic linear order of truth-values. A chiasm is defined as a tupel of order, exchange, coincidence and positioning relations.

therefore the semantics of PCL is not defined over a set of truth-values but over an order, a chiastically ordered structure of truth-values.

The difference becomes obvious for the semantics of ternary and general n-ary logical functions or logical compositions. This difference between set based and order based semantics is hidden for the typical binary case.

As a natural consequence the notions of sentence, model and satisfaction have to be distributed over the indices of their semantics.

13.2.4 Consequence relations and proof theory

For each single logical system of the PCL complex there exist a consequence relation and a proof theory for this logic.

The consequence relation for the whole system of logic is composed of the distributed single consequence relation of each logic.

We will choose the analytical tableaux method as our proof procedure.

13.3 Signatures and Vocabularies

$$\frac{Voc^{(3)}}{Voc^1 \mid Voc^2 \mid Voc^3}$$

The distribution of vocabularies in PCL is produced not only by the genuin vocabularies of the single logics but also by the interaction between the logics. Vocabularies in general are not only objects or sets but morphisms.

In this sense vocabularies are not the initial objects of their category. They can be initial but only from a local point of view,

$Voc^{(3)}$: $Voc^{(3)}$ —> $Voc^{(3)}$

with Id, Perm, Red, Bif as operators.

$$\frac{Voc^{(3)}}{Voc^1 \mid Voc^2 \mid Voc^3 \mid \mid Voc^1 \mid Voc^2 \mid Voc^3 \mid \mid Voc^1 \mid Voc^2 \mid Voc^3}$$

Monoform Voacabularies

For the sake of a gentle introduction I choose a monoform setting of the vocabularies (alphabets) and the sentences. All vocabularies have the same, but not the identical, signs. All sentences are balanced, that is, they have the same syntactical structure. That means, that everything is in full parallelism or accordance with the classical setting.

Local definitions

For all logical systems we have locally:
Voc^i = {p, q, r, ..., and, or, not, trans, ...) and the same type of brackets.

The sentences Sen^i are defined locally recursively over the Voc^i.

1.

2.

3.

4.

X, Y, Z, .. are metasymbols for the object symbols build up with p, q, r, ...

Global definitions or definitions of mediation

Composition of $X^{(3)}$

$X^{(3)} = X^1 * X^2 * X^3$

Decomposition of $X^{(3)}$

Symbols or variables:

$$\frac{X^{(3)}}{X^1 \mid X^2 \mid X^3}$$

Negations

$$\frac{X^{(3)}}{N_1 X^{(3)}}$$

$$\frac{X^{(3)}}{N_2 X^{(3)}}$$

Junctions and Transjunctions

Junctions

Composition

$$\frac{X^{(3)}, Y^{(3)}}{X^{(3)}\ \text{ooo}\ Y^{(3)}} \qquad \text{o=\{and, or, impl, ...\}}$$

Decomposition

$$\frac{X^{(3)}\ \text{ooo}\ Y^{(3)}}{X^1 \text{ o } Y^1 \mid X^2 \text{ o } Y^2 \mid X^3 \text{ o } Y^3}$$

Decompositions of Transjunctions

$$\frac{X^{(3)}\ \text{ooo}\ Y^{(3)}}{X^1 \text{ o } Y^1 \mid X^2 \text{ o } Y^2 \mid X^3 \text{ o } Y^3} \qquad \text{o=\{transj1, ...\}}$$

$X^1 \text{ o } Y^1 \mid X^2 \text{ o } Y^2 \mid X^3 \text{ o } Y^3$

$X^1 \text{ o } Y^1 \mid X^2 \text{ o } Y^2 \mid X^3 \text{ o } Y^3$

As in classical syntax where not all possible combinations of signs from the alphabet are defining sentences I have to introduce a new rule of combining signs, especially between logical connectives, negators, junctors and transjunctors from different logical systems. These rules are called in the literature of polycontextural logics the VB-Rules (Vermittlungs-Bedingungen = Conditions of mediation).

As in classical logic where the pure combinatorial possibilities are restricted by semantical considerations I have to do the same for the VB-rules. In relation to a specific choise of the concept of mediaton between logics or of logics which is reflected by the VB-rules the possibilities of combining logical connectives of different systems is ruled. A lot of combinations are ruled out. Not everything can be mediated whith everything.

#Comment:
To state it just at the very beginning of my exposition of PCL, if someone needs to collect these 3 different vocabularies under one new all 3 subsuming vocabulary, it would have been produced simply a new vocabulary as a basis for a new 4-contextural logic.#

#Comment:
Each system has its own vocabulary and is additionally able to reflect the terms of the neighbour systems in its own contexture. Therefore, the terms of the other systems can occur in its range. This doesn´t mean that all systems have the vocabularies of all other systems included in their own vocabulary. The representation of the vocabularies in other systems is the result of operations of interactions between the genuin vocabularies, like Bifurcation. The other vocabularies don´t occur in the vocabulary of a system but in the contexture or at the locus of the system. Therefore they are still disjunct or irreducible different by their belonging to another system, marked by their colour or their index.#

#Comment:
On a metatheoretical level we have to be aware about the following situation. If we speak about signatures as morphisms and as categories we are using category theoretical terms. And it sounds as if PCL turns out to be simply a special theory of category theory.
This is the case e.g. of fibered logics. *"However, the concept of abstract fiberings is general enough to allow us to "mix logics" in the sense that different logics can occur as local fibres."* Pfalzgraf

If we speak about a plurality of irreducibly different institutions and signatures or vocabularies we are forced to use also a multitude of irreducibly different category theories. We don´t have this polycontextural category theory at disposal. This very new theory would have to be developed on the basis of polycontextural logics in a similar sense as classical category theory is based on classical logic (many-sorted eqational logic). But I am just introducing PCL.
Therefore, we have to use terms and methods of classical theories as our first step tools of thematizing, modelling and formalizing the trans-classical domain of PCL.#

14 Modi of interactions: super-operators

It seems to be natural to accept that there are at least the following operations of interaction between the systems to observe.

ID: *Identity*. Mappings of a system onto itself

RED: *Reduction*. Mappings of systems into other systems (Acceptance)

PERM: *Permutation*. Transversions between systems, exchange of positions

BIF: *Bifurcation*. Mappings of systems onto themselves and simultaneously into others.

IDi: (G1G2...Gi...Gn) ===> (G1G2...Gi...Gn)

PERMij: (G1G2...Gi Gj...Gn) ===> (G1G2...GjGi...Gn)

REDij: (G1G2...Gi Gj...Gn) ===> (G1G2...GiGi...Gn)

BIFi : (G1G2...Gi...Gn) ===> (G1G2...(Gi1...Gin)...Gn)

I call these additional operators *super-operators*. In contrast to the first operators of the natural system, *zero* and *suc,* which are defined inside the system, e.g. *locally*, the superoperators are defined between the natural systems of a collection of cloned systems and are therefore of a *global* character.

Algebra of super-operators

ID(ID) = ID

ID (X) = X, X= {ID, PERM, RED, BIF}

cycles: PERM(PERM(...))

final reduction: RED(RED) = RED

iterative explosion (?): BIF(BIF) =

BIFi : (G1G2...Gi...Gn) ===> (G1G2...(Gi1...Gin)...Gn)

??: BIFij : (G1G2...(Gi1...Gin)...Gn) ==> (G1G2...((Gi1...(Gj1...Gjn)...Gin))...Gn)

?: This iteration does not iterate the representation diagram of the system

BIF2: ((G1, #, #), (#, G2, #), (#, #, G3)) ==> ((G1, G2, #), (#, G2, #), (#, G2, G3))

BIF2: ((G1, G2, #), (#, G2, #), (#, G2, G3)) ==>

((G1, G2, #), (#, G2, #), (#, G2, G3))

it is a step wise procedure inside of the scheme.

14.1 Super-operators in disseminated natural systems

DISS$^{(3)}$ (object $_{nat0}$) = object1$_{nat0}$ § object2$_{nat0}$ § object3$_{nat0}$

object1$_{nat0}$ —> object1$_{nat0}$
object2$_{nat0}$ —> object2$_{nat0}$
object3$_{nat0}$ —> object3$_{nat0}$

object1 —> object1
object2 —> object2
object3 —> object3

ID(3) (zero(3)) : —> nat(3)
zero1: —> nat1
zero2: —> nat2
zero3: —> nat3

(ID PERM2 PERM3) zero(3) : —> nat(3) with
zero1: —> nat1
zero2: —> nat3
zero3: —> nat2

(ID BIF1,3 ID)zero(3) : —> nat(3) with
zero1: —> nat1 simul nat2
zero2: —> nat2
zero3: —> nat3 simul nat2

(RED2 ID ID) zero(3) : —> nat(3) with
zero1: —> nat2
zero2: —> nat2
zero3: —> nat3

The same procedure is to apply to the successor function *suc* and later to other operators like the addition *add*.

In contrast to the purely parallel construction we have to introduce a more complex notation for the general case.

Untill now I have treated zero as an object and different types of zeros as objects belonging to different contextures. This is a quite conservative introduction. In correspondence to the idea of proemiality and polycontexturality, it is more appropriate to think of zero as an action. In this sense zero is the notation of the action of beginning. There are many beginnings but no single origin. Actions, and especially simultaneous actions, are not necessarily connected with the notion of identity. In contrast, objects are very close to the notion of identity. The classical concept of an object coincides more or less with this notion of identity. Actions are not given as ontological entities, therefore they have to be interpreted. An interpretation involves an interpreter, which is a point of view. Because there is no single privileged point of view there is a multitude of interpreters, interpreting successively or simultaneously the realizations of actions.

The Operator (ID BIF1,3 ID)zero(3) suggests that there are additionally to the genuine objects of the systems objects from the neighbor systems too.

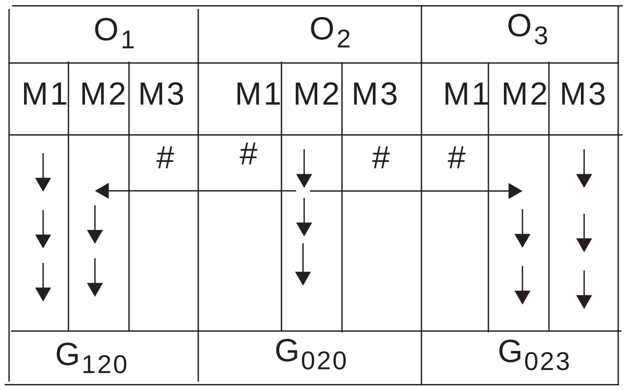
The first object of the application of the operator (ID BIF1,3 ID) to zero(3) is therefore:

(zero, zero, #), (#, zero, #) (#, zero, zero)

$$\frac{(ID,\ BIF^{1,3},\ ID)\ (OP1,\ OP2,\ OP3)}{(OP1,\ OP2,\ \#)\ (\#,\ OP2,\ \#)\ (\#, OP2,\ OP3)}$$

In this sense the Operator (ID, $BIF^{1,3}$, ID) is only a perhaps misleading abbreviation of the above more explicit notation.

Diagramm 18 **Transition-Diagramm of the Operation (ID BIF1,3 ID)**



What´s new?

In contrast to the previous ebook „SKIZZE“ the idea of unrestricted „cloning“ also of non-transjunctional functions is new. This idea is realized by a more radical approach to the *super-operator* construction between formal systems as defined before and additionally by a more radical application of the category of „sameness" to all objects.

The idea of reflectional modeling is more advanced and more explicit with the concept of *reflectional architectures* of formal systems.

The concept of *arithmetical neighbors* is more explicit and the status of the initial object zero in contrast to its neighbors is explained philosophically and formally.

The idea of categorial *metamorphosis* on the basis of the proemial relation is introduced.

The phenomenon of over-determination, that is of the possibility of a multitude of interpretations of arithmetically interpreted morphograms is linked to the metaphor of *togetherness*.

14.2 Towards a metaphor of togetherness

Time is coming that we have to learn to live together at the same place without any chances of excluding each other.

Earlier on we solved this problem of living together with the help of the operation of separation and exclusion. Nobody had to live at the exact same place as someone else. The separation of two beings has given the space and possibility for interaction and cooperation between these entities. The separation was the fundamental condition for the possibility of interaction (cooperation, communication, co-creation, etc.).

Now it seems that we have reached the point that we have to develop a concept of living together in which we have to take place together simultaneously at the exact same place.

It will turn out that this way of living together is prior to any separation and therefore to any form of interaction and cooperation.

In classical terms two objects must be identical if they are not different. They are different if it is possible to separate them.

How could togetherness be thought and conceptualized whithout the assumptions of identity and diversity and the procedures of identification and separation?

How could this be possible? First of all, it isn´t possible at all on the premisses of the traditional concepts of place, object, state, separation and interaction. The reason is obvious, all these concepts are fundamentally rooted in the ontological and logical principle of identity.

In technical terms, how could it be possible that two different states of a computation could occupy the very same place in the computing space of their machine?

Obviosly this is not possible at all. It isn´t possible neither from the point of view of the machine nor of the basic concepts of the programming languages. It is impossible for logical and physical reasons.

Simply take the example of the definition of EQ in the programming langauge LISP:

EQxy $=_{def}$ if (eval x) = (eval y) then true
else false

The equality EQ of x and y is strict, it is fullfilled or it is not – tertium non datur. The logic which is ruling these conditions is strictly binary. It is in whatever form a two-valued logical system which is ruling the conditions of equality. All in all, there are three levels of equality involved ruling this definition: the definitional ($=_{def}$), the defined (EQ) and the defining (=).

There is also no chance on the level of implementation on a more physical level of a machine. Two states are equal if they have the same address, and if they have the same address they have the identical physical realisation which is the equality =.

It seems that there is no chance to escape this situation.

Brian Smith has done a lot of work to clear and liberate this situation of strict ontological identity and bivalence in computer science. But at the end of his sophisticated work *"On the Origin of Objects"*, MIT 1996 he introduces again a classical foundation for his quite liberal pluralistic and relativistic concept of truth and identity.

It also doesn´t help much if we refere to our philosophers of the flux, Heraclit, Hegel, Whitehead, Deleuze / Guattari, Irigaray etc. because they don´t touch the topics of formalism and computation. The same is true for the more philosophical and theological work on togethernes by Heidegger, Buber, Binswanger, Levinas and others.

We will see, that togetherness in this study is not to be reduced to an anthropological category of, say Mitsein, Miteinandersein, Begegnung.

14.3 Kenogrammatic foundations of togetherness

Obviously we need a scriptural system which is beyond identity (of its signs and operations).

Everyone knows that the semiotical basis of any programming language is only possible because of the two fundamental operations of identification and separation of its signs.

Without separation there are no signs, and vice versa, without identification there is no separation. And without separation and identification there are no signs. And without signs there is no language. And so on...

Bad enough, my job has to be to develop a way of writing without or maybe even beyond the interlocking game of identification and separation.

How could this be possible? Try it with the idea of kenogrammatics!

There is not much but enough work done on this topic of kenogrammatics to understand at least the very idea of this quite radical exercise.

The procedure I propose works as follows:

1. Different from semiotics I introduce the concept of kenogrammatics and its morphograms with some definitions and rules.
2. A simple mapping of natural numbers onto the kenogrammatical system defines the trito-numbers of this kenogrammatics.
3. I introduce a decomposition of these trito-numbers into its binary parts. The result are the binary number systems with their rules of distribution and mediation based on the rules of kenogrammatics.
4. As a result we observe that the same „physical“ kenogrammatical „address“ gives simultaneously place to semiotically different objects (words).

14.3.1 A short introduction to the idea of kenogrammatics

Without jumping into the deepness of fundamental and foundational studies we can try to do a simple exercise.

In some sense we find „in a stream of signs“ a pattern and we are able to distinguish some features of it.

For short we introduce an arbitrary pattern MG with the feature (ABBCA),

MG = (ABBCA).

First we say that this is not a list or a chain of signs, at least we have not to presuppose it. May be we simply don’t know it now.

We call it a pattern. This means, that every other figure of the same structure is equivalent to this pattern. Patterns in kenogramatics are called *morphograms*. For convenience we accept the alphabetical order of the features for our notations.

MG = (ABBCA) = (BAACB) = (CAABC) etc. but also (*##+#) etc.

What can we do with such a morphogram? We can study it´s behavior!

Is the process of finding a pattern in a stream of signs not again involved in the same paradox of identification and separation as we know it from the previous introduction of signs? What is the difference?

Metaphor: Dynamical Dominoes and possible continuations

As a metaphor, think of a dominoes game where the dominoes are allowed to change size and the composition rule preserves the two-dimensionality in contrast to a Left-Associativ-Grammar (Hausser, 1989). The main principle is *possible continuations* in contrast to the well known *principle of substitution.* A crucial point of this principle is, as far as I understand it, that it doesn't appeal to a beginning (of recursion) and a pre-given alphabet of signs. It seems that this idea is more coalgebraistic than based on the concept of an algebra.

Evolution

We will observe that the morphogram MG has the possibility to change. One form of change may be its growing. But this will happen in a strict way ruled by its intrinsic structure. Our morphogram may grow by repeating elements of its structure or it may grow in producing a new feature according to the underlying principle of possible continuations.

MG0 = (ABBCA)

MG1 = (ABBCAA)
MG2 = (ABBCAB)
MG3 = (ABBCAC)
MG4 = (ABBCAD)

These are the only possibilities for the morphogram MG0 to grow.

Suppose we want MG0 to grow to the new morphogram with feature (ABBCAE). But this figure is equivalent to the pattern (ABBCAD). It produces nothing new in comparison to the MG4.

Emanation

Another way of change can be realized without growing. The morphogram MG has the possibility to reduce or augment its internal structure between a minimum and a maximum of differentiation.

Reduction: (ABBCA) —> (ABBBA) —> (AABCA) —>... —> (AAAAA)

Differentiation: (ABBCA) —> (ABBCD) —> (ABCCD) —> (ABCDE)

Emanation and Evolution

It turns out that each morphogram we observe is constituted as an interaction of an evolutional and of an emanational procedure.

The rules of emanation are symmetrical in contrast to the asymmetry of the rules of evolution.

This is not in contradiction to the fact that all single morphograms can be produced or listed by the procedure of evolution alone.

Some interesting properties of morphograms

Monomorphies
Verknüpfung, Verkoppelung, Verschmelzung
concatenation, composition, merging
A new form of equality
Addition
Multiplication
Reflection

14.3.2 Natural numbers in kenogrammatic systems

As a first step in our construction we can introduce a mapping procedure of the natural numbers onto the collection of morphograms.

NKG: Nat ––> MG

MG4 = (ABBCAD) may become TZ = (122314)

Because the equality of two morphgrams depends only on its structure and not on its symbols the same happens for the numerical patterns.

The following exemples are trito-numerical equivalent (tzeq):

(122314) tzeq (23324) tzeq (322435)

14.3.3 Binary numbers in kenogrammatical number systems

We say that each kenogrammatical number TZ can be decomposed in several ways into its n-ary components. We will work with *binary* decomposition of trio-numbers.

Each pair of numbers i, j of a kenogrammatical trito-number TZ belongs to a binary number system (BTZ) defined by theses two numbers.

(122314)

S1 = (1, 2) S4 = (3, 4)
S2 = (2, 3) S5 = (2, 4)
S3 = (1, 3) S6 = (1,4)

(122) is in S1 and (223) is in S2 but they are *overlapping* in the element (2). Also, (31) is in S3 and (1,4) is in S6 they overlap in the element (1) which belongs simultaneously to the binary system S3 and to the binary system S6.

We say composition and decomposition is restricted to the overlapping of only one element. The general case of overlapping with more than one element in common is called merging or fusion. Overlapping with no common element is called concatenation.

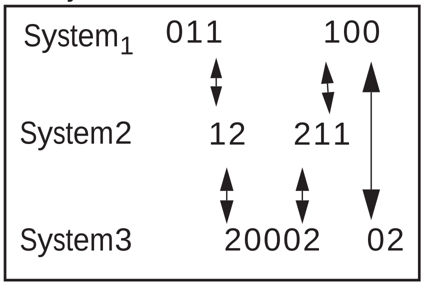
As a further example we use the operation of decomposition of a morphogram into its parts. The end and the start of two successive parts of two different binary number systems have a common element which is marking the jump from one system to the other.

Take the kenogrammatical number TZ= (0112000211002)

BTZ1 = 011/12/20/000/02/211/100/02
with $S_1S_2S_3S_1S_3S_2S_1S_3$, l=8

BTZ2 = 011/12/200002/211/100/02
with $S_1S_2S_3S_2S_1S_3$, l=6

Diagramm 19 **Systemwechsel BTZ1**



14.3.4 Chiasm of togetherness

What are the rules of the togetherness of binary structures (numbers) at the same place (locus) of a morphogramm?

We observe that we don't use the concepts of time and space to explain the behavior of our patterns. Obviously kenogrammatic patterns, morphograms, are not ruled by the principle of semiotical identity.

The attribute of an object depends of its context, as what it is thematized or as what it figures. An object has in this sense no fixed attributes. The most general attributes of our objects are their beginning and their end or in another, but quite different wording, their head and their tail. At this point we are observing only the behavior of head (begin) and tail (end) to describe the functioning of the chiasm.

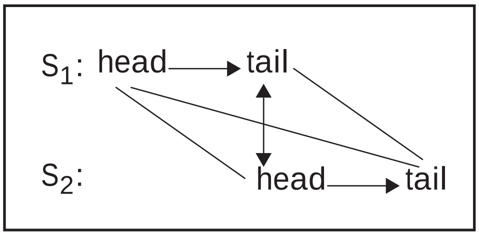
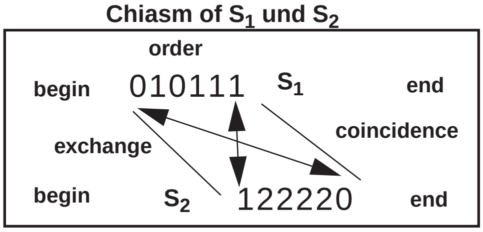


Diagramm 20 **Head-tail-Chiasm**

Diagramm 21 **Chiasm of S_1 und S_2**



Some importand relations between head and tail in a chiasm.

Between head and tail we observe an *order relation* in the sense that first there is the head and then follows its tail.

What figures as head in one system occurs as tail in the other system and vice versa. They realize an *exchange relation* between the two concepts, head and tail in respect of the two levels S_1 and S_2.

Between the head (tail) of one system and the head (tail) of the other system we observe a *relation of coincidence* in the sense that both occurrence of the concepts belong to the same kind or category distributed over two levels.

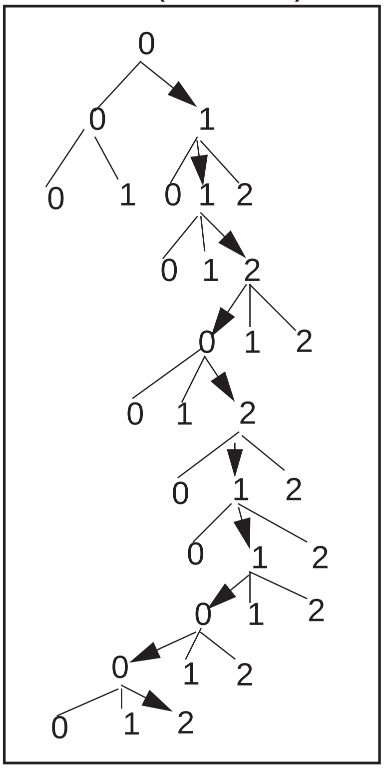
To summarize: A chiasm between the concepts of head and tail is characterized by the relations of order, exchange and coincidence distributed over two loci.

We accept the constraint for this case that the tail of System S_1 and the head of System S_2 has only one element, therefore the exchange relation has only one element in common in contrast to the common definition. With that we have an interlocking mechanism of the composition of objects of different levels.

14.4 Interpretations and specifications

Diagramm 22

TZ=(01120211002)



14.4.1 Dekompositionen

Erstes Beispiel

Die als Trito-Zahl notierte Ereignisfolge TZ im Gewebe dreier Binärsysteme S_1, S_2 und S_3 mit den 3 Elementen {0, 1, 2}. Je 2 Elemente definieren ein Binärsystem.

TZ=(01120211002)

lässt mindestens zwei Deutungen zu:

a) 011/12/202/211/100/02 mit der Systemfolge: $S_1S_2S_3S_2S_1S_3$
b) 011/112/202/211/1100/002 mit Systemfolge: $S_1S_2S_3S_2S_1S_3$
Hier ist zwar die *Subsystemfolge* der beiden Auflösungen die gleiche, die Auflösungen selbst sind jedoch verschieden in ihrer jeweiligen Länge.

Weiteres Beispiel

Die Trito-Zahl TZ= (0112000211002)
lässt Deutungen zu, die sowohl die Subsystemfolge als auch die Länge der Subsystemfolgen betreffen.

a) 01/12/20/000/02/211/100/02 mit $S_1S_2S_3S_1S_3S_2S_1S_3$, l=8

b) 01/12/200002/211/100/02 mit $S_1S_2S_3S_2S_1S_3$, l=6 a eqtz b

S1: {0,1}
S2: {1,2}
S2: {0,2}

TZ=(01120211002)

this number can be produced by two different successor sequences
##the order has to be inverted####

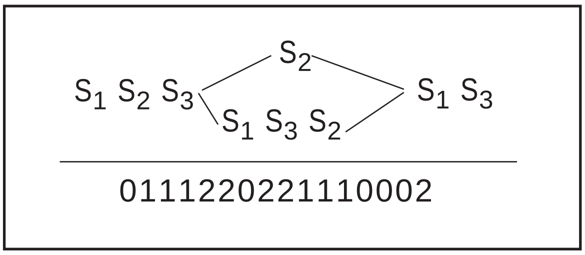
a)
suc1(suc2(suc2(suc3(suc3(suc3(suc2(suc2(suc1(suc1(suc1(suc3(suc3)...(zero1)...)

011/12/202/211/100/02 with the subsystem sequence: $S_1S_2S_3S_2S_1S_3$

and by
b)
suc1(suc1(suc2(suc2(suc2(suc3(suc3(suc3(suc2(suc2(suc2(suc1(suc1(suc1(suc1(suc3(suc3(suc3)...(zero1)...)

011/112/202/211/1100/002 with the subsystem sequence: $S_1S_2S_3S_2S_1S_3$

Different interprations of a poly-numerical event



Gaps in natural number series?

"The law which we applied was the principle of numerical induction; and although nobody has ever counted up to 10^{1000}, or ever will, we know perfectly well that it would be the height of absurdity to assume that our law will stop being valid at the quoted number and start working again at 10^{10000}.

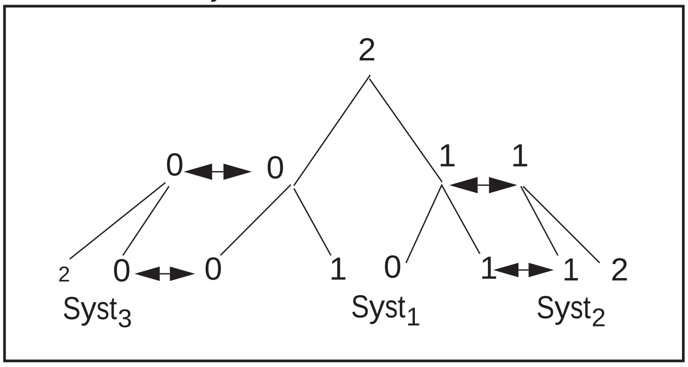
We know this with absolute certainity because we are aware of the fact that the principle of induction is nothing but an expression of the reflective procedure our consciousness employs in order to become aware of a sequence of numbers. The breaking down of the law even for one single number out of the infinity would mean there is no numerical consciousness at all!" Cybernetic Ontology, p. 360

Switches between arithmetics

Splits in the concept of natural number series, bifurcations of the natural

Diagramm 23

I Systemwechsel mit Bifurkation



Comments

"every element is distinct"

"infinite number"
plurality of "interpretations" vs. plurality of natural interacting systems

plurality of beginnings

steps inside and jumps between the distributed natural systems.

problems of circularities: the numbering (EMAN, EVOL) of the systems involves meta-arithmetics on the level of the superoperators.

the strategies of finding institutions and the strategy of combining, dissemination and fibring formal systems are complementary. One is searching for a common structure, homogeneity, the other is accepting diversity and is searching of modelling heterogeneity.

initial vs. final semantics, junk, confusion

abstract type, interpretations

inheritance: intra- vs. trans-contextural, inheriting by assimilation vs. by modification (accomodation)

inheriting by assimilation is the classical version.

inheriting by modification (of categorical distinction) is the chiastic change from one type in a contexture to another type in another contexture.

E.g. simple object in one contexture to complex object in another contexture. This is similar to inheritance by crossing the borders

This text is part of an excercise to write my thoughts in english instead in german as done before.

In parts it is also something like a paraphrase of a chapter in Michael Downward´s book.

14.5 Superpositions of the superoperators

In the case of the classical fundamental system with its operations *zero* and *suc* we have only a very limited possibility of applying the operator *suc*. It can be applied only iteratively on itself: *suc(suc(suc(……)))*. That´s it. That´s also the reason way this system is called natural. Nothing artificial happens to it. This changes in a very small sense with the introduction, say of the addition operation. Because two strings can now be equal or not equal and this possibility starts some kind of reflection. We have to introduce mechanisms of comparison and decisions.

The cloned systems I am introducing are still quite natural. Locally they behave at least to most parts the same as classical natural systems. Because the introduction of a plurality of natural systems and because of their interactivity between them I call these systems *ultra-natural* systems.

Internally these ultra-natural systems realize bravely their iterations of their *suc* function based on their introductory function *zero*. But between the single subsystems of the complexion of successor systems we observed the additional operations of interaction, e.g. SUP-OP: {ID, RED, PERM, BIF}. It is natural to understand that we can combine these operators in many ways. As a result we are producing a quite interesting system of superpositions of these operators applied to the classical operators zero and suc.

Because I am modeling the ultra-natural system along the lines of the classical natural system I have to develop step by step the evolvement of new concepts and to define the range of there applicability and combinatorics of the possibilities of the new system based on at least the 4 operators of interaction and the two internal operators of introducing the objects, zero and suc.

We are studying the dissemination of only three systems. Where does this number 3 comes from?

Additionally to the known super-operators I introduce the operators of diremption, EVOL for evolution and EMAN for emanation. It is exactly the operator EVOL which delivers the number of the distribution of the systems. That is the operator DISS turns out to be a non specified case of diremption (repetition, replication).

EVOL(3)nat0=

DISS(3)nat0=
nat(1)0
nat(2)0
nat(3)0

Cloning sameness and difference

EVOL and EMAN

15 Tactics of implementing polycontextural systems

Because we still don´t have the trans-classical computing systems we are forced to model and to implement our trans-classical formalisms in the framework of classical concepts. One obvious way of modeling the disseminated objects is done by using many-sorted logics (many-sorted abstract types).

The sorts of a many-sorted logic are treated as universes (names, contextures) of abstract objects.

Logic —> sorts/ /Logics ==> PolyLogic —> PolySorts

We shouldn't forget, that I am using in this implementation scheme the term sorts as a logical term, not to be confused with sorts as data-types. That means, that our sorts have also to include the *control structures* of the programming languages. Poly-sorts in this sense are not only different vocabularies and dictionaries but are also implementing different control structures. And other stuff too.

```
DISS(3)nat0:
name
      nat(3)0===

super-operators
      ID, PERM, RED, BIF
          nat(3)0 —> nat(3)0

poly-sorts
     sorts1
            nat0
     sorts2
            nat0
     sorts3
            nat0

poly-opns
     opns1
             zero: —> nat
            suc  : nat  —> nat
     opns2
             zero: —> nat
            suc  : nat  —> nat
     opns3
            zero: —> nat
           suc  : nat  —> nat

DISS(3)nat0===

3-contextures
           contexture1 § conttexture2 § contexture3


super-operators
ID
PERM
RED
BIF

contexture1
nat0 =
    sorts
       nat
opns
      zero: —> nat
      suc  : nat  —> nat

contexture2
nat0 =
    sorts
       nat
opns
```

```
    zero: —> nat
    suc  : nat  —> nat

contexture3
nat0 =
   sorts
     nat
opns
    zero: —> nat
    suc  : nat  —> nat
```

16 Towards poly-versal algebras

16.1 Universal algebra vs. universal co-algebra

16.2 Poly-versal dissemination of algebras and co-algebras

16.3 Neutrality of the disseminative mechanism

```
abstract polycontextural object===
type of dissemination:: linear ...star chiasm
complexity of dissemination:: DISS = {EVOL, EMAN}
complication of dissemination:: COMP = {iterative, accretive, integrative}
super-operator:: SUP-OPS = {ID, PERM, RED, BIF}
   sup-op: complexity ——> complexity
              contextures —> contextures
IDi:       (G1G2...Gi...Gn) ===>  (G1G2...Gi...Gn)
PERMij: (G1G2...Gi Gj...Gn) ===>  (G1G2...GjGi...Gn)
REDij:    (G1G2...Gi Gj...Gn) ===>  (G1G2...GiGi...Gn)
BIFi :     (G1G2...Gi...Gn) ===>  (G1G2...(Gi1...Gin)...Gn)
```

name per contexture :: name^1 ....name^s

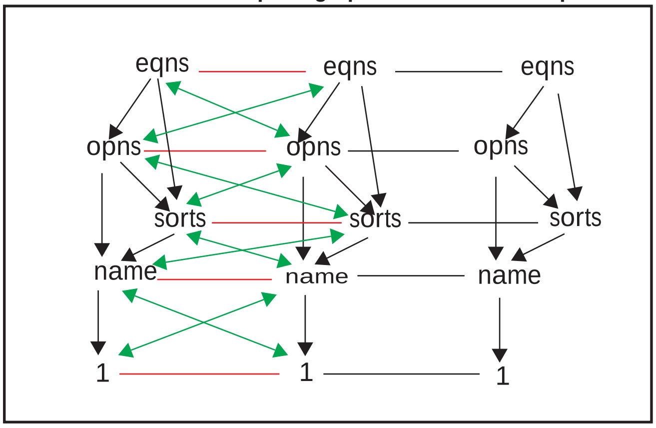
sorts sorts

opns opns

eqns eqns

DISS$^{(m)}$ abstract-object = abstract-object § abstract-object § .... § abstract-object

Diagramm 24 **Conceptual graph of disseminated specifications**



Short or canonical version of dissemination:

abstract polycontextural object===
trans-contextural head
complexity of dissemination:: DISS = n contextures, constant
super-operator:: sup-ops = {ID, PERM, RED, BIF}
sup-op: complexity —–> complexity
contextures —> contextures

IDi: (G1G2...Gi...Gn) ===> (G1G2...Gi...Gn)
PERMij: (G1G2...Gi Gj...Gn) ===> (G1G2...GjGi...Gn)
REDij: (G1G2...Gi Gj...Gn) ===> (G1G2...GiGi...Gn)
BIFi : (G1G2...Gi...Gn) ===> (G1G2...(Gi1...Gin)...Gn)

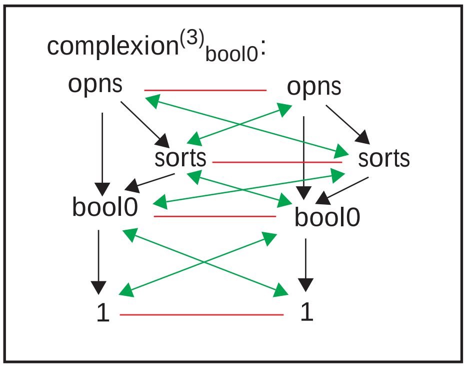
intra-contextural heads
name per contexture :: $name^1$ ....$name^s$
sorts sorts
opns opns
eqns eqns
with
type of dissemination:: linear
complexity of dissemination:: DISS = {EVOL}, constant = n
complication of dissemination:: iterativ

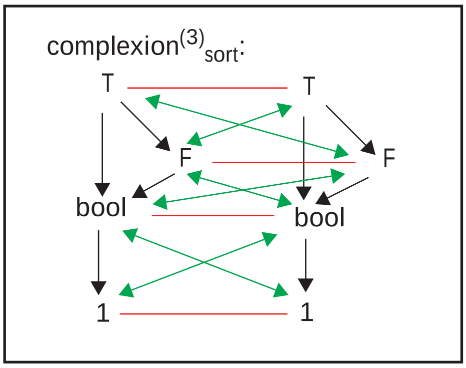
mono-form vs. poly-form
abstract algebras
single-sorted vs. many-sorted
homogenous vs. heterogenous universal algebras

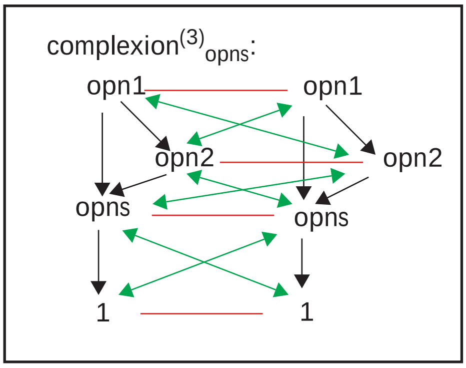
17 Polylogical abstract objects

The idea of proemiality and of poly-arithmetics can be clarified one step further if we introduce another very simple abstract object, the object bool0 and later the object bool1. To deal with equations in arithmetics we need this boolean object, because equations can be true or false.

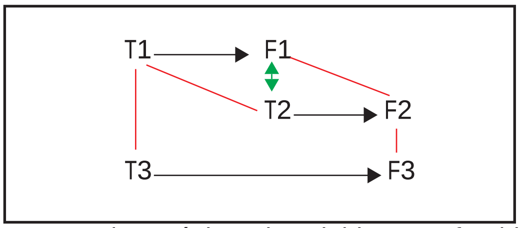
bool0=
sorts
bool
opns
T, F : —> bool







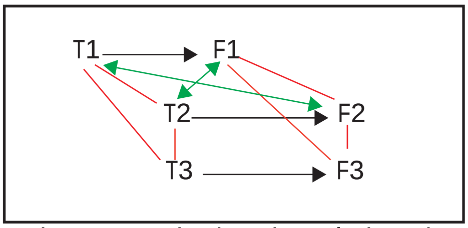
Short, reduced, chiastic diagram with a third, mediating system bool3



In the previous diagrams I haven´t introduced this type of a third mediating system.

Remarks on mediating systems

A more explicit characterization of the relationality of disseminated systems is given in the following diagram where the same type of objects, the initial and the final objects, are collected together.



The system3 is mirroring or monitoring the relations between system1 and system2, its function is reflectional towards the distributed and mediated systems. The whole figure could be called *"reflected chiasm"* because the difference between system1 and system2 is reflected in system3.

The relationship between the reflectional and the disseminational systems is not static or a pre-given order of hierarchy, first the dissemination, second the reflection. It is natural to accept an interplay of chiastic order between reflectional an disseminational compound of the polycontextural system. Insofar as the reflectional system is occupying a place for its own, it is also a distributed system, and mediated with the 2 other systems. This will become clearer in more complex situations.

This version of conceptualizing the situation will be later of some relevance, especially if we ask about the logical truth or initiality and the logical false or finality of the whole complexity at once. That is, the "truth" of the whole as (T1, T2, T3) and the "false" of the whole at once as (F1, F2, F3).

Now, we are ready to apply these distinctions to the general conceptual graph of disseminated systems or objects.

$sort^{(3)}$ ===

$bool^{(3)}$ ===

$bool^1$ § $bool^2$ § $bool^3$

$bool^i$ = {Ti, Fi}, i=1,2,3

remarks

Short notation for the diagrams of mono-form and linear disseminations

$bool0^{(3)}$ = $bool0^1$ § $bool0^2$ § $bool0^3$
$sorts^{(3)}$ = $sort^1$ § $sort^2$ § $sort^3$
$opns^{(3)}$ = $opns^1$ § $opns^2$ §$opns^3$

$bool0^{(3)}$ ===
    $opns^{(3)}$

$opns^1$
  T, F : —> bool
$opns^2$
  T, F : —> bool
$opns^3$
  T, F : —> bool
modula conditions of mediation.

bool is not simply a set of values but an ordered mediation of boolean values.

```
bool1 =
    sorts
        bool
    opns
      T, F : —> bool
      non : bool —> bool
      and : bool, bool —> bool

   eqns
      x  bool
      non T = F
      non non x = x
      x and T = x
      x and F = F
```

$bool1^{(3)}_{non}$ ===

```
    sorts
        bool
    opns
      Ti, Fi : —> booli
      noni : bool i —> booli

   eqns
      x  bool
      noni Ti = Fi

      noni noni x = x
```

noni nonj non1 x = nonj noni nonj x,   i,j = 1,2

more explicit

opns
Ti, Fi : —> booli
non1 : bool 1 —> bool1
bool 2 —> bool3
bool 3 —> bool2
non2 : bool 1 —> bool3
bool 2 —> bool2
bool 3 —> bool1

Equations

non1 T1 = F1
non1 T2 = T3
non1 T3 = F2

non1 F1 = T1
non1 F2 = F3
non1 F3 = F2

non2 T1 = T3
non2 T2 = F2
non2 T3 = T2

non2 F1 = T3
non2 F2 = F2
non2 F3 = T2

The operator neg can be composed by an internal negation and a corresponding permutation of the neighboring systems.

non1 $x^{(3)}$ = (negx1, perm(x2, x3)) = (negx1, x3, x2)

non2 $x^{(3)}$ = (perm(x1, x3), neg2) = (x3, negx2, x1)

opns
and : bool, bool —> bool

eqns
x and T = x
x and F = F

Some De Morgan formulas for 3-contextural algebras.

non1(non1 X et et et non1 Y) = X vel et et Y

non2(non2 X et et et non2Y) = X et vel et Y

17.1 Interpretations

$obj1^i$ = ({$false^i$, $true^i$}, T^i = $true^i$, F^i = $false^i$)

Initial semantics for the abstract theory bool0

Junk

Confusion

Final

Coments

"Every initial object in a class defined by an abstract type is said to be isomorphic because it has the same basic internal structure."

"A non-initial interpretation in which both T and F are mapped to true may be written as follows:

obj3 = ({true, false}, T = true, F = true, non = not, et = and)

but every expression now evalueates to true.
It allows a total collaps of the domain space and is an example of a final interpretation oposed to an initial interpretation, which allows no collaps of the domain space."

"Initial and final semantics are the limiting forms of interpretations; anything in between is described as loose semantics."

"We will be only concerned with initial semantics."

Downward, p. 178

Some Super-operators for $bool1^{(3)}$
ID, RED, PERM

ID: $bool0^1$ § $bool0^2$ § $bool0^3$ —> $bool0^1$ § $bool0^2$ § $bool0^3$

RED1: $bool0^1$ § $bool0^2$ § $bool0^3$ —> $bool0^1$ § $bool0^3$ § $bool0^3$

RED2: $bool0^1$ § $bool0^2$ § $bool0^3$ —> $bool0^1$ § $bool0^2$ § $bool0^2$

PERM1: $bool0^1$ § $bool0^2$ § $bool0^3$ —> $bool0^1$ § $bool0^3$ § $bool0^2$

PERM2: $bool0^1$ § $bool0^2$ § $bool0^3$ —> $bool0^3$ § $bool0^2$ § $bool0^1$

The ID, RED, PERM operators can by realized in polycontextural boolean systems by the operations of negation and conjunction.

17.2 Semantics of equality in polylogical systems

Arithmetical vs. logical equality

Equality in data systems and equality in control systems (logics)

```
nateq = nat1 + bool1 +
opns
    eq : nat, nat —> nat
eqns
    x, y  nat
    eq(zero,zero) = true
    eq(zero,suc(x)) = false
    eq(suc(x),zero) = false
    eq(suc(x),suc(y)) = eq(x,y)
```

Equality in local systems

For each system of the complexion under consideraton we have locally the well known characteristics of equality.

Equality of compound systems

System S113
eq(3) (zero, zero)(3) === (true1, true1, true3)

eq1 (zero, zero) = true1
§
eq1 (zero, zero) = true1
§
eq3 (zero, zero) = true3

eq1 (zero, suc(x)) = false1
§
eq1 (zero, suc(x)) = false1
§
eq3 (zero, suc(x)) = false3

eq(suc(x),suc(y)) =1 eq(x,y)
§
eq(suc(x),suc(y)) =1 eq(x,y)
§
eq(suc(x),suc(y)) =3 eq(x,y)

18 Some serious consequences of the superoperator construction

18.1 Splitting or not splitting atomar terms for valuations

Does the transjunctional super-operator BIF exist for $bool0^{(3)}$?

Booleans are atomic therefore they can not be split by the super-operator BIF. But this is true only under the condition that they are ruled by the law of identity. Why should they?

Cloned objects are the same but they are not identical.

How can we replicate systems properly if we cannot replicate their atomic terms? Dissemination as distribution and mediation, or as replication, gives us no answer about the splitting of atomic terms because the distributed systems are identical for themselves, e.g. bool1, bool2 and bool3 are all containing and conserving the classical elements, values and operations in their domain. Modeling "as such", "as other", ...

Normally we know transjunctional operations in polycontextural logics only from binary logical operations, the so called *transjunctions*. Bifurcations are generalizations of this binary concept to super-operators which are ruling poly-contexturality as such and not only internal or local operations. Also super-operators are well know in the polycontextural literature they have never been studied independently from their historical sources. In this sense the question of a generalized form of bifurcation, applicable to all objects of a system, was not focussed. Here I try to develop some ideas and constructions of these generalized bifurcations with the help of the metaphors of cloning, replication and not only by the metaphor of splitting, parallelism, simultaneity. In some sense, we even can say, that bifurcation, simultaneity etc. are cases of cloning. More explicite, these terms seems to form a system of complementarity. There is no bifurcation without replication–and vice versa. A free use of the idea of replication goes beyond the well known transjunctions in polycontextural logics. I can not go deeper into the development and explication of the metaphor „cloning“, but it has to be mentioned, for short, that the very idea of logical cloning and replication as well as of logical bifurcation and simultaneity is based on the kenogrammatic concept of morphograms. Morphograms are structural patterns invariant to logical negations, therefore byond identity and diversity of signs, which means outside the realm of signs.

From a more technical point of view I am abandoning in a further step the basic functional approach of the historical polycontextural logics based on a interpretation of multiple-valued logics. One of the earlier significant steps was to abandon the Cartesian product approach of n-ary functions and their problem of decomposition into two-valued subsystems (theory of place-value systems).

Example (ID, BIF, ID)(opn1, opn2, opn3)
arity-zero operations
ID1 T, F —> bool0^{1} : {T1, F1} simul {T2, F2}
BIF1 T, F —> bool0^{2} .simul. bool0^{1} : {T2, F2}
ID3 T, F —> bool0^{3} : {T3, F3}

unary operations (arity-one)
ID1 bool0^{1} —> bool0^{1}
BIF1 bool0^{2} —> bool0^{2} .simul. bool0^{1} —> bool0^{1}
ID3 bool0^{3} —> bool0^{3}

binary operations
ID1 bool0^{1} x bool0^{1}—> bool0^{1}
BIF1 bool0^{2} x bool0^{2} —> bool0^{2} .simul. bool0^{1}x bool0^{1} —> bool0^{1}
ID3 bool0^{3} x bool0^{3} —> bool0^{3}

Diagramm 25

O_1			O_2			O_3		
M1	M2	M3	M1	M2	M3	M1	M2	M3
1		#	#		#	#	#	1
2	2			2				
	3			3				3
G_{120}			G_{020}			G_{003}		

In a short notation we have:
(1, 2, 3)––>(12§23§13)––>((12%23)§23§13) :: ((12%23%#)§(#%23%#)§(#%#%13))

If we apply the unary operations of negation to this evaluation (mapping) (Belegung) we have to deal with the quite new situation of a reflection of the values of system2 in system1. It seems to be reasonable to accept that a negation of the values in system2 has to be mirrored one-to-one in system1 to be correct. We say that system1 has a model of the behavior of system2 in itself. The model is not the original, it differs in its place in the reflectional system. If system2 changes its state the mirrored model in system1 has to change in exactly the same way. Does this make any sense? Probably it is the most simple case of transjunctional or replicational distributions.

18.2 Bifurcational distribution of negations and junctions

If we start with bifurcation we are forced to distribute all operations in a transjunctional way. This is really a new and intriguing situation. Not only we have to involve constants, negations but also binary junctional operations like conjunction, disjunction into this transjunctional game. And the originary transjunctions of the old place-value system of logic are understood as a very special but quit explicit case of bifurcation.

Does it make any sense to repeat exactly the same logical situation of one system in another logical system which is distinguished from the first only by its different place in the complexion of the whole polycontextural logic?

From the point-view of a theory of argumentation (interaction or communication) this type of modeling corresponds to the situation when an actor is agreeing in all logical points with its partner of communication. The agreeing system has its own position and its own logical arguments but additionally it offers space to the other system to accept its logical arguments. Insofar the modeling has to be strictly one-to-one. To reduce the situation of agreement to the usual case the actor denies its space and accepts the arguments in the space of the other system. „I agree, but keep it for yourself. There is no logical space I can offer you for that.“

From the point of view of the model of reflectional programming this situation of mirroring the logical constellation of the environmental or partner system could be understood as an interpretation and modeling of the so-called „causal relation“ of a reflectional system on a logical, and not on an informational, level.

This is the special case of transjunctional behavior. The general case accepts a different logical behavior, a different sequence of argumentational steps at the locus of the accepting system. This is the real case for logical transjunctions as we know them from polycontextural logic.

In this new context the operation of bifurcation is distributing total functions and not partial functions as it is necessary for transjunctional operations. To distinguish the two concepts, this type of function should be called replicative transjunction or simply replication (of functions).

Mixing different types of logics

We also have to consider the case, that combination of logics has not to be homogeneous, that means, that we are mixing different types of logical systems together. Therefore the bifurcation operation of this different systems produces „inside“ of reflecting system a mapping of components which are from a different type of logic than the reflecting system itself. But this possibility is out of the range of this study, which is mainly introductory.

18.3 Problems of the beginning and the beginning of problems

18.3.1 The beginning as zero

```
nateq = nat1 + bool1 +
opns
    eq : nat, nat —> nat
eqns
    x, y  nat
    eq(zero,zero) = true
    eq(zero,suc(x)) = false
    eq(suc(x),zero) = false
    eq(suc(x),suc(y)) = eq(x,y)
```

Additionally to the successor operation let´s introduce the inverse function of the predecessor *pred*.

pred(suc(x)) = x
suc(pred(x)) = x

As we can see, our natural system introduces zero as an absolute beginning. There are no predessors of zero. And there is no number x with a succession to zero. That is, the function pred(x) is not defined for x = zero.

$0 \neq succ(0)$
More explicit: nonEx(x): suc(x) = 0

The term *zero* seems to be a very priviledged object. It is the beginning of everything, in this sense it is not only a beginning of many other beginnings, but an origin. It is called an initial object. And later we can show that there is one and only one such initial object, all others are strictly isomorphic to it. The whole richness of the pluralities of beginnings is reduced to the general and abstract initial object as the only origin. Plurality is possible only in a secundary sense of applied, that is concrete or even empirical systems.

And this is exactly how it has to be for human beings. There is one and only one beginning–and this beginning is the *origin* of everything.
Ask Aristotle why it has to be this way.

citation
And nobody should think that there has been the slightest change in this mono-contextural archeology since Aristotle.

citation

18.3.2 Are there any neighbors of zero?

All that is in sharp contrast to my construction of a plurality of natural systems. It also violates my principle, that there are many beginnings but no single origin.

Because I accept that locally for each natural system, zero is an initial object and therefore there is no predecessor of it, I have to introduce another wording. Despite the fact that the initial object has no predecessor it is more natural to speak of many neighbors. In other words, zero has predecessors but not in its own system but in its neighbor systems, therefore these predecessors are strictly speaking neighbors, that is neighbored initial objects.

And therefore in the strict sense, but maybe broken down, split, of the meaning of the term „initial“, there is no initial object left at all. Initiality occurs as cloned and dispersed in plurality. If there are many initial objects, the notion of initiality is changed and has lost its init and its unicity, uniqueness, singularity guarantied, before deconstruction, by isomorphism.

$$neighb^{i}(zero^{i}) = zero^{j}$$

In general

$$suc^{j}(neighb^{i}(x^{i}) = suc^{j}(x^{j})$$
$$neighb^{i}(suc^{j}(x^{j}) = suc^{i}(x^{i})$$

For 3 contextures:

$$neighb^{i}(neighb^{i}(x)) = x, \quad i=1,2,3$$
$$neighb^{i}(neighb^{j}(neighb^{i}(x))) = neighb^{j}(neighb^{i}(neighb^{j}(x))), \quad I,j = 1,2$$

For 4 contextures, we have additionally to the cyclical equations a commutative eqation:

$$neighb^{1}(neighb^{3}(x)) = neighb^{3}(neighb^{1}(x))$$

This approach to polycontextural arithmetics is still very static and presupposes that there are something like pre-given arithmetical objects and orders between these objects. It suggests, that the object „zero“ is a stable arithmetical entity.

A more dynamic approach is developed if we remember that we are much more focussed on the operations, and their operationality, than on their objects. In this sense zero is not an object but a function or operation which can be realized independently of a special object. As in the classical approach each object chosen as a beginning is considered as isomorphic to zero. But this is considered on the level of the models of the abstract system of natural numbers or the abstract word algebra. It is a model theoretic consideration and does not belong to the level of the abstract system itself.

In polycontextural system we rencontre a very different situation. Because we have a multitude of abstract systems there are complex possibilities of interaction between these abstract systems without leaving their abstractness for the purpose of modeling.

In the case of the static approach we have only the possibility of reaching the different zeros from a zero in a given system. That is, the zero of a neighboring system is reached as the neighbor of zero in a chosen system. Functions which are not zero do not have a neighbor in another system which is a zero function.

This statical situation is radically changed in a dynamic system. Each function can have its own zero neighbors. Arithmetically speaking each number in one system can change its functionality to a beginning in another system. And each beginning in one system can be an ending in another system. Therefore, there is no absolute beginning needed, and an ending has not to be connoted with attributes like potential or actual or factual or whatever type of infinity nor with the concept of finity. All this Greek heritage will be in the play in a much later step of arithmetical thinking.

Maybe we have to distinguish between the description of an arithmetical process, which leads to the well known formal systems, from Peano to Lorenzen, and the notion, the conceptualisation of an arithmetical system. The description tells us what natural numbers are doing, or better, what happens if we use numbers. This leads automatically to the problem of antropological problems of the limits of usage of numbers or in contrast to the non-antropological, but Platonist concept of usage or existence of numbers.

But there are other ways of thinking, too. It has its occurrence in Hegels *Logic* and its further development in Gunthers *Natural Numbers in Trans-Classic Systems.*

As I have shown before, the idea of proemiality is to inscribe the difference which constitutes all relations and operations as such. Proemiality is the prelude to all operations in formal systems. It is the constitution of all institutions as formal systems. A trans-classical approach to the problems of introducing natural number systems is therefore to apply the proemial relation, that is the strategy of chiasm, onto the arithmetical system. I have to suspend the questioning of the totality of the term *"all"* in these statements.

This leads to a characterization of a dynamical approach written, inscribed, as a chiasm between the four terms: inital, final, successor, predecessor.

This chiasm or proemial relation between initial and final, successor and predecessor, does not need a fixed beginning, it doesn't force us to accept a decisionist beginning or start of the abstract system by a privileged initial element, written as a introductory rule of level zero. On this level, there is also no need to be concerned about infinities of all sorts.

On the other hand, it offers a mechanism for a mediating interplay of cognitive and volitive structures and actions in a formal system.

Further, this chiasm makes it reasonable to speak of obstacles in arithmetical systems without being confused with the problem of the existence or nonexistence of numerical objects which should be and could not be the last number in a series of numbers.

As much as the first number is relative, as much the last number is relative, too. Beginnings and endings are an interacting couple of terms, and not an asymmetrical one, in the sense of one beginning and no end as in classical arithmetics.

After having introduced this idea of a chiastic interplay of the primary terms of the trans-classical concept of a plurality of arithmetics, it is not unnatural to specify a special case of this dynamics to define a very special statical system, the system of natural number series as we know it.

This would be the strategy of reducing the multitude of polycontexturality to a single elementary contexture of mono-contexturality. Another strategy which is not under consideration here would be to understand sign systems in general as "crystallizations" of kenomic events, studied in the system of kenogrammatics.

18.3.3 Deconstructing the origin

It is mentioned in philosophy by Derrida that a sign or a mark has to be able to be repeated, iterated, otherwise it cannot be a sign or a mark. It is the iterability of signs and marks which makes them a mark or a sign. Without going into the highly complex work of deconstructing the origin, the initiality, the initiality of the origin, of zero and other logocentric concepts I like to mention that the whole problematics of the origin is reassembled in the rule or function of introducing, or postulating, zero.

It can be mentioned that the zero can be repeated by the successor function. But this is not exactly the case, what is repeated is the successor function applied on the single and only object zero. This is clear if we reconsider that the suc function is of unary type and the zero is of arity-zero. It also can be mentioned from a semiotical point of view that the sign „zero“ can be repeated as often as we wish and that therefore „zero“ is obviously a sign. But repeatability shouldn't be an abstract concept. Iterability happens in the context of a defined system. And here I am thematizing the concept of an abstract natural object, that is a word algebra with its initial object zero.

In this sense, and not in a arbitrary other sense, it is the case that the fact that zero as the only start of the word algebra has no predecessor, means that this object „zero“ is not repeated, cannot be repeated and isn't allowed to be repeated. Zero is not an event of iterability. Iterability of a sign doesn't mean that the sign has to be iterated, but that the sign has the necessary possibility of being repeated.Obviously, the semiotic status of zero as an initial object in the word algebra, and similar in all other formal systems, is very unclear.

The decision to make a start with zero is not inscribed in the formal system which contains zero. It could be said, that the introduction of zero as the initial object can be understood as a one-step iteration. Iterability can be introducing and repeating. But this is in strict conflict with the concept of an initial object, insofar, as initiality is the start of repetition, and what is the starting point of repetition is not itself, in the same sense, repetition, too.

In this consideration, zero is the initial object of the word algebra and not the number „zero“. We have not to be confused with possible interpretations of the number zero as ontological notions like nothingness, or emptiness or as arithmetical place-holder for numerals in a positional system and so on. Zero here is simply the mark of a start. It could be a stroke in a stroke calculus with the initial object „stroke“ introduced by the arity-zero rule: introduce a stroke! Or, it could by a cross in the calculus of indication: draw a distinction! Mark it!

Nearly all philosophical or meta-mathematical studies are concerned about the problem of infinities of all sorts, later they consider the problem of finity, too. But there are no studies about the strange situation of their initial object, zero. Sometimes zero is thematized as an ontological problem, mixed with notions like nothing, nothingness or even emptiness. But the very character of the initial object to be the initial of the series of the, say, natural numbers, is not worth a reflection. The reason may be very simple, it makes obviously no sense at all to think of the initial, the zero, as a derivate of something else.

And by the way, Aristotle has cleared in his attacks against Pythagore and Plato the scene for ever.

I think that we are in the hypnosis of some quite odd or at least strange dichotomies. The Greeks had been very careful with the problem of the beginning of the series of natural numbers. Aristotle organized a lot of highly intriguing thoughts in defending his initial object, arche. The queer distinction they introduced is the dichotomy of beginning / infinity. Because Aristotle had to fight against mythological circular reasoning, he coulden´t introduce the more harmonic dichotomy of beginning / ending, arche / telos.

They also didn't introduce the dichotomy of zero / ininity as an interpretation of nothing vs. all, simply because the didn't work with the concept of zero.

Surely, the character of the inital object changed in history from the one (1) to the zero, but the pattern of the main questions remained. The questions, from theological, philosophical and mathematical point of view, had always been about the concept of infinity in all its disguises. Bad enough, all the work was done as a family affair between Aristotle and Plato. Some criticized everything which is not finite as bad platonism, the other ones wanted much more than infinities, but trans-finities and even more. And today we are criticizing the concept potential infinity in the name of a more terrestrial and anthropological concept of finity, which should fit much better with the scenario of finitarism in todays computer science.

Even my own hero Aleksander Yessenin-Volpin is more concerned about all his finities than with any deconstruction of the inital objects of his NNNS (Natural Number Notational Series).

?? I think, that my emphasis on the role and status of zero, as an arity-zero term, as I am developing here, is totally new in the discourse of the problematics of natural numbers in any sense. If not, I would be quite happy to hear more...

18.3.4 Brian Rotman´s attack

Although Brian Rotman supports the idea that numbers are made by counting, and are not to be presuposed as platonic entities, an idea which is well known by constructivists (Bishop, Lorenzen), he is not aware about the problem of starting to count. As it is well known, no formal system starts to work, to draw conclusions, to construct numbers, and so on, without a decision of a user to start the system. And this decision has no representation in the system itself. It is not included that a formal system has to realize its one start.

All these approaches are Platonist in the sense that they are primarily cognitivist. They deal only with the cognition of these processes and procedures. All volitive decisions and actions are excluded not only from the formal systems, but more important, from all rationality at all. Earlier on there was a big confusion between psychological, logical and even neurological aspects of logics. After Husserls big attack against psychologism in arithmetics and logics there was a new confusion which lead to the tabu of considering reflectional aspects of formal systems because of the fear of psychologism. Today, there is nearly no limit of confusing every thing with every thing.

From a strict formalist view-point the idea of an interacting mechanism of cognition and volition for formal systems is beyond any rational discourse and is strictly excluded from the academic world.

The other obvious restriction of the arguments of Rotman, and probably all other philosophers of mathematics and the mathematicians themselves, is simply the fact, that he is dealing with the numbers, in what ever onto-semiotical status, as objects. His concerns are the numbers and not the processe, or the processuality of counting in itself. The results of the process of counting are his objects under consideration and not the very process which is producing his objects. All in all, it is some productionism which determines his idea of incorporated numbers. But we have to accept, that this approach is very natural, because we have to use sings as objects and we don´t have any help to write a process as a process, there is no notational system for processes as such. In a process algebra, we are dealing with the names of processes and not with the processuality of the events. Similar in a musical notational systems, the notes are fixed, they tell the musician what to do, which process has to be started. In this sense they are a notational fixation of musical thoughts, in other words, they are signs or markers and ruled by the semiotical concept of identity. More explicite, it has to be mentioned, that the whole adventure of mathematics and philosophy is written in the framework of writing, even in the world view of the book. Maybe it is a principally absurd and totally Sysiphus-type of work to try to do something else than to write.

The strategy of Rotman´s intervention is to cut down the phantasies of Platon and Cantor in reducing the possibilities of repetition.

"One needs to see how and for what reasons one can refuse the idea of perfect repeatability; ..." , p. 54

From the point of view of thematizing the metaphorical arsenal of his argumentation, it is obvious, that he is in the metaphorizity of the steps, the stepwise counting, with and whithout restrictions. In contrast to Yessenin-Volpin, there is nothing to see like metaphors of jumping, switching, leaving...the series of natural numbers for other, neighboring, series of natural numbers.

"Counting , ..., is an activity involving signs. And, as an activity, counting works through–it is–*significant repetition."* p. 6

Who is counting? Is it not, again, a transcentental ego, disguised as a singular empirical anthropological subject, which is counting, all these numbers in his or her solitude? Why not an interacting multitude of counting agents?

As I argued, the start of the whole story, demands for a decisionist introduction of the inital object, zero. It turns out that zero as an arity-zero term is not a sign. It can´t be a sign in the proper sense because it is a non iterable term. This term is postulated, but not repeated in its word algebra. Again, what is repeated, and what can be repeated, is the arity-one term suc, the successor function, which, by its name means what it does. The suc-function is repeating itself applied on the initial term zero. As a result of this repeated application we get a series of repeated zeros. But again, this plurality of zeros is not a plurality of the initiality of the initial object zero, but a result of the application of the second rule, the suc-function and not a repetition of the first rule, the introduction of the initial object zero. The first rule will never be repeated.

Again, I am obviously not speaking about the impossibility of repeating the word "zero", because, as we can see, I am repeating this word "zero" quit a lot in my text. The word "zero" is not the initial object of the word algebra, because the word "zero" is not at all a arity-zero object but part of a very different grammatical system, which may have its own initial object or not. In paranthese I mention that there had been some paleo-linguists which had been searching for the Ur-words.

18.3.5 Deconstructing the origin/Cloning as a mode of iterability

"Il n´y a pas de mot, ni en general de signe, qui ne soit construit par la possibilite de se repeter. Un signe qui ne se repete pas, qui n´est pas deja divise par la repetition dans sa premiere fois *n´est pas un signe."* Derrida

Similar ideas about the repeatability of signs can be found in the semiotics of Peirce.

The beginning, the start, zero, is not repeatable.

It is not allowed to repeat the beginning.

The whole drama starts with the non-repeatability of the initial object.

This lack of repeatability of zero on one side and the characterisation of signs as repeatable objects on the other side shows some weakness in the argumantation of Rotman. This is the beginning of the whole story of thinking, in mathematics as well in philosophy.

"What would a mark be that one could not cite? And whose origin could not be lost on the way?" Derrida, Margins, p. 321

Derrida brought together in his concept of iterability, not only the stream of non-founded events– the *mis en abym*, but also the alterity of the „iter", the „alter". Some Derrida experts seems to be lost in the „Strudel des Denkens" (Heidegger) by the infinity of iterability and are blind for this little jump of contexture as hinted by *„And whose origin could not be lost on the way"* which at least is in conflict with the idea of repeatability and which is not necessarily well understand as *„the far out of sight"* by a *„interminable network"*.

Is it helpful to mention *„iteration alters, something new takes place"* (Derrida, LI, p.175)? in: Gasche, Mirror, p. 215

Citation goes together, like translation, with jumps to different contextures, all organized by their own and different rules and different origins. More technically, this procedure of citation is possible only in connection with the interplay of an identity and a neighbor function of a mark.

Only conservative translations and citations are covered by category theoretical morphisms. That is, translations which are not loosing their origin in the process of translation.

On the other hand, translations as jumps may be wild, out of hands, beyond rules, but not arbitrary. They are „ruled" by the „procedures" of chiastic change.

The problematics are getting more virulent. Behind this generalized idea of iterability and repeatability which is based on a generalized concept of signs and marks, there is something like the strict non-iterability of the non-signs and non-marks, the kenograms.

Singularity?

In misleading words, more Kantian terms, we can state, that the pure possibility of iterability itself is not iterable. Which doesn't mean at all, that this primordial possibility is unique.

Or even harder: In misleading words, more Kantian terms, we can state, that the pure necessity of the possibility of iteration itself is not iterable. Which doesn't mean at all, that this primordial necessity has to be unique.

Singularities?

Iterability of morphograms?

What´s about the non-iterability of morphograms? What could the term „non" mean in a field which is not ruled by negations at all? Is there a a-iterability of morphograms?

Can we repeat or cite or mention a morphogram?

May be, morphograms have the cruel characteristics of not being able to change their definition. Moving morphograms in a kenogrammatical system never changes the morphogram as a morphogram; it remains irrestible to change. To move could mean, to change the context of a morphogram, that is, to put it together with other morphograms, etc. First of all, morphograms are not involved in the game of meaning or significance. They don't mean anything.

How could a morphogram be changed if it is exactly defined as a pattern for all changes of itself as a morphogram?

Morphograms are structurally invariant under negation.

Morphograms can be changed into other morphograms. The reflector operation is a very simple operator to change a morphogram into its inverse morphogram. But that is not a case of iterability in the strict sense.

Only death does not change.

Is there an „iter" for morphograms as there is necessarily an iter for signs to be signs?

To cite a morphogram means to give it a name and to use this name. But this is a change of systems, from the morphogrammatic to the linguistic systems. And not a citation of a morphogram in a morphogrammatic system.

A word, a sentence, a text is part of the game of repeatability. Morphograms may be the invariant and non-significant structure or pattern of all this variations.

We are reaching here a domain or field of pre-semiotics or something like the *archi-ecriture*, *archi-trace* and its *differance*, as is was exposed by the „early" Derrida.

18.4 Where is the problem?

Counting robots

Why should I make such a fuss about natural numbers? Everyone knows how to use them, everybody is counting all the time, our computers are counting permanently, even some dogs are counting…

We have learned all that, and we have taught all that. We are all fit to use them.

But do we understand natural numbers? And why should we understand them? Is it not enough to know how to use them? And are our mathematical considerations not good enough to understand the concept of natural numbers?

As I tried to show, our understanding and our formalization of natural numbers is based on a very deep intuition and an insight in the very nature of numbers.

But how could I presuppose that my robot or my extra-terrestrial visitor has the same deep and well-founded intuitions? How could my robot even have any intuitions?

And surely, nobody has ever seriously asked a human child if it really wants to learn all this stuff.

Obviously, if we want, or have to, construct a robot, that is, an artificial system, which is able to use numbers, we have to be able to teach this system from scratch everything which is needed to understand and to use numbers.

Counting the counter

To say, numbers are produced by the process of counting, sounds at first quite good. Counting counts numbers. This pragmatical or constructivist opinion is surely helpful and less magical than the platonist one. But things are not as simple as we are taught they are. Who counts? This could open up an endless enquiry. And it happened and happens somewhere else. The question can be formulated much simpler. If the counter is counting, is then the counter not a product of his own counting? Is the counter himself not as well a product of counting as the numbers he counts? Who is first, the counter or the counted?

Why should we terrorize our understanding of the natural numbers by the simple structure of our language? Or grammar?

An intrinsic chiasm of counting

Goodsteins numerals

To overcome all these circularities, Goodstein has introduced another strategy. We should not deal with numbers, but with *numerals*. Numerals are elements of a formal game, called arithmetics. Numbers are the interpretations of the moves of numerals in a formal system. The formal system is defined by the rules of arithmetics.

But where are they from? We are back at the beginning of our discussion, the stroke calculus.

18.5 The natural between/beyond algebras and co-algebras

After having involved ourselves in some philosophical argumentations about the deconstruction of the origin and its ends, it is time, to ask the matematicians if there are not already some more a precise methods developed or under development to undestand this higly ambiguous and subversive scenario in a more operative and less suggestive setting.

It is, or would be, an extremely interesting enterprise, to compare and correlate or even to interweave this two movements of thinking. Regrettable, I have to say, this work is only at the very beginning. Neither from the side of the matematicians nor from the side of the philosophers or deconstructivists there is any sign that they have started to learn from each other. Surely, there are, as usual, some, extremely few, exceptions. Maybe Kent Palmer, ...

18.5.1 new tools

„New mathematical tools are needed to model stream-based computation, because inductive methods of definition and reasoning only work in domains of finite objects. The chief new notions are coinduction, coalgebras, and non-well-founded sets.

Inductive definitions provide three conditions:

(1) initiality,

(2) iteration, and

(3) minimality. (...)

While induction formalizes the metaphor of constructing finite structures from primitives, coinduction formalizes the observation metaphor of stream-based environments. Coinductive definitions eliminate the initiality condition of induction, and replace the minimality condition by a maximality condition. (...)

*Coinduction provides a mathematical framework for formalizing systems that interact with the external world though infinite interaction sequences. In addition to greatest fixpoints, the semantics of coinduction assumes lazy evaluation; the tokens of the stream are observed one at a time, rather than all at once. Hence, coinductive definitions permit us to consider the space of all processes as a well-defined set, even if the input streams are generated dynamically and cannot be predicted a priori.“ (*Dina Goldin und David Keil)

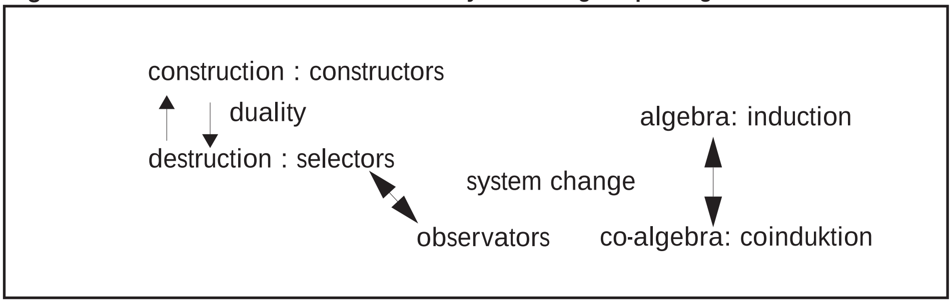
Algebra	Co-Algebra
induction initial constructor total algebra	co-induction final object destructor partial functions coalgebra
visible structure well founded Turing Machine Horn clauses	hidden behavior non well founded sets Persistent TM liveness axioms

18.5.2 Duality and some Subversions

„Elements of the General Theory of Coalgebras“ hingewiesen.

„But the theory is not just a simple minded dual to universal algebra. Structures such as e.g. bisimulations, *that don't have a classical counterpart in universal algebra, but that are well known from computer science, figure prominently in the new theory."* Peter Gumm

Diagramm 26 **Between duality and change of paradigm**



18.6 The Peters swinging tantra

The ultimate tantra about the interplay of algebraic and co-algebraic approaches in general is written by Peter Padawitz, the inventor of the swinging types.

„Initial structures are good for modelling constructor-based data types because they fit the intuition about these types and admit resolution- and rewrite-oriented inductive theorem proving. The corresponding specification and verification methods do not comply so well with non-free or permutative types such as sets, bags and maps and are still less appropriate when infinite structures like streams or processes come into play.

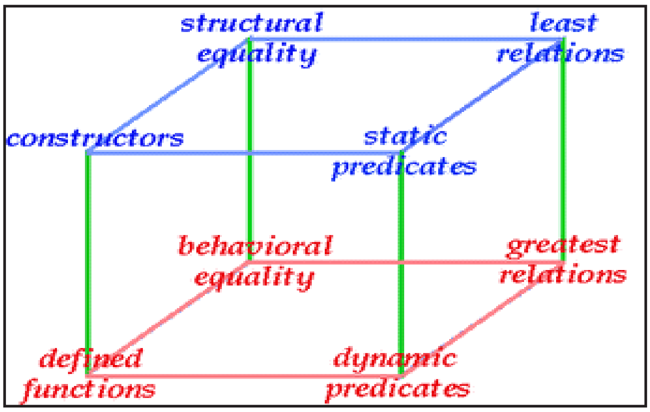
Non-free and infinite structure are better modelled as dynamic objects, which are identified through reactions upon actions (methods, messages, state transitions) rather than through constructors they might be built of. Extensional, contextual, behavioural, observational or bisimilarity relations model object equality and the suitable domains are final structures that are conservative with respect to visible subtypes.

Consequently, a collection of data types and programs should be designed hierarchically as a "swinging" chain of specifications each of which extends its predecessor by either constructor types or action types.

Constructor types introduce the visible domains and come with inductively defined totalfunctions, structural equality and safety predicates with Horn clause axioms, while action types provide the hidden domains together with coinductively defined partial functions, behavioural equality and liveness predicates with liveness axioms that are dual to Horn clauses. A swinging specification is interpreted as a sequence of initial and final models. General proof rules capture this semantics and exploit the duality of induction and coinduction to its outmost extent.

The deductive tractability is further enhanced by making both constructor and action types amenable to rewrite oriented proof methods so that we can reason about swinging specifications in the same way we are used to reason about exclusively constructor-based types."

Swinging Types provide a specification and verification formalism for designing software in terms of many-sorted logic. Current formalisms, be they set- or order-theoretic, algebraic or coalgebraic, rule- or net-based, handle either static system components (in terms of functions or relations) or dynamic ones (in terms of transition systems) and either structural or behavioral aspects, while swinging types combine equational, Horn and modal logic for the purpose of applying computation and proof rules from all three logics.



A swinging specification separates from each other visible sorts that denote domains of data identified by their structure; hidden sorts that denote domains of data identified by their behavior in response to observers; -predicates (least relations) representing inductive(ly provable) properties of a system; and -predicates (greatest relations) representing complementary ``coinductive'' properties, which often describe behavioral aspects ``in the infinity''.

A model that combines static with dynamic features and structural with behavioral aspects of a system is obtained naturally if all involved entities (objects, states, etc.) are presented as terms built up of constructors for visible or hidden sorts and if functions are specified by conditional equations (= functional programs), least relations by Horn clauses (= logic programs or transition system specifications) and greatest relations by co-Horn clauses. Term equivalences are either structural or behavioral, the former being least, the latter being greatest solutions of particular axioms derived from the type's signature.

http://issan.cs.uni-dortmund.de/~peter/Swinging.html

19 Computational Ontology and the Problem of Identity

„Already Heraclitus pointed out that the notion of identity is not completly clear. But mathematicans prefer to proceed as if Heraclitus had not lived. I cannot continue in this way, this situation when an infinite process can be imbedded in an finite object is anordinary one in investigations of distinct natural number series, and I shall need an apparatus for the explicit consideration of all identifications used in such cases.“ A. Yessenin-Volpin

"Real-world computer systems involve extraordinarily complex issues of identity. Often, objects that for some purposes are best treated as unitary, single, or "one", are for other purposes better distinguished, treated as several.
Thus we have one program; but many copies. One procedure; many call sites. One call site; many executions. One product; many versions. One Web site; multiple servers. One url; several documents (also: several urls; one Web site). One file; several replicated copies (maybe synchronized). One function; several algorithms; myriad implementations. One variable; different values over time (as well as multiple variables; the same value). One login name; several users. And so on.
Dealing with such ***identity questions*** *is a recalcitrant issue that comes up in every corner of computing, from such relatively simple cases as Lisp's distinction between eq and equal to the (in general) undecidable question of whether two procedures compute the same function.*
The aim of the ***Computational Ontology project*** *is to focus on identity as a technical problem in its own right, and to develop a calculus of generalized object identity, one in which identity -- the question of whether two entities are the same or different -- is taken to be a* ***dynamic*** *and contextual matter of* ***perspective****, rather than a static or permanent fact about intrinsic structure.“* Brian Cantwell Smith

„By the way, what is static and what is dynamic may be in the eye of the beholder. ‘We suggest…that many grammatical frameworks are static formalizations of intuitively dynamic ideas´,..“ Yuri Gurevich

„Current OO notations make no distinction between intra-application variability, for example, variability of objects over time and the use of different variants of an object at different locations in an application, and variability between applications, that is, variability across different applications for different users and usage contexts."
K. Czarnecki, U. W. Eisenecker, Generative Programming

19.1 Identity

19.2 Equality

19.3 Bisimulation

„By identifying two states with same external behavior, we get an extensional notion of equality, that can be captured by the following axiom:

Axiom 2.4. Two states are considered equal if they cannot be distinguished by (a combination of) observations.

To a user, again, the state may remain hidden, it is irrelevant, as long as the automaton implements the desired regular expression. Again, two states may be identified, if they behave the same way on the same input, which is to say, if they cannot be distinguished by any observation."

I am refering here to the great book *Modal Logic* (Blackburn et al.).

Bisimulation – the Basic Case

We first give the definition for the basic modal language.

Let M = (W, R, V) and M´= (W´, R‘, V‘) be two models.

A non-empty binary relation Z WxW‘is called bisimulation between M and M‘if the following conditions are satisfied:

(i) If wZw‘then w and w‘satisfify the same letters.

(ii) If wZw‘and Rwv, then there exists v‘(in M‘) such that vZv‘and R‘w‘v´ (the *forth condition*).

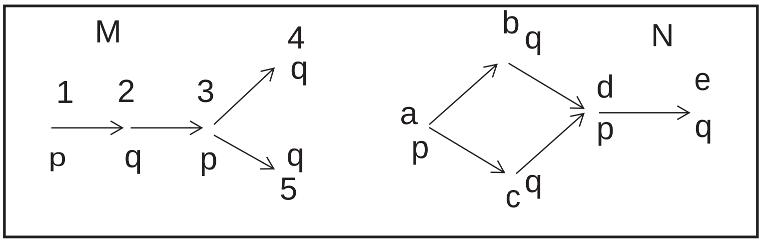
(iii) The converse of (ii): if wZw´and R‘w‘v‘. then there exists v (in M) such that vZv‘and Rwv (the *back condition*).

Example:

The two models M und N are bisimilar.

Z = {(1,a), (2,b), (2,c), (3,d), (4,e), (5,e)}

Diagramm 27 **Bisimilar Models**



To show the bisimilarity of M and N, we define the relation Z. Condition (i) of our definition is satisfied: Z-related states make the same prpositional letters true. Moreover, the back and forth conditions are satisfied too: any move in M can be matched by a similar move in N, and conversely.

The two models are showing the same behavior in respect to the relation Z, therefore they are bisimilar.

"Quite simply, a bisimulation is a relation between two models in which related states have identical atomic information and matching possibilities."

"Examples of bisimulations (...) disjoint unions, generated submodels, isomorphisms, and bounded morphisms, are all bisimulations."

Bisimulation, Locality, and Computation

"Evaluating a modal formula amounts to running an automaton: we place it at some state inside a structure and let it search for information. The automaton is only permitted to explore by making transitions to neighboring states; that is, it works locally.

Suppose such an automaton is standing at a state w in a model M, and we pick it up and place it at state w´in a different model M´; would it notice the switch? If w and w´are bisimilar, no. Our atomaton cares only about the information at the *current state and the information accessible by making a transition – it is indifferent to everything else. (…)*

When are two LTS (Labelled Transition Systems) computationally equivalent? More precisely, if we ignore practical issues (…) when can two different LTSs be treated as freely exchangeable (óbservationally equivalent´) black boxes? One natural answer is: when they are bisimilar.

Bisimulation turns out to be a very natural notion of equivalence for both mathematical and computational investigations." p. 68

Morphograms and Bisimulation

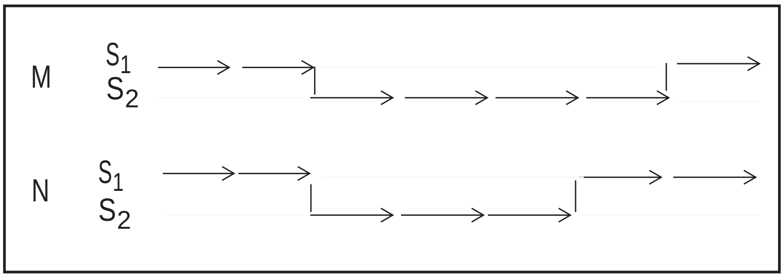
We can now apply the idea of Bisimulation directly to our study of the behavior of morphograms in kenogrammatical systems.

For example, lets interprete morphogram MG = (aabcbcbaa) as Trito-Number TZ = (00121211).

Das Verhalten dieser Trito-Zahl ist jedoch nur über ihre Aktionen in beobachtbaren Systemen bzw. Kontexten zugänglich und diese seien hier ihre binären Komponenten.

Die Trito-Zahl TZ zeigt zwei Verhaltensweisen, die sich in zwei Modellen des Verlaufs der Binärsysteme darstellen lassen.

M = ($S_{1122221}$) und N = ($S_{1122211}$). M und N unterscheiden sich an der zweitletzten Stelle bzgl. S_2 und S_1. Die Knoten bzw. states der Modelle werden als die Belegungen des Morphograms durch Zahlen, d.h. der Trito-Zahl interpretiert. Die Zahlen als states haben einen Index, der angibt zu welchem Subsystem S_1 oder S_2 sie gehören bzw. den Übergang (Sprung) markieren.



Da das Morphogramm MG als solches nicht direkt zugänglich ist, dafür jedoch die zwei Modelle des Verhaltens des Morphograms, lässt sich aus der Bisimulation der zwei Modelle M und N auf die Struktur des Morphogramms schliessen. D.h. die Bisimulation zwischen M und N erzeugt eine Äquivalenz bzgl. des Verhaltens bzw. den Manifestationen des Morphogramms.

In dieser Thematisierung erscheint ein Morphogramm als die Klasse aller seiner bisimilaren Modelle. Nach der Terminologie von *hidden* und *visible algebras*, sind die beobachtbaren Verhaltensweisen des Morphogramms *visible*, und die dahinterliegende Struktur *hidden*.

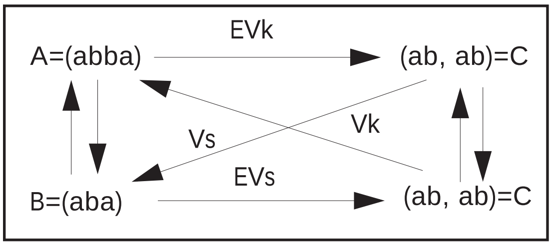
Die zwei Trito-Zahlen TZ_1= (001212) mit der Subsystemfolge S_{11222} und TZ_2 = (001012) mit der Subsystemfolge S_{11112} sind nicht bisimilar, da die Wertung des 4. Zustandes in TZ_1 und in TZ_2 mit "2" bzw. "0" differieren.

19.4 Kenogrammatic decomposition and bisimulation

"Wenn sie in zwei gleiche Teile zerlegt werden können…" heisst, wenn ihre Verhaltenspattern sich nicht unterscheiden lassen, sind sie gleich. D.h., die Idee der Dekomposition eines Morphogramms in gleiche Monomorphien durch Abstraktion über verschiedenen Dekonstruktoren lässt sich als Bisimulation verstehen.

Es wird hier ein spezieller Zusammenhang zwischen der Struktur des Morphogramms und seines Verhaltens bei einer Dekomposition hergestellt.

Diagramm 28 **EVk * Vs = EVs * Vk**



Different questions (EVk, EVs), equal answers (ab, ab)

19.5 Sameness in PCL-Systems

Identity vs. diversity.

Equality vs. sameness vs. non-equality (?)

Sameness as the basic category of polycontextural systems.

Gleichheit (Heidegger) likeness

Diagramm 29 **Modell von Selbigkeit-Gleichheit-Verschiedenheit**

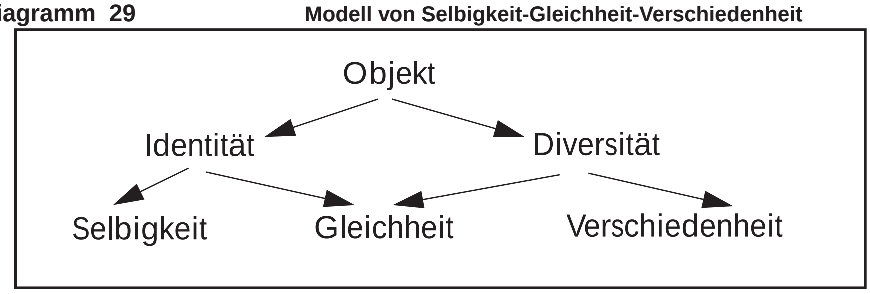


Diagramm 30 Identitäts-/Diversitäts-Relationen der Proto-Struktur

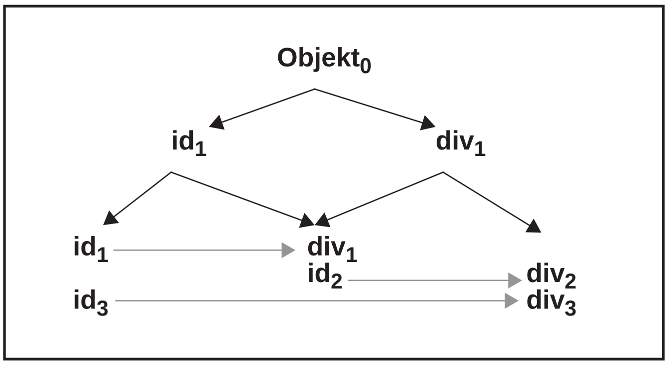
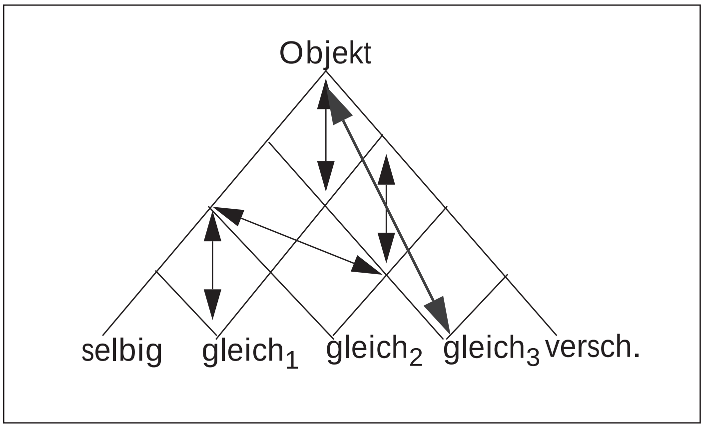


Diagramm 31 Unterschiede in der Gleichheit



20 Cloning the Ur-Logik

Not only in arithmetics with its natural numbers we are involved in the naturality of our basic terms of thinking but also in logic itself. All ways of thinking are founded in a single way of thinking, reducing all possible multitudes to a strict unizity, the famous Ur-Logik. In contrast to arithmetics logic has unfolded itself in a wide range of different logical systems. But there are good reasons to accept the strategy to reduce this plurality back to the singularity of the Ur-Logik.

combinatory Logic as introduced by Moses Schönfinkel (1924 in Moscow) and independently later by Haskell Curry (1930) was often called Ur-Logik, Schönfinkel's main operator was therefore U. And it was believed that with this Ur-Logik the very fundamental problems of the foundations of logic and mathematics could be solved for ever.

Today combinatory logic plays an important role in the construction of functional programming languages like ML, Haskell, Miranda. Combinatory logic is also crucial for the definition and exploitation of parallelism of functional programs.

As the great logician and magician Raymond Smullyan pointed out in his famous book *To Mock a Mocking Bird* (1985), combinators are programs and everything a program can do can be done by combinators. This is an enormous statement if we contrast the radical simplicity of combinatory logic with the complexity of programs. Obviously, combinatory logic must have a special power. This power is based in its abstractness which is surpassing our normal attitude to logic and which is not fearing logical paradoxes. Combinatory logic is not founded in ordinary language and perception (Anschauung) but in formal and formalist thinking and scriptural construction.

And again, ...

As Natural as 0,1,2

Philip Wadler. Evans and Sutherland Distinguished Lecture, University of Utah, 20 November 2002.

"Whether a visitor comes from another place, another planet, or another plane of being we can be sure that he, she, or it will count just as we do: though their symbols vary, the numbers are universal*. The history of logic and computing suggests a programming language that is equally* natural*. The language, called* lambda calculus*, is in exact correspondence with a formulation of the laws of reason, called* natural deduction*. Lambda calculus and natural deduction were devised, independently of each other, around 1930, just before the development of the first stored program computer. Yet the correspondence between them was not recognized until decades later, and not published until 1980. Today, languages based on lambda calculus have a few thousand users. Tomorrow, reliable use of the Internet may depend on languages with logical foundations. "*

http://homepages.inf.ed.ac.uk/wadler/topics/history.html#drdobbs

20.1 Disseminations of Combinatory Logics

Following E. Engler, (1983) combinatory logic as a proof theoretical system is build by *language*, *axioms* and *rules*. Combinatory logic is a system, or more exactly an algebra A = (A, *, S, K), therefore we can disseminate this algebra in a way we have introduced before by means of the proemial relationship.

The algebra A consists of the formulas A, the application "*", and the combinators "S" and "K". The *terms* of the language are build by atomic terms, with variables x, y, z...and the *constants* S and K and the binary *application* "*" of the combinatory algebra. *Formulas* are equations between terms. The universe of combinatory logic is single-sorted or untyped. That is, in this term algebra there is no distinction between functions and their arguments–except that * interprets the first expression as a function and the second as the function´s argument (R.W. Stark)

Axioms are
t = t for atomic terms
Sxyz = (xz) (yz)
Kxy = x

The deduction rules are the rules of equality.
t1 = t2 ==> t1t3 = t2t3

t1 = t2 ==> t3t1 = t3t1

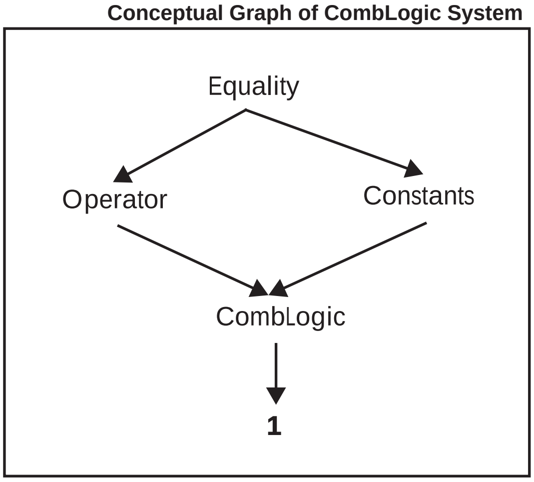
t1 = t2 ==> t2 = t1

t1 = t2, t2 = t3 ==> t1 = t3

Provability

Again, it maybe helpful to put the skeleton of combinatory logic together in a conceptual graph which visualize the conceptual dependencies of the main notions of the system of combinatory logic.

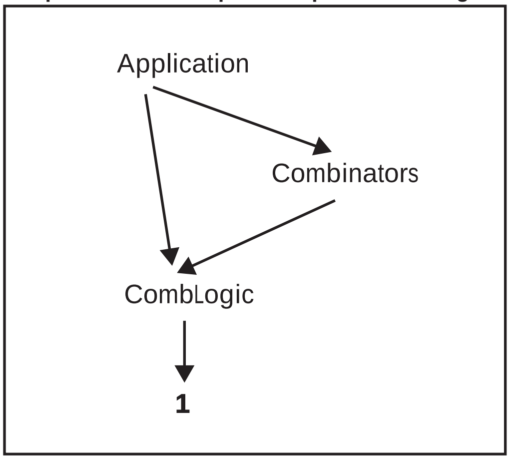
Diagramm 32 **Conceptual Graph of CombLogic System**



Equality: equations are formulas, Operator: application *, Constants: S and K.

The operational conceptual graph deals with the dependencies of operator and operand of combinatory operations (formulas).

Diagramm 33 **Operational Conceptual Graph of CombLogic**

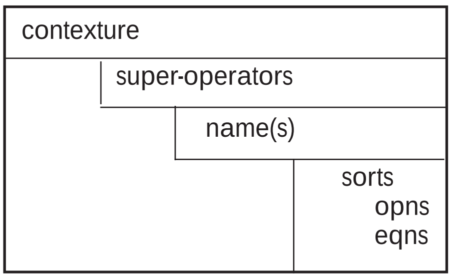


Combinators figures always in two different roles, one as operators and the other as operands. The whole term then is therefore the result of the application of an operator to its operands, that is an operation which is an application.

As a meta-theoretical result, the consistency of the combinatory system is provable.
There exists one and only one combinatory logic, thus Combinatory Logic is founded in the unizity of 1. The consisteny of the system gives a strong argument for its unicity.

As an abstract algebra combinatory logic is build up by the well known tectonics of "name(s)", "sorts", "operations", "equations". The basic module of the abstract algebra in a polycontextural setting includes "contexture", "super-operators". Therefore, the combinatory tectonics is reduced to the special type of: *contexture* is 1, *super-operator* is ID, and *name* is Combinatory Logic, *sorts* are one, the entities E, with constants K, S, I and *opn* is *, and *eqns* are "=".

Diagramm 34 **Basic Modul**

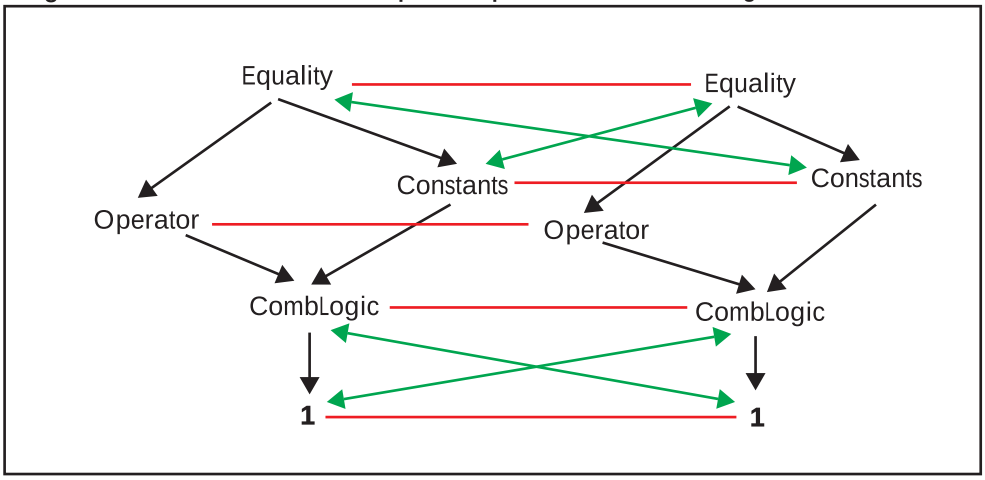


Because combinatory logic is an algebra A = (A, *, S, K), we can disseminate this algebra in a way we have introduced before by means of the proemial relationship.

The new polycontextural algebra of combinatory systems consists of the distributed original algebras and a set of super-operators {ID, PERM, RED, BIF} between these algebras.

By a conservative dissemination of combinatory algebras I understand a dissemination which is preserving the type-structure of the algebras, that is, conservative dissemination produces a strict parallelism of the combinatory systems. A conservative dissemination is a very special case of the general unrestricted dissemination of combinatory algebras including all possible transformations between different types.

Diagramm 35 **Conceptual Graph of combined CombLogics**



Meta-theoretical considerations about the disseminated combinatory logics like consistency etc. are not touched by the process of cloning. Each cloned or colored system inherits all the meta-theoretical merits of its original system. Even if the original maybe lost, or not easy to distinguish from its clones.

20.1.1 Toward combinatory poly-logics

After having convinced myself about the naturality of multiple beginnings I feel free to start our Ur-Logik simultaneously at different ontological places at once.

To give a introductory idea about a possible poly-contextural combinatory logic derived from the proemial distribution of classical combinatory logic I propose some constructions.

Poly-contexturality means that there are entities, atoms, objects belonging to different universes, all mediated together to build a complexity of combinatory logics. Classical combinatory logic which is obviously mono-contextural has only one global domain, its universe, the domain of PC-based combinatory logic is not a uni-verse, but a pluri-verse (or multi-verse).

A = (A, *, S, K)
A = (A, *, S, K)
A = (A, *, S, K)

In the mono-contextural setting we are only considered with elements belonging to their universe. Elements not belonging to U don't exist, therefore they have not to be studied, at least not at the beginning of the construction of the system. All elements which belong to their universe, and only one universe exists anyway, are elements which are identical with themselves, which means they exists, because elements which are not identical with themselves don't exist. This is surely a heavy-weight ontological assumption.

Existence is expressed by identity, or equality. Therefore the rules of equality of elements and combinations of elements has to be considered.

Reflexivity

Transitivty

This situation has to be transformed into the poly-contextural constructions.

Because in a poly-contextural situation there are several universes the question arises to which universe an element belongs and to which universes it doesn't belong.

The situation is similar to the question of belonging to a specific type or not. But here we are dealing with the very basic fundaments of the calculus, and not with a meta-level construction.

It is true, or it is the case that x belongs to the universe U_1 means x $=_1$ x, to shorten this wording we can write T_1x^1, short T_1x.

A(3) —> A(3)

If someone has a problem with the numbering of the different Ur-Logics and thinks I am reducing the whole construction back to the linearity of our unique natural numbers, don´t worry. First, which number system do you mean? Here we are offering more than a handful of different possibilities to chose one of the many natural number systems.

But equally I would say, yes, you are right! I am using numbers to number these Ur-Logics, but again, if I mention or thematize these numbers, they are invited to their own contexture and respected as members of their own arithmetic system, different to the counted number systems before. Therefore, there is no privileged number system. What is first and what is second in a temporary hierarchy is based only on its functionality, on its role in the complex game and not by any kind of primordial substantiality may it be as formal as possible.

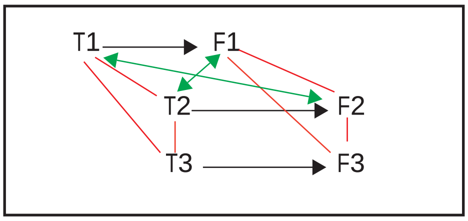
Wordings

What is not an atomic element of U1 is as such an atomic element of U2, what is not an atomic element of U2 is also not an atomic element of U3, and what is an atomic element of U1 is also an atomic element of U3. All that at once, not in the sense of set theory but in the sense of mediated atomic elements.

$x^{(3)} = (x^1, x^2, x^3)$, more explicit because it is not a tupel but a mediation and again visualized in the diagram below.

$x^{(3)} = (x^1 § \; x^2 § x^3)$

Signatures of $x^{(3)}$
$T_{1,3}$ x
§
$F_{1,2}$ x or F_1, T_2 x
§
F$_{2,3}$ x or $F_{2,3}$ x



If we want to be interested in the elements which don't belong to the universe2 and the universe3 we simply have to introduce a new universe, say universe4 to positively localize these elements, which are not existent in the former constellation. Non-existence means, existence in another universe.

Therefore, no element is lost, each has a place to be. There is no *Heimatlosigkeit* in polycontextural systems but also no *letztendlicher* origin.

This allows to introduce in combinatory polylogic a special kind of negation, the systemic negation (pemutation).

$x^{(3)} = (x^1, x^2, x^3)$

$n1(x^1, x^2, x^3) === (x^1, x^3, x^2)$

$n2(x^1, x^2, x^3) === (x^3, x^2, x^1)$

Combinatory logic as a calculus has also its linguistic rules. Not every possible combination of the atomic elements are allowed, also there are rules about technical signs like brackets. Therefore a calculus consists of expressions which can be well-formed or ill-formed, but it is dealing only with the well-formed expressions. As usual, from all the possible concatenations the well-formed have to be cut out. And further, not all well-formed expressions are theorems, they have to be cut out by the axioms and rules of the calculus. What we see is the tectonics, a hierarchically ordered system of linguistic levels.

Complex atomic terms

Atomic terms in poly-logic are complex. They are composed of local terms.
(x, x, x) .... (x, #, #)

Distribution of the Axioms of Combinatory Logic

Axioms

Operator-Axioms
Sxyz = xz(yz)
Kxy = x

General Axiom for atomic terms

Equality x = x

Distribution of axiom of identity of atomic terms:

Axa1. xi = xi for all i of s(m), that is (x1 = x1 § x2 = x2 §....§ xn = xn)

This is fine for separated parallel systems. Because the whole system is a mediation of different systems we have to analyse also the rules and axioms of non-equality.

x1 = non x1 is obviously in contradiction to the axiom Axa1.

The situation for complex equations is more intriguing

x1 = non? x2 or x1 non? = x2 or non?(x1 = x2)

To deal with these situations it is useful to use the introduced system-signatures.

We have to specify which negation is involved in the denial of the axiom. Complementary we have to specify the type of identification of the terms.

x = x means T1x = T1x, that is, the atoms x belong to the system, they are not from somewhere else. To start numeration, the system and its signature has the identifying number 1. In other words, x = x, means x is identifyied as x, therefore it is identical. In a mono-contextural setting nothing else then identification leading to identity and non-identification leading to deversity can happen.

Queer, or othogonal equality

But what is x1 = x2 ?

In a poly-contextural situation the process of identification is complex from the start. It enables to identify x as x1 or x2...or xn. The distinction between xi and xj is not diversity (non-identity) but sameness in contrast to difference and equality.

For ni (xi = xi) we can speak of intra-negation in contrast to inter-negation with nl(xi = xj).

S1: x = x
S2: x = x
S3: x = x

Inter-negation jumps from one system to another, this type of negation is involved with a permutation or transversion of the system index of the negated element.

x1 = x2 means more explicite:
S1: x = #
S2: # = x

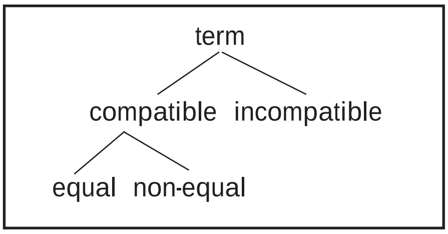
S3: x = x

(x, #, x) is not equal (#, x, x)

T1x = n? (T2x)

Global and lokal equality
Are two complex terms globally equal if all its aspects are locally equal?

The main distinction for atomic terms is compatiblity/incompatibility. And only the compatible terms can be equal or non-equal.



For compatible terms equality and non-eqality is ruled by negations. Incompatibility of terms is ruled by their distance between the systems they belong. This systemic distance is ruled by orthogonal permutations. Negation can be considered as the case of nil distance, the negational permutation is not leaving the system it belongs.

Between symmetry and asymmetry

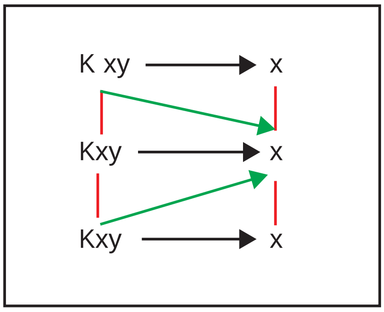
t1 = t2 ==> t2 = t1

Two ways of modeling: dissemination and fibering

Additionally to the approach of disseminating systems by means of proemiality, distribution and mediation, we can model this procedure in the category framework of logical fibering. This approach is well known by the work of Pfalzgraf (1988 –). Maybe it is possible and helpful to make the distinction that the disseminatory approach is corresponding more to a proto-theoretical thematization whereas the category theoretical approach reflects more a meta-theoretical point of view.

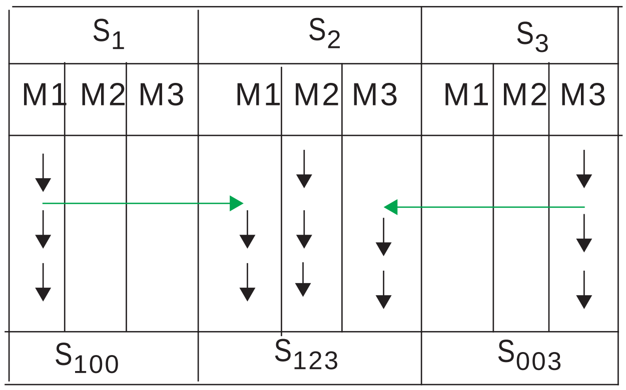
(BIF, ID, BIF) ((K1, K2, K3) (x1y1, x2y2, x3y3)) ==>

((x1, #, #), (x1, x2, x3), (#, #, x3))



Not reduction but overlapping. At the locus S_2 , with its own object x2, we have an overlapping of the objects of system1 and system3, therefore (x1, x2, x3).

Diagramm 36 **Transcontextural transitions of (BIF, ID, BIF)**



Logical transjunctions and togetherness

In logical semantic tranjunctions as we know them togetherness is organized by splitting the function into its partial parts. These parts are (more or less) disjunct resp. to their values. Therefore this type of togetherness is not depending on the metaphor of cloning but on the metaphor of splitting (of total function into its partial functions) too.

But even for logical transjunctions this emphasis of splitting in contrast to cloning and “gluing” is not the whole truth. Also transjunctions are highly over-determined, their truth-values are belonging simultaneously to different logical sub-systems

(T_1, F_1) trans==> ($\mathbf{F}_{2,3}$, $\mathbf{F}_{2,3}$). In this case **F** belongs at once to two sub-systems,

S_2 and S_3. A similar over-determination is produced for the values which are involved in the conditions of mediation of the sub-systems.

Strategies of Extensions

As we know, combinatory logic is not only consistent but complete, it is a sound system. To add any new combinators runs–from purely fomalistic point of view–into a dilemma. If the system should keep its consistency, a new combinator would not be of great novelty, because it could be defined with the existing combinators S and K. If this extension would introduce into combinatory logic something irreducible new it would destroy the soundness of the system.

If in parallel programming, operators like *par* and *seq* are introduced, they are in the dilemma of being reducible if they don't wont to destroy the basic system of combinatory logic. Are the parallel combinators strict or non-strict?

> Parallelism is introduced in GPH by the *par* combinator, which takes two arguments that are to be evaluated in parallel. The expression p `par` e (here we use Haskell's infix operator notation) has the same value as e, and is not strict in its first argument, i.e. bottom `par` e has the value of e. (bottom denotes a non-terminating or failing computation.)
> Its dynamic behaviour is to indicate that p could be evaluated by a new parallel thread, with the parent thread continuing evaluation of e.
> We say that p has been sparked, and a thread may subsequently be created to evaluate it if a processor becomes idle. Since the thread is not necessarily created, p is similar to a lazy future [MKH91].

Nevertheless, it is highly reasonable to introduce by definition new combinators especially for economic reasons. It would be nearly impossible, at least for a human logician, to deal with only the minimal set of operators, like K and S.

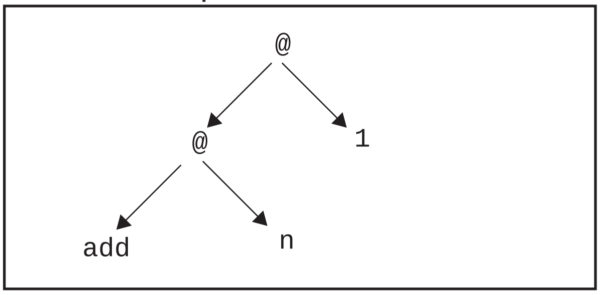
Proemiality is concerned with the cloning of the natural as it appears in arithmetics or in logics. Therefore, questions of efficiency don't matter on this level of thematization. Thus, it seems that there is no chance of changing or extending fundamental systems then by keeping their soundness and disseminating the system as such. In this sense, cloning or disseminating seems to be the only chance of escaping the narrowness of the pre-given naturality.

Obviously this approach of disseminating the naturality of the classical system is not involved in any manoeuvres of the heterodoxy of deviant or alternative or wild or non-orthodox strategies of what ever color. Again, this doesn't exclude that contextures can be involved in all sorts of non-classical systems and even in the distribution of different non-classical systems in the polycontextural framework. Combining different classical and non-classical systems are a new possiblity of PCL-systems.

20.1.2 Disseminating the successor function

As an example we are disseminating the graph of the successor function SUCC as defined by `succ n = n + 1` over two loci. The function SUCC is defined by add and the tags @ are representing the application.

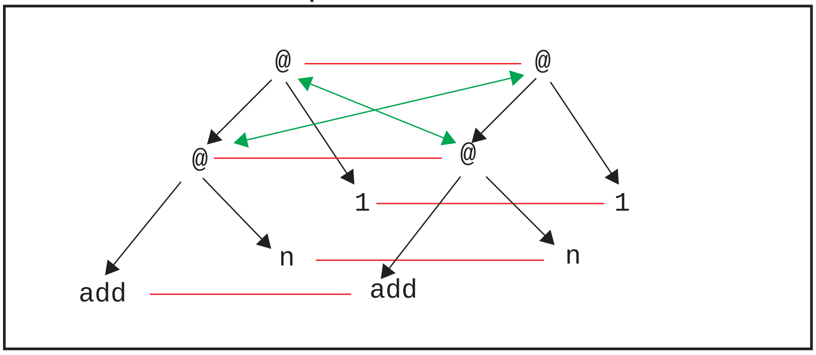
Diagramm 37 **Graph der Funktion succ**



Because the distribution is monoform, that is type-preserving, this distribution is conservative and shows a certain parallelism of the involved terms.

$\text{succ}^{(2)}$ with $\texttt{succ}^1$ `n = n + 1 .simul.` $\texttt{succ}^2$ `n = n + 1`

Diagramm 38 **Graph of two distributed SUCC-Functions**



Some trans-contextural dissemination of the successor function

(BIF, ID, BIF) (succ, succ, succ) (n, n, n) ==>

(succ n, #, #) § (succ n, succ n, succ n) § (#, #, succ n)

20.2 Two faces of the self-applicability of Y

A polycontextural explication of the self-application of the Y-operators has to distribute its application over two places, where the exchange between operator and operand is realized. With a cut in the very concept of identity the Y-operator is simply dealing with its two possible states as an operator and as a operand. These two states have to be realized simultaneously and this can be done by a dissemination over the two simultaneously existing systems.

A strict formalist point of view would exclude the distinction of Y as an operator and as an operand and argue that Y is a combinator and is applied on itself and nothing else.This PCL-construction doesn't exclude self-application in the mode of identity. This can happen freely in each contextural system locally.

The freed face of Y

If chiasm is a primary structure of pcl-systems, and the Y combinator can be understood in PCL as a chiastic operator, why shouldn't we take Y as a primary, not derived operator?

Self-referentiality in combinatory logic is secondary, in the theory of living systems self-referentiality is of primary character. This insight was the reason for the different formal approaches to self-refernetiality and circularity, especially by Francisco Varela to formalize the autopoietic and autonomous structure of living systems byond the classical cybernetic approaches of feedback loops and self-organiszation..

In PCL, circularity is interpreted as a chiasm. The slogan I introduced at this time of "Circulus Creativus" (v. Foerster) was *"Not all circles goes round."* ("Nicht alle Kreise gehen rund.")

Operators like S, K, I are not self-referential, but hetero-referential. They refer to there terms and not to themselves even if the terms consists of themselves. They will occur in a finite chain of superpositions referring to there operands. Self-applicability and superposition of operators is not the same as self-referentiality of operators.

The SKI-system is not only complete it also allows to define operators like Y without producing any inconsistencies inside the combinatory system.

On this probably most formal level we can study the underlying identitive formalist ontology of combinatory logic, Curry´s general theory of OBS.

Also there is a strict hierarchy between the notion of operator and operand for the formulas and applications of SKI, the Y operators forces to abstract from this distinction, because Y is at once considered as operator and as operand. But from a graphemic point of view this is an abstraction, denying the actual inscriptions of the calculus. Again, it is said that Y applies to itself. The presupposition is that Y is as an operator identical to itself. This may be true in a purely formalistic context which denies the behavioral aspect of the construction: Y as an operator and Y as an operand, roles be played by Y in the game of self-application.

To deny this difference means to realize an abstraction from it and this is a mental procedure. That is, the abstraction is not inscribed but thought by the logician. Maybe here is one of the reasons why combinatory logic was called idealistic by the materialist dialecticians of the former Soviet Union.

Finally, a PCL understanding of the mechanism of the Y operator can offer two graphematical realizations of it. One is the chiastic modelling of the functional change between Y as an operator and Y as an operand distributed over two logical loci, that is over two contextures. And second, the inscription of the abstraction from this chiastic mechanism in a third contexture mediated by the two first contextures which corresponds the purely formalist approach of Haskell Curry.

The PCL understanding of the Y operator doesn't deny the typefree construction of combinatory logic.

For historical reasons I mention this:

> *We can no longer "explain" a paradox by running away from it; we must stand and look it in the eye. Something is gained by the mere bringing about of this state of affairs. The paradoxes are forced, so to speak, into the open, where we can subject them to analysis. This analysis must explain the fact that F(F) does not belong to the category of propositions, an explanation which comes within the province of combinatory logic as here conceived.* Curry, Feys, 1956/67

(Similar wording by Heinz von Foerster, Howe and Varela)

WHY, the new turn?

The new turn would be to define the hetero-referential combinatory S, K, I by means of chiastic self-referential operators, like "Why". It would be possible to distinguish between derived self-referentiality, Y defined by combinators, and fundamental or architectonic self-referentiality, not reducible to hetero-referential combinators.

In other words, the super-operator (ID, PERM, RED, BIF) which are applied on the combinatory operators S, K, I for reasons of dissemination, have to be involved into the construction of a PCL-based combinatory logic.

links

http://www-fp.dcs.st-and.ac.uk/~kh/papers/pasco94/pasco94.html
http://burks.brighton.ac.uk/burks/pcinfo/progdocs/plbook/function.htm
http://www.di.unipi.it/ricerca/Report2002/node64.html
http://nfocentrale.net/miser/readings/
http://www.andrew.cmu.edu/~cebrown/notes/barendregt.html
http://www.haskell.org/humor/
http://www.angelfire.com/tx4/cus/combinator/birds.html

1 Pragmatics of cloned natural systems

After having produced a picture of the intuition of proemiality and polycontexturality of natural number systems, the obvious questions arises, what can we do with all that? and especially, what can we do with all that what we cannot do with the classical approach?

It is more than crystal clear, that everything would be changed if we would have been able to introduce, in a convincing way, the slightest change in the very concept of formal systems, say, logic and arithmetics. Logic and arithmetics have not to be confused with the big business of all sorts of logical and arithmetical systems or the immense multitude of formal systems based on the very concept of logic and arithmetics. (Whatever this exactly means.)

1.1 Relativization of Inductive Definitions (David Isles)

Turing Machines

First of all we should remember that the concept of a Turing Machine is a paper-and-pencil concept. More a program, than a physical machine. Its purpose was purely mathematical, that is to give a formal explanation of the intuition of the notion of algorithm in mathematics, especially in number theory.

This opens up the possibility of questioning Turing´s explication in the context of new mathematical intuitions and their own explications.
First of all it is also about *Gedankenexperimente* and not about computer science or technology.

On the other side, the today reality of computation is far beyond of what is conceived by Turing Machines. Instead of algorithms, one of the new metaphors and challenges seems to be *interactivity*, in all its forms.

Therefore, it is possible to start a more deconstructing reading of the concept of Turing machines and to introduce step by step a new type of machines, the *polylogic machines*, without being forced to enter a debate engulfed by the orthodoxy / heterodoxy dramaturgy of academic referees. Nothing is wrong with the classical concepts. Neither with the known extensions, like o-machines, etc., of Turing himself and others. And nevertheless there is no reason to not to try another approach, surely not to the exactly same challenges, but strongly related to each other and interwoven in some family resemblance(similarity, likeness).

Andrea Lynn Stein
Interactivity

David Isles Argumatation (1980)

"As a final example, let us consider the changes effected when one uses different NNN´s (Natural Number Notation Systems) in place of the intuitive natural numbers in a standard argument from recursion theory. In what follows a <—– n means that Turing machine a is given input n. Recall the standard

Theorem (unsolvability of the halting problem

Let T be the class of Turing machine programs and /a/ be the Gödel number of a T. There is no „test" Turing machine b T such that b <—– /a/ halts in state

Y if a <— /a/ halts
N if <— /a/ halts doesn't halt.

Proof
If there were, define the contradictory machine
b* = b v {<YSRY>/S any tape symbol of b} #

In this argument the intuitive natural numbers are used in at least three distinct constructions.
1) in the inductive definition of the class of Turing machine programs. Here a given inductive definition will have a length and we may speak of l(a), the shortest length of the Turing machine program a;
2) the class of inputs to the Turing machines; and
3) to measure the length of Turing machine computations (this is implicit in the words "halts" and "doesen´t halt").

Now whatever may be our preconceptions, there is nothing in this argument that requires the use of the "same" natural numbers in all three constructions.
Indeed all that is required is that if a T, then /a/ should be defined, that is, should be available as an inout. Hence it is consistent with the structure of the argument to suppose that we have three different NNN´s, N1, N2 and N3 and that for a prticular stage k we consider the class of Turing machine programs $T(N^k_1)$ (where a $T(N^k_1)$ means l(a) (N^k_1), the class of inputs N^k_2 and relativize the notion of "halting" to "halting as measured in N^k_3.

Theorem

The point of this example is to suggest that the peculiarly "absolute" character of a result such as the unsolvability of the halting problem may be chimerical and have its origin in certain unrecognized assumptions (the uniqueness of the natural numbers)."

in: F. Richman (Ed), Proc. New Mexico 1980, LNM873, 1981, p 133

Paleonymy of the wording

It is of great importance, to keep exactly the traditional wording in the process of deconstruction. Therefore it is pointed to "in this argument" and not in another agument. Often people change the wording and proof the classical result wrong. Deconstruction has nothing to do with this attitude of "*Besserwisser*". The classical argument is in no sense wrong; contrarily, it is correct in all steps.

The difference is in the decision of the preconditions. If we accept them, then everything is correct. If we don't accept them, e.g. the uniqueness of the natural numbers, then trivially our results will differ correspondingly.

On the other side we have the enormous problem to introduce the new and different concept, here the anti-traditional concept of NNNS. If it fails, the whole argument was only a *Gedankenexperiment* in the sense, suppose we have a well founded theory of NNNS then look what happens with our heavy weight theorems. And this (hypothetical) argumentation and its constructions is of enormous importance at least to learn how to over come classical limits of thinking.

Unfortunately, Yessenin Volpins introduction of his NNNS has not convinced many of his colleagues. This is, by the way, one of the reasons of my own research, based on Gunthers concept of kenogrammatics, in this field. May be my own attempt brings the whole idea and intuition some steps further to realization.

1.2 Polylogic Graph Reduction Principle

1.3 Computable Metamorphosis

2 Programming languages in the context of proemiality

2.1 polyLISP

Datatype: poly-lists, 4-pointer objects
Control type: poly-logical and poly-arithmetical operators

Agents and Ambiguity

"Understanding natural language also requires inferring hidden state, namely, the intention of the speaker. When we hear, *"John saw the diamond through the window and coveted it,"* we know "it" refers to the diamond and not the window—we reason, perhaps unconsciously, with our knowledge of relative value. Similarly, when we hear, *"John threw the brick through the window and broke it,"* we know "it" refers to the window.
Reasoning allows us to cope with the virtually infinite variety of utterances using a finite store of common sense knowledge. Problem-solving agents have difficulty with this kind of ambiguity because their representation of contingency problems is inherently exponential." Stuart Russell

A mechanical system doesn't know the intention of the speaker. Therefore it has to analyze the sentence and to chose in parallel all grammatically possible interpretations, also it has to go on in parallel with the two interpretations until there is new knowledge, from the past or from the new experiences, which enables a decision, which interpretation of the sentences should be preferred in the actual context, or situation. But the old interpretation will still be a possible choice for the case that the narratives turns back to a new context in which this interpretation will have its own significance and will prevail.

It is also possible, especially in esthetic texts, that both interpretations are of equal importance, and that there is an ambivalence played by the game of interchange between background and foreground positions of the interpretations. Maybe there is at this point a connection to Selmer Bringsjords project of artificial joke making programs.

All these maneuvers are possible only in a real parallel and grammatically or semantically multi-layered system. Probably the best candidate for this job, again is the poly-contextural logic.

From a technical point of view of poly-contextural systems there is no reason to think that the complexity of dealing with ambiguity has to grow exponential.

Is there a method in the poly-contextural approach to reduce complexity from the exponential to the linear type?

Concurrency in Communicating Object Petri Nets

Rüdiger Valk
Fachbereich Informatik, Universität Hamburg
Vogt-Kölln-Str. 30, D-22527 Hamburg
email: valk@informatik.uni-hamburg.de

From a Petri net view objects appear in the form of tokens. During the last decade tokens have been considered as more and more complex data objects. In this paper we continue our previous work [Valk 87a] by adding dynamical aspects to such tokenobjects. To integrate this approach into the systematics of Petri net modelling, it is quite natural to consider dynamical tokens as Petri nets themselves.

Before giving an introduction to the formalism used and an overview on the structure of the paper we motivate the approach by some less formal examples. In the first example there are three tasks A, B and C to be processed on three machines M1, M2 and M3 (fig. 1.1). There are limited resources for the machines of the following kind.

Machines M1 and M2 are operated by an operator O1. He can only operate one of the machines M1, or M2 at a given time. The same holds with operator O2 with respect to M1 and M3. Machine M1 can work, in mutual exclusion, only in a mode 1 with O1 or in a mode 2 with O2. Each of the tasks is divided in two subtasks, e.g. a1 and a2 in the case of A. The subtasks have to be executed by particular machines, as specified on the left-hand of fig. 1.1. In the case of task A the second subtask a2 must be executed on M2 after the execution of a1 on M1. We take an „object-oriented“ approach in the sense that the task is to be modelled as an object that enters machine M1 and leaves it after execution to be then transferred to machine M2. Attached with the object there is an „execution plan“ specifying the machines to be used and the order for doing so.

Also the current „status“ of the execution is noted in the execution plan. This situation is formalized by the Petri nets of fig. 1.2., where the two modes of machine M1 are modelled by two different transitions M11 and M12. Mutual exclusion is obtained by the places p2, O1 and O2. Initially, all three tasks A, B and C are in the place p1 (in the net on the right-hand side). By the „object -oriented“ approach they are not represented by an unstructured token, but by their entire execution plan, as given on the left-hand side, also as Petri nets. Note that by the marking in the nets A, B and C also their „status“ description is given.

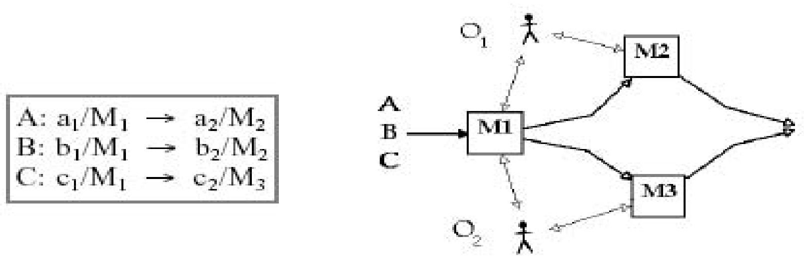


Fig. 1.1: Three machines with 3 tasks

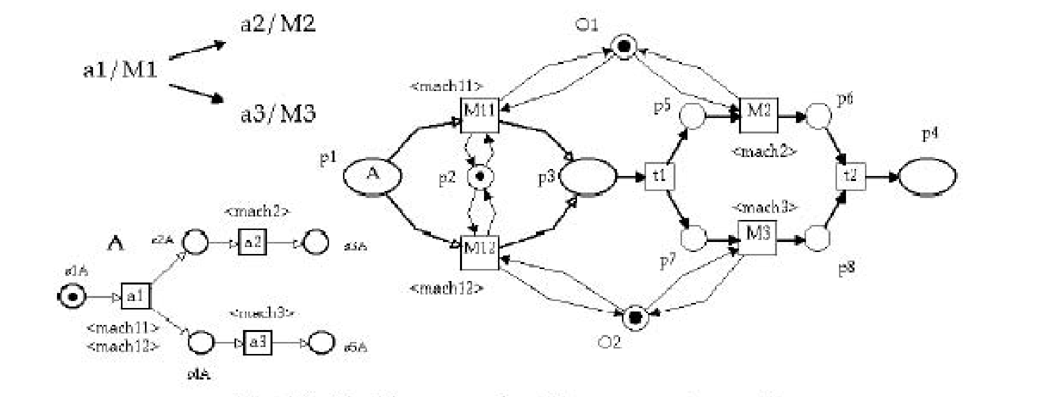


Fig. 1.4: Machine example with concurrent execution

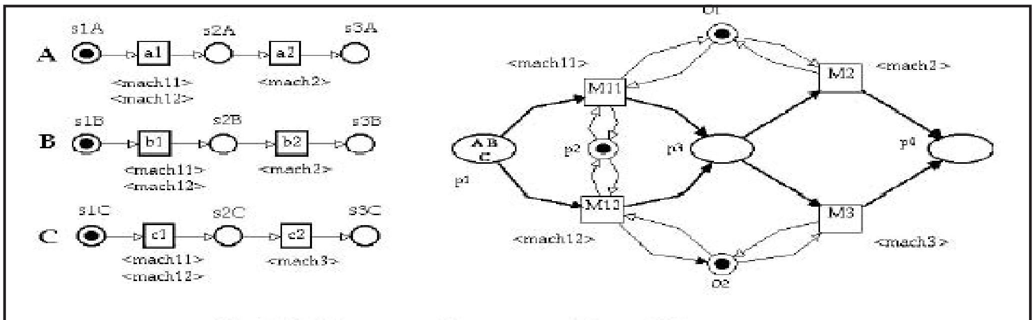


Fig. 1.2: Three machines example as object system

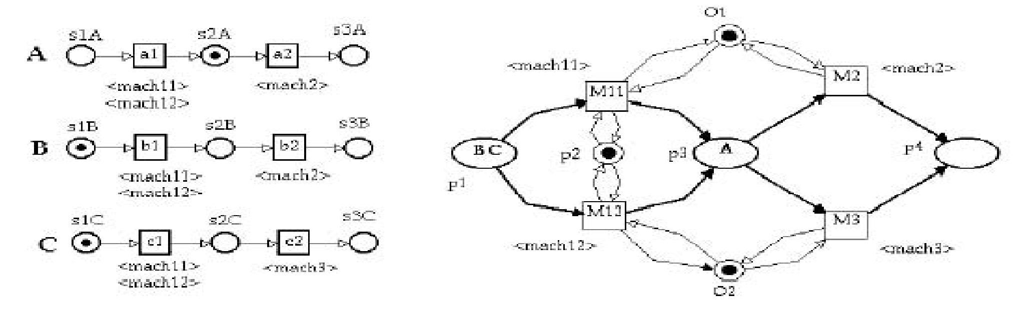


Fig. 1.3: Follower marking of the object system from fig. 1.2

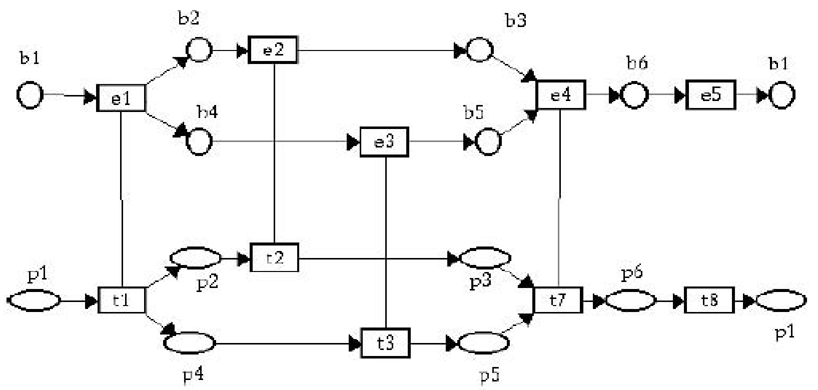


Fig. 3.3: Cooperating process representation of a subprocess from fig. 3.2

3 Internal vs. external descriptions of interactions

The above example gives us an excellent description of interaction between two systems observed *externally* by an observer which does not belong to the systems under consideration. The external observer is epistemologicaly neutral and does not take part for one or the other position.

The relation of interaction between the system net and the object net is simply a mapping between two sets. To analyze the structure of this mapping further it has only zoomed into the set of properties of the relation.

Does there exist an interaction? Is the question by the external observer. Where does the interaction take place, is the question of the internal observer.

To understand my own approach more properly it may be helpful to characterize it as an *internal description* realized from the standpoints of the agents of the interaction system. That is the reason why the agents have in themselves a place to locate the modelling of the neighbor systems and are not simply exchanging informations.

An internal description is a *second-order* concept in the sense of Secons-order Cybernetics, it marks the process of observing observing systems, that is, of decribing systems which themselves are observing, each other, and are not simply existing as objects for the observation of an external observer.

Interpretations: Chiasm of memory and processor

Kenogrammatics is not only neutral to the distinction of number and notion as Gunther pointed out, it is also neutral to the distinction of program and data, and on a hardware level it is neutral to the distinction of processor and memory. Kenogrammatics gives space for an interlocking mechanism between say, memory and processor. In other words, in trans-computation, which is close to the theory of living systems, the living tissue, to be a processor or a memory is purely functional and is ruled by the as-category.

The new scene of AI: Cognitive Systems?

Some brand new trends, similarities and complementaries to my work, even filiations, and other connections are listed.

Results of *Concept Mining Strategies* based on limited facilities of private internet research.

Nevertheless I hope it will give some orientation about new important trends and hints what to read

1 Some Gurus

1.1 Marvin Minsky

Push Singh

1.2 Aaron Sloman

THE COGNITION AND AFFECT PROJECT
http://www.cs.bham.ac.uk/research/cogaff/0-INDEX.html
http://www.cs.bham.ac.uk/research/cogaff/phd-theses.html

Keywords for google:
cogaff tertiary emotions
cogaff meta-management
cogaff diagrams
cogaff vision architecture
cogaff artificial intelligence toolkit

Catriona Kennedy

My PhD thesis topic is ``Distributed Reflective Architectures for Anomaly Detection and Autonomous Recovery''. Some technical reports are available in the Cognition and Affect Directory.

The aim of the research is to explore architectures which allow an autonomous system to detect and recover from anomalies without user intervention. An anomaly is any event that deviates from the model-predicted state of the world and may also occur in the system's own software or hardware. This means that the system must have a model of its own operation (reflection).

I am exploring forms of distributed reflection using a multi-agent network, where each agent may specialise in a particular aspect of the system's operation. The network is not intended as a team of cooperating agents but instead as a decentralised control system for a single autonomous agent (a "multi-agent agent"). The idea is inspired by various branches of philosophy and biology, in particular by autopoiesis theory , immune system models and Minsky's Society of Mind concept.

1.3 John Laird

Interactive game research

http://ai.eecs.umich.edu/people/laird/index.html

Research Activities:
My primary research interests are in the nature of the architecture underlying artificial and natural intelligence. Since 1981, my work has centered on the development and use of Soar, a general cognitive architecture. Over the years, this has led to research in both AI and cognitive science. Within AI my work has included research in general problem solving, the genesis of the weak methods, the origins of subgoals, general learning mechanism, interacting with external environments, learning by experience and by instruction, and integrating reactivity, planning, and learning, all in the service of constructing complete autonomous intelligent agents. Within cognitive science, my research has concentrated on detailed modeling of human behavior (reaction times and error rates) in visual attention, concept acquisition, and dual tasks. More recently, my research is concentrating on creating human-level AI agents for interactive computer games.

I am a founder of Soar Technology.

www.soartech.com

http://ai.eecs.umich.edu/people/laird/gamesresearch.html

1.4 Peter Wegner

www.cs.brown.edu/people/pw.

Interaction Machines

Lectures 1998 in Hamburg

http://swt-www.informatik.uni-hamburg.de/Lehre/ss98/PeterWegner.html

Why Interaction Is More Powerful Than Algorithms, Communications of the ACM., May 1997
Interactive Software Technology, Handbook of Computer Science and Engineering, CRC Press, 1996.
Frameworks for Compound Active Documents, Work in Progress, May. '97
Interactive Foundations of Computing, Final Draft, Theoretical Computer Science, February 1998
A Research Agenda for Interactive Computing, Work in Progress, Jan. '98
On the Expressive Power of Interactive Observers, Work in Progress, Jan. '98
Observability and Interactive Computation, Work in Progress , Jan. '98

Models of Interaction

"Interaction as a Conceptual framework for Object-Oriented Programming"

Abstract

The paradigm shift from algorithms to interaction captures the technology shift from mainframes to workstations and networks, from number crunching to embedded systems and graphical user interfaces, and from procedure-oriented to object-based, agent-oriented, and distributed programming. The radical notion that interactive systems are more powerful problem-solving engines than algorithms is the basis for a new paradigm for computing systems built around the unifying concept of interaction. This talk will extend Turing machines to interaction machines, show that interaction machines are more powerful than Turing machines and show that interaction machines are more natural as a model for objects, agents, design patterns and applications programming. More information can be found in an article in the May 1997 Communications of the ACM on "Why Interaction is More Powerful than Algorithms".

Observability and Empirical Computation

"Interactive Foundations of Computing Observability and Empirical Computation"

Abstract

Interaction machines, which extend Turing machines with input and output actions, are shown to be more expressive than Turing machines, both by a direct proof and by adapting Godel's proof of irreducibility of mathematics to logic. Observational expressiveness, defined by distinguishability of system behavior, provides a common metric for comparing the expressiveness of algorithms and interactive systems that also expresses the explanatory power of physical theories. The change in metric from algorithmic transformation to interactive observation captures the essence of empirical computer science. Observation in physics corresponds to interaction in models of computation. The relation between observers and the systems they observe is examined for both computation and physics. The evolution in computing from algorithms to interaction parallels that in physics from rationalism to empiricism.

Interactive extensions of Plato's cave metaphor and the Turing test confirm that interactive thinking is more expressive than logical reasoning. Turing test machines with hidden interfaces express interactive thinking and collaborative behavior richer than the traditional Turing test. Interactive (nonmonotonic) extensions of logic such as the closed-world assumption are exp

lored. Procedure call, atomicity of transactions, and taking a fixed point are techniques for closing open systems. Pragmatics is introduced as a framework for extending logical models with a fixed syntax and semantics to multiple-interface models that support collaboration among clients sharing common resources. Interaction machines with an associated model of observability provide a precise framework for empirical computer science and a basis for modeling physics, interactive extensions of logic and the Turing test, and practical application programs.
More information can be found in the May 1997 Communications of the ACM in an article "Why Interaction is More Powerful than Algorithms", in the February 1998 Theoretical Computer Science in an article "Foundations of Interactive Computing".

Interactive Software Technology

"Towards Empirical Computer Science"
Abstract
Part I presents a model of interactive computation and a metric for expressiveness, part II relates interactive models of computation to physics, and part III considers empirical models from a philosophical perspective.

Interaction machines, which extend Turing machines to interaction, are shown in Part I to be more expressive than Turing machines by a direct proof, by adapting Godel's incompleteness result, and by observability metrics. Observation equivalence provides a tool for measuring expressiveness according to which interactive systems are more expressive than algorithms. Refinement of function equivalence by observation of outer interactive behavior and inner computation steps are examined. The change of focus from algorithms specified by computable functions to interaction specified by observation equivalence captures the essence of empirical computer science.

Part II relates interaction in models of computation to observation in the natural sciences. Explanatory power in physics is specified by the same observability metric as expressiveness in interactive systems. Realist models of inner structure are characterized by induction, abduction, and Occam's razor. Interactive realism extends the hidden-variable model of Einstein to hidden interfaces that provide extra degrees of freedom to formulate hypotheses with testable predictions conforming with quantum theory. Greater expressiveness of collaborative computational observers (writers) than single observers implies that hidden-interface models are more expressive than hidden-variable models. By providing a common foundation for empirical computational and physical models we can use precise results about computational models to establish properties of physical models.

Part III shows that the evolution in computing from algorithms to interaction parallels that in physics from rationalism to empiricism. Plato's cave metaphor is interactively extended from Platonic rationalism to empiricism. The Turing test is extended to TMs with hidden interfaces that express interactive thinking richer than the traditional Turing test. Interactive (nonmonotonic) extensions of logic such as the closed-world assumption suggest that interactiveness is incompatible with monotonic logical inference. Procedure call, atomicity of transactions, and taking a fixed point are techniques for closing open systems similar to 'preparation' followed by 'observation' of a physical system. Pragmatics is introduced as a framework for extending logical models with a fixed syntax and semantics to multiple-interface models that support collaboration among clients sharing common resources. 19.06.98

http://swt-www.informatik.uni-hamburg.de/Lehre/ss98/PeterWegner.html

1.5 Joseph Goguen

Category theory and computation
Algebra vs. Coalgebra

http://www-cse.ucsd.edu/users/goguen/zoo/

Peter Padawitz

Swining Types
http://issan.cs.uni-dortmund.de/~peter/Swinging.html

Peter Gumm

Universal Coalgebra
http://www.mathematik.uni-marburg.de/~gumm/Papers/publ.html

1.6 Dov Gabbay

Combining logics
Fibring, labelled deductive systems

Jochen Pfalzgraf

http://www.cosy.sbg.ac.at/~jpfalz/literature.html

Polycontextural logics and fibred categories

CLC - FibLog project.

FCT Project POCTI/2001/MAT/37239 (January 1, 2002 - December 31, 2004)

A project of CLC on fibring and other constructions for combining logics with applications to the development of mixed logics, such as logics of hybrid systems and logics of authentication.

http://www.cs.math.ist.utl.pt/cs/clc/fibring.html

Fibring Logics: FAQ

A topic of research at CLC - FibLog project.

Why combine logics?

Besides leading to very interesting applications whenever it is necessary to work with different logics at the same time, combination of logics is also of interest on purely theoretical grounds.
The practical significance of the problem is clear, at least from the point of view of those working in knowledge representation (within artificial intelligence) and in formal specification and verification (within software engineering). Indeed, in these fields, the need for working with several formalisms at the same time is the rule rather than the exception. For instance, in a knowledge representation problem it may be necessary to work with both temporal and deontic aspects. And in a software specification problem it may be necessary to

work with both equational and temporal specifications.
From a theoretical point of view, one might be tempted, for instance, to look at predicate temporal logic as resulting from the combination of first order logic and propositional temporal logic. But the approach will be significant if and only if general preservation results are available about the mechanism used for combining the logics. For example, if it has been established that completeness is preserved by the combination mechanism @ and it is known that logic L is given by L'@L'', then the completeness of the combination L follows from the completeness of L' and L''. No wonder that much theoretical effort has been dedicated to establishing preservation results and/or finding preservation counterexamples.

Why fibring?

Among the different techniques for combining logics, fibring, as originally proposed by Dov Gabbay, deserves close study because of its generality and power. Fibring explains many other combination mechanisms as special cases and it is powerful enough for the envisaged applications.

What is fibring?

But what is fibring? The answer can be given in a few paragraphs for the special case of logics with a propositional base, that is, with propositional variables and connectives of arbitrary arity.
The language of the fibring is obtained by the free use of the language constructors (atomic symbols and connectives) from the given logics. So, for example, when fibring a temporal logic and a deontic logic, mixed formulae such as

((G a) -> (O(Fb)))

appear in the resulting logic. Naturally, in many cases we want to share some of the symbols. The previous example involves the constrained form of fibring imposed by sharing the common propositional part.
At the deductive system level, provided that the two given logics are endowed with deductive systems of the same type, the deductive system of the fibring will be obtained by the free use of the inference rules from both. This approach will be of interest only if the two given deductive systems are schematic in the sense that their inference rules are open for application to formulae with foreign symbols (from the other logic). For instance, when we define MP by the inference rule

{(x -> y), x} |- y

in some Hilbert system, we may implicitly assume that the instantiation of the metavariables x, y by any formulae, possibly with symbols from both logics, is allowed when applying MP in the fibring.

The semantics of fibring is more complicated and, therefore, it is better to consider the special case where both logics have semantics with similar models. A possible, quite general, model for many logics with a propositional base is provided by a triple

< U, B, v >

where U is a set (of points, worlds, states, whatever), B is a subset of the parts of U, and v(c) is a map from B^n to B for each language constructor c of arity n. In what follows we consider only logics with a semantics given by a class of such triples.
Given two such logics L', L'' what should be the semantics of their fibring? It will be a class

of models of the form above, such that, at each point u of U, it is possible to extract a model from L' and one from L''. If symbols are shared, the two extracted models should of course agree on them.
In order to visualize the semantics of fibring, consider the fibring of a propositional linear temporal logic with a propositional linear space logic. Each model of the fibring will be a cloud of points where at each point we know the time line and the space line crossing there. For instance, at the point

< Lisbon, 10h15m 28 January 2002>

we know the time line (of past, present and future) of Lisbon and the space line (the universe taken as a line for the sake of the example) at 10h15m 28 January 2002.

Where to start learning more about fibring?
P. Blackburn and M. de Rijke. Why combine logics? Studia Logica, 59(1):5-27, 1997.
D. Gabbay. Fibred semantics and the weaving of logics: part 1. Journal of Symbolic Logic, 61(4):1057-1120, 1996.
D. Gabbay. Fibring logics. Oxford University Press, 1999.
A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorial construction. Journal of Logic and Computation, 9(2):149-179, 1999.
A. Zanardo, A. Sernadas, and C. Sernadas. Fibring: Completeness preservation. Journal of Symbolic Logic, 66(1):414-439, 2001.

Last update: January 29, 2002.

Publications of the project FibLog
http://www.cs.math.ist.utl.pt/cgi-bin/uncgi/bib2html.tcl?proj=fiblog

1.7 Brian Smith

Reflection
Introspection
Embeddednes
On the Origin of Objects

Others

http://krusty.eecs.umich.edu/cogarch3/CapabilLists/SelfReflect.html
http://krusty.eecs.umich.edu/cogarch3/CapabilLists/SelfReflect.html

http://www.sciences.univ-nantes.fr/info/perso/permanents/martinez/Researches/EDBT99/StudentWorks/SSrinivasa.html

TUNES

http://www.tunes.org/Review/Reflection.html#Blurb

Introductory Blurb about Reflection

Reflection is the ages old concept of someone thinking about oneself. Yes, there are other meanings to the word; this is the one we consider here. In Computer Science, Reflection is a powerful conceptual tool with such various applications as simplifying logical proofs enough to make them physically tractable, enabling dynamic (run-time) evolution of programming systems, transforming programs statically (at compile-time) to add and manage such features as concurrency, distribution, persistence, or object systems, or allowing expert systems to reason about their own behavior.

Sylvie Kornman

http://www2.parc.com/csl/groups/sda/projects/reflection96/abstracts/kornman.html

Pattie Maes

Computational Reflection

Damjan Bojadziev

Self-referentiality
http://nl.ijs.si/~damjan/phen.html

Artificial Life Approaches

Kampis, Georg (1991): Self-modifying Systems in Biology and Cognitive Science. Pergamon Press, New York.
Claus EMMECHE
http://alf.nbi.dk/~emmeche/cePubl/compnolife.html

2 Some comments on Hypercomputing

Selmer Bringsford

german vs. usa approach, grid computing vs. hypercomputing
The german approach of hypercomputing is working with clusters of workstations to build super-computers and is therefore grid computing.
In contrast, the usa approach is more concerned with the building of a hybrid construction of different sorts of computing, neural, symbolic, etc.

cit.

GRAND CHALLENGE (Kennedy)

A new proposal dealing with a model of 10^{13} elements is on the way. Not mentioning the principle problems of such an approach.

How can we build human scale complex systems?
Julian F. Miller and Catriona Kennedy
School of Computer Science
University of Birmingham

"Devise techniques for constructing software and hardware systems consisting 10^{13} subprograms (approximately the number of cells in the human body) that carry out useful functions that in some sense are as complex as that of a mature human being."

Background and Motivation
The complexity and sophistication of living creatures dwarfs human designed systems. Humans are constructed using a development process that starts with a single fertilised cell. Somehow, given the right conditions, this cell replicates and differentiates itself to form a human baby that learns how to talk, think, do mathematics, and write Grand Challenge proposals. We know that in some sense a human can be produced from the interpretation and

decoding of a string of information, yet we have almost no idea how to construct large scale systems using this type of mechanism. Human top-down design methods appear to be unable to create stable complex systems on anything like the scale of living organisms. We appear to be faced with huge combinatorial problems. As we move toward the construction of nanoscale physical computing systems we will face enormous problems that originate in the problem of getting the information from our macroscopic world into the microscopic world of the computational elements. Living systems have an extremely elegant solution to this problem: the information for the construction of the whole is contained in all the basic computational elements (cells). Computer algorithms formulate computation as a transformation of inputs into outputs. Living systems are not best viewed in this way. In computer algorithms programs and data are rigidly divorced from one another, in living systems there appears to be no clear boundary between programs and data. Currently we have virtually no idea how to construct systems of this nature. Computer algorithms and electronic hardware are extremely sensitive to errors that can have catastrophic consequences. Living systems do not exhibit this propensity for error and are massively robust. Trying to build systems that are in some formal sense as complex as human beings may shed light on many fundamental questions concerning the nature of computation and could be useful: selfrepair, intrusion detection, adaptive behaviour, intelligence…

The PCL approach to neural complexity

Gunther´s strategy in contrast to the neural science approach.

In other words: there are not only theoretical but also practical reasons why research in the neural system of the brain will never unrevel how the brain contributes to the solution of the riddle of subjectivity.

However, there is another way to approach the problem. Instead of working uphill from the neuronic level we may ask: what is the highest achievement of the brain? In other words: what mental world concept does it produce? We can describe this world concept in semantic and structural terms and work down from there posing the question: how must a brain be organised in order to yield such images with their peculiar semantic significance. This types of investigation has hardly started, but it is as important and necessary as the other one.

Gunther, Cognition and Volition

Obviously, the PCL approach takes the other option of producing cognitive systems, not denying the reasonability of the more conventional empiricist approach of modeling, that is cloning nature, here the brain of a little child.

It is helpful to use a decades old distinction Gotthard Günther´s of *homunculus* approach versus the *robot* approach in AI & AL.

From the point of view of the PCL approach the grand challenge is not so much to build an artificial child but some small animals, maybe insects, not only as mini-robots, but as cognitive and volitive systems.

Complementary to Gunther´s strategy I mention another strategy.

Living systems have an extremely elegant solution to this problem: the information for the construction of the whole is contained in all the basic computational elements (cells).

This situation is well mirrored in the polycontextural approach. All contextures, say as logical systems, contain the operators of the local and global interactions.

Computer algorithms formulate computation as a transformation of inputs into outputs. Living systems are not best viewed in this way.

Contextures are together by mediation as a form of structural coupling. Informational procedures like input/output communications play a secondary. role

In computer algorithms programs and data are rigidly divorced from one another, in living systems there appears to be no clear boundary between programs and data.

This corresponds clearly to the proemial relationship between data/programs and distributed systems, that is contextures. But this description is correct only from a external point of view. Internally there is no fuzziness between data and programs but a complex chiastic interchange of both. The functionality of being a program or being a set of data is strict and dualistic. But a single program can change into data and vice versa in a more fundamental sense than we know it already from programming.

It is not an accident that Catriona refers in her Ph.D. dissertation to some work about reflection and the “Blind Spot” problem I published many years ago.

Currently we have virtually no idea how to construct systems of this nature.

Maybe, the polycontextural approach can offer some implementable ideas.

Therefore, the PCL approach offers a strategy which is beyond well known procedures of producing complexities, like recursion, fractal and chaotic processes.

3 Some special institutions

MIT

Grand Challenge UK

http://umbriel.dcs.gla.ac.uk/NeSC/general/esi/events/Grand_Challenges/proposals/

DARPA

4 Similar or complementary work to the PCL-Project

4.1 On Architectonics

4.2 On Reflectionality

meta-level architectures

4.3 On Interactivity

intentionality
communication
cooperation
cocreation

4.4 On Positionality

blind spot

4.5 On Proemiality

switches

4.6 On Polycontexturality

combining logics
fibred categories

5 What are the decisive advantages of the PCL approach?

More formalizations than simulations

The main advantage of the PCL approach lies in the fact that it´s mode of thematization is basically formalization. Formalization in the sense of PCL produces a very strong connection to operativity on the levels of implementation and realizations.

The PCL approach can be understood as a complementary project to the classical and neo-classical approaches of AI. Thus it is not an isolated and incompressible endeavour. There are a multitude of very actual approaches in computer science which can be seen as complementary to the PCL approach. These approaches are more concerning the modi of implementation with the means of classical formalisms and programming languages than with formalization and developping new complex programming languages and tools.

The Ultimate Diamond Strategies of PCL

Given the algebraic and the coalgebraic approaches, that is the structural and the processual view of formalisms and computations, the PCL approach is decisively beyond this strong duality, establishing a way of thinking beyond this dichotomy, in the deconstructional sense of functioning neither with the one nor with the other, and at the same "time" with with both positions at once.

A new epoch of production lines

Therefore, a whole range of working prototypes of new products involving complexity, self-referentiality and interactivity are conceivable.

TransComputation: A new paradigm of computing

Nevertheless, the PCL approach is new and distinct from other well known modern approaches. More exactly, the PCL approach is developping not only some new programming languages as tools but a new medium of thinking and programming in the sense of a change of the very paradigm of computation. The Grand Challenge today is to understand computing beyond information processing and beyond the framework of classical mathematics.

General Intersubjective scientifity

The PCL approach is not only trans-disciplinary in its methodology but also understandable across the disciplinary boundaries because of its strong emphasis on operative formalisms. This is guaranteeing an intersubjective, that is, strongly scientific approach, which has no need to depend on any Gurus and / or mystical insights.

The mega-procedure of proemiality

The main method of the polycontextural approach can be seen in the concept and apparatus of proemiality. The proemial relationship can be instrumentalized to a strong tool in dealing with the project of extending the scope of computing.

Interactive computing and paraconsistent logics have been brought together by Goldin. Why is paraconsistency not enough?

No-nonsense; but well-founded in scientific avant-garde

Also the PCL approach is genuine new and contemporary, it has a well founded history in the development of philosophy, cybernetics and logics. Even if the main stream of second-order cybernetics is not well aware about the fact that PCL, by the work of Gunther, Pask, von Foerster and others, is a decisive framework of cybernetical thinking since the very beginning of the new cybernetics as it was developed at the BCL.

6 Comments on the Grand Chalenge Project

http://wetware.hjalli.com/000040.html

The 'Journeys' proposal does not suggest that the classical methods should be abandoned, and acknowledges their great success. It rather suggests that computer scientist also look outside this classical frame for solutions. Even though it's probably longer than most of you will read, I think the proposal's list of paradigms that define classical computing and possible non-classical extensions, give the best possible picture of the aims:

1: The Turing paradigm

classical physics: information can be freely copied, information is local, states have particular values. Rather, at the quantum level information cannot be cloned, entanglement implies non-locality, and states may exist in superpositions.

atomicity: computation is discrete in time and space; there is a before state, an after state and an operation that transforms the former into the latter. Rather, the underlying implementation substrate realises intermediate physical states.

infinite resources: Turing machines have infinite tape state, and zero power consumption. Rather, resources are always constrained. substrate as implementation detail: the machine is logical, not physical. Rather, a physical implementation of one form or another is always required, and the particular choice has consequences.

universality is a good thing: one size of digital computer, one size of algorithm, fits all problems. Rather, a choice of implementation to match the problem, or hybrid solutions, can give more effective results.

closed and ergodic systems: the state space can be pre-determined. Rather, the progress of the computation opens up new regions of state space in a contingent manner.

2: The von Neumann paradigm

sequential program execution. Rather, parallel implementations already exist.

fetch-execute-store model of program execution. Rather, other architectures already exist, for example, neural nets, FPGAs.

the static program: the program stays put and the data comes to it. Rather, the data could stay put and the processing rove over it.

3: The output paradigm

a program is a black box: it is an oracle abstracted away from any internal structure. Rather, the trajectory taken by a computation can be as interesting, or more interesting, than the final result.

a program has a single well-defined output channel. Rather, we can chose to observe other aspects of the physical system as it executes.

a program is a mathematical function: logically equivalent systems are indistinguishable. Rather, correlations of multiple outputs from different executions, or different systems, may be of interest.

4: The algorithmic paradigm

a program maps the initial input to the final output, ignoring the external world while it executes. Rather, many systems are ongoing adaptive processes, with inputs provided over time, whose values depend on interaction with the open unpredictable environment; identical inputs may provide different outputs, as the system learns and adapts to its history of interactions; there is no prespecified endpoint.

randomness is noise is bad: most computer science is deterministic. Rather, nature-inspired processes, in which randomness or chaos is essential, are known to work well.

the computer can be switched on and off: computations are bounded in time, outside which the computer does not need to be active. Rather, the computer may engage in a continuous interactive dialogue, with users and other computers.

5: The refinement paradigm

incremental transformational steps move a specification to an implementation that realises that specification. Rather, there may be a discontinuity between specification and implementation, for example, bio-inspired recognisers.

binary is good: answers are crisp yes / no, true / false, and provably correct. Rather, probabilistic, approximate, and fuzzy solutions can be just as useful, and more efficient.

a specification exists, either before the develop-ment and forms its basis, or at least after the development. Rather, the specification may be an emergent and changing property of the system, as the history of interaction with the environment grows.

emergence is undesired, because the specification captures everything required, and the refinement process is top-down. Rather, as systems grow more complex, this refinement paradigm is infeasible, and emergent properties become an important means of engineering desired behaviour.

6: The "computer as artefact" paradigm

computation is performed by artefacts: computation is not part of the real world. Rather, in some cases, nature "just does it", for example, optical Fourier transforms.

the hardware exists unchanged throughout the computation. Rather, new hardware can appear as the computation proceeds, for example, by the addition of new resources. Also, hardware can be "consumed", for example, a chemical computer consuming its initial reagents. In the extreme, nanites will construct the computer as part of the computation, and disassemble it at the end.

the computer must be on to work. Rather, recent quantum computation results suggest that you don’t even need to "run" the computer to get a result!

A thought provoking list of ideas indeed and as you can see the reference to nature and biology is all over it, or as put in the proposal paper:

Many computational approaches seek inspiration in reality (mainly biology and physics), or seek to exploit features of reality. These reality-based computing approaches hold great promise. Often, nature does it better, or at the very least differently and interestingly. Examining how the real world solves its computational problems provides inspirations for novel algorithms (such as genetic algorithms or artificial immune systems), for novel views of what constitutes a computation (such as complex adaptive systems, and self-organising networks), and for novel computational paradigms (such as quantum computing).

...

Meta-heuristic search techniques have drawn inspiration from physics (simulated annealing), evolution (genetic algorithms, genetic programming), neurology (artificial neural networks), immunology (artificial immune systems), plant growth (L-systems), social networks (ant colony optimisation), and other domains.

Movable Type 2.64, Copyright © 2003, Hjalmar Gislason

Non-Academic Projects

Peter Voss
Ben Goertzel

Transhumanism
General Artificial Intelligence Machine
Seed AI

Schmitdhuber ?

Schmitthuber´s Gödel Machine

Peter Krieg´s Polylogic (von Foerster) Machine
Peter Cariani´s Pask Machine

Also these projects don't touch the fundamentals of computation they try to develop a new paradigm of AI and AL.

There are some correspondences between these transhumanist approaches and what is envisaged by Aaron Sloman and Marvin Minsky. It is the idea of "child computing" and Seed AI, which means not to develop a highly intelligent machine as an analogy to adult human intelligence but to invent the framework of learnability and growing which would be able even to develop interactively with its environment its own logic.

From a general philosophical point of view these new trends go together with an "empirical turn" in computer science, back to nature (esp. today: biology) which is more reflecting the engineering aspect in contrast to the mathematical perspective.

Wegner´s approach, also highly mathematical, is in its aims and its paradigm, explicitly connected with empiricism.

Obviously, the PCL approach is not empiricist in the sense mentioned by Wegner, Sloman, Minsky and the Transhumanist, it is by its polycontexturality and kenogrammatics beyond this dichotomist situation of rationalism and empiricism realized today by logic based symbolism and learnable and adaptive neural networks and other approaches.

It is easy to mention that the polycontextural approach is beyond any sign systems, symbolic or statistic, or whatever, thanks to the pre-semiotic system of kenogrammatics. But it is an enormous difficult task to give an acceptable introduction to kenogrammatics as such.

A lot of fundamental problems are not solved in natural science. The danger of the empirical turn is easy to see, it transports these problems to computer science, mostly without knowledge about them.

May the classical approach of AI be rooted in logical positivism, a highly idealistic doctrine, also positivist would deny to be idealists, the new approach is based on empiricism. From a PCL point of view there is no necessity to accept this dichotomist situation. The search for foundation of computer science and AI esp. in other doctrines is denying its autonomy.

6.1 Seed AI, another myth?

Seed Computing is intrinsically interwoven with a fundamental belief in nature. We have to organize the conditions for learning and growing, nature will do the rest. This homunculus approach, which goes back to the medieval alchemists, has given up to understand and to know how to construct an intelligent system. Even with the assumption of a well reflected scientific knowledge about nature in the sense of molecular biology, neurophysiology etc. it is nature who has to realize the intelligence of the machine and not the engineer.

Maybe the metaphor of the seed in Seed AI sounds friendly to nature but it denies from the very beginning the definition of seeds. Seeds don´t come as singularities, seeds comes in the pluralities of thousands and more to realize some single succeeding plants. How can the metaphor of seeds be a leading metaphor to an engineering project? Probably only as the singular successful seed in denying its possibility as one of millions of unsuccessful seeds. It´s great to see the successful growing of a seed. Hopefully it will happen, even against the (mis)leading metaphor, to Seed AI.

6.2 Child like Computing, one more myth?

As Seed Computing, this approach favored by Minsky and Sloman is rooted in the metaphor "homunculus" and not in the metaphor of the "robot", constructed by engineers.

"Child computing” seems not to be mainly rooted in (neuro)biology but in development psychology. Not only Heinz von Foerster but also Minsky refers to the famous Swiss behavioral psychologist Piaget as one of his main empirical source.

The problem of inheritance is repeating itself again. The unsolved problems of Piaget´s psychology will appear as a serious obstacle in AI research.

My favorite problem in Piaget´s psychology is the relationship between accommodation and maturation.

“There is no accommodation without maturation and there is no maturation without accommodation”.

Does it help to classify Piaget´s psychology as a dialectical approach? Or as a second order cybernetics approach, as Heniz von Foerster proposed? As long as we don´t know what it means to be dialectical and second order cybernetic, I guess, it is not more than a hint in a direction which is probably not fully misleading. But to construct a system on the base of a vague hint is clearly not working.

What happens? It is the typical situation of confusing description language with construction language.

There is a lot to learn about the dialectics of Piaget´s psychology. But not much about dialectics and its apparatus. The reason is simple, and well known, dialecticians have the strong belief that there is no formal dialectics as there is a formal logic. Each formalization of dialectics would automatically kill its basic dynamics. How can it be possible to construct an artificial developing intelligent system on the base of Piaget´s psychology if this psychology is fundamentally based on dialectics and that the possibility of an operative dialectics is strictly denied by the dialecticians of all colors?

A classic in english language: “The Development of Dialectical Operations” (Ed. Klaus F. Riegel), Basel 1975

In German: Thomas Kesselring, Die Produktivität der Antinomie, Suhrkamp 1984

It is exactly here the place where the work of Gotthard Günther comes into play. Before to construct an intelligent machine, he tried to develop a constructive and operational theory and formalism of dialectic systems, short of dialectics.

Obviously, this position is in all aspects beyond the dichotomic paradigm of rationalism and empiricism, and therefor of the double metaphor of homunculus / robot.

Slide-diagrams PCL-CogSys

DERRIDA'S MACHINES

Cloning the natural

– and other fragments

Proemiality and Panalogy

The new scene of AI: Cognitive Systems?

ThinkArt Lab Glasgow

Institute of Speculative and Experimental Informatics

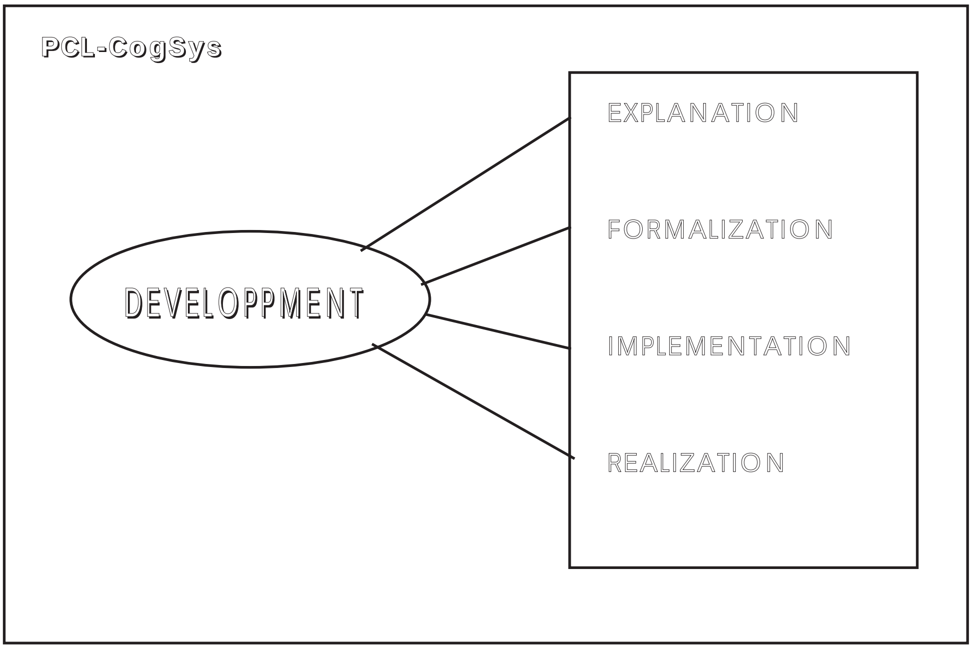
Dr. Rudolf Kaehr

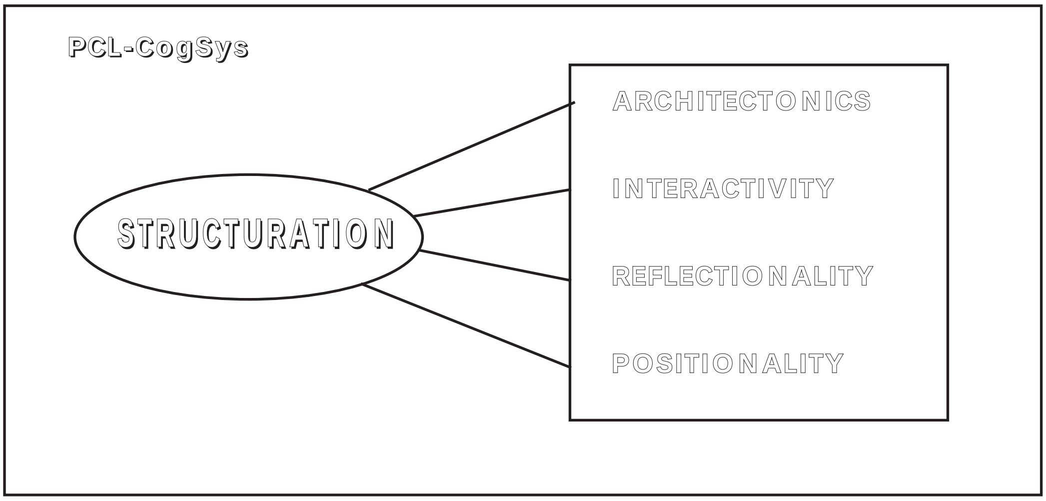
40 St. Enoch Square

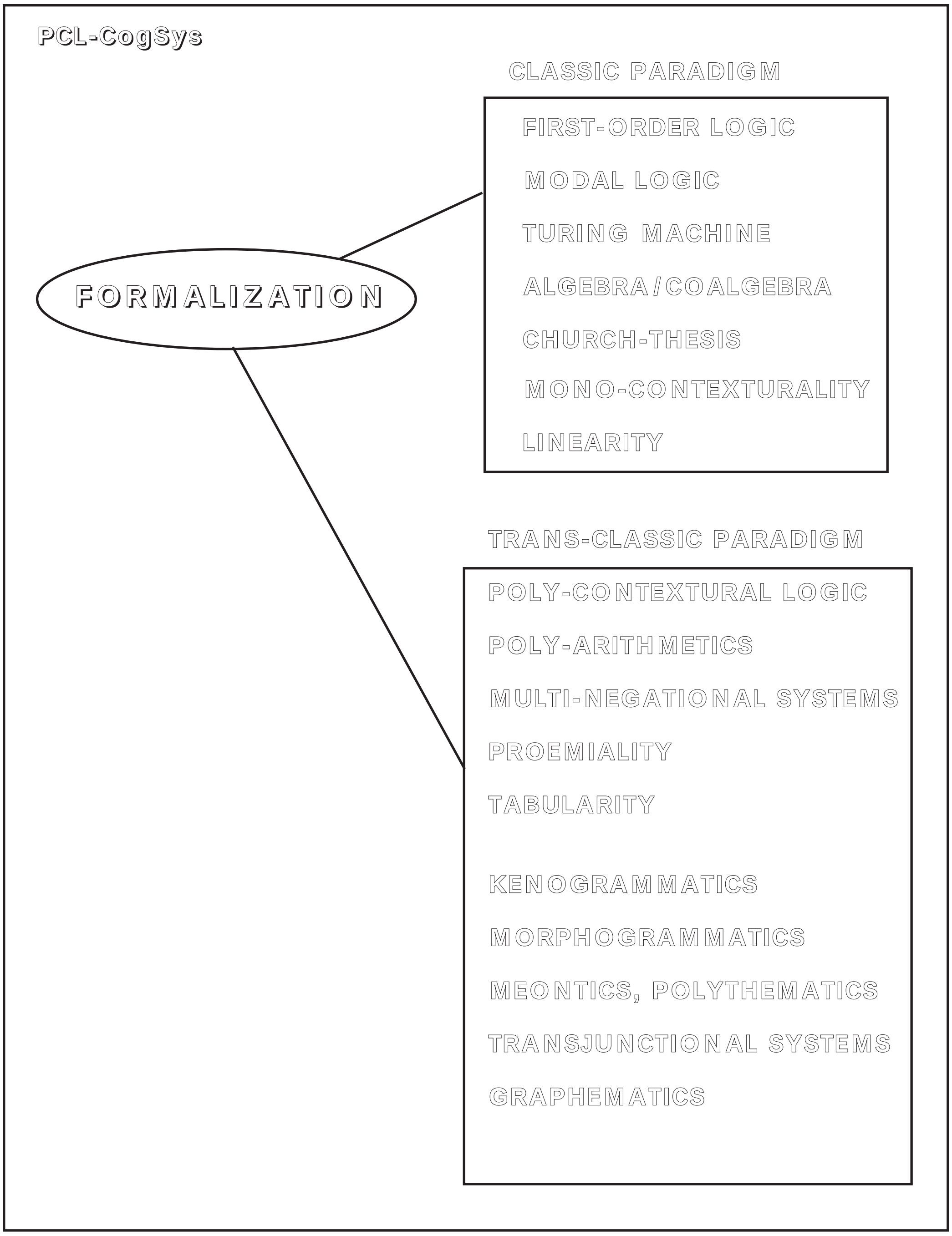
Glasgow G1

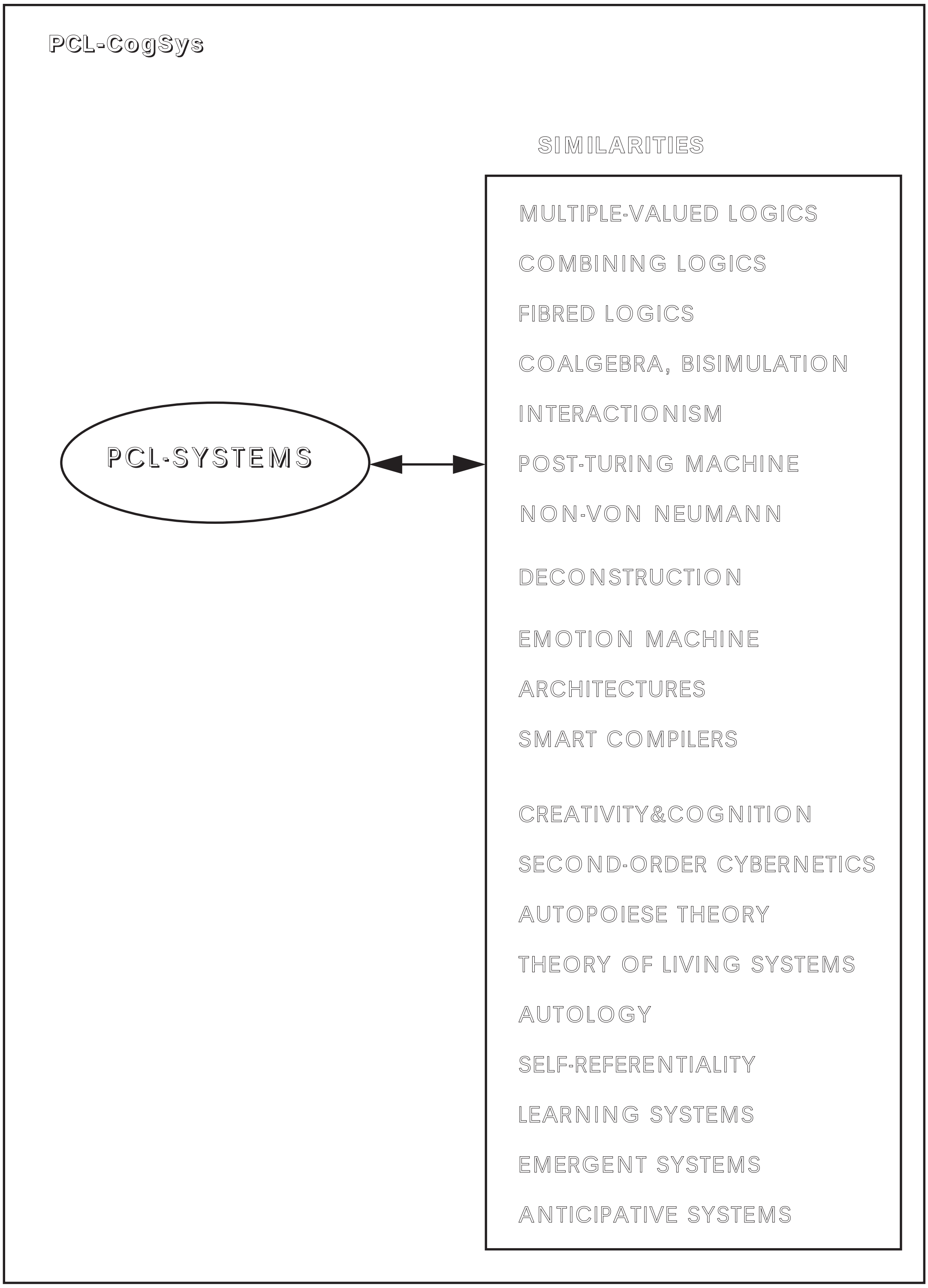
Scotland, UK

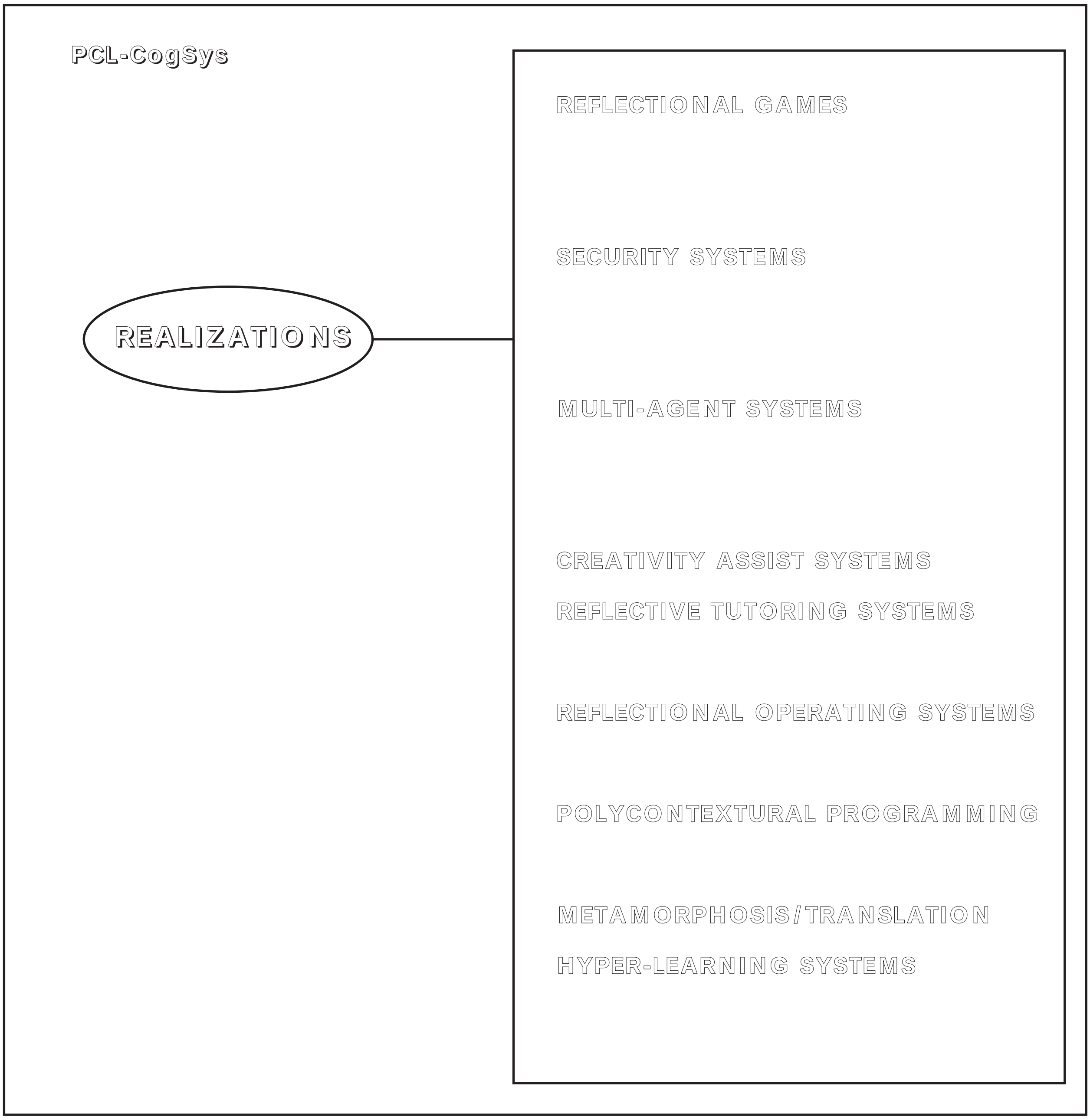
September 2003


PCL-CogSys
DEVELOPPMENT
EXPLANATION
FORMALIZATION
IMPLEMENTATION
REALIZATION

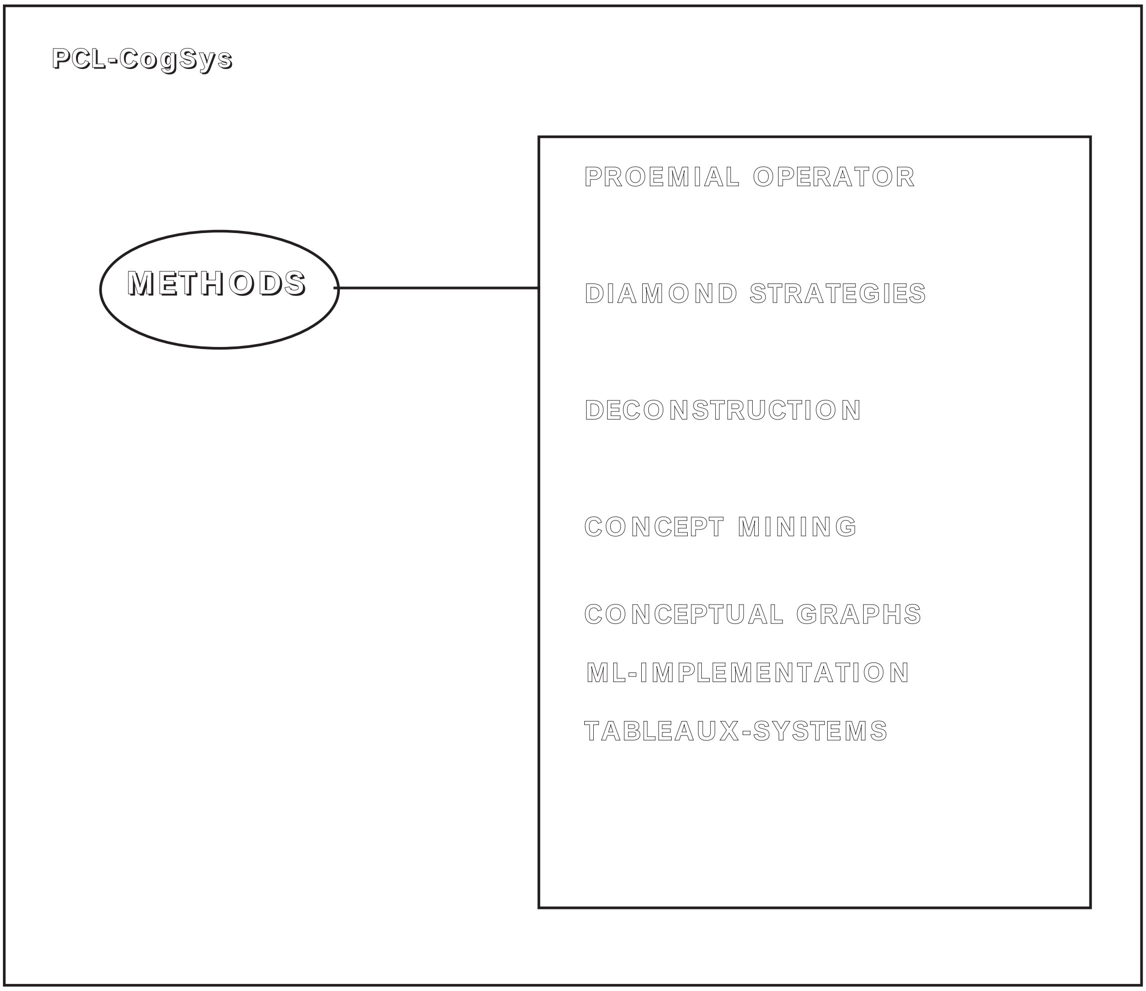

PCL-CogSys
STRUCTURATION
ARCHITECTONICS
INTERACTIVITY
REFLECTIONALITY
POSITIONALITY


PCL-CogSys
CLASSIC PARADIGM
FIRST-ORDER LOGIC
MODAL LOGIC
TURING MACHINE
ALGEBRA/COALGEBRA
CHURCH-THESIS
MONO-CONTEXTURALITY
LINEARITY
FORMALIZATION
TRANS-CLASSIC PARADIGM
POLY-CONTEXTURAL LOGIC
POLY-ARITHMETICS
MULTI-NEGATIONAL SYSTEMS
PROEMIALITY
TABULARITY
KENOGRAMMATICS
MORPHOGRAMMATICS
MEONTICS, POLYTHEMATICS
TRANSJUNCTIONAL SYSTEMS
GRAPHEMATICS


PCL-CogSys
SIMILARITIES
PCL-SYSTEMS
MULTIPLE-VALUED LOGICS
COMBINING LOGICS
FIBRED LOGICS
COALGEBRA, BISIMULATION
INTERACTIONISM
POST-TURING MACHINE
NON-VON NEUMANN
DECONSTRUCTION
EMOTION MACHINE
ARCHITECTURES
SMART COMPILERS
CREATIVITY&COGNITION
SECOND-ORDER CYBERNETICS
AUTOPOIESE THEORY
THEORY OF LIVING SYSTEMS
AUTOLOGY
SELF-REFERENTIALITY
LEARNING SYSTEMS
EMERGENT SYSTEMS
ANTICIPATIVE SYSTEMS



PCL-CogSys
REALIZATIONS
REFLECTIONAL GAMES
SECURITY SYSTEMS
MULTI-AGENT SYSTEMS
CREATIVITY ASSIST SYSTEMS
REFLECTIVE TUTORING SYSTEMS
REFLECTIONAL OPERATING SYSTEMS
POLYCONTEXTURAL PROGRAMMING
METAMORPHOSIS/TRANSLATION
HYPER-LEARNING SYSTEMS



PCL-CogSys
METHODS
PROEMIAL OPERATOR
DIAMOND STRATEGIES
DECONSTRUCTION
CONCEPT MINING
CONCEPTUAL GRAPHS
ML-IMPLEMENTATION
TABLEAUX-SYSTEMS

CogSys	PCL-CogSys
PANALOGY	ARCHITECTONICS
BRAINSTORMING	DIAMOND STRATEGIES
WAYS OF THINKING	POLY-CONTEXTURALITY
CHANGE	PROEMIALITY
EMOTION MACHINE	COGNITION&VOLITION
IMPLEMENTATION	FORMALIZATION
COMMON SENSE	HERMENEUTICS
PSYCHOLOGY	PHILOSOPHY
INFORMATION	STRUCTURATION
SEMIOTICS	GRAPHEMATICS

Some non-technical background texts to PCL

Limitations and Possibilities of Communication and Co-Creation

(In this paragraph I give a programmatic text (with minor changes) which I have proposed together with Eberhard von Goldammer in 1986 for a Conference at Brussels „The Challenge of the Future“. Translated by Bob Cleaveland. I don't remember of any response.)

The development of productive forces has led to a situation where it is no longer permissible to collectively gather social, scientific and technological actualities or data into a single unified or unifying context. The idea of a „Summum Bonum“, – whatever form this concept may want to assume, e.g. a „deep structure“ of the universe – has forfeited its synthesizing power.

It is necessary to develop a way of thinking for the future which does not simply replace as the basic premise unity with diversity but which deals with a more fundamental interplay supporting and relating both these factors operatively and textually. With the introduction of the „proemial relationship“ a first effort to realize this trans-logical interplay of unity and diversity is made.

The opening of new resources in thinking also depends substantially upon whether the transition–a paradigm change–can be achieved from the restrictive Gödel-McCulloch-Pitts era (the „mathematizing power of Homo-Sapiens“ E. Post) to a post-Gödelian epoch of scriptualizing and the methods of symbolising.

1 American Second-Order Cybernetics

The American second-order cybernetics points in particular to the necessity of such a shift:

Heinz von Foerster:*"The logic of our western industrial corporate society (with limited liability) is unidirectional, deductive, competitive and hierarchical, and the keystones of its paradigm are the claim of objectivity and the theory of types, which exclude in principle the autonomy of paradox and of the individual. In the scientific revolution that we know create and experience, however, we perceive a shift from causal unidirectional to mutualistic systemic thinking, from a preoccupation with the properties of the observed to the study of the properties of the observer."*

That which cannot be grasped or expressed from the fundament of logo-centric scientific thinking is an operative time-structuring in which linearity and tabularity, the fields of ruptures, emanation and evolution are communicated as complementary communication structures. Time as a complex system of emanation and evolution is not thought of or conceived as „present“ but from the difference, differance (J. Derrida), i.e., the discontexturality of the contextures „past“ and „future“ whereby time is freed from the concept of being (G. Gunther)

A further self-incapacitation of thinking occurs not only through the prohibition of basic self-references in formal systems, but also through the presupposition, the apriori of

„Potential Realisability“ (Markov) forming the basis of all operative systems. An additional hindrance results from its idealistic concept of infinity which absorbs considerable brainware energy and wastes communication possibilities.

Future problems of communication will not alone be mastered through new media because these problems are now pushing their way to the surfaces between the paradigms themselves. Both in scientific discourse as well as in social realities. In particular the communication between the natural sciences and humanities has not yet been developed. The problem is one of an „Ecology of Mind“ which must be solved by this communication, i.e., through the development of a position in which the reason for the differences between the two is recognized and the communication barriers are de-constructed.

The so-called „New Logic of Information and Communication“ belongs to the old paradigm of logocentrism if it is performed as a field of the „New Rethorics“ (Perlman), the „Dialogik“ (Lorenzen), or the „Dialectical Logic“ (Apostel, Arruda). Simply because their aim is the unification of the proponents and the opponents under the summum bonum of rationality and truth. This „new logic of information“ is not polylogic but still remains monologic.

Transformation of Man-Machine Communication

The purely instrumental understanding of technology which predominates today both in engineering and in the humanities is insufficient. A new understanding of the man-machine symbiosis as a heterarchical interplay between mechanism and creativity needs to be developed and practised in connection with the development of new architectures in computer technologies.

Future art should take a position of sovereignity of creation and production in co-operation with computer technology and new methodologies of thinking. An option could be a co-operation with the theory of polycontextural thinking and operating.

2 In a nutshell: Proemiality and Polycontextural Logic

The idea of an extension of classical logic to cover simultaneously active ontological locations was introduced by Gotthard Gunther (1900-1984, us-american thinker, born in Germany; colleague of Heinz von Foerster at the BCL (1961 -1972), Urbana, Illinois, USA). The ideas of Polycontextural Logic originate from Gunthers study of Kant, Hegel, Schelling and the foundation of cybernetics in cooperation with Warren St. McCulloch. His aim was to develop a philosophical theory and mathematics of dialectics and self-referential systems, a cybernetic theory of subjectivity as a chiastic and heterarchical interplay of cognition and volition in a (de-)constructed world.

Polycontextural Logic is an irreducible many-systems logic, a dissemination of logics, in which the classical logic systems (called contextures) are enabled to interplay with each other, resulting in a complexity which is structurally different from the sum of its components. Although introduced historically as an interpretation of many valued logics, polycontextural logic does not fall into the category of multiple valued logics, fuzzy or continuous logics or other deviant logics. Polycontextural logics offers a framework for new formal concepts such as multi-negational and transjunctional operators.

The world has „indefinitely“ many logical places, and each of them is representable by a two-valued system of logic, when viewed locally and isolated from there neighbour systems. However, a coexistence, a heterarchy of such places can only be described by the proemial relationship in a polycontextural logical system. We shall call this relation according to Günther the proemial relationship, for it prefaces the difference between relator and relatum of any relationship as such. Thus the proemial relationship provides a deeper foundation of logic and mathematics as an abstract potential from which the classic relations and operations emerge.

The proemial relationship rules the mechanism of distribution and mediation of formal systems (logics and arithmetics), as developed by the theory of polycontexturality. This relationship is characterised as the simultaneous interdependence of order and exchange relations between objects of different logical levels.

According to Gunther: „*The proemial relationship belongs to the level of the kenogrammatic structure because it is a mere potential which will become an actual relation only as either symmetrical exchange relation or non-symmetrical ordered relation. It has one thing in common with the classic symmetrical exchange relation, namely, what is a relator may become a relatum and what was a relatum may become a relator. Or to put it differently: what was a distinction may become something which is distinguished, and what has been distinguished may become a process of distinction. The proemial relationship crosses the distinction between form and matter. […] We can either say that proemiality is an exchange founded on order; but since the order is only constituted by the fact that the exchange either transports a relator (as relatum) to a context of higher logical complexities or demotes a relatum to a lower level, we can also define proemiality as an ordered relation on the base of an exchange.*“

The proemial relationship implies the simultaneous distribution of the same object over several logical levels, which is not covered by classical theories of types – because it doesn't distinguish between the sameness and the identity of an object.

The proemial relationship enables a trans-classical framework of polycontextural and tabular thinking useful for the study and the development of a human based information revolution and the development of a revolutionary technology.

Discontexturality: The Art of Thinking Art in ThinkArt

1 Creativity and Computability beyond Science and Metaphor

Art in all its forms has always been a producer of Metaphors. From Plato's writings against Art to Cyberspace there is a celebration of Metaphors and a strict fight against them.

The opposite of Metaphor or Metaphoricity is the Formality of mathematical Science.

Joseph A. Goguen reports that the famous logician E. Dijkstra *"is also famous for walking out of lectures as soon as an intuitive visual diagram appears"*. Any use of Metaphors instead of proofing scripturally the statement is against the strictness of mathematical reasoning and Scientifity.

The E2 Science Festival *"Science and Metaphor"* (Edinburgh) states that Metaphorical work could help to understand and even to solve main problems in Science.

But the role of Science is changing, too. Computer Science is not well understood as the Science of Problem Solving for real world Problems. The new Paradigm of Science is Construction. Computer Science appears now more as Reality Construction than as Problem Solving. And Reality Construction is more an Art then a Science.

The common ground of Metaphors and concepts is their relation to (the) one and only one idea of rationality, truth, pleasure, beauty and mankind (humanity); its logocentrism and mono-contexturality. (J. Derrida: White Mythology)

After having been experienced in a reversal of the order between Science and Metaphor we are now forced to invent a new interplay between Art and Science which rejects the common ground of Metaphor and concept; its Digitalism. Otherwise we would play the same game in reverse. Art has to refuse to be a servant of Science in producing Metaphors.

For Art a new continent of creativity and invention appears: Beside the traditional material of Art work there is a new stuff: *the Body of Science.* Scientific concepts, models, strategies, methods, etc. appears now as a possible material for creative work. Formal concepts as "relation", "linearity", "number", "computability", „net", „information", „interaction", „interface" or „system", „recursion", „re-entry" etc.etc. are discovered and unmasked as belonging to the new continent of Metaphoricity.

It seems to be necessary to invent/discover a new transdisciplinary cooperation between creativity and computability, between Art and Science, Concept and Metaphor.

This would open up a new use of Metaphors in Science and Art in surpassing the old barriers between creativity and computability. It would enable new possibilities of Thinking and of thinking Art and to build an interface between Thinking and Art in/as ThinkArt.

I would like to introduce this view point of poly-contexturality for the understanding of the new constellation of the interplay between Concepts and Metaphors. And to discuss the structural consequences it could have for Art in the Information Age.

2 Some reflections about the structure of the new Art Material

2.1 A certain chain of -ISMs

Ontological developments in recent history confronts us with a chain of primary or leading concepts: substance (material), function, system and structure. This basic concepts are organizing and determining the paradigms of Substantialism, Functionalism, Systemtheory and Structuralism –or better: Contexturalism, Kenogrammatics, Proemics.

What really works today is functionalism. The most effective representativ of functionalism is to find in the paradigm of Functional Programming.

Systemtheory is today more an ideology, a intention and aim for research, especially for culture studies, but not a working paradigm in the sense of hard science.

Structuralism, in the sense of our chain of paradigms, is not yet a paradigm with working methods and concepts.

Classical Structuralism, in the sense of the french movement, is more a functionalism or relationalism (Bourbacki) then a concept of structure in our sense.

System theory lives from its syntax and more concrete from its vocabulary, its set of signs. This signs are pregiven, they are the real starting point of the tectonics of the formal system. The same is valuable for all applicative systems, too. Even for Autopoietic Systems its components are determining the identity of the system; the system or the operations in it are not defining there components.

In contrast to system theory for structural or kenogrammatical systems there are no elements, components or vocabularies outside of the system. There is no sign repertoire as the very first level of the tectonics of the system. And there is no proper starting point of its syntactical structure. The operations – „over the components“ – of the systems are producing there proper components; but not as a self-referential and circular organization, but as a chiastic and proemiell event or mechanism of self- and co-creation.

This sounds very non-formal and has to be introduced in a very complex double strategy of formalization and metaphorical transformation of the old formal concepts and working metaphors.

2.2 The formal as the new material

From the point of view of kenogrammatics (kenos, Greek empty) strict formal terms as „relation“, „linearity“, „numbers“ etc. used in mathematics and formal logic are discovered and unmasked as being purely metaphorical.

The prototype of a formal system, of operativity and calculability: the Arithmetics or formalized theory of natural numbers is unmasked as a metaphorical system. Its basic terms are not defined formally they are used in the general context of common sense in a non-formal but figurative and metaphorical sense. *„Natural Numbers are given by God“* (Kronecker)

But, what's the status of being unmasked? The (W)hole western metaphysics is disguised and encrypted in it!

The mechanism of masking should be formalized and ånd at new metaphorized – and therefore un-masked as such. As such: there is now ontological decision for a *ov* or *me on*; there „is“ only the mechanism of the interplay between (*methexis*) the masking / un-masking – beyond propositions (*apophansis*).

The mask of the mask; the masking of the mask, the mask of the masking. To mark the mask as the mask of the mark and as the mark of the mask. There is no mask without mark and there is no mark without mask.

The whole mathematics is unmask as a masquerade of marks and masks.

At this epoch the sign was always in the role of the repeater, the *singe.*

To use signs and to sign with signs needs a subject as an actor of the process of signing. To mark is un-masked of subjectivity; marks are marks of marks – and only marks are marks of marks; of theme-selves, themselves without a self of them.

Therefore, we begin the game again with the kenogramm; it is neither mask nor mark.

About the Art of Programming Art

Rudolf Kaehr
Academy of Media Arts Cologne
WEB: www.techno.net/pcl, Email: kaehr@khm.de
Self-modifying media lectures: http://smml.khm.de

The process of preparing programs for a digital computer is especially attractive, not only because it can be economically and scientifically rewarding, but also because it can be an aesthetic experience much like composing poetry or music. Donald E. Knuth, The Art of Computer Programming

Hackers are Artists Ars Electronica 2000

1 Open_SourceThinking Project Glasgow

It seems to be necessary to invent/discover a new transdisciplinary co-operation between Creativity and Computability, Concept and Metaphor, Art and Science. This would open up a new interplay between creativity and computability surpassing its dichotomic obstacles.

We invite artists and researchers of these topics to take part to the Open_SourceThinking Project by the ThinkArt Lab Glasgow in co-operation with the Academy of Media Arts, Cologne to study the state of the art in post-classical computing of programming art.

2 "Well-defined" problems in creating problems

Computers are traditionally designed to solve *well-defined* problems. This understanding of computer systems as tools for supporting repetitive and mechanical works was for a long time a satisfying paradigm in the domain of business and engineering. But meanwhile the computer has entered the realm of creative work. Software tools are ubiquitous and indispensable for all kinds of creative processes. We now begin to realise that creative work is not so much a question of solving well-defined problems.

But what sort of problems do artists solve? We think that the work of an artist or any other creative group is only vaguely connected with problem solving. Creative processes are not special cases of problem solving. Even in the fields of business and engineering there are wide areas which are at least "ill-defined problems". Therefore computer systems should be understood more as a medium of co-operation and co-creation in assisting creative work and not primarily as tools for solving problems in creative processes.

3 What could we understand by creativity?

Is there any framework to surpass the philosophically relevant dichotomy of creativity as unstructured flashes of insights or applications of procedures of intuition for inspiration versus computation? What are the limits and the challenges of support systems for creativity? Are there possibilities for Creativity of Thinking and of thinking Art by inventing an Interface between Thinking and Art in/as ThinkArt?

Is it then necessary for artists to be creative? From a more philosophical point of view it turns out that it will become more the task of computer systems to be able to realize creative prozesses and to support artist with creative solutions. Not the artist has to be creativ, it´s the machine which will be creative. The self-understanding of artists has to be transformed to a new form of activity beyond creativity, innovation and problem solving. The role of the artist is then not primarily to create but rather to ´thematize´ a world which is already in existence assisted by creative machines.

The tools we use in creating computer art at the moment are to a great deal responsible for the quality of our aesthetic products. But even more important is the usually underestimated fact, that the basic concepts behind the tools guide our thinking, determine our perception and constrain our creativity. Computers today are generating objects which artists have to identify, to select and to evaluate as aesthetical objects. Tools for supporting creative work have to be adaptive to the needs of the user, emergent in functionality, and powerful in expressiveness.

4 Cutting the Human/Machine-Interface again

Developments in Computer Science and Second-order Cybernetics shows us, that the new movements in Information Technology can be understood as a radical new structural cut between "subjectivity" and "objectivity". We propose that this structural view of the developments opens up a more relaxed understanding of cyberculture than it is defined by the paradigm of information processing.

It seems that the well known attributes of cyberculture like speed, omnipresence, virtuality, globalization, digitalism, complexity etc. are more the effects of a radical new structural cut in our self-definition than the cause or the organizational determinants of the historical formation of the information age. The new cut of computer revolution is purely "contextural" and is not well understood in terms of space, time, reality, identity and information and its deviants which concepts still belongs to the classical paradigm of modern science which itself is based on the first cut of cartesian philosophy and science.

Many domains which were believed to belong to the side of "subjectivity" now have moved to the side of "objectivity" by simultaneously changing the structure of objectivity and of subjectivity. This change in the epistemological cut transforms therefore the concepts of space, time and information and their logic too.

Gotthard Gunther wrote in "Cognition and Volition. A Contribution to a Cybernetic Theory of Subjectivity" (1970):

... since the Aristotelian epistemology required a clear cut distinction within subjectivity between subject as the carrier or producer of thoughts and the thoughts themselves, it was reasoned that the subject of cognizance could have rational thoughts without being a rational entity itself.

It should be kept in mind that, if we postulate a polycontextural Universe, the barriers which now cut through this empirical world, have lost nothing of their intransigency by being multiplied.

In order to integrate the concept of discontexturality into logic we have introduced the theory of ontological loci. Any classical system of logic or mathematics refers to a given ontological locus; it describes the contextural structure of such a locus more or less adequately. But its statements-valid for the locus in question-will be invalid for a different locus.

How can artists help to revolutionise the new technologies? And how can new technologies help to transform art and the self-understanding of the artist? Which new framework of logics and rationality do we need to formulate and to formalise this new form of thinking beyond classical dichotomies?

From the point of view of a polycontextural framework of logic questions about forms of interactivity, creativity and invention arise in a new light and will have to surpass the classical paradigm of information processing and computing (Shannon, Turing, First-order Cybernetics) being guided by the question "What«s after Digitalism?".

The Human-Machine-Interface today has at least two faces. One for the classical relationship between man and machine in the framework of the first cut, another one after the second cut. It«s the Janus face of the Human/Machine-Interface. One looking back to the history of mankind as the inventor and ruler of all sorts of machines, this is the attitude which governs the epoch from the archimedean to the cybernetic types of machines, the other face looks forward to a new future of humanity and its relationship to technology beyond cultural pessimism and computer euphoria.

5 To use and to be used by technology and beyond

As long as we only use technologies - as a toolbox for creativity - we believe in the simple cut between subjectivity and objectivity. Therefore human beings are just the users of computer systems and in the use of them they remain unchanged in their own structural definition. Therefore the anthropological structure of human beings like autonomy and identity doesn't change in being involved in using computer systems.

This orthodox position is clearly described by Donald A. Norman in Worsening the Knowledge Gap:

The Computer should Be a Tool*. Computers have the capability to act as tools for us everyday folks, as knowledge amplifiers. This means they should help us with our everyday tasks, making it possible to do things we could not do before. Computers are tools, and should be treated as such; they are neither monsters nor savants, simply tools, in the same category as the printing press, the automobile, and the telephone.*

But first we should be able to answer the question of J. Frazer proposed from the viewpoint of Second-order Cybernetics: *"Can computers be 'just a tool'?"* (in: mutual use of cybernetics and science, Amsterdam 1991)

The question 'Are computers just a tool?" or can they be? raises the general question of can any tool be neutral? I believe most of the issues are centred on the innocent sounding cliche "the computer is just a tool". A psychological trait in the naive, usually reserved for human authority has been transferred in an antropomorphic manner to the computer. But that just implies that it can be no more than a tool and this I believe also to be very dangerous. Tool is equally loaded because the word is taken to imply that it is under the control of the user and working for and in sympathy with the user.

6 Questions and Outlooks faced at the Academy of Media Arts

After ten years of experiences at the Academy of Media Art, Cologne we learned that there is no chance for artists than to be trained in programming and to be able of questioning the fundamental concepts of programming (languages, models of computation, logics) and the constructions of interfaces. Not only new ways of teaching computer science and programming have to be developed but also new types of systems, which are open-ended and semantically emergent have to be considered.

Taking this fact serious means that artists should not only be trained in using tools but should learn to design and program their own computing environments in a creative manner, i.e. to develop their own systems. This activity of personal programming and constructing interfaces by artists - which are different from those of innovative engineers - has to be supported and guided by a new framework of conceptual orientation.

Here are some of our questions:
- How can we support the user in his / her / its search for problems?
- What are the generic architectures for systems with emergent functionalities?
- How can individualised long-term relationships between systems / artists look like?
- What are the alternatives to modularization and hierarchization in complex systems?
- How can we dissolve and distribute our global point of observation and manipulation?
- How can we gain usability without loosing the full potential of the machine?
- How can we design systems capable of self-development and nevertheless keep control?
- How can we combine reliability and determinism with uncertainty and surprise?

6.1 What are the new Paradigms of Computation ?

Our method of *concept mining* (in contrast to data mining) produced an interesting list of developments in post-classical computing in the field of *Beyond Computation*. It´s all about surpassing the limits of algorithms and Turing Machines. The heroes of classical computing and their work is now history and has to be studied by historians and media archaeologists. What we need are new horizons of computing in the open framework of the second cut.

Here are a few, hopefully helpful, non-technical hints about our directions of research and teaching.

Reflectional Programming

Systems which are capable of self-reflection are able to examine their own internal processing mechanisms. They can use this capability to explain their behaviour, and modify their processing methods to improve performance. Such systems must have some form of Meta-Knowledge available, and in addition, they must actively apply the Meta-Knowledge to some task.

Thus, the appearance of second order cybernetics is the appearance of a new dimension -- reflexion. However, this dimension was developed differently in the Soviet Union and the West. In the Soviet Union, the idea of reflexion was combined with the idea of structure; as a result, reflexive analysis appeared. In the West, the idea of reflexion was combined with the idea of computation; as a result, calculations with self-reference appeared. Vladimir A. Lefebvre 1986

Computational Ontology beyond Identity

Real-world computer systems involve extraordinarily complex issues of identity. Often, objects that for some purposes are best treated as unitary, single, or "one", are for other purposes better distinguished, treated as several. The aim of the Computational Ontology project is to focus on identity as a technical problem in its own right, and to develop a calculus of generalized object identity, one in which identity -- the question of whether two entities are the same or different -- is taken to be a dynamic and contextual matter of perspective, rather than a static or permanent fact about intrinsic structure. Brian Cantwell Smith

Polycontextural Logic: Transjunctions of viewpoints and contextures

The digital abstraction is not a statement about how things are; it is merely a way of viewing them. A combinational circuit may be analyzed in terms of boolean logic, but it is voltage, are not a collection of ones and zeros. (...) At best, the digital abstraction tells us that the combinational circuit is amenable to analysis in term of ones and zeros; but it does not change the reality of what is there. Andrea Lynn Stein

Logical fiberings prove to be particulary suitable for modeling communication and interaction between co-operating agents, due to the possibility to switch between a local/global point of view which is typical for this framework. Jochen Pfalzgraf

Topics in Co-Operation, Interaction, Co-Creation

Algorithms and Turing machines (TM) have been the dominant model of computation during the first 50 years of computer science, playing a central role in establishing the discipline and providing a deep foundation for theoretical computer science. We claim that TMs are too weak to express interaction of object-oriented and distributed systems, and propose interaction machines as a stronger model that better captures computational behavior for finite interactive computing agents. Moreover, changes in technology from mainframes and procedure-oriented programming to networks and object-oriented programming are naturally expressed by the extension of models of computation from algorithms to interaction. Peter Wegner

Patterns of Self-(Organisation, Reference, Amendment, Reproduction)

Ideas of self-reference (and its self-modification), and their application to cognition have a much longer history, however. (...) The cognitive and philosophical significance of such issues was first raised by the so-called BCL school, whose members and associates included W. McCulloch, W.R. Ashby, G. Günther, L. Löfgren, H. von Foerster, and H. Maturana. George Kampis

6.2 How to Organise our Work of Programming Art?

XP as a specific method of generating software

eXtreme Programming as a heterarchical and chiastic mode of inter/trans-personal programming is a real challenge for artists programming art work. Pair programming and collective ownership goes much beyond classical teamwork. It involves a new understanding of the identity and subjectivity of artists and programmers in their mutual work. It displays well the intimate relationship between creation, control and implementation as a semiotical process beyond personal psychologies.

Some important features: XP teams focus on validation of the software at all times. Programmers develop software by writing tests first, then software that fulfils the requirements reflected in the tests.

Pair Programming. XP programmers write all production code in pairs, two programmers working together at one machine. Pair programming has been shown by many experiments to produce better software at similar or lower cost than programmers working alone.

Collective Ownership. All the code belongs to all the programmers.

UML as a general method of modelling projects

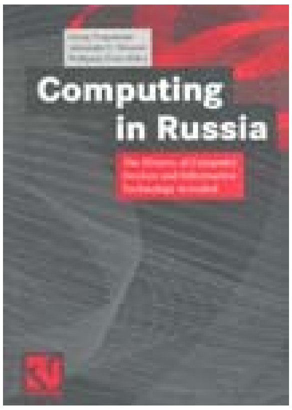
The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems. Using UML, programmers and application architects can make a blueprint of a project, which, in turn, makes the actual software development process easier.

Mostly computer art projects are much too complex and too ambitious to be realised in the context of the usual art funding. UML can help to design a conceptual model of the project. It could be reasonable to accept this UML modelling as a first realisation of the concept of / as the art work.

6.3 Which Languages for the Art of Programming?

Interactive Programming as organizing an interactive community of objects goes far beyond the traditional concept of programming in the sense of inventing algorithms for solving well-defined problems.

Interactive Programming In Java *provides an alternate entry into the computer science curriculum. It teaches problem decomposition, program design, construction, and evaluation.* Andrea Lynn Stein



Rudolf Kaehr, Computation and Metaphysics

IN

Trogemann, Georg / Nitussov, Alexander Y. / Ernst, Wolfgang (Eds.)

Computing in Russia

The History of Computer Devices and Information Technology revealed

Vieweg Verlag 2001, 350 pp. With 128 Fig., Hardc.

ISBN: 3-528-05757-2

Preis: 49.90 / CHF 81.50

Out of print 2003

This book is the first compendium on the development of the computer in Russia to appear in the West. After briefly illuminating the history of Russian mechanical calculation devices, the book largely focuses on the first generations of (military and civilian) electronic computers, most of which were developed in the Soviet Union during the "Space-Race" and the Cold War, simultaneously with similarly fundamental developments in computing in the U.S.A. The reader is introduced to computers and cybernetics from mathematical, technical, social and cultural perspectives through archive material and through texts by some of the preeminent veterans of Russian computing (historians, engineers, military historians). This alternative history and pre-history of information processing and of the computer ends with the adopting of the IBM standard and of Western technologies around 1970. Under the title Arifmometr (the name of the first Russian calculation device), a critical part of Eastern European technological culture is (re)-discovered for the reader; at the same time, the reader is reminded of the alternatives to the Western hemisphere's concept of the computer, which are of decisive historical interest.

Computation and Metaphysics

„There is little doubt that our present „thinking“ machines are hardly more than wooden horses.“ Gotthard Gunther

Beyond Marxism and Cybernetics

Gotthard Günther (1900-1984) was a man and a thinker of the in-between and frontiers[1]. This was not the result of his emigration from Germany to USA – he was not a victim lacking a Heimat – this was his decision as a result of radical thinking. However, he did not see himself as fitting within the contemporary movements of his time, rather he perceived himself as being much ahead of an arriving future.

Also one eyed he could see far more into the landscape of the future than most of us could ever see with three eyes.

1. Gotthard Günther, Selbstdarstellung im Spiegel Amerikas. In: L.J. Pongratz (Hrsg.), Philosophie in Selbstdarstellungen Bd. II, Meiner: Hamburg 1975, 1-76

For his colleagues at the BCL[2] he was a continental philosopher, for his philosophy colleagues in Germany he was an American cybernetician, for the GDR ideologists he was a western metaphysical idealist and for the BRD philosophers he was a dialectical materialist. For the German New Left he was a logical positivist, for the positivists he was a Hegelian transcendentalist. For himself he was a transcendental logician but then discovered that he was a dialectical materialist, but in the sense of Lenin and Schelling emphasizing the heterarchical polycontexturality of grounds. In his thinking he didn´t accept any compromise, but for his *special food* he had to go to a lorry drivers inn. *Günther* was never a name in the singular, they had always been called *The Günthers*: Gotthard and Marie[3]. He was a good friend of thinkers of very different origins like Ernst Bloch the Marxist philosopher at the time of his emigration in the USA and his main work *„Idee und Grundriss einer nicht-Aristotelischen Logik"*[4] written in the 50´s was supported by the Platonist Kurt Gödel.

Günther himself was never involved in politics. He liked the clear sky and the fresh air of his gliding and skiing. In the fresh air of the winter mountains of New Hampshire and focussed with only one eye, he was able to make distinctions which would have been confused by more disturbance. This was the place he came to his radical metaphysical and logical decisions about the future of thinking. Back from the mountains down in the cities there was mismatches everywere.[5]

With the „Hyäne des Pentagon" at the Checkpoint Charley

With his passion for skiing - he had to give up gliding - he became an academic spy, even a double spy; at least their were some people who liked to belief that. After he became a professor emeritus in1972 he gaves lectures in philosophy at the university of Hamburg and he made his home there. The Academy of Science in Berlin, former Capital of the GDR, wanted his secrets about the newest developments of US cybernetics as developed at the BCL - they received from Günther a hard lesson about the necessity to change their dialectical materialism towards a transclassical operational dialectics. The US Air Force paid his trip back to the mountains. Some philosophical reports about cybernetics in Berlin (Ost) were delivered. The BCL was known only by a few specialists in the West as well as in the East in the 70´s. Today it is the source of the new German ideology: Radical Constructivism, Second Order Cybernetics and Autopoiesis with Heinz von Foerster and Humberto Maturana as the leading figures.[6]

At this time - I invited Günther to the Free University of West Berlin and accompanied him to his lectures at the Academy of Science - we had a crucial point in common: both of us had to pass the mysterious Checkpoint Charley; now part of a museum. By passing this place of technological secrets Gotthard told me that he is a „Hyäne des Pentagon" (or that the other side told him this). I didn´t really understand, probably because I was hearing something sounding more like Princeton then Pentagon. I couldn´t believe that there could be any precious secrets at the Academy.

2. BCL: Biological Computer Laboratory, University of Illinois, Urbana, Ill., USA, 1957-1976
3. Dr. Marie Günther-Hendel, jewish, teacher and founder of a free school in Italy
4. Gotthard Günther, Idee und Grundriss einer nicht-Aristotelischen Logik, Meiner Verlag Hamburg 1959
5. Gunther-WEB: www.vordenker.de and www.techno.net/pcl
6. Realitäten und Rationalitäten , A. Ziemke, R. Kaehr (eds) , Selbstorganisation, Bd. 6, Dunker&Humblot Berlin 1995

Also fully involved in multiple-valued logic and perfectly informed by the JPRS[7] Günther did not mention anything about the first implementation of a *ternary* computer in1958 by a Russian team at the Computing Center of the Moscow State University[8] but had to respond to a hard critique from the Moscow logician Alexander Zinovjev about some problems involved in his place-value system of logic[9].

Günther, a lifelong emphatic skier, earned his money as a research professor for the foundations and philosophy of computation and cybernetics. Since the appearance of symbolic logic in the 30´s he was convinced that dialectics could only succeed and prove its supremacy over Aristotelian logic if it could find a formalism beyond all logical formalisms for its realisation[10]. He was one of the very first philosophical readers of the „*Introduction*" by the Polish logician Alfred Tarski.[11]Again that was in contradiction to the mainstream of German transcendental logic and philosophy. With the raise of Cybernetics in the USSR and its emphasis in the GDR by Georg Klaus, Manfred Buhr and Günther Kröber[12] there was a hope of some possible co-operations in the project of formalizing dialectics. After having given a lecture in Moscow Günther wrote one of his last works *„Identität, Gegenidentität und Negativsprache"* which tracks back to a lecture given 1976 in Belgrade. In this text Günther makes a step beyond the dichotomy of number and logos in introducing the concept of a new type of language for the notation of non-designational realities.[13]

We know at least since his book *„Das Bewusstsein der Maschinen"*[14] that Günther was proud to be on the payroll of the US Air Force Office of Scientific Research. This fact was surely one of the main reasons why he was totally ignored by the German New Left Movement. I remember a wild night with some Maoist comrades in West Berlin. I told Alfred Sohn-Rethel, then a late member of critical theory, he just arrived from exile in Birmingham, U.K. that by the irony of history the real thinkers of dialectics are not in the revolutionary underground of a socialist country but at the BCL sponsored by the US Air Force.

Since the very beginning of his academic life Günther was interested in the philoso-

7. JPRS: Joint Publications Research Service, RAND Corporation, Santa Monica, Cal.,USA
8. *„In 1958 the first full scale implementation of a ternary computer was completed by a Russian team at the Computing Center of Moscow State University, and named Setun´. It was used for some time, but both poor hardware reliability and inadequate software hampered its usage."* Computer Science and Multiple-Valued Logic, (ed.) David C. Rhine, North-Holland, 1984, p. 7, cf. Cybernetics and the Dialectic Materialism of Marx and Lenin, footnote 18, this book
9. Nachlass: Zinovjev
10. Gotthard Günther, Logistik und Transzendentallogik, in: Beiträge zu einer operationsfähigen Dialektik, Bd. I, Felix Meiner Verlag, 1976
11. A. Tarski, Einführung in die mathematische Logik, Verlag Jul. Springer, Berlin 1938
12. *„Aus heutiger Sicht sind diese Versuche einer mathematischen Modellierung dialektischer Widersprüche bestenfalls von historischem Interesse. Sie haben weder die Philosophie noch die Kybernetik substantiell bereichert."* K. Günter Kröber, Kybernetik als mathematische Theorie dialektischer Widersprüche, in: Kybernetik steckt den Osten an - Wiener's Ideen in Osteuropa und der DDR, Kolloquium der Gesellschaft für Kybernetik e. V. , Nov. 2000; cf. http://www.kybernetiknet.de/
13. Gotthard Günther, Identität, Gegenidentität und Negativsprache, Hegel-Jahrbuch , Pahl-Rugenstein, 1979
14. Gotthard Günther, Das Bewusstsein der Maschinen, Eine Metaphysik der Kybernetik, Agis Verlag, Baden-Baden 1963

phy of history. It is no surprise that there are several unpublished papers and book manuscripts about Russia and Marxism of the former USSR in the Nachlass[15].

„Cybernetics and the Dialectic Materialism of Marx and Lenin“[16] is not simply a literal translation of the german paper, from a lecture at the University of Cologne in 1964, but a transformation for the purpose of the US reader. And also this English version exists in several forms and intentions.

„Das Bewusstsein der Maschinen“ first published in 1957 and then in 1963 with a new chapter *„Idealismus, Materialismus und Kybernetik“* gave as some of his Western friends thought, dialectical materialism too much of a positive image. His answer was that he took both Apostel Paulus and Lenin with the same seriosness. When attacked by a readers-letter in *Astounding Science Fiction* he replied that it is more dangerous to be a metaphysician than to be a Marxist in the USA of today (McCarthy era).

With the Science (Fiction) Avantgarde

Gotthard Günther was always into techniques. Not only was he involved in the science fiction avant-garde with John W. Campell Jr. and published in *Astounding Science Fiction* and *Startling Stories* in the 50´s he also was the first to introduce American science fiction to Germany and he had a license for professional gliding and skiing. All this was too early for the Germans and the books *„Weltraumbücher“* published by Karl Rauch Verlag Düsseldorf 1952 had to be taken off the shelf. It was surely enormous luck but perhaps not a total surprise that Warren Sturgis McCulloch[17] discovered the importance of Günthers work for the logical foundations of cybernetics. He arranged for him a professorship at the BCL where Heinz von Foerster was the director.

Some years before *„Cybernetics and the Dialectic Materialism of Marx and Lenin“* Günther presented his fundamental work *„Cybernetic Ontology and Transjunctional Operations“*[18] on the 1 April 1962. Later published in the famous *„Self-Organizing Systems“*[19]. In this work he proposed a far-reaching formalization of dialectical und reflectional structures able to give a foundation for the implementation of subjective behaviours in machines. As a main step there is the formalisation of the transclassical operators of rejection and transjunction embedded in his morphogrammatics. With this background of polycontextural logic, his refutation of the whole alternative of idealism and materialism, which he had a deep knowledge of, and the design of a transclassiscal Worldview finally got its scientific foundation.

The Vietnam War and the End of Switching

When Günther was proposing machines capable of self-generating alternatives he was not only fully rejecting the alternative of Western idealism and Eastern materialism but trying to implement this same gesture into his idea of a trans-classical machine able of making refutations. His proposal *„A Study of new Development in Dialectic Theory*

15. Gotthard Günther, Die amerikanische Apokalypse, (ed.) Kurt Klagenfurt, Profil Verlag München Wien 2000
16. this book, pp.
17. Gotthard Gunther, Numbers and Logos, Unforgettable Hours with Warren St. McCulloch, in: Selbstorganisation, pp. 318-348
18. Gotthard Gunther, Cybernetic Ontology and Transjunctional Operations, Technical Report No. 4, Electrical Engineering Research Laboratory, University of Illinois, Urbana, Ill., Sponsored by: National Science Foundation, Grant 17414, Washington 25, DC..
19. Self-Organizing Systems, M.C. Yovits et al (eds.), Spartan Books, Washington, D.C., 313-392, 1962

in Marxist Countries and their Significance for the USA"[20] ran in parallel to the complementary proposal for the *„Investigation of a Mathematical System for Decision-Making Machines"*[21].

„On the other hand, a machine, capable of genuine decision-making, would be a system gifted with the power of self-generation of choices, and then acting in a decisional manner upon its self-created alternatives. (...)
A machine which has such a capacity could either accept or reject the total conceptual range within which a given input is logically and mathematically located. It goes without saying that by rejecting it the machine displays some independence from the programmer which would mean that the machine has the logical and mathematical prerequisites of making decisions of its own which were not implied by the conceptual range of the programme. But even if we assume that the machine accepts affirmatively the conceptual context of the programme qua context, this is by no means the same as being immediately affected by the specifique contents of the programme that the programmer feeds into it. If we call the first attitide of the machine critical acceptance of the programme and the latter naive acceptance, then it mus be said that the difference of their handling a given input in both cases are enormous. In the first case a conceptual and therefore structural context is rejected this does not necessarily imply that also the specific content of the programme are rejected. They still may be accepted, but moved to a different logical or mathematical contexturality."[22]

Unfortunately lack of money and the need for more serious military R&D caused by the Vietnam War made a bitter end to this story[23]. After one last grant to complete the final archiving of the work done, the BCL closed in1976 with its *„BCL Publications"*[24].

As an expert in skiing, for whom water skiing was a perversion[25], Günther wanted at least to have a transclassical computer system able of reflection, cognition and volition in his studio before dying. Skiing was his obsession and to build a transclassiscal computer his profession.[26]

Computers in the sense of transclassical cybernetics are not simply a tool or a medi-

20. Gotthard Gunther, A Study of new Development in Dialectic Theory in Marxist Countries and their Significance for the USA, 13 S., 1970
21. Gotthard Gunther, Proposal for the Continuation of a Mathematical System for Decision Making Machines, Under Grant AF-AFOSR 68-1391 for One Year From 15 October 1970, July 31, 1970
22. Gotthard Gunther, Proposal for the Continuation, p. 6-7
23. *"But then came the Mansfield Amendment. Most of the early work on cybernetics had been supported by the Office of Naval Research and the Air Force Office of Scientific Research. But in about 1968 the Mansfield Amendment put an end to research projects supported by the Department of Defense, which were not clearly related to a military mission. It was intended that the National Science Foundation and other agencies would pick up the support of projects that had been funded by DOD. The problem of course was that these agencies did not have the people who were familiar with the work in cybernetics. There followed several frustrating years of searching for new sources of support. Meanwhile Ross Ashby and Gotthard Gunther had retired and left the University. Finally in 1975 Heinz retired and moved to California."* Stuart A. Umpleby, Heinz Von Foerster, A Second Order Cybernetician, in: Cybernetics Forum, Vol IX, Fall 1979, N. 3, p. 5/6
24. BCL, The Complete Publication of the Biological Computer Laboratory, (eds. Wilson, von Foerster), Illinois Blueprint Corp., Peoria, Ill 61603, 1976
25. personal remark: I gave him a beautiful book about water skiing from the American thrift shop Berlin.
26. Gotthard Günther, Lebenslinien der Subjektivität, Kybernetische Reflexionen, CD, c+p 2000 suppose Köln

um but much more a radical new step in the understanding and transformation of the world and human nature in a trans-terrestrial world game.[27]

Computation and Metaphysics today

Questions of cracking identity in formal logical and computing systems are finally recognized now by leading computer scientists.

"Real-world computer systems involve extraordinarily complex issues of identity. (…)
Dealing with such identity questions is a recalcitrant issue that comes up in every corner of computing, from such relatively simple cases as Lisp's distinction between eq and equal to the (in general) undecidable question of whether two procedures compute the same function.
The aim of the Computational Ontology project *is to focus on identity as a technical problem in its own right, and to develop a calculus of generalized object identity, one in which identity -- the question of whether two entities are the same or different -- is taken to be a dynamic and contextual matter of perspective, rather than a static or permanent fact about intrinsic structure. "*[28] Brian Cantwell Smith

27. Gotthard Günther, Beiträge zu einer operationsfähigen Dialektik, Bd. I-III, Felix Meiner Verlag, 1976 ff.
28. Brian Cantwell Smith, SMITH-bio.html, 1999 , cf. B.C. Smith, On the Origin of Objects, MIT Press , 1996

Exploiting Parallelism in PCL-Systems

1 Polycontextural Strategy towards the Challenge of Parallelism

"Cooperation is expensive, yet it is the only way to get large tasks done quickly."
Peyton Jones, 409

Motivations

Living systems

Living systems are not based on expensive exploitation of nature. They are naturally non-expensive but highly complex. Complexity in contrast to complication (=nowadays complexity) is a strategy of nature that is not repeated in science. Science behaves quite non-natural in thinking and replicating natural systems.

Robert Rosen has introduced as a first step into a more natural science of the natural, the distinction between simple and complex systems.

Today's reality of computing isn't mirrored properly in the framework of mono-contextural concepts, models and methods. To realize parallelism, it would be much more strait forward to understand that parallelism is a quite special case of *interactivity* between more or less autonomous systems.

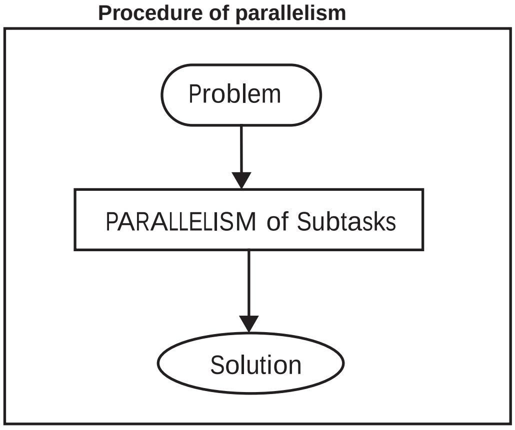
Interactivity is well understood in the framework of poly-contextural systems. Therefore I propose that parallelism is a special case of polycontexturality.

Parallelism comes in computer science in many incarnations. In this study I will start with a quite vague notion of parallelism and I will restrict this notion mainly to software concepts and programming languages. Standard examples for functional and logical programming are introduced for the purpose of deconstruction and introduction of a new understanding of parallelism as guided by poly-contextural strategies. This new understanding of parallelism as interaction of autonomous systems is not in conflict with the well known approaches to handle parallelity. In contrary, the poly-contextural approach is seen as a complexion of complementary strategies enriching the field of parallelisms.

1.1 Parallelism in hierarchies

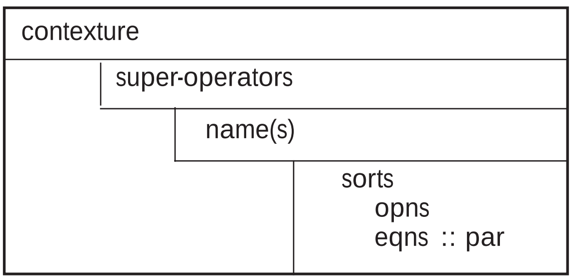
Parallelism in classical systems is obviously defined by its operations and equations. The operations and functions can have to property of being parallelized. The head of the basic module, or the abstract algebra in which this functions are defined is not involved in the procedure of parallelism. Neither the main function which heads the parallelized child functions. The head of the algebra and the main function is hierarchizing the construct of parallelism. In this sense parallelism occurs only secondary and involved in a hierarchy of the whole structure. This approach is reasonable because parallelism is involved here in solving a well-defined problem and delivering a well-defined solution, and parallelism is considered only as a tool or method to solve the problem more economically than in a linear setting.

Diagramm 39 **Procedure of parallelism**



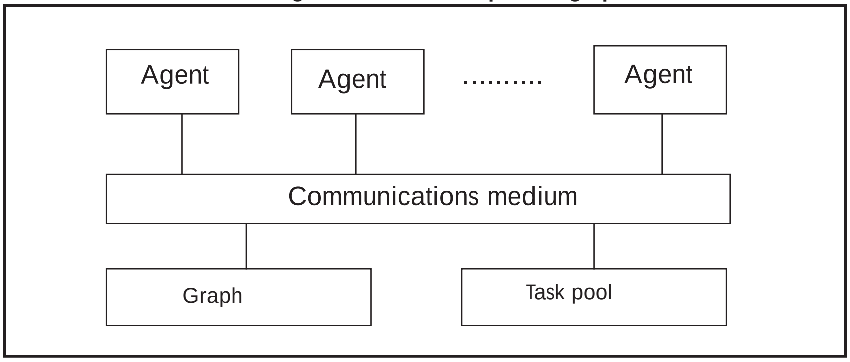
It seems to be of no value to think about a distribution of the problem and the solution statement during the procedure of parallelism. The main task and its solution is unique and a distribution of it would only be a repetition of the same problem at different places. In other words, parallelism in combinatory logic based programming languages is limited to the parallelism of the strict combinators. The non-strict combinators, like S, K, I are not involved in this type of parallelisms.

Diagramm 40 **Basic Modul e**



For CL, the basic module is reduced to: *contexture* is 1, *super-operator* is ID, and *name* is Combinatory Logic, *sorts* are one, the entities E, with constants K, S, and *opn* is *, and *eqns* are "=". And parallelism is concerned only in respect of operations and equations. In contrast, poly-contextural parallelism is distributing the whole concept of abstract algebra with its head and body over different places.

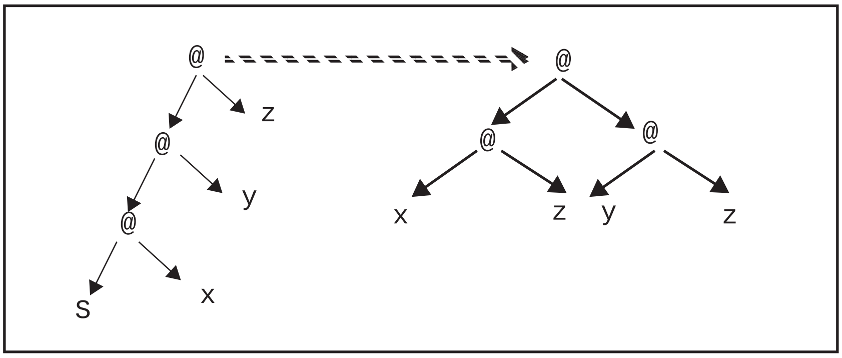
Diagramm 41 **Logical structure of a parallel graph reduction machine**



"A task *is executed by an agent. Typically an agent will be implemented by a physical processor.* Agents *are concrete pieces of hardware (we can point to one!), whereas a task is a virtual object (a piece of work to be done). An agent is employed if it is executing a task. An unemployed agent will look for a task to execute in the* task pool *which contains all the tasks awaiting execution."*

"Synchronization between tasks is mediated entirely through the graph*, so that the tasks do not communicate directly with each other at all."* Peyton, p. 414

Obviously the reduction of the S commbinator, Sxyz = (xz) (yz), gives the key for parallelsim in classical systems. The graph reduction of Sxyz is shown in the diagram.



Parallelity in CL-based programming languages is at first depending on the combinator S. The graph gives the logical structure of the formula.

Is there any parallelity involved with the other main combinator, K? Obviously not: Kxy = x is reducing (xy) to (x). This process doesn't show any possibility of parallelism.

Remark. From a PCL point of view, which is dealing with complexities, and where terms are complex and in some sense ambigue, the reduction opens up the question, to which x of the other, but same x´s of the complexion are we reducing? The intra-contextural understanding of the rule Kxy = x asks for an additional identity relation

ID, which rules the process of staying in the contexture and not leaving it. In analogy, K can reduce at once into different contextures, $K^{(3)}xy = (x, x, x)$ depending on the super-operator BIF. Every action or transition of a system is involved into the question of a contextural staying / leaving-decision.

Parallel Graph Reduction

"We have seen that functional languages can form a basis for parallel programming.
(i) Graph reduction is an inherently *parallel* activity. At any moment the graph may contain a number of redexes and it is very natural to reduce the simultaneously.
(ii) Graph reduction is an inherently *distributed* activity. A reduction is a (topologically) local transformation of the graph, and no shared bottleneck (such as an environment) need be consulted to perform a reduction.
(iii) All communications mediated trough the graph. This gives a very simple model of the way in which concurrent activities cooperate, and it is a model in which we have considerable confidence (because it is the same as our sequential implementations!)
(iv) The entire state of the computation at any moment is well defined – it is the current state of the graph.
Graph reduction gives us a *rocked-solid model of parallel computation* which can underpin the complexities of a parallel machine." Peyton Jones, p. 413

Also functional programming languages are not inherently sequential as conventional imperative languages but in principle parallel, in order to produce interesting results the program must contain *algorithmic parallelism*.

This inherent parallelism of functional languages offers to program parallelism with the means of the language itself without the need of adding new operators of parallelism.

This concept is widely known and has been explicitly studied by S.L. Peyton Jones, Kevin Hammond and Hans-Wolfgang Loidl.

Problems of resources: sparking, blocking, strategies

In living systems redundancy is inherent.

In cloned systems tasks don´t have to block each other.

Problems of granularity.

Parallel HASKELL

To exploit algorithmic parallelism in mono-contextural parallel functional programming languages like parallel HASKELL two main operators are additionally introduced: the operators *par* and *seq*.

Simulating parallelism in the framework of poly-contextural systems has to deal with this two operators.

Modeling: seq, par ==> diss

The parallel operators par and seq in GranSim

Consumer/producer

For example in the expression

```
let
  squares = [ i^2 | i <-– [1..n] ]
in
squares `par` sum squares
```

the list of squares will be produced by one thread (the *producer*) and the result sum will be computed by another thread (the *consumer*). Note that this producer-consumer structure of the algorithm is solely determined by the data-dependencies in the program. All the programmer has to do is to annotate a named subexpression (in this example squares), which should be computed in parallel. Thread placement, communication and whether the thread should be created at all are up to the runtime-system.

Algorithmic parallelism of Fibonacci

```
parfib :: Int –-> Int
parfib 0 = 1
parfib 1 = 1
parfib n = nf2 `par` (nf1 `par` (nf1+nf2+1))
          where nf1 = parfib (n-1)
                nf2 = parfib (n-2)
```

The drawback of this program is the blocking of the main task on the two created child-tasks. Only when both child tasks have returned a result, the main task can continue. It is more efficient to have one of the recursive calls computed by the main task and to *spark* only one parallel process for the other recursive call. In order to guarantee that the main expression is evaluated in the right order (i.e. without blocking the main task on the child task) the *seq* annotation is used:

```
parfib :: Int –-> Int
parfib 0 = 1
parfib 1 = 1
parfib n = nf2 `par` (nf1 `seq` (nf1+nf2+1))
          where nf1 = parfib (n-1)
                nf2 = parfib (n-2)
```

The operational reading of this function is as follows: First spark a parallel process for computing the expression parfib (n-2). Then evaluate the expression parfib (n-1). Finally, evaluate the main expression. If the parallel process has not finished by this time the main task will be blocked on nf2. Otherwise it will just pick up the result.

http://www.dcs.gla.ac.uk/fp/software/gransim/user_2.html#SEC6

1.2 Parallelism in Heterarchies

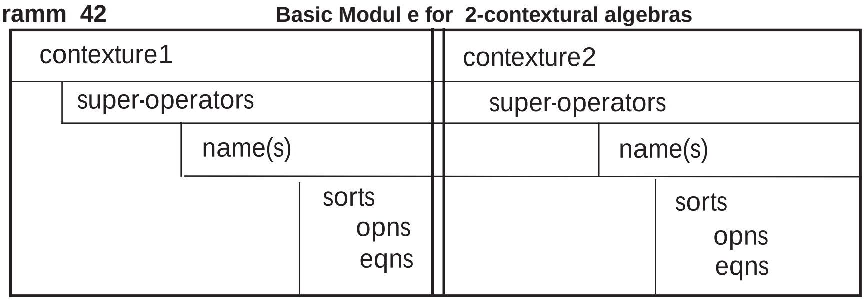
Parallelism in polycontextural systems is not hierarchical but heterarchical. Heterarchies in PCL systems are strict, there is no summon bonum, no main head or main task which could have the power to subsume the disseminated systems under a common general system. Such a concept of heterarchy is not involved in any reduction to a meta-system of what ever generality. In other words, a general system which would reflect as a mediating system the behavior of two contextures would be itself a contexture like the observed contextures.

The difference would depend only on their different functionality or roles but not on their structure. The meta-system would simply be another contextures together with the observed contextures. This result is a consequence of the proemiality of dissemination as opposed to the abstracting and homogenizing power of morphisms.

Parallelism in polycontextural systems has at least two main aspects. One is the autonomous parallelism of the systems involved. Because polycontexturality consists of a mediation of distributed basic systems parallelism is not only inherent inside of the system but fundamental in the sense that the architectonics of the system as an heterarchy is parallel.

The second aspect is the trans-contextural interactivity between the systems involved. The first can be seen as a radicalized form of classical parallelity changing from the strict to the non-strict functionality, the second is unknown in classical systems and could have only a vague analogy in message passing between tasks.

Diagramm 42 **Basic Modul e for 2-contextural algebras**



Because poly-contexturality, in this study, consists of a dissemination of combinatory logics, at each contextural occurrence of such a combinatory logic there is obviously also an occurrence of the S combinator responsible for intra-contextural parallelism. Therefore, it has to be distinguished between the architectural parallelity of the whole system and the intra-contextural parallelities, types of parallelism, defined by the S combinator. This distinction of two types of distributed parallelism, inter- and trans-contextural, opens up a great flexibility of modelling parallelism in PCL systems.

In following the S-parallelism with its strict separation of parallel procedures we can speak of a similar strict disjunct parallelism of distributed autonomous functions for each contexture. That is, this strict parallelism is defined without the trans-contextural super-operator BIF, consisting only of intra-contextural operations. If we exclude any permutation PERM and reduction RED, an absolute strict separated parallelism will be introduced. For logical operations this can even be strengthened by the restriction to strictly mono-form functions, like (and,and,and), (or,or,or) etc.

1.2.1 Interaction model of consumption and production

cf. Mahler, Disseminatorik

Karl Marx , Ware-Geld-Chiasm

Example: consume list *par* produce list

list as produced and list as consumed

list as list: list as neither produced nor consumed and as both at once.

Production and consumption are understood as interacting procedures. They are building a system of mutual interaction. If we are concentrating only on their common data then we are running into ontological or semiotic problems of their identity. Because in a classical setting we cannot consume and produce at the same time, that is at once, our data under consideration. Data-sharing under the principle of identity easily produces contradictions.

Production and consumption

Both procedures are per se, at least in a classical setting, well-defined and independent from each other.

Together their functionality is contrary, they are as notions opposites.

Neither-nor

The objects they are producing and consuming as such are neither produced nor consumed, they can be involved into this interaction but there is no necessity for there existence to be involved in this special game. Therefore they are localized as patterns in the system of kenogrammatics.

Both at once

To produce and to consume the same object at once is obviously violating the principle of identity because both processes, production and consumption, are contrary procedures related to common objects. Working together at once in an interaction they are localized as behavioral patterns in different contextures. In this distributed functionality they are independent of each other. Referring at once to the same object, as an object of consumption and an object of production, their common object is over-determined and ambiguous.

Metaphor *basket*

Why can we not at once *empty* the basket, *filling* it, *check* and *administrate* its content, *change* its seize, etc.?

1.3 Hierarchical parallelisms in Heterarchies

One of the main task of the proposed approach of introducing polycontextural systems is to model classical parallelism in a polycontextural framework. This is the strategy of applying complex methods to non-complex problems.

What are the intended merits and advantages of this approach?

To be ruled by a common head means that the control structure, the logic and all basic non-strict operators are in common and cannot be used or be involved freely. Also classical parallel processes can run concurrently without sharing special data and methods involving elaborated communications, what they are sharing anyway is their common data type, their common logic and arithmetics as superposed in their common head. For that reason, a complicated strategy of distributing the access to the computational resources is needed. Because the abstract poly-contextural or polylogic machine as such is parallel in itself it is natural and not expensive to model parallel functions on it. The costs are somewhere else, in the decision to give up the hegemony of classical logocentric reasoning and to develop on a new level of abstraction and the thematization of a system of polycontextural computation.

Wordings

“I will deal with the problem but I will solve only some parts of it. You can deal with the other parts of the whole problem.”

This does not only mean that an agent is dealing with the sparked parallel task but that he also knows the whole problem, that is, he accepts the head of the problem too. This is automatically given in the classical situation, because the tasks are hierarchically organized by their common head. But the isolated parallel tasks don´t know anything about the main problem, they are reduced to their parts of the whole problem. Not only the isolated parallel task have to be distributed over different contextures but also their head and with it their main task. Also the whole problem is distributed it doesn’t mean that in each contexture the whole problem has to be solved. In accordance to the modeling of the parallelity each contexture will be concerned only with the parallel task which it has accepted to deal with. This decision can be changed but this is not in the focus now. The distribution of parallel tasks over different contextures involves a process of negotiation and agreement which will not be modeled in our examples.

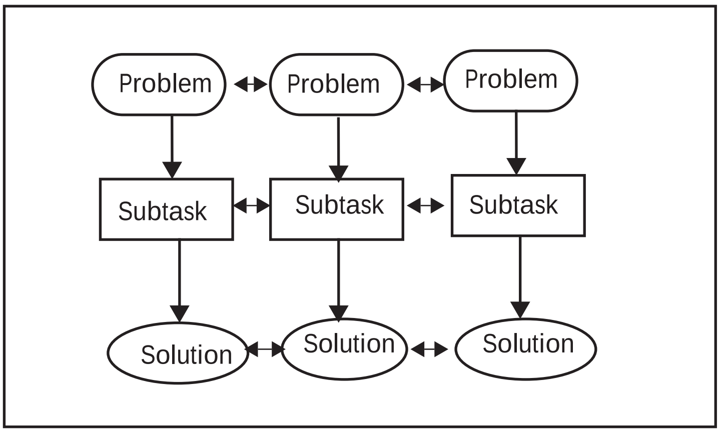
Towards a strategic metaphor

To solve a problem we have to clone it into its multitudes. This could give a hint to a metaphor of *poly-graph reduction*.

A direct step to model the classical type of parallelism in the polycontextural framework I propose to map the intra-contextural functional nodes @ of the hierarchical graph into the mediating poly-contextural operators § of the heterarchical graph. With this mapping we are distributing the hierarchically organized parallel tasks of the source to the heterarchically organized tasks of the polycontextural target system. This further step from the hierarchical to the heterarchical graph reduction, could be called *poly-graph reduction.*

In the same sense as the hierarchical organization of the tasks is ruled by the graph, the heterarchical organization rules the polycontexturally distributed tasks. As the hierarchical graph of parallelism guarantees the functioning of the computation, the heterarchy of the polycontextural graph is guaranteed by the operator of mediation § of the distributed tasks. Both, distribution and mediation, are the characteristics of dissemination. Thus, hierarchical parallelism is mapped into disseminated parallelism.

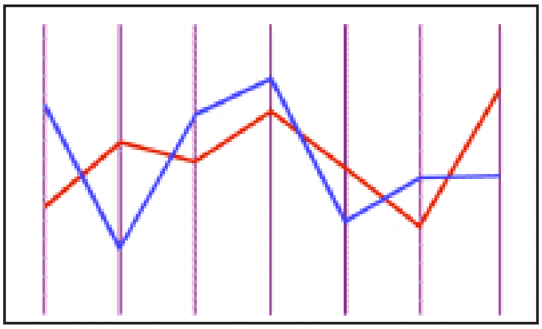
This picture of a mapping from a hierarchy to a heterarchy has two strategic functions, it gives us a *visualization* of the idea, that is a metaphor of our vision, and second a hint to a strict *formal realization* of the idea of disseminating intra-contextural parallelism to trans-contextural parallelism.



Parallel coordinates

Alfred Inselberg: PARALAX

1985 Alfred Inselberg began using parallel coordinates for representing highly multidimensional data sets. This remarkable achievement given its simplicity has produced, together with interaction techniques, some interesting and intuitive tools, like City'Oscope for the selection and finding of items in multidimensional spaces.



Parallel coordinates. They represent multidimensional data. Each axis corresponds to one dimension (one variable). For example, when evaluating automobiles they could be the volume of the cylinders, number of doors, the consumption of fuel per mile, etc. The axis are placed in parallel and the line that joins the values corresponding to a particular car model represent one "point" in this n-dimensional space.

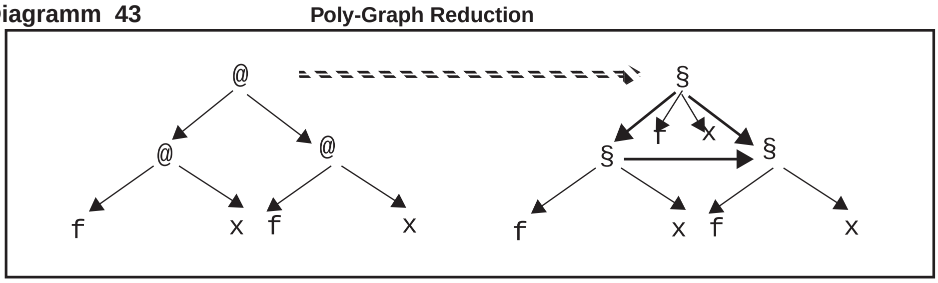
http://www.infovis.net/E-zine/2003/num_112.htm

Comment

From a logical and ontological point of view each dimension of the multidimensional data structure corresponds in itself to a two-valued logic.

1.3.1 A very first step of modeling poly-contextural parallelism

Diagramm 43 **Poly-Graph Reduction**

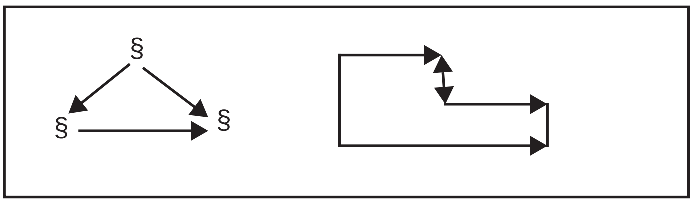


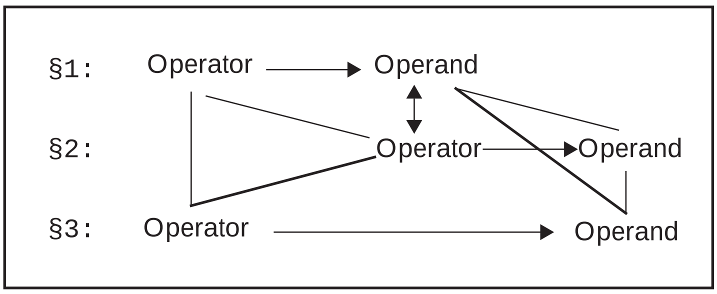
$$\text{Sffx} =_{par} (fx) @ (fx) \quad ===> \quad \text{Sffx} ===_{PAR} (fx) \S (fx) \S (fx)$$

Poly-graph reduction: Operator@ ==> Operator§

The @-structure is hierarchical, and well known. The §-structure is heterarchical, and has to be introduced. It is the structure of distributed and mediated, that is, disseminated systems as developed before.

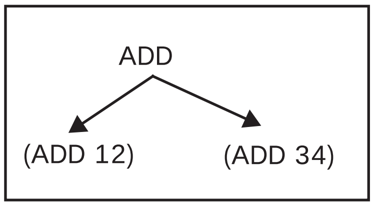
Poly-graph reduction is modeling intra-contextural parallelism of a single contexture, mono-contexture, into individual, that is elementary contextures, of a poly-contextural compound. What is inside a contexture, what is part of a contexture becomes itself a contexture in a poly-contextural system. At least, this is the main strategy without mentioning the transformations which happens to these parts.





As an introductory example lets take the formula x = (ADD(ADD 12)(ADD 34)) as well studied in Mahler, p. 143.

The tree of x is obvious.

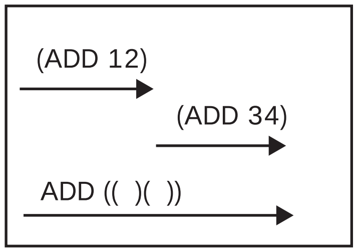


Task : ADD((ADD 12)(ADD 34))

Also this example shows algorithmic parallelism it is modeled in a framework of mono-contextural combinatory logic. Parallelism is restricted to strict operations.

This task is easily distributed over 3 contextures and its short cut notation maybe:

ADD^3 $((ADD^1)(ADD^2))$



In reality, 3 different number systems are involved in this model. Also they are not identical they are the same. And the number system of contexture3 takes the role of representing the resulting arithmetics, perhaps forgetting the genealogy of its components.

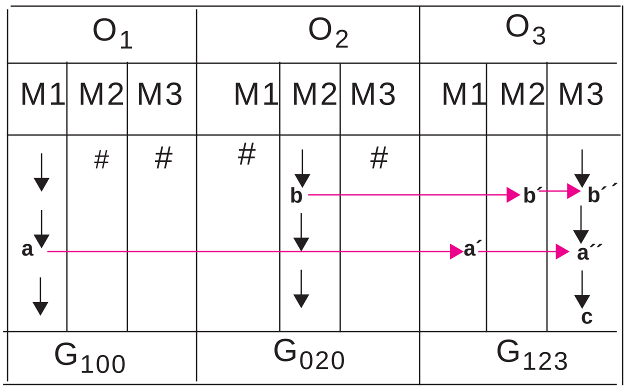
As in the intra-contextural parallel modeling of x the child tasks can be executed in parallel without any delay on a multi-processor machine. The main task has to wait in both cases of modeling under consideration. But there is an important difference even in this elementary example of modeling parallelism. The main task in the poly-contextural model is in reality not waiting the results of the distributed child tasks. It is an autonomous ADD-processor, it may add what ever is to add at the time. That this processor takes the role as the executor of a main task is secondary. If he gets his values from the neighbor tasks which represents the child tasks he will operate on them. Therefore this process is not involved in a waiting mode at all. Nevertheless there is a structural dependency between the child and the main task in this example, the main task can only be executed if the results of the child tasks are accessible to the main task.

As in the classical case the graph takes the organization of the tasks. The classical graph is hierarchic, the poly-contextural graph is heterarchical. In the hierarchical case the organization of the work seems to be easy because the main task simply has to wait for its results. But this happens on the costs of efficiency. Because the processor is identified with its role in the graph there is no autonomy to process something different, therefore, in contrast to the heterarchical case, it has to wait.

The price of the autonomy of the processors in the PCL model is its need of reflection. The processor has to decide to take the job if it is free and the neighbor processors have delivered their results. The results of the neighbor systems is not put directly into the scope of the main processor but to the reflectional space of the neighbor systems inside the main system. As the results of the neighbors the main agent can accept and take this results for further calculation. In the same sense the processors which play the roles of the child tasks are free to continue other jobs using or not the results they send to the main processor.

All that seems to be quite complicated and dangerous compared to the hierarchical model which lacks any freedom, and is therefore very stable, but with its own costs of identifying different roles which are separable in the heterarchical case.

This example shows again the purely functional definition of poly-contextural operations. They are not so much hetero-referentially oriented to their arguments, values and objects but more to their own functionality and to their place and locality inside the whole scenario of poly-contexturality.



In this example **a´** represents to the machine M3 at the place O_3 the value of processor M1which delivered the value **a** for itself in its own space.

The value or object **a´** is semiotically not identical with the object **a** but it is the same, that is, it is a duplication of **a**. This duplication of **a** is not well understood in terms of data processing. It is a transition from one contexture to another one. Such operations are part of the poly-contextural framework independently of the systems which are localized in it. Therefore the proposed model is nevertheless not a model of data-sharing or similar.

The mirrored objects in contexture3 of contexture1 and contexture2 are not identical to the objects **a´´** and **b´´** in use by the main task of machine3, they are, again, only the same, that is, they are objects used as objects for computation by machine3 as they are delivered by the other machines. These objects are having 3 different appearances in respect to their 3 different roles in the whole game. All these roles are played by the objects more or less simultaneously. As a final result of this interaction, machine3 is delivering the result **c**.

The metaphor of cloning in contrast to the metaphor of transport and transfer of data maybe helpful again.

Implementing the mirroring metaphor

Following the steps I developed above it should be evident that the mirrored objects are not stored in some memory outside the logical construction. This process of mirroring belongs entirely to the logical construction of the interaction.

To give this construction a more physical metaphor we can speak about an implementation of the calculation and the reflections on a stack machine. This means that not only the calculations are realized by their stack but also the positioning of the results at other places are realized by new separate reflectional stacks, stacks inside the polycontextural stack machine which are simultaneously dealing with the mirrored data of the neighbor systems. Each intra-contextural stack has simultaneously a reflectional stack of each of its neighbor systems.

1.3.2 A further step of modeling poly-contextural parallelism

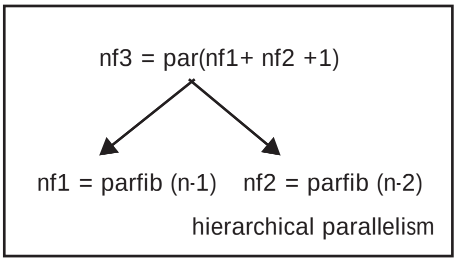
The idea is to simulate classical parallelism by means of polycontextural methods.

To simulate parallelism in classical systems we don't need transcontextural operators, it is sufficient to use monoform or polyform intra-contextural operators. These operators are defined by the restricted set of the super-operators {ID, RED, PERM}.

The Fibonacci function in the example is a function defined by double recursion. This function is primitive recursive.

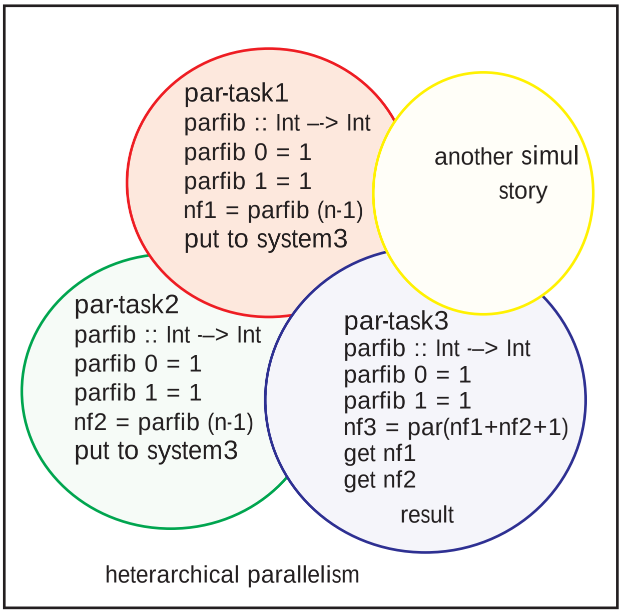
Classical parallel formulation:

```
parfib :: Int –> Int
parfib 0 = 1
parfib 1 = 1
parfib n = nf2 `par` (nf1 `seq` (nf1+nf2+1))
        where nf1 = parfib (n-1)
              nf2 = parfib (n-2)
```



Poly-contextural distributed formulation:

```
par-task1
  parfib :: Int –> Int
  parfib 0 = 1
  parfib 1 = 1
  nf1 = parfib (n-1)
  put to system3
                 .simul.
par-task2
  parfib :: Int –> Int
  parfib 0 = 1
  parfib 1 = 1
  nf1 = parfib (n-2)
  put to system3
                 .simul.
par-task3=main task
  parfib :: Int –> Int
  parfib 0 = 1
  parfib 1 = 1
  nf3= par(nf1+nf2+1)
  get nf1
  get nf2
result
```



Contexture3 represents the main contexture with the full instruction set. Contexture1 and contexture2 are calculating the two independent parallel tasks nf1 and nf2. The operation *seq* has a representation in the transaction (put / get) of the results of nf1 and nf2 to nf3. Because the Fibonacci functions "fib" are distributed over 3 different contextures which are architectonically parallel, it is not necessary to repeat the operator “par” in the distributed functions nf1, nf2 and nf3.

contexture(3)
contexture1
super-operator: ID
name: arithmetics fibonacci
sorts: Integer
operations: nf
equations:

Main scheme for PCL-parallel Fibonacci

$parfib^{(3)}: Int^{(3)} \longrightarrow Int^{(3)}$

$parfib^{(3)}$ (0, 0, 0) === (1, 1, 1)

$parfib^{(3)}$ (1, 1, 1) === (1, 1, 1)

$nf^{(3)}{}_3$=== $par^{(3)}(nf^{(3)}{}_1$+++ $nf^{(3)}{}_2$+++ (1,1,1))

get $nf^1{}_1$ for $nf^3{}_3$

get $nf^2{}_2$ for $nf^3{}_3$

Reflectionality in parallel processing.

Instead of the function get in “get nf2 for nf3” we could use “mirror”. Mirror the value of nf1 and nf2 at the place of nf3.

Comments

At a first glance such a mapping seems to be rather trivial or even meaningless. But we shouldn’t forget that the structure of the framework into which we are embedding these different arithmetical functions is ruling their locality and the possibilities of their interaction. In another wording we can say, that the distributed arithmetics of the polycontextural framework are autonomous in their status and are receiving some jobs, like to calculate parts of a Fibonacci function from another contexture and are delivering their results back to this arithmetical contexture.

What are the advantages of this modeling parallelism in polycontextural systems? Even if it seems that the difference is nearly invisible both, the hierarchical and the heterarchical parallelism, are fundamentally different in their conception and operativity.

In both models the calculation is decomposed into 3 steps realized, say by 3 processors. But the difference in the mode of decomposition is crucial, one is algorithmically the other is architectonically. Therefore their parallelism is set on different operational levels. Also this difference is sufficiently formally introduced to make this difference more explicit and clear we should introduce a new more complex example. The difference between the arithmetical and the architectonic parallelism has its influence on the way the parts are communicating together, how the child tasks are delivering their results to the main task. Obviously the architectonic model is incorporating in its basic logical and arithmetical operations the mechanisms of communication, the algorithms model has to introduce special non-logical and non-arithmetical operations to fulfill communication. Communication in this model seems not to be guarantied by the graph of the function alone. But the connection graph in the hierarchical model rules only the conceptual dependencies of the parts and gives no mechanism for the arithmetics of the connection of the parts. In contrast, the graph in the heterarchical model represents the rules of the mediation of the different contextures to a complex polycontextural system and represents insofar the architectionical base of interaction and communication between the parts.

Because the Fibonacci example proposes to solve a problem and especially a hierarchical problem–main task, child tasks, result–the procedure is to bring together heterarchical components to a hierarchical solution. This is an interplay between hierarchical and heterarchical functionality of parts of a complex system.

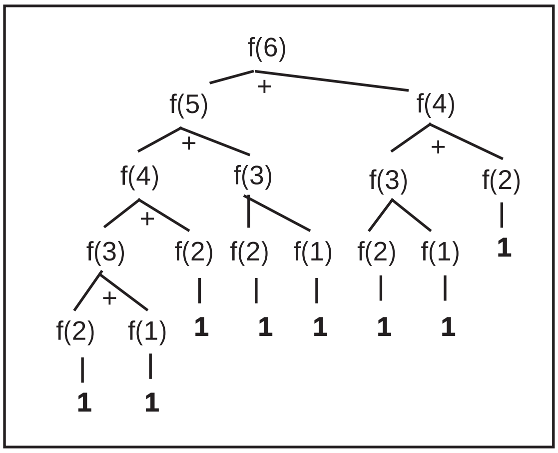
get / put aspect as a reflectional mechanism:

At each contexture the whole problem statement is mirrored. The other programs, machines, loci knows what is needed and can deliver it if they are ready with their own calcultation. This is different to a retrieval system like in Prolog where results can be retrieved from a common data base. This method needs for itself a store and search mechanism which has to be managed in favor to the main task

1.3.3 A more explicite modeling of the Fibonacci example

f(5) par f(4)
―――――――
f(6)

(f(4) par f(3)) par (f(3) par f(2))
――――――――――――――――
f(5) par f(4)
―――――――
f(6)



Usually we would reduce any redundancy in the program by means of storing and retrieving mechanism. But this strategy relays on other mechanism which are used to solve the task. Here the Fibonacci example is taken to show a new modelling which depends only on the formula and its logic, no special tricks are involved. Redundancy here is only a part of the example and not of the concept.

The brackets in the scheme are not necessary because of the parallelism of the tasks.

Trees mapped into binomials.

(f(4) et f(3)) par f(3) par f(2)
――――――――――――――――
f(5) par f(4)
―――――――
f(6)

$S_1S_2S_4$
S_3S_5
S_6

An even cleaner mapping would be to model the arithmetical subtasks only into disjunct logical subsystems, like S1, S4, S7, etc.

1.3.4 Putting the model into a tabular design

In a tabular design of the distributed parallelism it will be visible how the procedure of the parallel computation is working. In the quasi-algebraic modeling we are concentrated mainly with the algebraic structure of the proposed parallelism and not with its processuality.

Because of the constitutional or architectonic parallelism of PCL systems it is adequate to speak of parallelism even in situation where we don't have any algorithmic parallelism. Even the arithmetical successor function which is much too elementary to have any intrinsic structure to give place for parallelism has in poly-contextural systems qua distributed function its parallelism. Poly-arithmetics are institutional parallel.

Two actors are processing independently each a function, one in system1 and the other in system2, the main task is processed by the actor in system3.

Also a system can know, that is, mirror the whole problem in its contexture, an actor can be seen as a processor which processes a function regardless where it is from. The actor has not to be focused on the problem in full.

Zig-zagging parallelism

Mapping classical parallelism onto polycontextural systems doesn't deny the possibility of developing inside a contexture again classical parallelism based on intra-contextural S-operators. These distributed intra-contextural parallelities themselves can now be handed over to new contextures for trans-contextural parallel computation. There is no need for an explosion of contextures but also no necessity for an encapsulation of rapidly growing internal parallelisms, they can be exported if necessary to other contextures.

Exploding parallelisms

In the hierarchical model each formula containing a S-combinator leading to parallelism can be treated inside of its subformulas. Therefore the danger of an explosion of intra-contextural parallelism emerges.

Exploding parallelism may produce unreasonable fine granularity.

Sxyz ==> (xz)(yz), for x=Sabc and z=Sdef

Reductions

One posibility to reduce this explosion is to model it into poly-contexturality.

1.4 Complexity measures of calculations

LIPS for different Prolog systems

2 Parallelism in Polycontextural Logic

Additionally to the well known OR- and AND-parallelism, polylogical systems offer two main extensions to the *logical* modeling and implementation of parallelism. First the distribution of the classical situation over several contextures and second, the trans-contextural distributions ruled by the different transjunctional operators. The distribution over several contextures corresponds to a concurrent parallelism where the different processes are independent but structured by the grid of distribution. The trans-contextural parallelism corresponds to a parallelism with logical interactions between different contextures. The logic of parallelism is to distinguish from parallelism in logic (viz. Kurfess, 1991) as it is developed in classical logic programming languages.

> “The tree corresponding to the search for a solution to a question seems open to various kinds of parallelism. The most obvious technique, called OR parallelism, allows processes to search disjunctive subtrees in parallel, reporting back to the parent node the result(s) of the search.
> The advantage of OR parallelism is that the searches are completely independent of each other and may execute concurrently (except that both may share access to a common data base storing facts and rules). The process performing the search of one subtree does not communicate with processes searching other subtrees.” Michael J. Quinn, 212, 1987

Prolog is based not only on its logic, used as an inference machine, but also on its semantics or ontology, realized as a data base. Therefore the process of parallelising has to deal with a deconstructive dis-weaving of the data base´s ontology.

2.1 Strategies towards a polycontextural parallelism in Prolog

Like in the case above, where the number systems had to be cloned, in the Prolog case, the data base has to be decomposed into disjunct parts. These separated conceptual parts, or conceptual subsystems, have to be distributed over different contextures in a mediated polycontexturality.

Additionally, the Prolog parallelism which is based mainly on OR- and AND-parallelism has to be mapped into distributed logics, that is, into a polylogical system.

The Prolog example allows to explain in more a plausible way the decomposition or cloning of the common universe of discourse, that is, the data base of facts, into different subsystems. And secondly it is easier to introduce parallelism based on polycontextural logic than on arithmetics and combinatory logics.

Polycontextural logic is not widely known but more accessible than combinatory poly-logic and poly-arithmetics, which I am just introducing. Additionally there exists since 1992 a working implementation of a tablex proof system of an interesting subsystem of polycontectural logics in ML, running on Unix systems like NeXT.

2.1.1 An intermediate step with Metapattern

As an intermediate step in the shift of conceptualization from a hierarchical to a heterarchical way of concept building it maybe helpful to use the strategy of metapattern (Wisse). Metapatterns are used as an new modeling strategy for complex informational systems. Metapatterns are not involved in changing the basic assumptions of programming languages or even their logic as with the PCL approach.

Metapatterns could be helpful to move the process of parallelisation from the OR- and AND-level, that is, from the logical level to the deeper level of the data base, with its facts and rules, shared by the classical parallelism.

> She can relax on a fixed object orientation because — the metapattern determines that — situation and object are relative concepts (Wisse 2001). A particular situation is also object in another, higher-level situation. Likewise, an object can act as situation in which another, lower-level object resides. Situation, then, is a recursive function of object and relationship. Wisse

Hierarchy or chiasm?

> It is this concept of situation that characteristically sets the metapattern apart from traditional object orientation (and provides it with advantages over OO; Wisse 2001). Compared to an object that (only) exists absolutely, an object believed to exist in a multitude a different situations can unambiguously be modeled – to be equiped – with corresponding behavioral multiplicity. Wisse 2001

> The radical conclusion from the orientation at situational behavior is that an object's identification is behaviorally meaningless. The modeler does not have to explicitly include something like an original signature in all her models. Essentially a privileged situation may implied. It serves the only purpose of guaranteeing sameness or, its equivalent, persistent identity across (other) situations. Being a situation in its own right, when included in a model it is represented by a seperate context. Made explicit or not, its role is to authenticate an object's identity in other situations by establishing the signature in other contexts.

Identity as a network of nodes

> Traditional object orientation assigns identity at the level of overall objects. Context orientation replaces this view of singular objects with that of plurality within the object; the object always neds a context to uniquely identify the relevant part of an overall object, which is what identifying nodes regulate. When behaviors are identical, no distinction between contexts is necessary.

2.2 Deconstruction of a typical PROLOG example

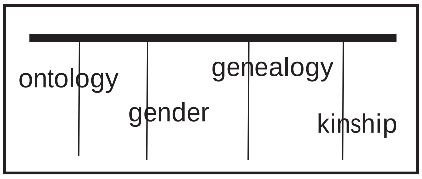
The classical prolog example to prove an “*aunt*”-relationship can be decomposed from its hierarchical ontology into different situations mapped into different contextures and visualized in the metapattern.

kinship: married / not-married, in-law, aunt

gender: male, female

genealogy: parent, sibling

ontology: different / not-different



It is also possible that there is some overdetermination because *parent* and *sibling* could also be part of *kinship.*

In Prolog all the facts belong to one ontology or to one semantic general domain or universe. All the rules are based on this mono-contextural ontology and on the corresponding logical operators AND and OR of the again, mono-contextural logic. Everything therefore is linearized and homogenized to a global or universal domain. This, if corresponding fairly with the real world situation is of great practicality and efficiency in both direction, in the case of the formal system, Prolog, and in the case of its data base.

But often, if not always, real world applications are much more complex than this. Even the fairly classical example is presupposing all sorts of facts which are not mentioned in the definition and which would belong to a different real world situation.

Instead of linearizing the above separated contextures *kinship, gender, genealogy, ontology* into one universal domain, for the example here represented by *kinship*, the polycontextural modeling is asking for an interweaving and mediating of these different contextures together to a complex poly-contexturality.

Compared to the original mono-contextural modeling this is involving much more complicated mechanisms than it is necessary in the classical case.

Reduction of complexity in knowledge acquisitions

Why should we model a simple situation with highly complex tools into a complex model if we can solve the problem with much simpler tools? Simply because the classical approach lacks any flexibility of modeling a complex world. The truth is, that the simple approach needs an enormous amount of highly complicated strategies to homogenize its domains to make it accessible for its formal languages.

To decompose the basic classical ontology into different disjunct domains is a well known procedure and should not be confused with the decomposition, or de-sedimentation of an ontology in the PCL case. In PCL the domains are not simply disjunct and embraced by the general ontology but interwoven in a complex mechanism of interactions based on their autonomous logics. This is similar to problem parallelism (parallel problem specification).

Reductions of computational complexity

What are the further advantages PCL decomposition strategy? Obviously, decomposition goes in parallel with strong reductions of complexity of the domain, that is the ontology, which is producing directly strong reductions of computational complexity. This goes together with the notion of architectonic parallelism which is in contrast to algorithmic parallelism.

2.2.1 Polylogical modeling of the metapattern

The metapattern approach has helped to dissolve the hierarchical conception of the "aunt"-relation into different aspects.

In Prolog, the aunt-relation is defined as follows:

ant(x,y):= female(x), sibling(x,z), parent(z,y).

additionally the rule for sibling is:

sibling(x,y):= parent(z,x), parent(z,y), (x/==y).

The aunt-function is fullfilled and is true, if all components which are connected by the conjunction *et* (AND) are true.

true(aunt(x,y) iff (true(female(x)) et true(sibling(x,z)) et true(parent(z,y)))

Metapattern distribute the AND (or: et) over different heterarchical places but gives no formalism to handle this distribution. Polylogics is also distributing these conjunctions but in transforming them at the same time into operators of mediation. Polylogics is shortly defined as a distribution and mediation of classical logics.

ant(x,y) := female(x) § sibling(x,z) § parent(z,y)

sibling(x,y):= parent(z,x) § parent(z,y) § (x/==y)

Therefore the polylogical truth-function is transformed to:

aunt(x,y) eTrue ==> aunt$^{(3)}$e(x,y) e (T_1,T_2,T_3)

The metapattern of parts of the formulas can be transformed into the diagram.

S1	female(x)			
S2		sibling(x,z)		
S3			parent(z,y)	
S4				aunt(x,y)

How to read the transformation?

In Prolog, each term as such has an identical meaning. If the variable x is denoted with “mary” and mary is female, then the relation or attribute female(mary) is true. Also the variables x, y, z,... are identical. Obviously no “x” will be read as an “y”; we don´t make a "x" for a "u".

In polylogic the situations are happily a little bit more flexible. The variables are flexible to occur as variables in different systems. The variable “x” can occur as the variable x in system S1, that is the variable x can occur *as* variable x1.

In the same sense the denotation “mary” can occur as female or as sibling or as parent or as something else. Mary *as* Mary, again something else, maybe a secret.

Our model suggest the following reading:

x as female: x1 and mary as female: mary1

x as sibling: x2 mary as sibling: mary2

z as sibling: z2 stuart as sibling: stuart2.

y as parent: y3 kathleen as parent: kathleen3

z as parent: z3 edward as parent: z3

The result: aunt(mary,kathleen).

x as aunt: x4 mary as aunt: mary4

y as -aunt: y4 kathleen as beeing in relation to her aunt: kathleen4

Also the simultaneity for "mary" of being female and sibling, which is ruled in the Prolog model by the conjunction “et”, is realized in the polylogical model, obviously by the mediation rule “§”.

This example is very simple because the elements of the partition are simple, there are no composed formulas included. Insofar there is no need to involve polycontextural negations, junctions and transjunctions. Only the operator of mediation "§" between distributed attributes and relations are involved.

Only if we freeze the scenario to a static ontological system all the flexibility of the as-function, not to confuse with the as-if-function, can boil down to the well known non-flexible structure. But to allow a flexible ontology with x as x1, as x2, etc. or mary as female, as sibling, etc. allows to change ontology and to be ready for new situations without starting the system from scratch. It is easy to freeze complexity, but there are no known rules how to make a frozen and dead systems alive. Maybe that’s the reason why artificial life is nevertheless so hard.

2.2.2 Prolog´s ontology

Prolog refers as it has to do as a programming language based on First Order Logic (FOL) on attributes, relations between attributes and inference rules etc. and not on behaviors and contexts.

To be a *parent* is classically an attribute of a person, described as a relation to other persons, in PCL this attribute becomes a *behavior*, maybe of a person, in a complex situation. To be parents is not necessary connected with the attribute to be *married*, to be a *sibling* has not to be restricted to have the same parents, to be married has not to involve different *gender*, and so on. And even that a person is different to another person, or that the person is identical to itself is not as natural as it seems to be. All these presumptions are reasonable, and are corresponding to possible real world models only if all the possible ambiguities and over-determinations are ruled out in favor to a very special model of kinship.

Take the example of the sportswomen who is excluded from making sport in a female team because she was before classified as a man. So the gender difference or even the sexual difference of masculine / feminine of a person is depending on contexts.

The solution to this situation of complexity is not so much to enlarge the given ontology and to introduce the new differences and attributes to cope with the new situation.

Because this strategy is based on the exact same ontological presuppositions and is therefore only repeating the old scenario again.

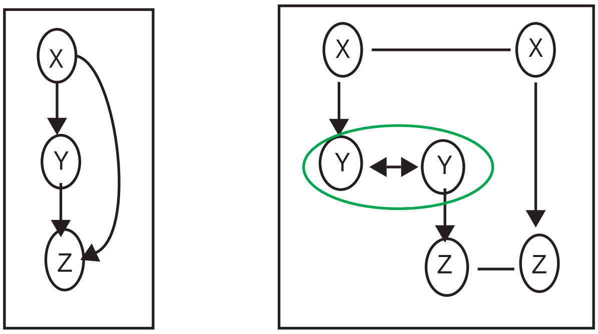
In the framework of PCL mechanism are offered for a great flexibility in interlocking and interweaving different points of view, situations, and modeling.

The decomposition of an universal domain into its different components is not only introducing a conceptual advantage for the process of modeling but also on a computational level a new form of parallelism is introduced.

The whole manoeuvre is quite similar to what I proposed as a proemial relation between sorts and universes in many-sorted first order logics.

2.2.3 The devil is in the detail

Polycontexturality is not starting somewhere in a complexity, it is virulent at the very beginning of the basic definition of relationships.



Y as child of X and Y as the father of Z has to be mediated, synchronized, realized. Only in a stable hierarchical ontology this relationship of Y as “child of” and “father of” is automatically connected. And therefore “father of father” can be equal to “grandfather” and realized by a conjunction of the two relations, father(X,Y) et father(Y, Z) eq grandfather(X, Z).

In a polycontextural setting this identity of Y, as child and as father, can not be presupposed but has to be established in a possible context. Y as child and Y as father has to be brought together in a way that the transitivity can hold. It is easily possible that the transitivity is broken for some reasons and that it has to be re-established. The reason why the transitivity can be broken lies in the poly-contextural assumption that a entity or a relation is not a simple identity but involved in a cluster or an intersection of a multitude of possible contextures. Only for restricted and regulated situations a complex situation can be reasonably reduced to a mono-contextural one in which transitivity holds unrestricted. Therefore, identity can not be presupposed it has to be realized from case to case.

Because of the relative autonomy of both relations in a complex kinship system, we can calculate and study them simultaneously, realizing some elementary parallelism. This is obviously not possible in a strict biological interpretation of the father-child-relation. There we have to accept the hierarchical dependencies of the relations. But again, we have to be aware that this is the case only because we restrict the setting to a mono-contextural case. In contrast, real world social relations are always highly complex.

Therefore we have two options, the mono- and the polycontextural. The advantage of the later one is flexibility, the advantage of the first one is stability. Both have there weakness, flexibility is risky and dangerous, stability is restricting and killing.

2.3 Prolog´s double parallelism dismantled in polylogical systems

As mentioned above, Prolog has additionally to its well known parallelism of OR- and AND-procedures, and some others, a new form of parallelism which is introduced by the process of deconstructing, dis-weaving, decomposing, de-sedimenting its basic ontology as presuposed in Prologs data base. This poly-contextural thematization of Prologs ontology goes together with the possibility to modell its classical parallelism into the architectonic parallelsim of polycontextural logic systems similar to the modeling of the graph parallelism of functional languages into the polycontextural architecture.

2.3.1 Polylogics

Each dimension, each level, each contexture of a complex object or system has its own ontology and its own logic. Furthermore, on a model-theoretic level, each contexture has its own theory of complete lattices in the sense of Garret Birkhoff which allows a very detailed analysis of the coneptual space of the objects belonging to each contextures. But lattices are not mapping the interactions between lattices of different contextures.

2.4 Polycontexturality proposed or produced

Instead of postulating polycontexturality as a new option it is possible to show the mechanism who to pass from a mono-contextural ontology to a poly-contextural one. This mechanism is described as a proemiality between logical sorts and their common logical universe. Sorts of a logic can be changed into universes of another polycontextural logic. And from a polycontextural logic the inverse procedure is possible, universes can change their role to sorts.

To postulate polycontexturality is legitimate because the mono-contexturality of classical logic and ontology is itself postulated and can not be proofed. Their is no proof which deceds between mono- and poly-contexturality inside the framework of classical logic and ontology.

It seems that Gunther is postulating polycontexturality, referring sometimes to reality, my emphasis on proemiality on the other hand is to produce a mechanism of introducing or generating polycontexturality.

The question is moved to the legitimation problem for the mechanism of proemiality. A positive answer is given by the fact of its hyperactivity: proemiality is working, viable, living.

2.5 Deconstruction Hierarchies in OOP frameworks

3 Tableaux Logics

Zurest die Regeln zur Herstellung der DNF

0) a) a1 & (a2 | a3) -> (a1 & a2) | (a1 & a3)

b) (a1 | a2) & a3 -> (a1 & a3) | (a2 & a3)

Die Regeln fuer die alpha Beziehung von Termen mit T-Anteilen

1) a) (t || ta) & (t' || ta') -> (t & t') || (ta &' ta')

b) t & (t' || ta') -> (t & t') || ta'

c) (t || ta) & t' -> (t & t') || ta

<t1..tn> &' <t1'..tn'> := <(t1 & t1')..(tn & tn')>

Die Regeln fuer die alpha Beziehung von Termen mit T-Anteilen

2) a) ([t] || ta) | ([t'] || ta') -> [t | t'] || (ta |' ta')

b) [t] | ([t'] || ta') -> [t | t'] || ta'

c) ([t] || ta) | [t'] -> [t | t'] || ta

<t1..tn> |' <t1'..tn'> := <(t1 | t1')..(tn | tn')>

Die Regel fuer verschachtelte ||-Terme

3) (t || ta) || ta' -> t || (ta &' ta')

Die Regeln zur propagierung der ||-freien Terme

4) a) t ist atomare signierte Formel

=> [t]

sonst expandiere sf unter Anwendung des zugehoerigen Tableaus.

b) t und t' sind ||-frei

=> [t] & [t'] -> [t & t']

c) t und t' sind ||-frei

=> [t] | [t'] -> [t | t']

Die Regel 1a) kann man auch weglassen, da sie sich aus 1b) und 1c) und 3) herleiten laesst. Dieses gilt nicht fuer die Regel 2a) !!

Easy OR- and AND-parallelism
Easy mix of OR- and AND-parallelism
Poly-And and Poly-Or Parallelism
Polycontexturality as a Form of Parallelism
Parallelism inside and between formal systems
Transjunctional Parallelism
Parallelism of the Proemial relationship
Parallelism in Kenogrammatics

Applying Poly-Parallelism to Parallelism

PolyGraph Reduction Machine

4 Why not Grid Computing?

Some fashions are coming and going. Grid computing is back again and hype.

I found a nice example which also gives a clear hint why we should celebrate grid computing. Mainly because of the ubiquity of computing devices.

http://www.spbsoftwarehouse.com/dev/articles/pocketcluster/images.html

Again, the main problem will be coordination and protection of the grid and the local systems. As long as interaction between computer systems is reduced to information processing there seems to be no big hope to solve these problems.

The PCL approach offers a concept and maybe a strategy which goes beyond information processing, the trans-contextural interaction, ruled by the trans-logical operators of transjunctions for solving the problems of interaction and separation.

Interactivity and contextural abstraction

The PCl approach offers at least two advantages.

First, the contextural abstraction offers a level of interaction beyond information or data processing, that is, the interactivity between systems is ruled by special contextural operators, the trans-contextural operators.

Second, the contextural abstraction is introducing the structural difference between local and global embeddedness. This contextural abstraction allows to draw a strict difference between a system and its computational environment. Each system has from a logical point of view its own internal logic. This logic is an abstraction of the whole polylogical structure of combined systems. With this abstraction the classical logico-arithmetical structure is separated from the interactional part of the full polylogic. Having established the difference between this local and global aspects of an interactive agent, the structural possibility is opened up to distinguish between data belonging to the system and data which are not yet accepted, which maybe new or antagonistic and hostile.

To be able to decide between self and non-self data on a structural and not only on a peripheral level offers the mechanism of self-protection.

Classical computing systems are principally lost in this situation because their logic doesn't offer operators to distinguish between self, that is local and non-self, say global interactions.

The question of embeddedness of interactive agents is tackled on a structural level. Each machine has its own logic which is not identical to the other machines but the same in the sense of likeness or similarity. From a conceptual point of view there is no difference in logic between different classical computing systems, they are ruled by the application of the identical logical system. Applied logical systems are structurally identical and subsumed to the abstract notion of logic per se.

Polylogical systems are not applications of an abstract logic but themselves realizations of different logics.

The big question arise, how can we implement polylogically distributed systems? How can the same logic be different to the logic of the other systems?

It is important to see that the operation of *contextural abstraction* enables a clear cut between the local logical operation and the global interactive logical operations inside the very kernel of a polylogical system.

The proposed contextural abstraction should not be confused by a possible contextual or context abstraction. Contexts are parts of contextures like sorts are parts of a universe (of discourse).

Minsky´s new Machine

1 Proemiality and Panalogy

1.1 Cognitive Systems and Panalogy Architectures

The DARPA label *Cognitive Systems* maybe interpreted as a interplay of cognitive and volitive aspects of a living system. Such an interplay was described by Gunther in his *Cybernetic Theory of Subjectivity* as the mechanism of a proemial relation.

This idea of a proemiality between structurally different systems can be brought to a more concrete level as an interplay of the four aspects of a living system „architectonics“, „reflectionality“, „interactivity“ and „positionality“. I choose these terms because they show a possible connection to existing work in the fields of AI, robotics, living systems, etc.

None of these aspects is prior to the other. They are simultaneously founding and generating each other. There is now need for a complex architectonic if there is no need for complex reflection, and so on.

The polycontextural approach to cognitive systems postulates that cognitive systems are from the very beginning involved in the interplay of (at least) these aspects of specifications. Cognitive systems don´t exist in the world isolated for themselves, and are starting from time to time to interact and to reflect, etc. In contrary, they simply don´t exist if they are not principally involved from the very beginning simultaneously in all these actions.

At the time there are some very interesting developments in AI, robotics and other branches, collected by terms like *"Cognitive Systems"* (DARPA), *"Architectures"* (Sloman, Minsky) and *"Emotion Machine"* (Minsky), *"Common Sense Interfaces"*, etc.

The main background idea and strategy seems to be to introduce multitudes against single monolitical concepts and methods. Slogans like "Multiple ways of thinking", "Diversity of ways of thinking", "Parallel ways of thinking", etc. The introduction of different agents like critics are part of the dissolution of monoliticity of classical modeling in AI.

Minsky calls one important case of multitudes "parallel analogy" or short *panalogy*.

Push Singh is on the way to write and implement in his Ph.D. dissertation *The Panalogy Architecture for Commonsense Computing*.

> *The Panalogy Principle:* If you 'understand' something in only one way then you scarcely understand it at all—because when something goes wrong, you'll have no place to go. But if you represent something in several ways, then when one of them fails you can switch to another. That way, you can turn things around in your mind to see them from different points of view —until you find one that works well for you now. And that's one of the things that 'thinking" means!

> We cannot expect much resourcefulness from a program that uses one single technique—because if that program works in only one way, then it will get stuck when that method fails. However, a program with multiple 'ways to think'—the way we'll describe in Chapter §7—could behave more like a person does: whenever you get frustrated enough, then you can switch to a different approach—perhaps through a change in emotional state.
>
> Minsky

Sloman´s Email

One aspect of the broader view is the way in which a growing interest in architectures and varieties of forms of representation replaces, or rather subsumes, the older emphasis on algorithms and data-structures.

By 'architecture' I don't mean what computer engineers used to mean: the low level organisation of a kind of hardware device, e.g. a Turing Machine architecture, or a VonNeumann architecture, or a Vax architecture. Rather the study of architecture includes the study of all sorts of ways of constructing complex functioning systems from many, possibly diverse, components. This includes software architectures, virtual machine architectures, hybrid architectures, etc.

I expect the study of architectures, especially layered virtual-machine architectures will continue to grow in importance. We may need entirely new kinds of mathematics for this.

From: Aaron Sloman (A.Sloman@cs.bham.ac.uk)

Date: May 26, 2003 23:22

Push Sings´s main questions

In order to explore how to build an architecture of diversity for commonsense computing, my thesis will explore these questions:

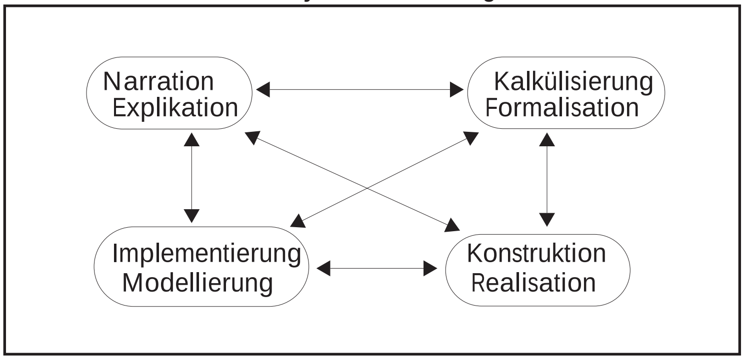
- How can we represent a “way of thinking”?
- How can we map types of problems to types of ways of thinking?
- How can we detect problems with the current way of thinking?
- How can we switch efficiently between ways of thinking?

1.2 Ways of thematicizing

In earlier papers I introduced some distictions to characterize how we are thematicizing our subject.

Explanation (Narration, Metaphors, Notions)
Formalization (Mathematics, Logics)
Implementation (Modeling, Computerimplementation)
Realization (Construction, real-world performance)

Diagramm 44 **Ways of thematicizing**



These categories of thematicizing are understood as purely functional and structural and not fixed in any sense. Therefore we can study the implementation of the explanation or the realization of the formalization, etc. In this sense even a poem can be thematicized by its power of formalization, its level of realization, etc.

From this point of view the project *"Cognitive Systems"* with its architectures are situated in the field of Explanation, see Minskys book "Emotion Machines", Implementation, Modeling, see SADE, CoGaff, SOAR and Realization in the domain of Modeling. This implies Formalization too. But here we observe a very classical situation without any attempts in the direction of the slogan "Multiple ways of thinking". The whole monolitical concept and apparatus of mathematical and logical thinking and reasoning is accepted, at least it is not a topic of the new panalogy program. The same happens to the realization of the model, it has to run on a classical computer, accepting the paradigm of algorithms as formalized e.g. in the Turing machine and the concept of information as formalized by Shannon.

This situation is surely not surprising, because it mirrors the current situation of technology and mathematics. Any denial of these presuppositions would sabotage the whole project of developing today on a reasonable base more complex systems.

Nevertheless, to discuss and surpass the limits of the formalization power of mathematics for the realization of artificial living systems was one of the aims at the Biological Computer Laboratory (BCL) in the early days of AI researches.

The only voice concerning mathematics in connection with the Grand Challenge Project I found in Sloman´s email. *"We may need entirely new kinds of mathematics for this."* But this statement can have itself a multitude of interpretations.

The open question remains, is the study of cognitive systems, more explicit, is informatics part of mathematics? Or is it the grand challenge of informatics to deliberate itself from the paradigm of mathematics? It seems that Margret Boden is thinking in this direction too.

2 Complementary Work?

From the point of view of strict foundational studies in this field, I try to realize some *complementary* aspects of the contemporary situation.

Therefore I have to focus mainly on the aspects of *formalization*. What are “Multiple ways of thinking” in logic and arithmetics? One actual answer to this question we can find in the growing approach of *Combining Logics* (fibring, labelling, weaving formal systems, Pfalzgraf, Gabbay). This trend is not yet recognized by the vision of panalogy. Mainly probably, because this work is not only very recent but also extremely technical and even mathematicians will have some problems to understand and to use it. Also, at least from my standpoint, this work is not radical enough at all. Because it is based on a monolitical kernel of classical logic. The diversity comes here to a stop at the bottom and ends in monolicity. On the other hand, it is based in its meta-language, category theory and multi-sorted logics, on a monolitic monster at the top.

That classical logics in all its forms are not enough for the study of cognitive systems maybe well known. Not only Kant and Hegel discovered it, but also Peter Wegner was criticizing the Japan Project from this point of view. Prolog, based on first order logic, is to weak to cope with interaction. But the common strategy of Hegel and Wegner is to avoid logic and to switch to a more speculative or empirical level of modeling, instead of transforming the paradigm of logic itself. There is no reason to believe that logic is something natural like the natural numbers of arithmetics and that it could not be changed as the naturality of the natural numbers can be de-mystified. This challenge is not accepted at all, the result is, again, some regression into non-formal thinking.

Because of my focus on foundational studies my realization of the category of explanation is worked out in a more philosophical sense, and the category of implementation too, is more foundational than empirical, that is, it is implementing new formalisms and programming languages with the help of today´s monolitical methods (existing programming languages) on monolitical machines.

> *"I first presented the idea that Turing machines cannot model interaction at the 1992 closing conference of the Japanese 5th generation computing project, showing that the project´s failure to reduce computation to logic was due not to lack of cleverness on the part of logic programming researchers, but to theoretical impossibility of such a reduction. The key argument is the inherent trade-off between logical completeness and commitment. Commitment choice to a course of action is inherently incomplete because commitment cuts off branches of the proof tree that might contain the solution, and commitment is therefore incompatible with complete exploration.”*
>
> Wegner, ECOOP ´99, p.1/2

As mentioned above, Wegners strategy to surpass this limiting situation is not to deliberate the paradigm of formality which is defining the very concept of logic and all the concrete logical systems, but some form of regression to empiricism.

> *“Logic can in principle be extended to interaction by allowing nonlogical symbols to be interactively modified (reinterpreted) during the process of inference, for example by updating a database of facts during the execution of logic programs. However, interactive discovery of facts negates the monotonic property that true facts always remains true.”*
>
> Wegner, p. 25

This strategy of extending logical systems by non-logical symbols for modeling interaction introduces into logic some non-logical elements of empiricism. For practical reasons this approach has its merits. Nevertheless, from a structural point of view of operativity and formality nothing has changed. Still the old logic is ruling the situation.

This strategy of extending the concept of pure classical logic is well know, at least by the work of mathematical linguists. I named this tendency of concretizing pure logic to more mundane tasks the “parametrization of logical constants”. Every element in a logical system which has some constant definition can by parametrized to a more dynamic notion. In a strict sense, all these extensions are conservative extensions of classical logic. At least, they are all based on a kernel of classical logic. But this kernel is taboo.

To empathize on this should make clear that this situation offers at least two options, one to accept classical logic and to parametrize it as far as possible and one who opts for a radical change of the kernel itself. This latter option doesn’t deny the reasonability of the first strategy. The difference should be recognized and also that there is no possibility to deny the reasonability of the second way of thinking. I have not to mention, that the first decision is the established way of thinking. The second way of thinking is closely connected with idea of polycontexturality and proemiality.

The complementary aspect of Minsky´s approach to the polycontextural approach is expressed by the statement

> *“We'll try to **design** (as opposed to **define**) machines that can do all those 'different things'." Minsky*

The question of definition is a logical one, the process of design belongs to the domains of modeling, simulation, implementation and not to formalization.

It seems not to be easy to escape the challenge of logics. All the tools and methods of design, programming languages, LISP obviously too, are based on logic. The same is the case for the machines.

Why should the process of design be restricted by the structure of its classical tools?

Some complementary aspects of MIT related and PCL related work.

Diagramm 45

	explanation	formalization	implementation	realization
MIT	psychology Piaget linguistics common sense mono-contextural parallel analogy	classical logics monoton vs. non-monoton semiotics as applications meta-level	AI programming languages, LISP	SOAR
PCL	philosophy foundational studies logic, mathematics deconstruction poly-contextural proemiality	polycontextural logics + arithmetics morphogrammatics kenogrammatics coalgebra, fibres	as modeling ML	as simulation and real appl

3 Minsky´s Architecture: The Six Level Model

Marvin Minsky offered the Six Level Model from his forthcoming book The Emotion Machine as an initial proposal for such an architecture. This architecture is being developed jointly by himself and Aaron Sloman, and is based on several key ideas:

1. Use several approaches, at once, to each problem. When one method begins to fail the system can quickly switch to another. We represent each fragment of knowledge with several different representations. By always maintaining several viewpoints (in contrast to all traditional systems), our processes will rarely get stuck.

2. Have many ways to recognize and respond to internal and external problems.

The architecture consists of layers of agents, where each layer is concerned with coordinating, managing, and responding to problems in the layers beneath it. Within each layer, there are ‘critics’ that detect types of problems in the layers beneath or in the outside world. These then turn on ‘selectors’ that invoke methods for resolving these problems.

3. Support many different “ways of thinking”. The most important high-level operation is mapping types of problems to large-scale “ways of thinking”. Each way of thinking disposes the system to use certain types of knowledge, methods of reasoning, types of critics, and other kinds of resources to solve the problem at hand. This architecture is really a kind of meta-architecture, one that invokes more specific architectures in response to the kinds of problems the system encounters.

THE ST. THOMAS COMMONSENSE SYMPOSIUM

Marvin Minsky, Push Singh, MIT, May 13, 2002

4 The Polycontecturality Approach

As the name suggests, polycontectural logic, polycontecturality in general, is interwoven with multiplicity from the very beginning.

Because of the complementary thematization of cognitive systems I am working on, it is easy to confuse common terms, like architecture, reflection, interaction, etc.

As a trial I define a cognitive system as a chiastic entity with the following structural aspects.

Architectonics
Reflectionality
Interactivity
Positionality

More details can be found in the following chapter *Proemiality and reflectional architectures.*

4.1 Architectonics

The operator of architectonics is the cut.

Classical science and computing is based on a single cut, the Cartesian cut. This cut is producing the difference of internal and external domaines, states, events. And most other dichotomies are based on this Cartesian decision.

Architectonics is defined by structural, that is, epistemic cuts. The cut between internal and external domains, the cut of the internal as a self and a model of another self.

Classical computing is still imprisoned by the simple Cartesian cut: inside / outside. Computational reflection in the sense of Smith tries to escape this frame in introducing differences in the inside of the system.

A short philosophical remark. Mostly, we are occupied by temporal studies, studies of the temporal behavior of systems, short with time. Even the circus of self-referentiality was a drama of time. Our concept of space is not welcomed because of the fear of objectification of subjective events. Only in recent time, a new emphasis for architectures in the theory of living systems emerges. Architectures are not tectonics.

Interestingly, the Hegel-Marx based Soviet concepts had been more architectonically oriented than the more temporal notions of self-referentiality of Second-order Cybernetics by the Western research (Levebvre)

Towards some cuts inside the cartesian cut

Object systems
Meta-systems
Meta-level systems

Because there is no theory about the process of cutting, reflectional programming is forced to introduce meta-circular interpreters.

Architectonics vs. Tectonics

Sign systems, as the scriptural medium of computation, are structured by their tectonics. This type of hierarchical tectonics is based on a single cut architectonics. But this cut is not part of the sign system, it is much more its blind spot.

From the point of view of polycontecturality, sign systems, semiotics, are not structured by architectonics.

Architectonics are understood as a form of mediation of structural and prozessual, algebraic and coalgebraic, principles.

Architectonics is not a Ur-ground, a static and eternal fundament, origin of everything. Architectonics is complex and dynamic, giving space for a multitude of beginnings of interacting tectonic systems.

Starting a list of questions and possible answers about reflectional blindness.
What is the blind spot of a program? Answer: Its operating system.
What is the blind spot of an OS? Answer: Its hardware system.
What is the blind spot of a computer system? Answer: Its users. Or: Its environment.

4.1.1 Reflectionality

In the history of cybernetics and computer science reflectionality was reduced mainly to recursive and self-reflectional concepts. The most famous approache is surely the Y-operator of combinatorial logic for the whole of AI esp. LISP and also well known, but more in the circles of second-order cybernetics the re-entry concept of Spencer-Browns Calculus of Indication. This re-entry concept has destroyed by its simplizity and mysticism all germs of complex architectonics of the early second-order cybernetics as they had been introduced by the research of Gunther and Pask.

Introspection
Reflection
Awareness
Self

4.1.2 Interactivity

Communication
Cooperation

Cocreation

4.1.3 Positionality

Embededdnes
Situation
Incorporation
Embodiment

Classification of "Cognitive Systems"

Today´s approaches to Cognitive Systems are therefore classified as
Architecture: one cut, internal / external
Reflectionality: Intentionality, representation of the external world
Interactivity: communication by means of information
Positionality: unizity as blind spot

*Classical computing system*s are well described as systems with a single cut, where reflectionality is reduced to representation of the world producing information, interactivity occurs as communication, communicating information and the *blind spot* of classical computing is its positionality.

This characterization shows clearly the conflict of introducing panalogy architectures in a mono-contextural paradigm.

What´s about some traditional specifications of our understanding of ourselves and the world?
Metaphysics
Ontology
Epistemology
Gnoseology,
Logic

The term cognitive, in cognitive systems, seems not to be very clear. The aspects of affect, emotion, decisions are not necessarily components of cognition. It would be more adequate to name such systems *subjective* systems as composed by cognitive and volitive functions (Gunther, Cybernetic Theory of Subjectivity).
Is the very concept of Cognitive Systems or Emotion Machine in itself panalogic? In other words, is the pananalogy of the new approaches, Cognitive Systems and Emotion Machine, mono- or polycontextural?

5 Togetherness of living systems

Some philosophical remarks about the concept of togetherness as it is used in this text. If we ask the internet about “togethernes”, we are quickly involved in all sorts of spiritual groups and mental health projects. Next we find us together with philosophers of togetherness like Martin Buber, Rosenstock-Hüssey and others. A step further we join the work of Martin Heidegger about Mitsein, then Ludwig Binswanger´s Miteinander-sein. And so on. Also this connections are of importance, this text tries to go radically beyond anthropomorphic implications. A more genuine reading of Heidegger gives us some hints to not to confuse anthropology with his strict structural deconstruction of ontology.

The desire to build a machine with cognitive, emotive and volitive behaviors shouldn’t try to implement some sort of classical anthropology and its (child)psychology, but should be guided by the structural analysis of the conditions of being in the world of living systems. This excludes not only (phenomenological) psychology but also biological approaches.

Architectonics maybe a hard but strict interpretation of "Mit-Sein".

The Blind Spot problem a heritage of solipsism

The so called “Blind Spot” exists only for an analysis of living systems as cognitive systems that is, on the base of representations (Vorstellungen) and information. It doesn’t change much if the framework of cognition is set in a more constructivist manner. The same problems of “reflective blindness” emerge. Simply because the restriction of living systems to cognition and all the optical metaphors of mirroring, reflection, and view points arise. The blind spot is mainly a result of cognitive solipsism. The Blind Spot problem is not solved by duplicating it by two cognitive systems instead of only one, as proposed by Kennedy (2003). The reason is obvious, there is no structural difference between the two cognitve systems as cognitive systems they are the same, and have in common the general idea of being a cognitive system. In other terms, its performance is an Ego-Ego-relation and not an Ego-Thou-relation.

It was exactly this Cartesian burden which Heidegger was deconstructing. His Daseinsanalyse is much more volitive, pragmatic than cognitive.

Cognitive Science as a base of cognitive AI is still dreaming in the Cartesian cage.

Embodiment, embeddedness, situatedness, etc. are terms in the direction of an abandonment of the dominance of cognitivism.

Togetherness as structural interactivity. Maturanas structuaral couplings.

Mismatches of architectures in interaction

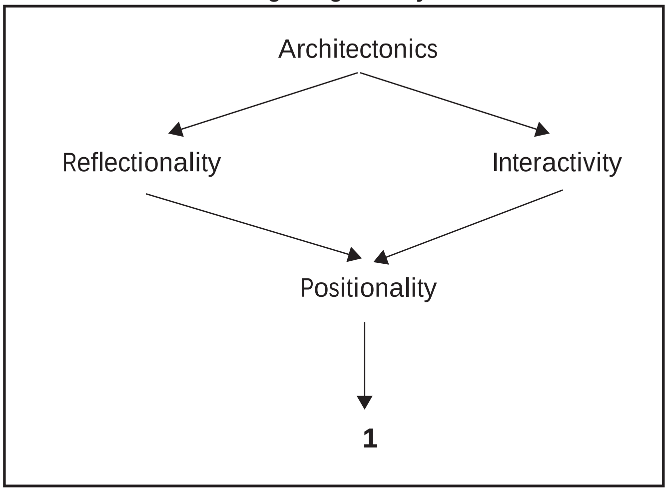
5.1 Conceptual graph of togetherness

Togetherness can be thematicized first as an interplay of different cognitive systems as Ego- and Thou-systems. Second togetherness can be understood as the mechanism of over-determination as simultaneous realizations of different events at the same “ontological” locus which has its inscribtional realization in morphogrammatics.

> "Since the classic approach to identify cognition and volition separately in a closed unit of individual subjectivity has failed we shall approach the problem from a different side. We shall assume that the phenomenon of subjectivity, as manifested by thought processes and decision making, cannot be looked for inside the skin of an individual living body - be that animal or man. We propose instead the following theorem:
>
> *Subjectivity is a phenomenon distributed over the dialectic antithesis of the Ego as the subjective subject and the Thou as the objective subject, both of them having a common medi-*

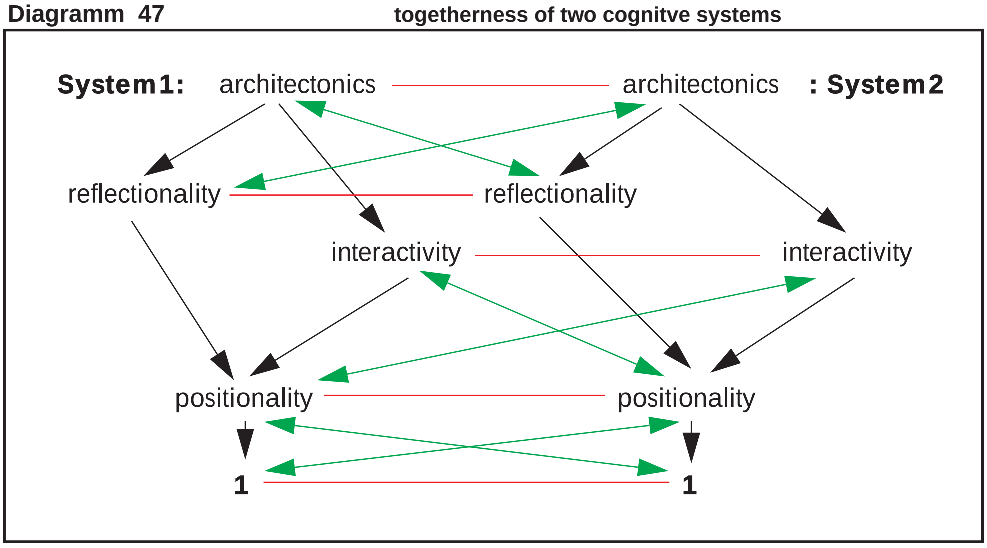
ating environment." Gunther, Cognition and Volition

Diagramm 46 **Single cognitive system**



The minimal structure of togetherness is the proemiality of the quadrupel (volition, cognition, subjetive subject, objective subject) in a co-created common world.

Diagramm 47 **togetherness of two cognitve systems**



The full graph of the chiastic interplay between two cognitive systems would have to consider all possibilities of combinations of order and exchange relations between the components of the two systems. Therefore we would have to study 5 coincidence relations and 25 exchange relations on the base of 10 basic order relations.

Cognitive System1	Cognitive System2
Architectonics	Architectonics
Reflectionality	Reflectionality
Interactivity	Interactivity
Positionality	Positionality

Some interpretations

unizity vs. positionality

The unizity 1 of the positionality of system1 is mirrored as a position, that is as one of several possible positions, in the positionality of system2.

The unizity 1 of the positionality of system2 is mirrored as a position, that is as one of several possible positions, in the positionality of system1.

Despite the fact, that the unizity of both systems for themselves is absolute, the change of functionality, ruled by the exchange relation, produces some kind of context dependent relativity between these absolute unicities.

A system therefore can accept the uniqueness of another system without getting into the problem of denying its own position and to have to be identified with the other system. It can offer in its own positionality space for the positionality of the other system (giving space, einräumen, espacement).

positionality vs. reflectionality

reflectionality vs. architectonics

interactivity vs. architectonics

6 Intra- and trans-contextural proemiality of/between cognitive systems

Intra-contextural proemiality occurs in the process of *introspection* of a system.

Interaction vs. interactivity

Interaction in the so called paradigm shift of computing and computation (Goldin, Stein, Wegner) is mainly understood as *informational interaction*.

cit.

The flow of information in the new paradigm is not restricted to the internal flow of information in computers and computer systems but also allows informational communication with a non-computational environment. In this sense computer science finds home to cybernetic approaches, mainly to concepts of old first-order cybernetics.

Despite of the strong differences between interactive and non-interactive computation as open and closed systems, the informational approach to interactivity is not aware that with the use of the general concept of information the difference between the inside and outside of interacting systems is niviledged to a homogen system of information flow. This information flow of algorithmic and non-algorithmic processes which is basic for the model of interactive computation is the common and homogenizing mechanism of internal algorithmic and external input-output streams.

If the difference of inside and outside should have any meaning it should be clear that the difference as such doesn't belong to the concept of information. There may be an informational process inside a system and there may be informational processes too in the environment of that system but the change as such from inside to outside or simultaneously from the outside to inside is itself not well understood as an informational process.

It has taken cybernetics a longtime of research to understand the problematics of this constellation. And only in a few approaches of second-order cybernetics some advance could be achieved (von Foerster, Gunther, Varela).

If interaction is understood as perturbation of structurally coupled systems the very concept of information with its information unit, channel, transmitter, receiver, etc. gets obsolete.

reflection vs. feedback

"The ability to transform its own representation because of a non-adequation between representation and reality is a typical reflective mechanism which regulate its activity by comparing results to the simulated world. But reflection is present only if this comparison and the way to reduce the difference is explicit and not defined in an ad hoc way. For instance, a simple feedback loop is not aware of its behavior, and does not define a reflective system (even if reflective systems often do use some sort of feedback)." J. Feber, in: Meta-Level Architectures and Reflection, p.192

6.1 An example: Switches between arithmetics

Why not simply ask the experts from the MIT?

Implementing Panalogy

I will use the term Panalogy to refer to a family of techniques for synchronizing and sharing information between different ways of thinking concerned with the same or similar problems. The term derives from 'parallel analogy'. By maintaining panalogies between ways of thinking, we can rapidly switch from one way of thinking to another.

We can also make more partial changes like the representation language they are using, the types of assumptions they are making, the methods that are available to them for solution, and so forth. The key idea is to support representing multiple problem solving contexts simultaneously and the links between them. A graphical depiction of panalogy at work is shown below in Figure 6.

Diagramm 48

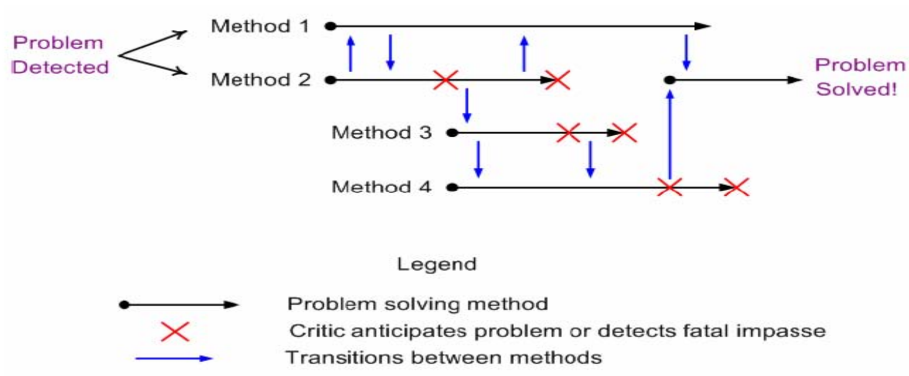


Figure 2. Switching between parallel methods or ways of thinking.

But is this exactly what I am looking for? Obviously not. To have the same wording and to have the same diagram doesn't yet mean that we are thematicizing the same situation in the same way of thinking and implementation.

The main difference between panalogy and proemiality is this. Panalogy is mono-contextural, always only one method is running, not several at once and there is no interactivity between successively different methods. They are applied only one after the other. If one method doesn't work, take another. Proemiality is ruling the interplay of different methods running and cooperating together at once.

Here, my distinction of different modi of thematicizing comes into play: narrative *explanations*, *formalizations*, *implementations* and *realizations*. I am introducing such patterns of multiple thinking directly into the very concepts and methods of semiotics, logics and arithmetics. And this happens step-wise on all 4 modi of thematization.

The Minsky approach is still mostly in the mode of *modeling* of some psychological and linguistic concepts from metaphorics into implementations.

Modeling means, that there is some knowledge about the subject, e.g. the way of thinking of a child, maybe with the help of Piaget, and then this knowledge has to be transformed into computer simulations.

The opposite or complementary approach of polycontextural logic is more concerned in constructing new ways of formal thinking and producing new formalisms, formal methods and apparatus, to help to understand the structural problems of natural science, e.g. child psychology and the unsolved paradoxes of the Piagetian approach.

The wording *"Switching between parallel methods of thinking"* sounds quite promising, but it doesn't gives us a hint how the switch is working, what is the mechanism of the switch, and, how do we know that we are dealing with the same problem after the switch to another domain. How much is the problem itself transformed by the switch of context? And what is the notion of sameness involved in this switch? What do we mean by "parallel" in this context?

> *I will use the term Panalogy to refer to a family of techniques for synchronizing and sharing information between different ways of thinking concerned with the same or similar prob-*

lems.

The common term between the different domains of panalogy is obviously information. But how can we know that all the domains are ruled by the very same concept of infromation? Why is the term information not in itself panalogical?

By maintaining panalogies between ways of thinking, we can rapidly switch from one way of thinking to another.

This sounds really good! But, again, how does it work and who is operating these deliberating switches?

Still, one thing seems common to every such change: In each of our different emotional states, we find ourselves thinking in different ways—in which our minds get directed toward different concerns, with modified goals and priorities—and with different descriptions of what we perceive. Thus, although we use ‘love’ for so many things, there’s one feature that most of those meanings share:

When a person you know has fallen in love, it's almost as though someone new has emerged—a person who thinks in other ways, with altered goals and purposes. It's almost as though a switch had been thrown, and a different program has started to run.

What could cause so dramatic a change? What makes our minds keep switching around? What happens inside a person's brain, to cause such a transformation? This book will argue that when we change what we call our ‘emotional states,’ we’re switching between different “Ways to Think.” For some of these, we have distinctive names—such as ‘suffering, 'anger', 'fear' or 'pain'—but others are harder to classify.

Why don’t we stick with one way to think? What are the functions of all those emotions? Our answer is that no one, single technique will help us face every predicament.

We'll try to **design** (as opposed to **define**) machines that can do all those 'different things'.

Marvin Minsky, Emotion Machine

Minsky´s question is “*What* could cause the change?” and not “*How* does it happen?” or "What is the *mechanism* of change?"

6.2 Brainstorming vs. Diamond Strategies

Critics

I will propose that there exist two important classes of agents that I will call critics and advocates.

Critics are the agents responsible for producing such a negative way of thinking, one that prevents actions from being considered or complains about actions under consideration, as opposed to retrieving or promoting the taking of those actions.

Advocates

A commonsense system is unlikely to be effective if it only sees flaws and never opportunities. On the positive side we need ways of thinking that are more optimistic and suggest courses of action. As I mentioned before, positive knowledge includes the space of ordinary effective procedures, ones that suggest things to do, as well as all sorts of positive knowledge about how to propose things to try.

Let us consider planning, as one example of a cognitive task a commonsense system should be able to perform. Positive agents produce ideas, analogies, inferences, plan and anything else that may suggests a path—both of mental inferences and of worldly actions and events—towards a goal.

Push Singh

The pananalogy architecture consits of the following components and agents.

Ways of thinking

Reflective

Deliberative

Reactive

Brainstorming

critics

advocates.

facilators

Panalogy

This architecture depends on the ability to rapidly switch between different ways of thinking, as depicted below in Figure 5. While the central operation in brainstorming is to select advocate and critic agent in ways that move us towards our goals, these agents may use many different ways of representing knowledge. If transitioning from one way of thinking to another requires a large-scale reconfiguration of the currently active society of agents, then this may involve a great many reformulations from the current way of representing things into suitable new representations.

Sing is introducing an interesting list of panalogy operators.

Environment panalogy.

Procedural panalogy.

Sensory panalogy.

Operator panalogy.

Category panalogy.

Ontology panalogy.

Composition panalogy.

Realm panalogy.

Sense panalogy.

To each key idea I have tried to associate new words: ways of

thinking, brainstorming, critics and advocates, reflective critics, and panalogy.

It is interesting to compare these concepts of “Ways of thinking”, “Brainstorming”, and “Panalogy” with similar concepts known from the theory of polycontexturality. A possible first correspondence maybe:

Ways of thinking vs. Polycontecturality
Panalogy vs. Architectonics
Panalogy transitions, switches vs. Proemiality
Brainstorming vs. DiamondStrategies

Brainstorming vs. Diamond Strategies

The complexity of a situation, e.g. a problem to be solved, can be elicited by the method of diamond strategies. Diamond strategies are not restricted to positive and negative agents, and possible mediators, but are evolving the whole range of possible meanings of a situations under consideration. A situation is not restricted to linguistic entities, each accessible event can be questioned by the diamond strategies.

Diamond Strategies are distinguishing two main types of questions. One is asking the enabling (Ermöglichung) question: “What enables X?” and “What disables X?”. The other type of questions is asking “What is the opposite to X?”. Additionally to the question of the opposite, there are two strong further questions “What is neither the position nor the opposition?”, this question is asking for something beyond the duality of position and opposition and is producing some reflectional distance to the problematic situation. The fourth question is asking for the common, the at once of position and opposition producing a higher form of unity of both beyond identification with a single one.

In a situation of problem solving, the opposite of the problem statement can be the context in which the problem is posed.Because a problem statement is a sentence which always can have several meanings, even if the statement is strictly well-formed, we have to analyze the context of the statement to get more information about further possible meanings which may be helpful to find a solution of the intended problem.

For a well-defined problem its solution lies in the domain of the problem statement. Most problems are not so strictly well-defined to be solved without some creativity. Diamond Strategies are helpful to support creativity.

The brainstorming approach, advocats and critics are mainly concerned about the truth or falshood of statements, or of the usefullnes or uselesness of some strategies and not the opening up of new spaces of choices.

Brainstorming is involved with the *goal*-oriented approach of problem solving.

6.3 Panalogy transitions and proemilaity

But Singh is keeping his mechanism of switching between different modes as a well regarded secret. It seems that the very idea of multiplicity, of a multitude of ways of thinking, etc. is in itself interesting enough.

Singh refers to Minsky´s *Emotion Machine*, there we can find a lot of examples which shows the phenomenon of changing positions. But there is no explanation how these transitions are working. What is missing it seems is an operator which is not only introducing these multitudes but also operates the switches between different levels, standpoints, ontologies, ways of thinking and so on.

My impression is that all these multitudes have to be pre-given by the designer of the system. It is not clear how the system itself can evolve and change its own framework of complexity.

In Singh´s model a critics agent suppose to change the way of thinking for the purpose of problem solving. Mainly there are only two attidudes involved, the negative and the positive way of thinking, critics and advocates. A new domain is chosen, where is it from?, and some adaption of the notions has to be realized to fulfill the transition. After the realized change, what happens structurally to the old system? Where are they localized structurally?

Diagramm 49

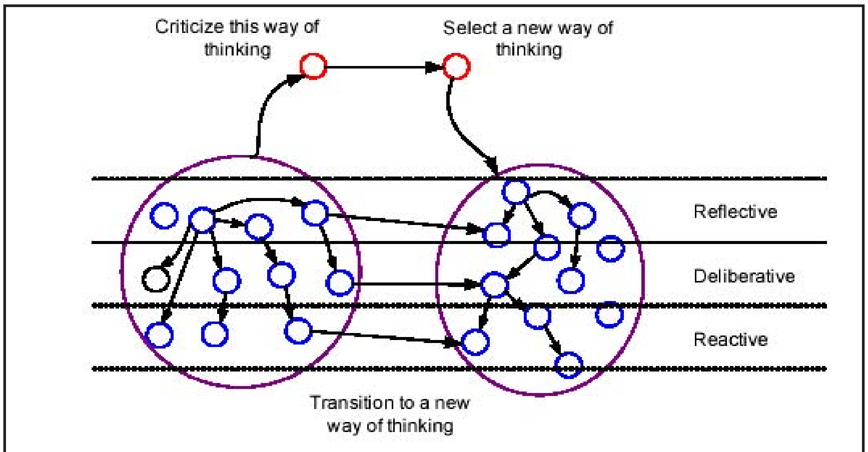


Figure 5. Transitioning to a new way of thinking

7 Cognition and Volition

"The world is a tragedy to those who feel, but a comedy to those who think."
Horace Walpole

> What could cause so dramatic a change? What makes our minds keep switching around? What happens inside a person's brain, to cause such a transformation? This book will argue that when we change what we call our 'emotional states,' we're switching between different "Ways to Think." Marvin Minsky

Gunther seems to be more concerned with the question *"How is it possible"* and not so much with Minsky´s question *"What could cause so dramatic change?"*. Obviously, both, the how- and what-questions are working together.

To explain this in detail, I make use of extensive citations.

In the proemiality chapter I have given a semi-formal explanation of the concept of proemiality. Now, cognition and volition, will give an interpretation of this formal concept and will put it into the more familiar context of the cognition/emotion interplay as we know it especially from Damasio and Minsky.

Gunther:

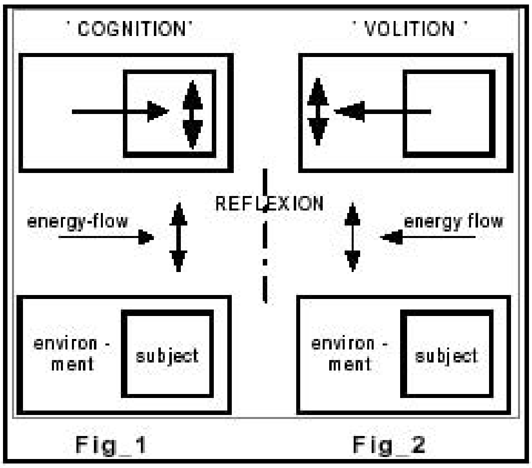
> Since the classic approach to identify cognition and volition separately in a closed unit of individual subjectivity has failed we shall approach the problem from a different side. We shall assume that the phenomenon of subjectivity, as manifested by thought processes and decision making, cannot be looked for inside the skin of an individual living body - be that

animal or man. We propose instead the following theorem:

Subjectivity is a phenomenon distributed over the dialectic antithesis of the Ego as the subjective subject and the Thou as the objective subject, both of them having a common mediating environment.

Since the present author is vigorously opposed to the prevailing methodological aim of total re-objectivation of life processes the following analysis of the fundamental relation between subjectivity as cognition and subjectivity as active volition is intended to be a contribution to a cybernetic theory of Life.

Diagramm 50



Our two figures show that the mutual relations of a cognition and a volition with regard to their environment are exactly inverse. It goes without saying that figure_1 and figure_2 represent an abstract separation of the interlocking mechanisms of cognition and volition. In reality there is, of course, a constant interplay between the two and it goes without saying that one of them cannot operate without being continuously supported by the other. There is no thought without an essential admixture of volition and vice versa volition without an intrinsic component of theoretical awareness would be totally blind.

We may now say that a system of subjectivity is a mechanism - albeit not a classic one - in which two interacting programs of cognition and volition regulate its relation to the environment concurrently. In one program the living system has to behave under the supposition that the environment represents the superior force of the factum brutum to which reason has to submit; now subjectivity finds itself placed at the bottom rung of a hierarchical ladder as long as the connection between subject and object is cognitive. In the other, the volitive program, the environmental objectivity is merely a nebulous field of potentialities which only the Will can transform to solid objective realities.

Since this paper is devoted to the problem of the mutual relation between cognition and volition some remarks should be added as to how the proemial relationship unites these two faculties and melts them together in a system of self-referential subjectivity. We stated that the proemial relationship presents itself as an interlocking mechanism of exchange and order. This gave us the opportunity to look at it in a double way. We can either say that proemiality is an exchange founded on order; but since the order is only constituted by the fact that the exchange either transports a relator (as relatum) to a context of higher logical complexities or demotes a relatum to a lower level, we can also define proemiality as an ordered relation on the base of an exchange. If we apply that to the relation which a system of subjectivity has with its environment we may say that cognition and volition are for a subject exchangeable attitudes to establish contact but also keep distance from the world into which it is born. But the exchange is not a direct one. If we switch in the summer from our snow skis to water skis and in the next winter back to snow skis, this is a direct exchange. But the switch in the proemial relationship always involves not two relata but four!

Not only two subjective faculties, called cognition and volition, are exchanged, but the order of subject and object also suffers a reversal. What had to be interpreted as subjectivity in the cognitive attitude of the subject, namely the symmetry of position and negation, becomes, in the volitive faculty, a property of the objective world which offers a physical alternative for the will. And where, for the cognitive attitude, the whole Universe is content of the consciousness the volitional act is a content of this very same Universe. In other words: the symmetrical exchange relation between cognition and volition implies a reversal of the non-symmetrical order of subject and object.

Gotthard Gunther, Cognition and Volition, A Contribution to a Cybernetic Theory of Subjectivity, in: Cybernetics Technique in Brain Research and the Educational Process, 1971 Fall Conference of American Society for Cybernetics, Washington D.C., 119-135

The interplay of cognition and volition doesn’t restrict the reasons of switching from one “way of thinking” to another to only emotional events like falling in love, etc. Also cognition as thinking can produce exiting emotions which are motivating or even forcing volition and cognition to a switch. The mechanism of proemiality also guaranties that both modi of existence, cognition and volition, are always simultaneously active, only changing their role of dominance from background to foreground functionality.

On the other hand, the concept of proemiality is open for the interplay with other behaviors additional to cognition and volition.

7.1 Gunther’s Conceptual Graphs in Proto-Structures

An interesting analogy between Minsky´s panalogy and Gunthers mappings of conceptual trees onto his architecture of proto-structure can be seen in the following diagram.Transitions between different ways of thinking can be realized in the framework of polycontexturality as mappings of different conceptual trees or semantic nets onto the kenogrammatic architecture of proto-structure.

From the point of view of polycontexturality, transitions between different ways of thinking can be seen as a switch between conceptual systems in a polylogical complexity. The actual system is the system under attention, the new system as a possibility is in the background. The transition is in this sense also a change of focus between fore- and back-ground of simultaneously existing parallel systems. Each point of transition belongs simultaneously to different logical systems. Therefore, a transition is not simply an exchange of information but a structural change of logical systems ruled by the operator of proemiality. Such proemial switches are not restricted to single systems or single ways of thinking. A switch can in itself be of complex structure entailing a multitude of ways of thinking and their changes.

Proemiality in a multitude of ways of thinking opens up the possibility of non-hierarchic, that is, *heterarchic* thinking and decision making. In a hierarchical system, the way down and the way up coincide, they have to be the same. And at the end, all path have a common origin as its start or as its end.

Why do we need a kenogrammatic system like the proto-structure in Günther´s diagram? A careful reading of Minsky´s and Sing´s introduction of pananalogy as a strategy of dealing with "different ways of thinking" shows that they don´t offer an answer to the question "Where are these different ways of thinking localized?". We can switch from one method to another, but we are not informed where they are localized structurally. What is the logico-structural difference between the different ways of thinking? They must occupy at least a different place in the whole system. The problem now is, that the places are not parts of methods, ways of thinking, but their condition. Places are opening up the possibilities of different ways of thinking, but they don´t belong to a way of thinking in the sense of the distributed methods.

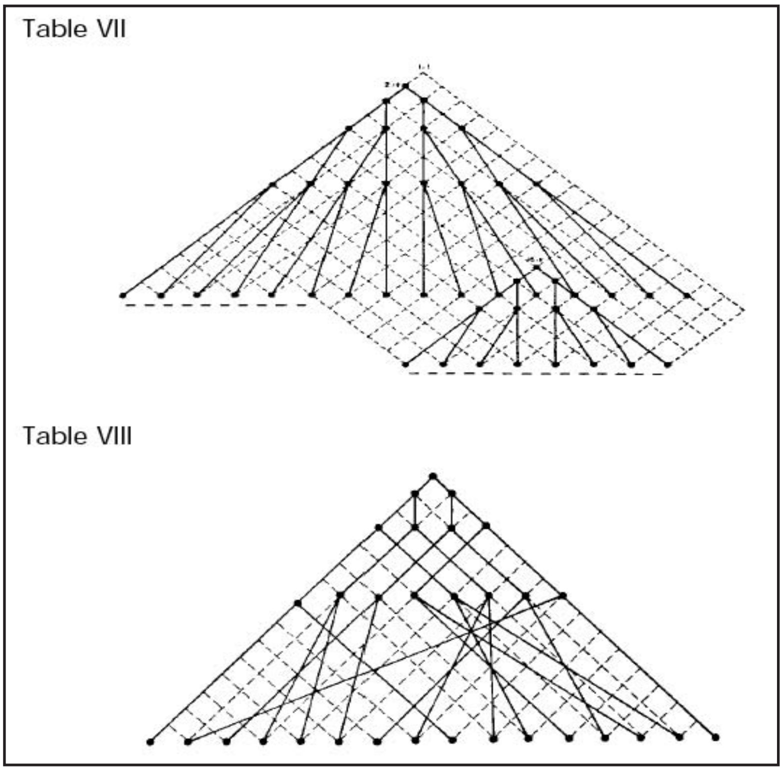
It is ecxactly the job of the proto-grams of the kenogrammatic proto-structure to offer a location to these different methods and logics, that is, to the different hierarchical conceptual graphs. This is not the only aspect but probably the most fundamental.

Each hierarchical conceptual graph starts with its root. All roots are different from each other. There is no common root in this scenario. To realize their interplay they have not only to be different but also to be located at different kenomic loci. Their mutual interplay is ruled by the proemial relationship, their difference is inscribed by their kenogrammatic localization in the proto-structure. A more complex differentiation of the kenomic locus is given by the deutero- and the trito-structure of kenogrammatics.

Therefore, the interactive interplay between different hierarchical conceptual graphs with its tree structure is ruled by the proemial relationship, also understood as chiasm, between different roots and their branches distributed over different loci. Obviously, locally, each tree is realizing an order relation, between roots and branches we observe an exchange relation, and the relationship between roots of different loci and branches of different loci is realized by the coincidence relation.

Despite the graphic form of the diagram of proto-structure, it is not a hierarchical system, and there is also no need to postulate a beginning as an ultimate root of the system. It is a grid of different kenomic loci.

Diagramm 51



7.2 Common Sense Agents and Ambiguity

> "Understanding natural language also requires inferring hidden state, namely, the intention of the speaker. When we hear, *"John saw the diamond through the window and coveted it,"* we know "it" refers to the diamond and not the window—we reason, perhaps unconsciously, with our knowledge of relative value. Similarly, when we hear, *"John threw the brick through the window and broke it,"* we know "it" refers to the window.
>
> Reasoning allows us to cope with the virtually infinite variety of utterances using a finite store of common sense knowledge. Problem-solving agents have difficulty with this kind of ambiguity because their representation of contingency problems is inherently exponential." Stuart Russell

A mechanical system doesn't know the intention of the speaker. Therefore it has to analyze the sentence and to chose in parallel all grammatically possible interpretations, also it has to go on in parallel with the two interpretations until there is new knowledge, from the past or from the new experiences, which enables a decision, which interpretation of the sentences should be preferred in the actual context, or situation. But the old interpretation will still be a possible choice for the case that the narratives turns back to a new context in which this interpretation will have its own significance and will prevail.

It is also possible, especially in esthetic texts, that both interpretations are of equal importance, and that there is an ambivalence played by the game of interchange between background and foreground positions of the interpretations. Maybe there is at this point a connection to Selmer Bringsjords project of artificial joke making programs.

All these maneuvers are possible only in a real parallel and grammatically or semantically multi-layered system. Probably the best candidate for this job, again is poly-contextural logic.

From a technical point of view of poly-contextural systems there is no reason to think that the complexity of dealing with ambiguity has to grow exponentialy.

Is there a method in the poly-contextural approach to reduce complexity from the exponential to the linear type?

Comparatistics of Models and Metaphors of Machines

1 Minsky Machines vs. Gunther Machines

1.1 Emotion Machine and Volition Machines

„On the other hand, a machine, capable of genuine decision-making, would be a system gifted with the power of self-generation of choices, and the acting in a decisional manner upon its self-created alternatives. (...) A machine which has such a capacity could either accept or reject the total conceptual range within which a given input is logically and mathematically located.“ Günther, Decision Making Machines, 1970

> *„We linked many–valuedness with self–reference. No self–reference is possible unless a system acquires a certain degree of freedom. But any system is only free insofar as it is capable of interpreting its environment and choose for regulation of its own behavior between different interpretations. The richness of choice depends on the magnitude of the value–excess offered by the logic which follows.“ (Günther 1968, 44)*

2 Turing Machines vs. Gurevich Machines

3 Wegner Machines vs. Turing Machines

4 Minsky Machines vs. Derrida Machines

4.1 The root problem

Many ways of thinking, Panalogy, does they have a common root, or even a ultimate origin?

What´s about Deleuze / Guattari and all their machines?

5 Keno Machines vs. Sign Machines (Markov Machines)

Schmitthuber´s Gödel Machine vs. Kaehr´s Gunther Machine

DYNAMIC SEMANTIC WEB

ULTRA-DRAFT

DERRIDA´S MACHINES PART II

The TransComputation Institute

ThinkArt Lab Glasgow

Dr. Rudolf Kaehr

April 1 2004

www.thinkartlab.com

Wozu Dynamic Semantic Web?*

SAP INFO 10/2003
20.10.2003 / Interview mit Prof. Dr. Jürgen Angele, ontoprise GmbH
Werden Computer uns einmal verstehen?

**Schaffen Sie mit semantischen Technologien den Sprung von der Verarbeitung von Daten zur Verarbeitung von Wissen?*

Angele: Ja, denn semantische Applikationen "verstehen" Informationen. "Verstehen" setzt eine gemeinsame Sprache voraus, um konzeptuelle und terminologische Verwirrungen, Unklarheiten und Mehrdeutigkeiten auszuschließen. Und genau das lässt sich mit semantischen Technologien erreichen. In einer Ontologie werden die für einen Anwendungsbereich relevanten Begriffe und deren Zusammenhänge exakt definiert. Die Ontologie beschreibt ein allgemein anerkanntes Verständnis dieses Anwendungsbereichs, das alle Personen und Anwendungen gemeinsam teilen und verwenden.

Ist es das, was wir mit dem DSW wollen?

1 Ziel: Was soll erreicht werden?

Es soll ein *Framework für ein Dynamic Semantic Web* entwickelt werden, das den Charakteristika des WWW entspricht und nicht bloss auf die Exteriorisierung von Datenbank Systemen aus ist.

Das WWW wird hier nicht nur als ein offenes System mit den Eigenschaften distribuiert, dynamisch und quantitativ massiv verstanden (Hendler), sondern zusätzlich als ein global-kulturelles, komplexes sich selbst organisierendes und selbst-modifizierendes Medium artifizieller Natur. D.h. auch, dass das WWW nicht vorgegeben (vorhanden) ist, sondern sich nur einer Interpretation in seiner Zuhandenheit erschliesst.

Die bestehenden Methoden konzentrieren sich auf die Vorhandenheit der Daten im WWW, DSW hat sich der Herausforderung der prinzipiellen Deutbarkeit des WWW, d.h. seiner Zuhandenheit zu stellen.

Daher ist Wissen (knowledge) und Bedeutung (meaning) in einem WWW als kulturellem System grundsätzlich nicht auf Eindeutigkeit, Disambiguität und Dekontextualisierung zu reduzieren. Dies ist möglich einzig für sehr spezielle Erfordernisse.

DSW hat somit zum Ziel, Mechanismen zur Handhabung, Implementierung, Formalisierung und Realisierung von ambiguen, kontextbezogenem und vieldeutigem Wissen, das nichtsdestotrotz einer machinalen Verarbeitung zugänglich ist, anzubieten.

Einige konkretere Ziele

Es sollen Methoden zur Erstellung komplexer evolutiver Ontologien entwickelt werden, die den Erfordernissen etwa der folgenden Kriterien gerecht werden können.

1. Ontology Engineering

Aus der komplexen Datenvielfalt, realisiert in heterogenen Ontologien, einer Organisation, eine vertikal strukturierte einheitliche Ontologie zu generieren, die dann mit den Methoden des Semantic Web verarbeitet werden können, stellt ein grosses und weitgehend ungelöstes Problem dar. Die Effektivität einer Implementierung misst sich jedoch auch an der Effektivität der Aquisition ihrer Daten.

Eine zusätzliche horizontale Organistionsform kann hier aus Engpässen einer aufgezwungenen Hierarchisierung entgegen wirken.

2. Distributed inferencing, architectonic parallelity

Distribuierte Inferenzmechanismen lassen sich aufgrund der polykontexturalen Logik ohne Komplikationen direkt realisieren. Je Kontextur bzw. je Modul, lässt sich eine eigene und autonome Deduktionsregel einführen. Dies geht weit hinaus über klassische Ansätze der Parallelisierung und der durch Mehr-Sorten-Logiken fundierten Distributionen.

3. Meta-Reasoning, Reflektionalität

Reflektionalität ist der polykontexturalen Architektonik, sowohl auf logischer wie ontologischer Ebene, inhärent. Entstammt sie doch dem Bestreben, eine Theorie und einen Apparat der Reflexionsformen zu realisieren.

4. Reusability

Wiederverwendbarkeit erhält durch die tabulare Anordnung der Module eine neue Dimension, die durch die vertikale Konzeption allein nicht realisiert werden kann.

2 Einschränkung: Was soll nicht erreicht werden?

Es geht bei dem DSW Projekt, trotz des fundamental neuen Ansatzes, nicht darum, Bestehendes in seiner konkreten Definition und Funktionalität zu kritisieren. Oder gar als falsch aufzuweisen. Einfach deswegen nicht, weil der PKL-Ansatz einzig und allein versucht, von anderen, eventuell allgemeineren Voraussetzungen, jedoch mit weit weniger ausgereiften Technologien, an eine gemeinsame Problematik heranzugehen.

Es geht aber auch nicht darum, mit den bestehenden Ansätzen, die sich auf spezifische Fragestellungen spezialisiert haben, wie etwa ontoprise, in Wettlauf oder gar Konkurrenz zu treten.

3 Methode: Wie und womit soll DSW erreicht werden?

Web Ontologien bestehen aus Modulen, die *vertikal* organisiert werden und somit eine Dynamik der Evolution, Adaption und Erweiterung im Rahmen einer systematischen Hierarchie ermöglichen.

DSW erweitert dieses Konzept der Modularität dahingehend, dass alle, auch die Basis-Module, *horizontal* organisiert werden können. Damit entsteht ein System ontologischer und logischer Parallelität und Nebenläufigkeit, das vertikale Interaktion zwischen den Ontologien und deren Modulen ermöglicht.

Die horizontale Organisation ontologischer Module soll mit den Methoden der polykontexturalen Logik realisiert werden. Die Polykontexturalitätstheorie stellt logische und ontologische Methoden der Vermittlung und Distribution modularer Systeme bereit.

Dabei kann jeder Modul innerhalb einer horizontalen Organisation selbst wiederum vertikal hierarchisch strukturiert sein. Damit ist ein flexibler und kontextbezogener Wechsel zwischen der horizontalen und der vertikalen Funktionalität gewährleistet.

Die Möglichkeit des Wechsels zwischen horizontaler und vertikaler Organisiertheit, oder in a.W. zwischen Hierarchie und Heterarchie, stellt die Grundstruktur der Dynamik des DSW dar. Dieses Verständnis von Dynamik stellt ein Novum in der Konzeptionalisierung und Implementierung von logischen und ontologischen Systemen dar.

Die konkrete Realisierung einer Implementierung von DSW hat sich mit den sich entwickelnden Methoden und Programmiersprachen des Semantic Web produktiv kritisch auseinander zu setzen und Strategien der Erweiterung, geleitet durch die Ergebnisse der polykontexturalen Logik- und Ontologie-Forschung, zu entwickeln.

Vererbbarkeit und Verwendbarkeit von Methoden

Damit ist, trotz der Novität des Ansatzes des DSW, Anschluss und Vergleichbarkeit, aber auch Verwertbarkeit des Bestehenden gewährleistet. Denn wenn Module, die in sich vertikal organisiert sind, in eine Distribution und Vermittlung horizontaler Art gebracht werden, lassen sich die Konzeptionen, Methoden, Formalismen und Techniken übertragen. Die vertikalen Methoden vererben sich, wenn auch ev. in modifizierter Form, in die horizontale Struktur. Insofern braucht nicht alles neu erfunden zu werden, um das Projekt des DSW zu realisieren.

4 Nutzen: Wozu soll DSW erreicht werden?

Eine tabulare Organisation ontologischer und logischer Module eröffnet automatisch strukturelle Vorteile einer linear organisierten Struktur gegenüber.

Transparenz

Horizontal verteilte Module und Ontologien unterstützen Transparenz aufgrund ihrer relativ autonomen Modularität, die eine Komplexitätsreduktion darstellt.

Flexibilität

Horizontal verteilte ontologische und logische Module unterstützen Flexibilität aufgrund ihrer Möglichkeit zwischen vertikaler und horizontaler Organisation zu wählen.

Disponibilität

Horizontal verteilte Module und Ontologien unterstützen durch ihre Verteilung über die zwei Dimensionen ihrer Positionierung.

Effektivität

Horizontal verteilte Module und Ontologien unterstützen die Effektivität sowohl ihrer Etablierung wie auch der Abläufe ihrer Prozesse, dank ihrer architektionalen Parallelistät.

Insbesondere werden die Prozesse der *Navigation, Negotation und Mediation* von und zwischen vertikal und horizontal verteilten Ontologien aufgrund der polykontextural verteilten Organisation unterstützt.

Navigation

Navigation zwischen Modulen erhält eine neue Dimension, wenn diese in ihrem Spielraum nicht mehr eingeschränkt wird durch eine übergeordnete, allen gemeinsame Basis-Ontologie.

Mediation

Mediation von Modulen ist in vertikalen Organisationsformen äusserst beschränkt und setzt eine allen Modulen gemeinsame Basis-Ontologie voraus. In diesem Sinne handelt es sich bei der vertikalen Mediation letztlich um eine Form der Subsumtion, die nicht in der Lage ist, Fremdes zu akzeptieren und mit Fremdem zu interagieren.

Negotation

Wenn auch DSW auf machinelle Assistenz setzt, ist immer noch genug Raum für Verhandlung zwischen menschlichen Subjekten. Diese Negotationen können sich nun aber auch auf formale Modelle der Vermittlung stützen und sind nicht der reinen Willkür bzw. dem blinden Vertrauen (Trust) ausgeliefert.

Evolution

DSW soll Grundprobleme der Evolution des WWW und der Semantic Web Ontologien aufweisen und zu polykontexturalen Lösungen verhelfen. Die bestehenden Methoden der Handhabung von Evolution von Ontologien sind auf die vertikale Organisation ihrer Methoden beschränkt.

5 Institutionen: Wo und mit wem soll DSW erreicht werden?

Zusätzlich zu Wiesbaden, Daniel Inc. und CNLPA ist involviert ThinkArt Lab Glasgow in Zusammenarbeit mit dem Computer Departement und dem Center of Critical Media Studies des Goldsmiths College, University of London.

In Planung: Gründung von Creative Industries Lab, London, Singapore und Kontakt zu McLuhan Institute, Maastricht, NL.

Die Manpower hängt von den Kontakten und den möglichen Finanzierungen ab.

6 Zeitrahmen: Wann soll DSW erreicht werden?

In einer ersten 3 Jahresplanung soll im ersten Jahr eine Konsolidierung der bestehenden Forschungsarbeiten geleistet werden, die in den folgenden zwei Jahren zu einem ausgereiften Prototypen führen sollen.

Die Emanzipation von den Methoden und Formalismen des Semantic Web in Richtung auf ein polykontextural fundiertes DSW kann nur Schrittweise geschehen.

Ein erster Schritt ist die kritische Aufarbeitung der bestehenden Tendenzen der Implementierung des Semantic Web bezogen auf Ontologiebildung, Web-Logiken und Impelementierungssprachen.

Ein weiterer Schritt ist die Abgrenzung von diesen Methoden und die Entwicklung von Erweiterungen der bestehenden Konzeptionen und Methoden des Semantic Web.

Dies soll in einem vorläufig letzten Schritt zur Entwicklung eines Prototypes einer DSW Implementierung führen.

7 Abgrenzungen: Wogegen soll DSW erreicht werden?

Angesichts der wachsenden globalen kulturellen Dominanz des WWW soll gegen einen reduktionistischen und technizistisch verstandenen und staatlich implementierten Begriff von Bedeutung und Wissen angegangen werden. Damit soll die relative Adäquatheit reduktionistischer Methoden für beschränkte industrielle, administrative und militärische Zwecke nicht geleugnet werden.

Das WWW ist hier jedoch als ein kulturelles und globales Medium verstanden. DSW versteht sich daher als ein nicht durch den Eurozentrismus reduzierte und auf Aristotelischer Metaphysik basierende Strategie der Eröffnung eines globalen kulturellen WWW.

Es soll mit dem DSW Denkmodelle und Verhaltensstrategien im Umgang mit dem WWW zur Hand gegeben werden, die eine Verabschiedung vom Aristotelismus in der Ontologie und Logik wie auch der Fixierung des Machinalen auf das Turingmodell zu unterstützen in der Lage sind.

Es kann nicht übersehen werden, dass nach dem Sieg der technizistischen Denkweise in der und durch die Computertechnologien nun eine entsprechende Vereinnahmung von kulturellen Schichten des Wissens durch das internationale Semantic Web Projekt in Gang gesetzt wurde. Dagegen sind die Bildungseinrichtungen noch gänzlich mit der Adaption an den Digitalismus und seiner Multimedia-Kultur beschäftigt. Die Hilflosigkeit dem Phänomen gegenüber zeigt sich leider auch in der sonst hervorragenden kritischen Arbeit zum Semantic Web des McLuhan Institute, Maastricht.

**Die vorliegende Arbeit ist ein Bericht zur Zielfindungsphase eines Joint Venture Projects mit der Firma DANIEL, Inc., Wiesbaden, Deutschland*

Towards a Dynamic Semantic Web

Dynamic Semantic Web (DSW) is based at first on the techniques, methods and paradigms of the emerging *Semantic Web* movement and its applications. DSW is advancing one fundamental step further from a static to a dynamic concept of the Semantic Web with extended flexibility in the navigation between ontologies and more profound transparency of the informational system. Web Services are now redefinded by Semantic Web. To proof the advantages of DSW, it is the main aim of this project to developed the tools and methods necessary to develop a DSW based Web Service (DSW business application).

The existing framework of the Semantic Web has only very limited possibilities of realizing dynamism. It´s dynamism is reduced to inter-ontological transactions (translations, mappings, navigation) between different local taxonomies and ontologies.

DSW is based on the genuinely dynamic first order ontologies and logics founded in kenogrammatics of the theory of polycontexturality allowing evolution and metamorphosis to create complex interactivity and new domains of interaction.

A General Metaphor

Peter van Dijcks overview

Themes and metaphors in the semantic web discussion.

http://poorbuthappy.com/ease/semantic/

http://petervandijck.net/

Joseph Goguen's help to not to be lost in the chaos of bricolage and the hype:

http://www.cs.ucsd.edu/users/goguen/projs/onto.html

http://www.cs.ucsd.edu/users/goguen/pps/lisbon04.pdf

http://www.cs.ucsd.edu/groups/tatami/seek/

1 The Semantic Web

“Semantic Web: a machine-processable web of smart data.” Daconta

Today, the Semantic Web is becoming an important reality. Not only in research centres but also in industrial, business and governmental organizations, Semantic Web applications are advancing. Semantic Web is understood as the “Next Web”.

“There´s a revolution occurring and it´s all about making the Web meaningful, understandable, and machine-processable, wether it´s based in an intranet, extranet, or Internet. This is called the Semantic Web, and it will transition us toward a knowledge-centric viewpoint of éverything´.” Stephen Ibaraki

As the WWW is based on HTML, the Semantic Web is based on *XML* as its frame language mediated by ontologies. *Ontologies* are the new key to meaning in information processing. Also deriving from philosophy where ontology is representing the most general theory about being and the formal structure of everything, in the Semantic Web, ontologies are of a very pragmatical value. *"Ontologies are about vocabularies and their meanings, with explicit, expressive, and well-defined semantics–possibly machine-interpretable."* Daconta

XML is the corner stone of the Semantic Web. "XML is the syntactic foundation layer of the Semantic Web." It is not a programming language; it is "actually a set of syntax rules for creating semantically rich markup languages in a particular domain. In other words, you apply XML to to create new languages."
"Why is XML so succesful? XML has four primary accomplishments, (...):
XML creates application-independent documents and data.
It has a standard syntax for meta data.
It has a standard structure for both documents and data.
XML is not a new technology (not a 1.0 release)."
More explicit, XML is characterised by following principles:
First: "Markup is separate from content."
Second: "A document is classified as a member of a type by dividing its parts, or elements, into a hierarchical structure known as a tree." Daconta

The Semantic Web is possible today and in reality it is a natural consequence of the fact of the Internet, the WWW, the knowledge about databases and the ubiquity of powerful computing facilities.

Two years ago the Gartner Group has given a marketing projection that *“By 2005 lightweight ontologies will be part of 75 percent of application integration projects”*.

International Investments

DERI-Centres: Ireland and Insbruck (Austria)
Leibzig
Dortmund
Edinburgh

Semantic Web and AI

The merits of the Semantic Web is that it is in its concepts and in its vision very pragmatically oriented. It is in sharp contrast to the sometimes very speculative aims of Artificial Intelligence.

A sharp distinction between Semantic Web and AI can be made between the relevance and understanding of *data* and *programs*. AI is concerned with highly complex programs being at the end able to understand data, e.g. texts and common sense. Semantic Web is more concerned in making its data "smart" and giving them some machine-readable semantics. AI tends to replace human intelligence, Semantic Web asks for human intelligence.

On the other side it seems that Semantic Web is lacking, at least today, strong and complex logics, automated deduction systems and inference machines. Topics which are well developed in AI research and applications.

Semantic Web inferencing machines are mostly based on F-Logic, which is a subsystem of First-Order Logic (FOL).

It is well known that AI has produced a lot of knowledge about Knowledge Representation systems, Concept Analysis and many other semantic based endeavours. Nevertheless, Semantic Web takes a new start on a more pragmatic level, with a more business oriented vision and from an other angle of the whole spectre of "mechanizing" knowledge and interactivity.

Ontologies

The Semantic Web is based on its ontologies. Ontologies are playing the key role in the process of realizing semantic information processing. Ontologies are themselves classified in several types. The most general case is the distinction between core ontologies and upper-level ontology. There are many core ontologies but only one upper-level ontology. The structure of ontology (and ontologies) is strictly hierarchical.

What are the promises?

"What are the real values for using ontologies? The real value of using ontologies and the Semantic Web is that you are able to express for the first time the semantics of your data, your document collections, and your systems using the same semantic resource and that resource is machine-interpretable: ontologies. Furthermore, you can reuse what you´ve previously developed, bring in ontologies in different or related domains created by others, extend yours and theirs, make the extensions available to other departments within your company, and really begin to establish enterprise- or community-wide common semantics." Daconta, p. 237

RDF (Resources Description Framework)

Additional to the link structure of HTML, RDF (Resource Description Framework) comes with a pointer to the resource of the data (object, information) introducing a semantic dimension to the strict syntactic definition of HTML.

A description is a set of statements about the resource.

The RDF model is often called a “triple” because it has three parts: subject, predicate, object.

Subject: This is the resource that is being described by the ensuing predicate and object.

Predicate: This is a function from individuals to truth-values with an arity based on the number of arguments it has.

Object: This is either a resource referred to by the predicate or a literal value.

Statement: This is the combination of the three elements, subject, predicate, and object. (Daconta)

All this is governed by the principle of identity.

“We should stress that the resources in RDF must be identified by resource IDs, which are URIs with optional anchor ID.” (Daconta, p. 89)

This linguistic characterization of the RDF triple is defining a statement and adding to its syntax some meaning guarantied by the identifiable IDs. This relation is decidable, that is, the connotation exists or it exists not, therefor it is true or false–TND.

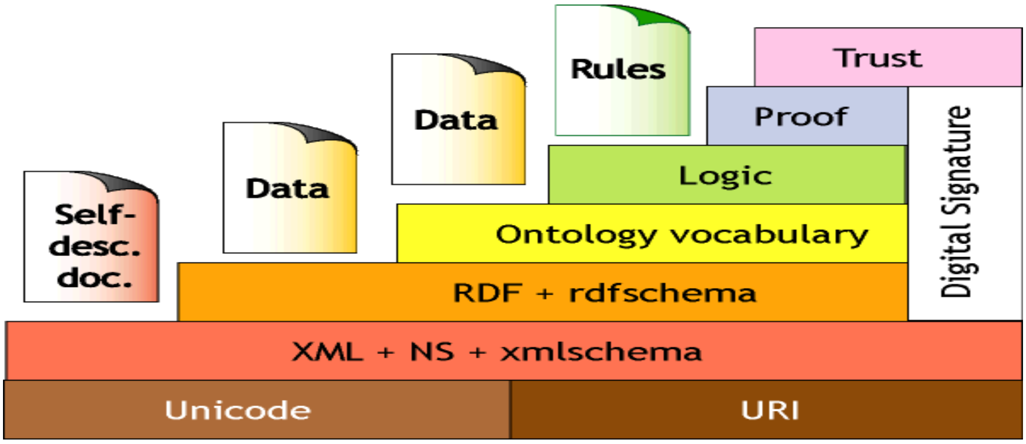
Missing linguistic contexts

At this point I would like to mention, that despite of its semantic relation and its foundation in a generally accepted ontology, this RDF triple is defining a statement in isolation, excluding its context. Later, contexts are introduced by ontologies. But the RDF definition is not involving them. As a consequence, all pragmatic points of views have to be introduced secondarily. It would be helpful, if we could introduce this contextual information at the very beginning of our construction. Without this we will simply repeat the paradoxes of knowledge engineering of the AI projects. That is, meaning of a sentence is context-dependent and contexts are defined by meaningful sentences.

The Semantic Web Stack

In this proposal I will concentrate myself on the basics of Semantic Web as it is proposed by its inventor Tim Berners-Lee:

Tim Berners-Lee´s three-part vision: (collaborative web, Semantic Web, web of trust).



Trust
Proof
Logic Framework, Rules
Ontology, Contexts
RDF Schema
RDF M&S
XML; Namespace
URI; Unicode
and
Digital Signature: Signature, Encryption

Problems with trust and signature

To begin with the top: trust. Let´s have a look to an example.
BMW-Example:
Trust or Distrust? Serious or a joke? How serious is the joke? Or is it simply stupidity?

Hierarchies everywhere

Taxonomies

A taxonomy is a semantic hierarchy in which information entities are related by either the *subclassification of* or the *subclass of* relation.

One of the basic distinctions of GOL is the distinction between *urelements* and *sets*. We assume the existence of both urelements and sets in the world and presuppose that both the impure sets and the pure sets constructed over the urelements belong to the world. This implies, in particular, that the world is closed under all set-theoretical constructions. Urelements are entities which are not sets. They form an ultimative layer of entities without any set-theoretical structure in their build-up. Neither the membership relation nor the subset relation can unfold the internal structure of urelements.

In GOL, urelements are classified into two main categories: *individuals* and *universals*. There is no urelement being both an individual and a universal.

Diagramm 1 **UML hierarchy diagram of a General Ontology**

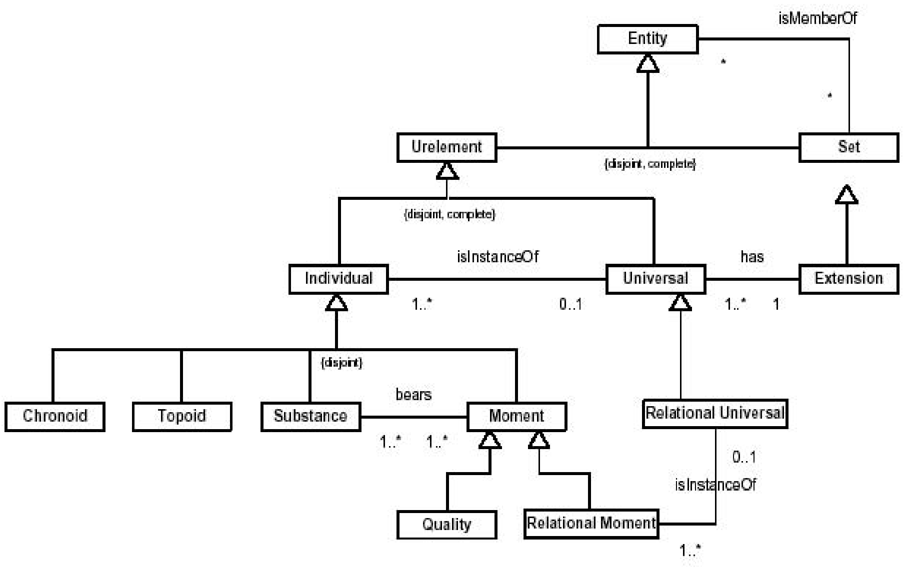
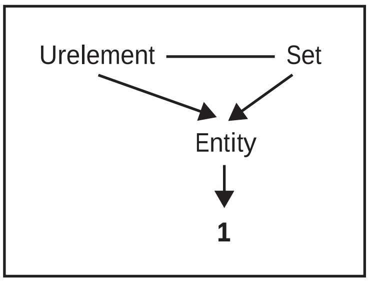


Fig. 1. A UML class diagram describing the basic concepts of GOL.

Conceptual graph of the basic triple (Entity, Urelement, Set) and its uniqueness 1.



Uniqueness means that there is one and only one ontology defined in terms of Urelement, Set and Entity. This also means, there is only one World, and at the end it means, there is only one WWW, too. But this is homogenizing complexity and diversity, and is simply a monstrous nomiminalisation. In other word, it is one and only one way of thematizating the world, the mono-contextural one.

> The development of an axiomatized and well-established upper-level ontology is an important step towards a foundation for the science of Formal Ontology in Information Systems. Every domain-specific ontology must use as a framework some upper-level ontology which describes the most general, domain-independent categories of reality. For this purpose it is important to understand what an upper-level category means, and we proposed some conditions that every upper- level ontology should satisfy. The development of a well-founded upper-level ontology is a difficult task that requires a cooperative effort to make signicant progress.

Diagramm 2 **Axiomatic Foundation of Upper-Level Ontologies**

Axioms of Basic Ontology

(a) Sort and Existence Axioms

1. $\exists x(Set(x))$,
2. $\exists x(Ur(x))$
3. $\forall x(Set(x) \vee Ur(x))$
4. $\neg\exists x(Set(x) \wedge Ur(x))$
5. $\forall x(Ur(x) \leftrightarrow Ind(x) \vee Univ(x))$
6. $\neg\exists x(Ind(x) \wedge Univ(x))$
7. $\forall xy(x \in y \rightarrow Set(y) \wedge (Set(x) \vee Ur(x)))$

(b) Instantiation

D. $ext(x, y) =_{df} Univ(x) \wedge Set(y) \wedge \forall u(u :: x \longleftrightarrow u \in y)$.

1. $\forall xy(x :: y \rightarrow Ind(x) \wedge Univ(y))$
2. $\forall x(Univ(x) \rightarrow \exists y(Set(y) \wedge \forall u(u \in y \longleftrightarrow u :: x)))$

(c) Axioms about sets

1. $\forall uv\exists x(Set(x) \wedge x = \{u, v\})$
2. $\{\phi^{Set} \mid \phi \in ZF\}$, where ϕ^{Set} is the relativization of the formula ϕ to the basic symbol $Set(x)$.[3]

Contributions to the Axiomatic Foundation of Upper-Level Ontologies, Wolfgang Degen, Heinrich Herre

All these axioms of the formal general ontology GOL are not only defining a (probably) consistent framework for all possible applicative, core ontologies, but are also asking a hard price for it: there is no dynamics in this framework of ontology. Everything is what it is, e.g. Urelement or Set. Any dynamics is secondary and localized in “chronoids”, “topoids”, etc. which are special cases of Individuals. In other words, no Urelement can become a set and vice versa, simply because this ontology is mono-contextural, lacking any fundamental perspectivism and interactivity with diversity.

2 How to introduce the Dynamic Semantic Web?

The Semantic Web movement is not only strong and inevitable, it is also open to the future. On a pragmatic level it is open for an increasing multitude of local and personalized systems. It´s general definition is monitored by the W3C, but in encouraging new developments and not restricting its future progress.

In this sense the Semantic Web movement includes without problems a spectre from Aristotelian fundamentalists to Rhizomatic Anarchists.

In other words, it is not in contradiction to the guidelines of the Semantic Web to develop as a new branch the paradigm of DSW.

It is a philosophical question if this branch is well understood as branch and should not be better thematized as something quite different, namely as an interlocking mechanism between core and upper ontologies and their logics distributed over different irreducible upper ontologies.

From a pragmatic point of view, DSW is better localized as a new branch or discipline of the Semantic Web.

The map of the Semantic Web assembles all sorts of theories, methods, implementations from philosophy to hard core programming, including AI and data-base technologies, logics, semantics, context theory, linguistics, neural networks, etc. on all levels of scientifity and scholarship, not excluding some confusions and other cocktail events.

This is allowing a great diversity of different approaches to be involved in the development of the Semantic Web and its extension to the Dynamic Semantic Web, and many other invention, too.

Decentralization and Heterogeneity

To deal in a flexible and controllable way with decentralized heterogeneities, hierarchies are not delivering the best possibilities. Here is the moment where *heterarchies* come into the play.

Decentralization and Heterogeneity is obviously in conflict with the strict reglementations of upper-level (first order) ontology as it is formalized in the general ontology GOL.

> Two different contexts relating respectively to species and environment point of view.
> With such different interpretations of a term, we can reasonably expect different search and indexing results. Nevertheless, our approach to information integration and ontology building is not that of creating a homogeneous system in the sense of a reduced freedom of interpretation, but in the sense of **navigating alternative interpretations**, querying alternative systems, and conceiving alternative contexts of use.
>
> To do this, we require a comprehensive set of ontologies that are designed in a way that admits the existence of **many possible pathways** among concepts **under a common conceptual framework**. This framework should reuse domain-independent components, be flexible enough, and be focused on the main reasoning schemas for the domain at hand. **Domain-independent, upper ontologies** characterise all the general notions needed to talk about economics, biological species, fish production techniques; for example: parts, agents, attribute, aggregates, activities, plans, devices, species, regions of space or time, etc. (emphasis, r.k.)

http://www.loa-cnr.it/Publications.html

2.1 Heterarchies, in general

In contrast to the Semantic Web with its tree structure, that is, with its fundamental hierarchic organization on all levels of conceptualization and realization, the Dynamic Semantic Web comes with a strong decision for heterarchies.

Heterarchies are not fully understood if we are not studying the interactivity between hierarchies. In this sense heterarchies are the framework of the interactivity of hierarchies. In other words, heterarchies are ruling the interplay between an irreducible multitude of different trees.

One great advantage is, each of these trees is inheriting the well known and proven methods and technologies of their classical predecessor, that is, logics, taxonomies, proof systems etc.

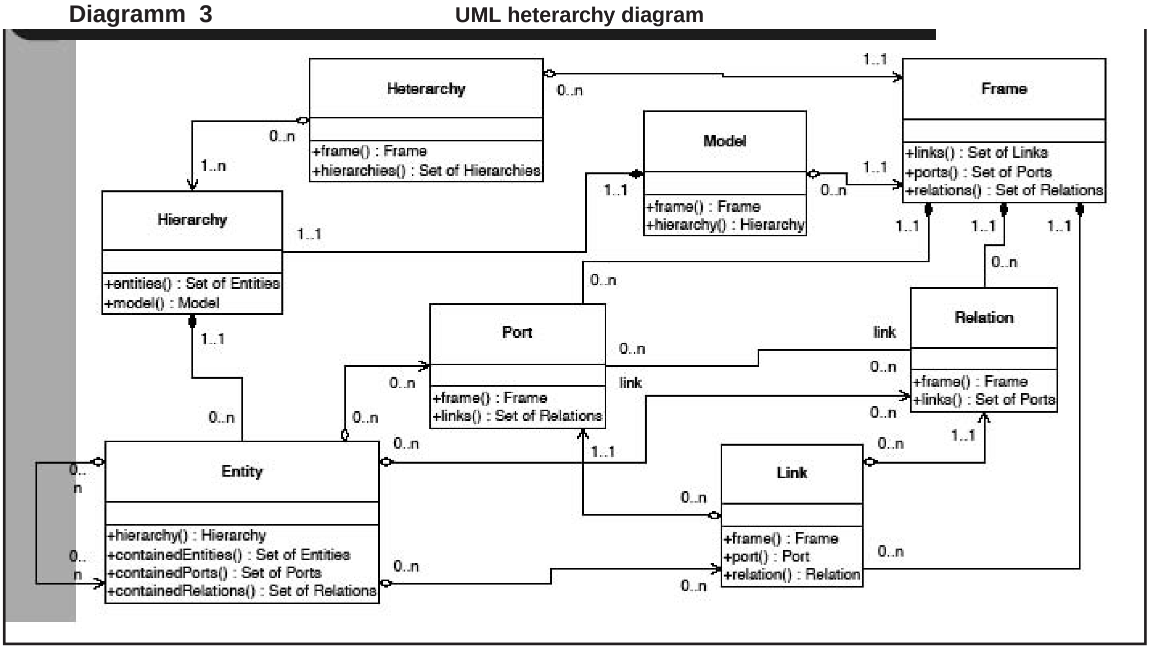
> "Whereas hierarchies involve relations of dependence and markets involve relations of independence, heterarchies involve relations of interdependence."
>
> "Stark has proposed "Heterarchy" to characterize social organizations with an enhanced capacity for innovation and adaptability.
>
> Networked or lateral organizations are in direct contrast with the tree-like, vertical chains of control of traditional hierarchies. The second feature means that heterarchies require diversity of components and building blocks." [Stark, 1999, page 159],

http://www.c3.lanl.gov/~rocha/GB0/adapweb_GB0.html

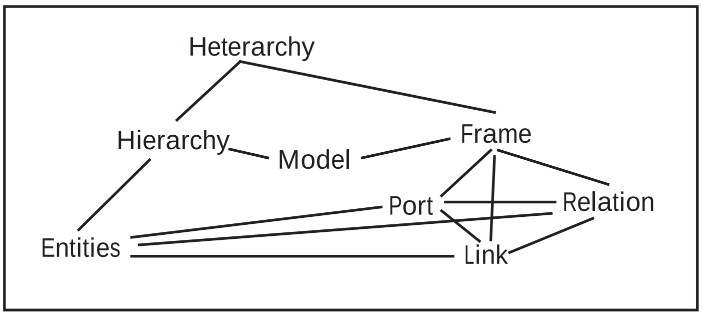
To give a more transparent modeling of the interactivity between hierarchies as it is proposed by the *proemial relationship* it maybe helpful to set the whole construction and wording into an UML diagram and to use the modeling of heterarchy worked out by Edward Lee as a helpful tool to explicate proemiality in terms of UML modeling.

Also the proemial relationship is not restricted to ontology and the distribution of hierarchical ontologies in a heterarchic framework and despite the fact that UML has no mechanisms of category change, metamorphosis and mediation it seems to be a helpful exercise to find a correspondence between the UML heterarchy diagram and the construction of proemiality which is more based on elementary terms of relationality. The heterarchy diagram is a class diagram which models the static structure of the system. Proemiality has, also it is fundamentally dynamic, its static aspects. It is this static aspect we can model with the help of the UML heterarchy diagram. A further step of UML modeling of proemiality will have to involve more dynamic models like interaction and activity diagrams.

Diagramm 3 **UML heterarchy diagram**



The conceptual graph of the UML heterarchy diagram may highlight its structure more directly.



It shouldn't be misleading to read the diagram as (methodological) hierarchy between the terms Heterarchy, Hierarchy and Entities. The additional terms Model, Frame, Port, Relation and Link are defining the structure of the interaction of the different hierarchies.

Abstract theories

Each hierarchy has its own ontology, logic, algebra, proof systems etc. To give an idea of the concept of interactivity between hierarchies let´s introduce the terminology of abstract objects or types or theories.

```
name=
  sorts s
    opns
        f: S^n --> s
        p: s^n
    eqns
       variable declaration
        L = R
```

“First of all, a name is given to the theory so that it becomes an identifiable unit binding together a number of operations and their properties into useful modules.

Keyword sorts opens the theory, listing the sorts or types of objects being defined in the abstract type.

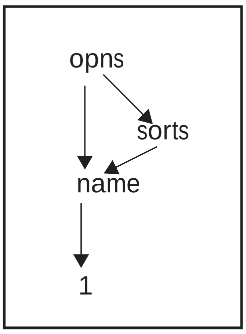
Next we have keyword opns followed by one line for each of the operations or predicates being defined in the abstract type.

Constants are seen as zero-arity operations.

The equations are defining equivalences between strings.” (Downward, p.179)

Short, the abstract theory consists of the categories name, sorts, operations, equations which build, again, a strict hierarchy of their tectonics:

```
name=
   sorts
       opns
        eqns
```



The arrows in this diagram represents conceptual dependencies in the notion of name. The notation

opns —> sorts

for example, means that:

the concept of opns varies as the concept of sorts varies.

In particular, it means that the concept of opns, the one that we have in mind, cannot be independent of the concept of sorts and neither can a particular opn be independent of its particular sort.

The notation

sorts —> name

means that the concept of sorts varies as the concept of nat0 varies.

Therefore the notion of opns varies as the notion of nat0 varies:

opns —> name.

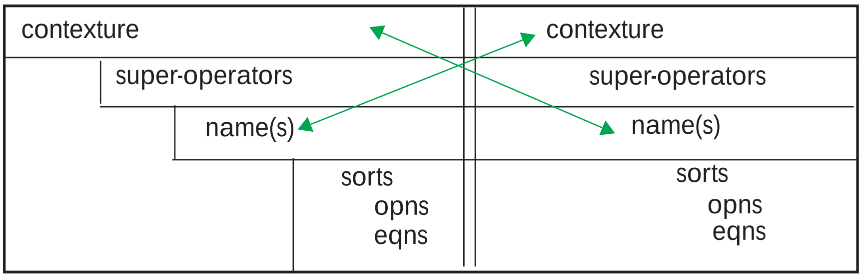
In a conceptual diagram, 1 represents the absolute. The notion

name —>1

expresses that the name notion is absolute, for it tells us that the name notion varies as the absolute varies – which is not at all.

Heterarchies are managing distributed hierarchies, therefor we are able to distribute abstract theories as such. This in itself would produce an interesting type of parallelism, architectonic parallelism. But more interesting are the interactions between hierarchies. A very conservative interaction is a one-to-one translation from one abstract theory to another abstract theory, based on morphisms. This form of interaction is basic for a successful realization of DSW applications.

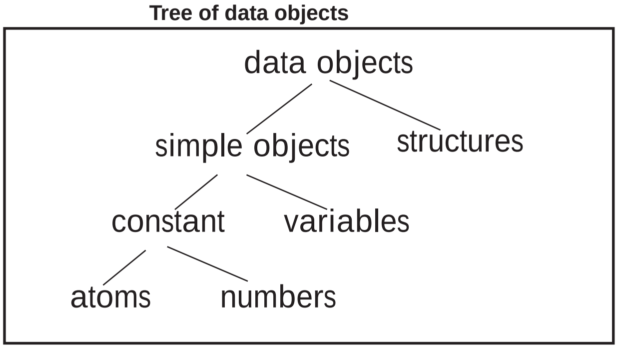
But the advantage of DSW come into play with the possibility of metamorphosis, that is the change of categories. This capability of DSW enables evolution of the system, discovery and creation of new domains, and marks the distinct difference to other architectures of a Semantic Web.



A simple example

There is an easy way of producing conflicts in a dialogical system, if e.g. L1 declares A as a simple object and L2 declares simultaneously A as a complex object, that is a structure. Obviously it is possible, in the polycontextural approach, to model this conflict and to resolve it in another logical system, say L3, this without producing a meta-system subordinating L1 and L2.

Diagramm 4 **Tree of data objects**



Furthermore, the conflict has a clear structure, it is a metamorphosis of the terms „simple object“ in L1 and „structure“ in L2. This metamorphosis is a simple permutation between sorts over two different contextures based on the chiastic structure of the mediation of the systems. But it respects the simultaneous correctness of both points of view in respect of being a „simple object“ and being a „structure“. In this sense it can be called a symmetrical metamorphosis.

Today computing is often characterized by its interactivity. But the programming languages have not changed to respond to this situation. They are still, in principle, monologic.

Ontology and the Semantic Mapping Problem

Why do we need all these abstract theories of translation and metamorphosis?

“One important issue in understanding and developing ontologies is the ontology or semantic mapping problem. We say “or semantic problem” because this is an issue that affects everything in information technology that must confront semantic problems–that is, the problem or representing meaning for systems, applications, databases, and document collections. You us always consider mappings between whatever representations of semantics you currently have (for systems, applications, databases, and document collections) and some other representation of semantics (within your own enterprise, within your community, across your market, or the world).

“This semantic problem exists within and without ontologies. That means that it exists within any given semantic representation such as an ontology, and it exists between (without) ontologies. Within an ontology, you will need to focus on a specific context (or view). And without (between) ontologies, you will need to focus on the semantic equivalence between different concepts and relations in two or more distinct ontologies.” Daconta, p. 218/19

This citation shows us the importance of mappings (translations, morphisms) between distinct ontologies. But don´t forget, these ontologies are applied, core ontologies, regional, and not general ontologies. They are parts, subsystems, instantiations of the one and only one general ontology, as formulated in GOL. This is an enormous restriction. Because, before we can interact with each other we have to agree to this general and global framework of GOL. But this is not always reasonable at all.

The mechanism of metamorphosis

DSW is introducing mappings, morphisms, translations and metamorphosis between first order ontologies, and is not concerned with regional, core ontologies only.

How does it work? The basic framework is given by the proemial relationship (Günther 1970).

"The answer is: we have to introduce an operator (not admissible in classic logic) which exchanges form and content. In order to do so we have to distinguish clearly between three basic concepts. We must not confuse

a relation

a relationship (the relator)

the relatum.

The relata are the entities which are connected by a relationship, the relator, and the total of a relationship and the relata forms a relation. The latter consequently includes both, a relator and the relata.

However, if we let the relator assume the place of a relatum the exchange is not mutual. The relator may become a relatum, not in the relation for which it formerly established the rela-

tionship, but only relative to a relationship of higher order. And vice versa the relatum may become a relator, not within the relation in which it has figured as a relational member or relatum but only relative to relata of lower order.

If:

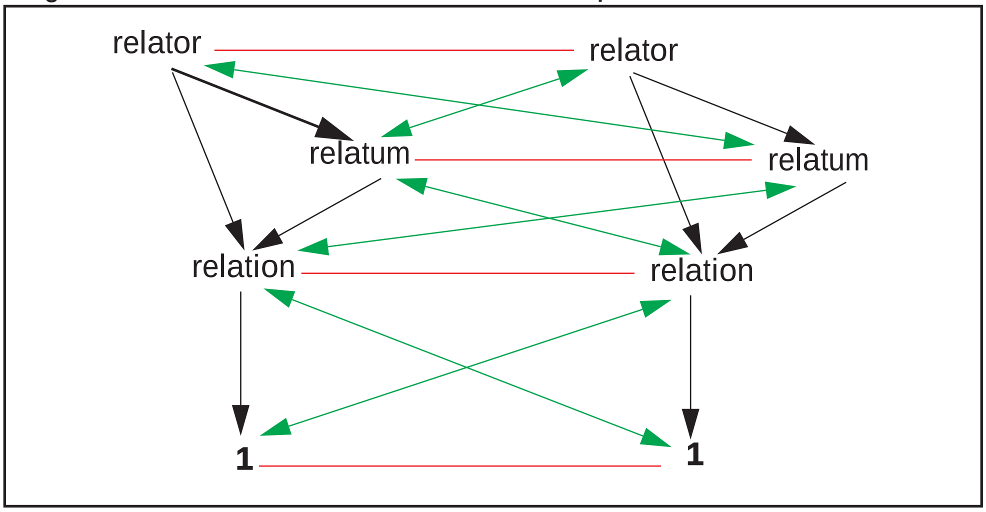
R_{i+1}(xi, yi) is given and the relatum (x or y) becomes a relator, we obtain

R_i (xi-1, yi-1) where Ri = x_i or y_i. But if the relator becomes a relatum, we obtain

R_{i+2}(xi+1, yi+1) where R_{i+1} = x_{i+1} or y_{i+1}. The subscript i signifies higher or lower logical orders.

We shall call this connection between relator and relatum the 'proemial' relationship, for it 'pre-faces' the symmetrical exchange relation and the ordered relation and forms, as we shall see, their common basis.“ Günther

Diagramm 5 **Proemial relationship**



PR: Rel(X,Y, Z,1) —> Rel(X,Y, Z,1)

Coincidence relation: id(Xi) eq Xj

Order relation: ord(Xi, Yi)

Exchange relation : exch(Xi) eq Yj

3 Development of a DSW Prototype Business Application

Increase in effectivity

This “killer application” will show a significant increase in flexibility, which goes hand in hand with an increase in speed and transparency of semantic information processing.

Attributes of a given static or stable, synchronic system

flexibility
speed
security
transformation

Attributes of dynamic evolving system

The dynamics of the semantic information processing in DSW opens up thew possibility to create new scenarios, invent new forms of interaction between business partners.

evolution
metamorphosis
co-creation
self-modification

How are the chances to develop a DSW Web Service?

Happily the Semantic Web community has developed lots of useful tools, free or commercial, to be used to develop the prototype of a DSW business application.

3.1 Web Services and Semantic Web, the classical view

http://www-106.ibm.com/developerworks/xml/library/x-ebxml/

Diagramm 6 **Web Service Scenario**

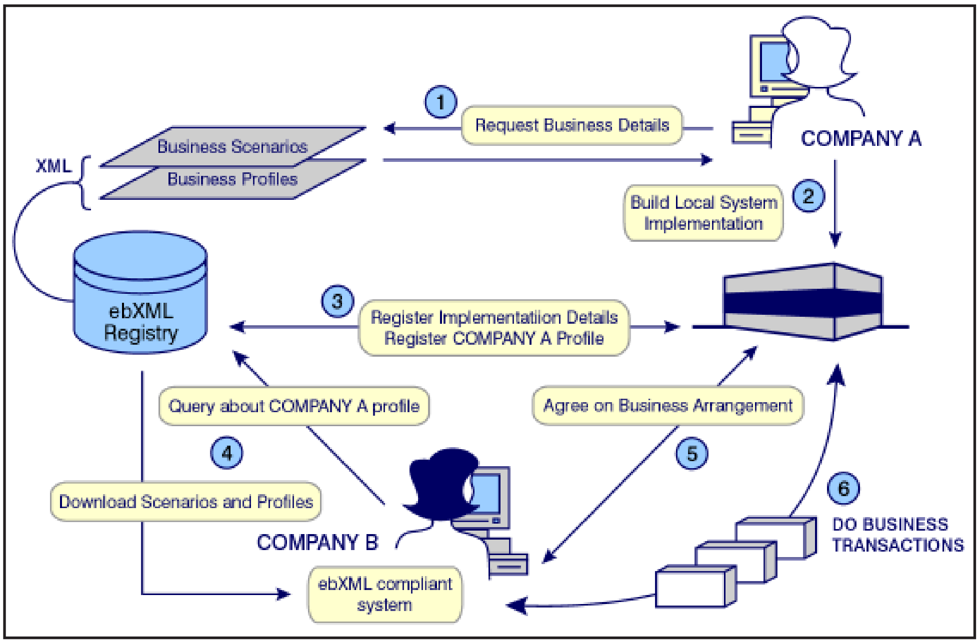
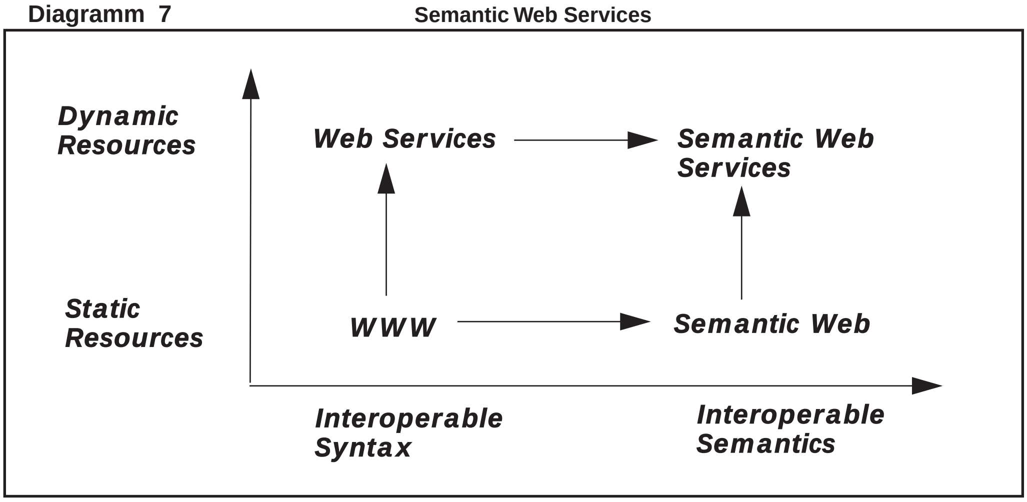


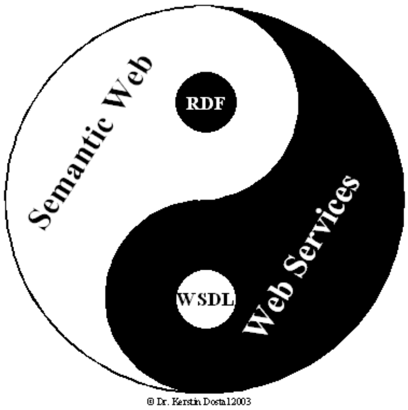
Diagramm 7 **Semantic Web Services**



Daconta, p.7

A metaphor of the internal dynamics of the components *Semantic Web, Web Services* and *RDF*, *WSDL* is given by the chiastic figure of the Ying-Yang-Picture by Wolfgang Dostal and Mario Jeckle, Semantik, Odem einer Service-orientierten Architektur.
http://www.jeckle.de/semanticWebServices/intro.html

Diagramm 8 **Ying-Yang-Picture**



3.2 A DSW business application is a DSW Semantic Web Service

THE Internet and THE WWW doesn't exist. THE WWW is a crude and awfully misleading nominalisation and abstraction from the evolving heterogeneous complexity of what we call the WWW.

THE Web Services are not a homogeneous business. They come in different and not homogeneous forms, that is, again, in heterogeneous definitions.

Heterogeneity itself is not a static term, too. It is a nominator for a flexible, loosely coupled evolving complexity of decentralized systems.

The Web is not only defined by its abstract specification but also by its use. The meaning of a sentence is not given by a catalog of administered meanings, but by its pragmatic use. And the administration of meaning is one and only one very special use of sentences and their meaning.

The picture of the situation has to be enlarged from Syntax&Semantics to, at least, Syntax&Semantics&Pragmatics (Hermeneutics).

Pragmatics or Hermeneutics is introducing different points of view, different irreducible contexts, that is, contextures, different approaches etc.

Syntax&Semantics&Pragmatics&Mediation

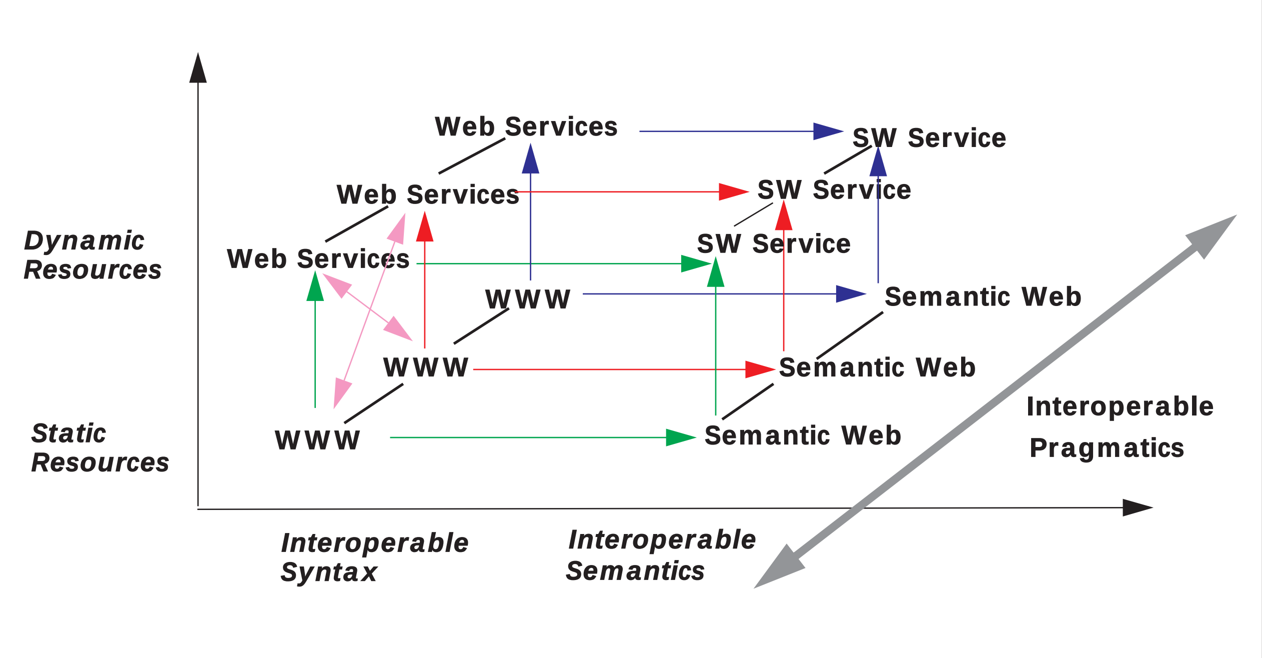
Mediation (Proemiality, Chiasm) is introducing the interlocking mechanism, the interactivity of all these different contextures.

Negotiation (Berthold Daum) is realized by human beings. But it is strongly supported by the mechanisms and rules of mediation. Insofar, DSW is not only introducing computer-aided semantics, but also several levels of computer-assisted negotiation.

This is in contrast, or better, in positive addition to Daum´s statement: *"Also obvious is that by the default the communication between observers can only be of informal nature. Consistent logical systems are only defined within a given context and, in general, cannot be used for knowledge transfer between different ontologies. The conse-*

quence is that some means of informal communication, such as natural language or heuristic mediation systems, is inevitable." Daum, p.185

Diagramm 9 Dynamic Semantic Web as a Pragmatic Web



Maybe that the structure of the metaphoric dynamism of the Ying-Yang-Picture is captured and formalized by the dynamics of distribution and mediation of contextures containing the basic quadruple of its different realizations.

3.3 What has do be developed to realize DSW?

Dynamic Semantic Web (DSW) consists in general of two main parts:
1. poly-Semantics
2. inter-Semantics or Pragmatics of mediation and navigation

Remember the Semantic Web hierarchy:
Trust, Proof, Logic Framework, Rules, Ontology, Contexts and
RDF Schema, RDF M&S, XML; Namespace and
URI; Unicode and
Digital Signature

poly-Semantics deals with the decomposition and distribution of different heterogeneous taxonomies, ontologies and their methods.

inter-Semantics deals with the interlocking mechanisms between the different heterogeneous contextures and their methods.

poly-Ontologies: Development of polycontextural ontologies

poly-Logics: Development of polycontextural logics and proof systems

3.4 How to establish a DSW system in a existing company?

It is not necessary to transform at first a business information system into a Semantic Web and in Semantic Web based Web Services. We can directly create a Dynamic Semantic Web transformation of the knowledge management system of an organisation.

What we can do on a informal non-technical level

Discover the heterogeneity of your data base.

Instead of trying to homogenize the different data systems it is more reasonable to understand them as an interacting system of heterogeneous parts. As a mediating tool to the full decomposition of a monolitic database into its heterogeneous parts, the method of *Metapattern* introduced by Pieter Wisse maybe a helpful methodology.

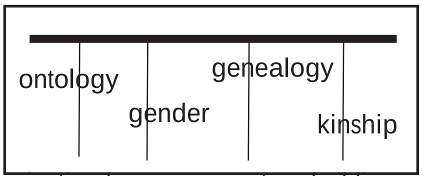
The classical Prolog example to prove an "aunt"-relationship can be decomposed from its hierarchical ontology into different situations mapped into different contextures and visualized in the metapattern.

kinship: married/not-married, in-law, aunt

gender: male, female

genealogy: parent, sibling

ontology: different/not-different



It is also possible that there is some over-determination because parent and sibling could also be part of kinship.

In Prolog all the facts belong to one ontology or to one semantic general domain or universe. All the rules are based on this mono-contextural ontology and on the corresponding logical operators AND and OR of the again, mono-contextural logic. Everything therefor is linearized and homogenized to a global or universal domain. This, if corresponding fairly with the real world situation is of great practicality and efficiency in both direction, in the case of the formal system, Prolog, and in the case of its data base.

But often, if not always, real world applications are much more complex than this. Even the fairly classical example is presupposing all sorts of facts which are not mentioned in the definition and which would belong to a different real world situation.

Instead of linearizing the above separated contextures kinship, gender, genealogy, ontology into one universal domain, for the example here represented by kinship, the polycontextural modeling is asking for an interweaving and mediating of these different contextures together to a complex poly-contexturality.

Why should we model a simple situation with highly complex tools into a complex model if we can solve the problem with much simpler tools? Simply because the classical approach lacks any flexibility of modeling a complex world. The truth is, that the simple approach needs an enormous amount of highly complicated strategies to homogenize its domains to make it accessible for its formal languages.

Decompose your data jungle into heterogeneous contextures.

Build your ontologies out of the distinct heterogeneous contextures.

Discover the interlocking mechanisms between heterogeneous systems.

Learn to navigate between different contextures and points of view.

With the help of the tools of implemented chiasms you have control and transparency about your navigations.

Navigation is more than translations (semantic mapping) or merging of local ontologies it opens up the possibility to access distinct "foreign" ontologies for cooperation which would otherwise be undiscovered.

To make business is not restricted to one business model, like the US american one. Globalization has not to homogenize different other ways of making business. Dynamic Semantic Web opens ways of mediating heterogeneous approaches on all levels of information processing.

Find leading metaphors for decomposition, mediation, navigation, negotiation which are accepted by your group and organization.

What we can do on a formal, engineering level

What are the Tools?

Research and commercial tools for creating ontologies

OntoEdit

Protege

OilEd

Evolving and self-modifying systems

Dynamics between Ontologies and contexts

Goguen on Semiotics and Category Theory

Further Extension of the Smartness of objects (data) (p. 3)

Logically it is a chiasm of Universe and sorts in many-sorted first order logics.

Heterarchies, in ontologies

Heterarchies, in logics

Heterarchies, in proof systems

Heterarchies, in taxonomies

Cybernetic Ontology and Web Semantics

There's more than one way to describe something

"No, I'm not watching cartoons! It's *cultural anthropology.*"

"This isn't smut, it's *art.*"

"It's not a bald spot, it's a *solar panel for a sex-machine.*"

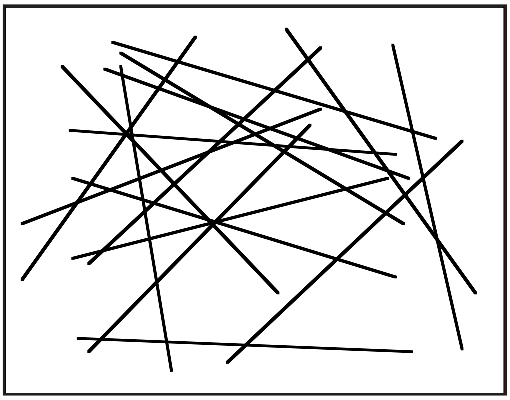
Reasonable people can disagree forever on how to describe something. Arguably, your Self is the collection of associations and descriptors you ascribe to ideas. Requiring everyone to use the same vocabulary to describe their material denudes the cognitive landscape, enforces homogeneity in ideas.

And that's just not right.

Metacrap: Putting the torch to seven straw-men of the meta-utopia
Cory Doctorow
http://ontology.buffalo.edu/

1 Life as Polycontexturality

By showing how Becoming has a component of Being as well as Nihility, he (Hegel) unwittingly laid ground to a theory of "poly-contexturality". Because, if we want to establish such a theory, we should not assume that all contexturalities can be linked together in the way a geographical map shows one country bordering on the next in a two-dimensional order. If the contexturality of Becoming overlaps, so to speak, the contexture of Being as well as of Nothingness, and the contexture of Becoming in its turn may be overlapped by a fourth contexture which extends beyond the confines of the first three, we will obtain a multi-levelled structure of extreme logical complexity.
Table I



Hegel´s logic further shows that if a plurality of contextures is introduced one cannot stop with three. In fact, one has to postulate a potential infinity of them. If one believes Hegel and there are most convincing arguments that one should - then each world datum in the contexturality of Being should be considered an intersection of an unlimited number of contextures. Table II with its seeming chaos of straight lines crossing each other at all possible angles may illustrate what is meant. Each contexture is logically finite insofar as its structure is confined to two values. But their respective ranges are infinite because one can generate, within the respective domain, a potential infinity of natural numbers. We have indicated the logical finiteness of the different contextures by having them represented by lines no longer than 2 inches.

The concept of contexturality illustrates the age-old logical distinction between identity and sameness. If I count 1, 2, 3, 4, º and so does my neighbor, then the numbers we both count are the same. However, insofar as these numbers have their existence only in the counting process, they are not identical because the two counting procedures can be clearly distinguished as having different origins in two separate organic systems. In other words: in the situation described above the sequence 1, 2, 3, 4, º turns up in two separate contextures. And no matter how far I count there is no number high enough to permit me to cross over to the psychic space of my neighbor.

Gunther, Life as Polycontexturality

New ontology, new Logics

This essay presents some thoughts on an ontology of cybernetics. There is a very simple translation of the term "ontology". It is the theory of *What There Is* (Quine). But if this is the case, one rightly expects the discipline to represent a set of statements about "everything". This is just another way of saying that ontology provides us with such general and basic concepts that all aspects of Being or Reality are covered. Consequently all scientific disciplines find their guiding principles and operational maxims grounded in ontology and legitimized by it. Ontology decides whether our logical systems are empty plays with symbols or formal descriptions of what "really" is.

The following investigation arrives at the result that our present (classic) ontology does not cover "everything". It excludes certain phenomena of Being from scientific investigation declaring them to be of irrational or metaphysical nature. The ontologic situation of cybernetics, however, is characterized by the fact that the very aspect of Being that the ontologic tradition excludes from scientific treatment is the thematic core and center of this new discipline. Since it is impossible to deny the existence of novel methods and positive results produced by cybernetic research, we have no choice but to develop a new system of ontology together with a corresponding theory of logic The logical methods that are used faute de mieux in cybernetics belong to the old ontological tradition and are not powerful enough to analyze the fresh aspects of Reality that are beginning to emerge from a theory of automata.

Gunther, Cybernetic Ontology

1.1 System Architecture in XML

"Also obvious is that by the default the communication between observers can only be of informal nature. Consistent logical systems are only defined within a given context and, in general, cannot be used for knowledge transfer between different ontologies. The consequence is that some means of informal communication, such as natural language or heuristic mediation systems, is inevitable." Daum, 185

Interactivity, between trans-contextural and transjunctional operators

Inevitably *"of informal nature"* only from the point of view of the local logical systems, but not under consideration of the more global logical operations of transjunction, which are exactly introduced for the purpose of trans-contextural interactions. Polycontexturality in the sense of Gunther, which is quite different from followers like Niklas Luhmann, is not only a *"combined system of multiple ontologies (polycontexturality) with a multileveled logic calculus"* as Daum recognized well, but also a complex system of interactivity between different contextures ruled by trans-contextural operations. These transjunctional and trans-contextural operators are operators in a exact formal sense, not only defined logically inside a contexture but also between contextures. The concept and formal definition of transjunctions had been introduced by Gunther in his famous paper *Cybernetic Ontology and Transjunctional Operations* (1962) even before he radicalized his position to a transition from multiple-valued ontologies to poly-contexturality. A more general approach of interactivity between contextures was introduced by Gunther in *"Natürliche Zahl und Dialektik"* (1972) but this concept goes back at least to the concept of an *inter-ontology* as considered in *"Natural numbers in Trans-Classic Systems"* (1970), *"The philosophical theory on which cybernetics may rest in the future may well be called an inter-ontology."* Following Gunther´s work I developed a complex philosophical and mathematical theory of interactivity in the framework of polycontexturality, developing and using notions like proemiality, chiasms, diamond strategies and co-algebras (SKIZZE-0.9.5).

We shouldn't forget to distinguish between different switches of contextures and bifurcational transitions of trans-contextural operations.

Bifurcations
Replications
Merging

1.2 Heuristic mediation of contextural switches

Also the introduction of trans-contextural operations is formal and operative, this interactions are not mechanical and predictable, but possible. Each decision a system takes to change contextures or to split into different contextures is spontaneous and creative. But this creativity is not based on chaotic “Willkür” it is not ruled but rule-guided by the trans-contextural operators. If we speak about the speechless of the counting process of natural numbers, the change from one contexture to another contexture of distributed natural numbers has to be commented, it is open to negotiation and interpretation, therefore we can speak not only about but of numbers. This way of speaking about trans-contextural changes, in other words of creativity, is not the free flouting way of speaking reclaiming deep insights about negativity and irrationality as opposed to mechanical rule-systems, but a new interweaving and interlocking process of speaking, conceptual writing and formal notations.

Rational decision-making of creative systems is in itself a polycontextural procedure, it is an interlocking mechanism of cognition and volition, a double gesture and not reducible to ultimate meta- or proto-systems.

2 Heideggers radical deconstruction of ontology

2.1 self-modifying media

Gunther´s chain of notions deliberating thinking from ontology:
ontology
meontics
poly-thematics
poly-contexturality
morphogrammatics
kenogrammatics
proemiality
negative languages

2.2 Freezing and melting ontologies

Ontology based web semantics, Semantic Web, is in danger to freeze the processuality of the development of the Internet.

Classical ontology, with pluralities in score and upper dimensions are not prepared for self-referential processes: the arrival of Web Semantics in the Internet is changing the Internet in introducing itself. It is a self-modifying media.

Heidegger, Whitehead, Gunther on self-modifying media processuality.

Web Semantics as based on ontologies is accepting classical logic in its Proof procedures as an ultimate system of rational reasoning. But logic itself is based on ontology, maybe analytic philosophy has forgotten this. Ask Quine.

Conflicts between flexibility, navigation and normation.

2.3 The world as a grid of upper-level ontologies

The significance of Heideggers questioning of classical ontology has a very practical reason for Web Semantics: It opens up the possibility of a multitude of interacting fundamental ontologies, that is of upper-level ontologies. Aristotelian ontology as proposed by the "hierarchy movement" of Web Semantics is blind of its restriction to one and only one contexture.

The world as the place in which a historical event like the development of Aristotelian ontology is possible does not consists of ontological entities, neither Urelements nor sets. The world gives or opens up the space and the fundamental possibility of ontologies of different types. Therefore, the loci where different ontologies are placed, positioned and situated are in a radical sense empty of any ontological, logical, semantical, arithmetical etc. meaning; they are empty places, written, inscribed as kenograms. The world as a kenogrammatic grid offers a structure for the distribution and interaction of different ontologies. Kenogrammatics, therefore, is the study of the structure and behavior of these grids of empty places. Trivially, because I am using a language to express these thoughts which is highly hierarchical it is natural to think that now the term "world" is the ultimate being. But this is wrong insofar as the whole mechanism, say of kenogrammatics, which is inscribed in a "trans-mathematical" formalism, shows a totally different behavior, that is a heterarchical in contrast to a hierarchical.

2.4 Ontology and logics of multi-media

2.5 Morphogrammatics of XML

3 Ontologies in different fashions

3.1 many-sorted logics

3.2 fibred category systems

3.3 polycontexturality

Fibres and navigation

4 Revival of classic ontology in Web Semantics?

The four systems concerned by this project provide this structure in very different ways and with different conceptual 'textures'. For example, the AGROVOC and ASFA thesauri put "aquaculture" in the context of different thesaurus hierarchies: according to AGROVOC the terms more specific than "aquaculture" are "fish culture" and "frog culture", whereas in ASFA they are "brackishwater aquaculture", "freshwater aquaculture", "marine aquaculture". Two different contexts relating respectively to species and environment point of view. With such different interpretations of a term, we can reasonably expect different search and indexing results. Nevertheless, our approach to information integration and ontology building is not that of creating a homogeneous system in the sense of a reduced freedom of interpretation, but in the sense of **navigating alternative interpretations**, querying alternative systems, and conceiving alternative contexts of use.

To do this, we require a comprehensive set of ontologies that are designed in a way that admits the existence of many possible pathways among concepts under a common conceptual framework. This framework should reuse domain-independent components, be flexible enough, and be focused on the main reasoning schemas for the domain at hand. **Domain-independent, upper ontologies** characterise all the general notions needed to talk about economics, biological species, fish production techniques; for example: parts, agents, attribute, aggregates, activities, plans, devices, species, regions of space or time, etc. On the other hand, the so-called core ontologies characterise the main conceptual habits (schemas) that fishery people actually use, namely that certain plans govern certain activities involving certain devices applied to the capturing or production of a certain fish species in certain areas of water regions, etc.

Upper and core ontologies provide the framework to integrate in a meaningful and intersubjective way different views on the same domain, such as those represented by the queries that can be done to an information system.

http://www.loa-cnr.it/Publications.html

Some links:
http://www.ifomis.uni-leipzig.de/People/People.html
http://ontoweb.aifb.uni-karlsruhe.de/
http://www.websemanticsjournal.org/

Ontology Groups
http://www.cs.utexas.edu/users/mfkb/related.html

Flexibility ruled by an upper framework?

"To do this, we require a comprehensive set of ontologies that are designed in a way that admits the existence of many possible pathways among concepts under a common conceptual framework."

Why should the common *"conceptual framework"* be thought in a hierarchical way? There are two possible ways of dealing with the task of finding an "upper ontology" which is "domain-independent" and so on. One is the classical way of *hierarchy*, as well established and studied and transformed to new applications like the search for a semantics of the Web. The other possibility which is able to cover all mentioned attributes of the "upper ontology" is offered by the strategy of *heterarchy* and *proemiality*. Heterarchy is neither hierarchy nor anarchy.

The classical approach seems to guarantee a good flexibility on the core base, the regional ontologies, by stabilizing its concepts on the upper level of the "common conceptual framework" which includes basic ontological and logical terms like "*parts, agents, attribute, aggregates, activities, plans, devices, species, regions of space or* *time, etc."* but the game doesn't stop here. What are "*parts*" from one vantage point can be "*wholes*" from another, "*agents*" can be understood as "*attributes*", "*activities*" as "*plans*", etc.

"Nevertheless, our approach to information integration and ontology building is not that of creating a homogeneous system in the sense of a reduced freedom of interpretation, but in the sense of ***navigating alternative interpretations****, querying alternative systems, and conceiving alternative contexts of use."*

What is the range of navigation? To navigate between alternative interpretations sounds quite polycontextural. But where are the limits, if not in the supposed basic logic and how does the navigation work? What are the rules of navigation? Are they ontological or logical or spontaneous?

Navigation and negotiation

The conflicting restless of interactivity between different ontology can come to a rest in a common upper ontology based on negotiation and agreement. But this upper ontology turns out to be a lifeless abstraction. Another result of negotiation could be a mediation between different ontologies which accepts the differences between the ontologies but is able to find intermediating rules of interactivity. Only in well established and simple situation we can discover a translation from one ontology to an other ontology conserving their ontological categories, like sorts to sorts, operations to operations, and so on.

Kenogrammatics as a common base of different ontologies

Different ontologies, if not anyway based on a common upper ontology and common first-order logic, have, even if they are incomparably different, irreducible to a common ground, one thing in common, they have, each for itself, a position. They take a position, occupy a position, a locus, where?, in some very general sense, in the world. This does not mean that they have in common a general concept of the world. This would be released by a general ontology and logic. But even general ontology and logic are taking place, are placing themselves in the world. It also does not mean that they share in abstracto a common empty locus. Each ontology is based on its own locus. And also the loci are empty they are not the same.

These loci have no attributes, no predicates, no relations, no processualities etc. nevertheless they exist, in a non onto-logical sense, but give place for ontology and logic, and ontologies and logics. There is also not a single primordial place, like nothingness or ultimate emptiness, there is multitude of empty places, differentiated between the same and not the same, in a non-logical sense.

These monsters of negative conceptuality are inscribed as kenograms (kenos, gr. empty). The grid of kenograms is the non-basic base of the distribution and mediation, the interactivity and navigationality of different ontologies.

Formal ontology, category theory and kenogrammatics

Formal upper ontologies are often described in terms of set theory. A more general approach would be to formalize ontologies with the means of category theory. The most basic and abstract distinction in category theory is the distinction between morphisms and objects.

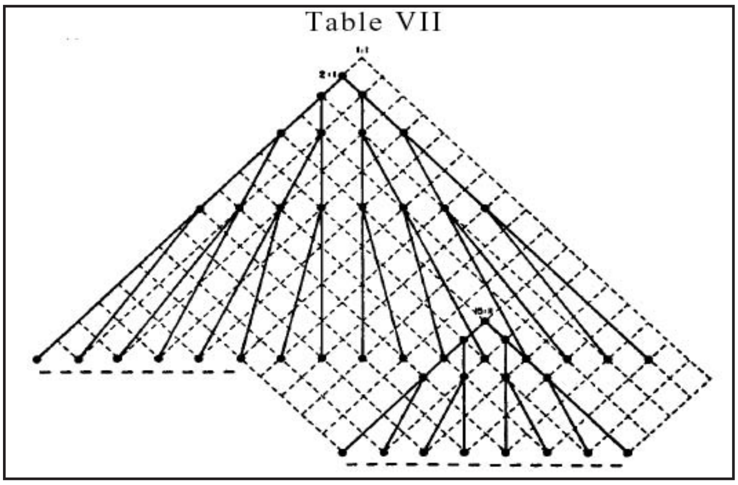
With this, another introduction of the empty positions, kenograms, of formal upper ontologies can be offered. Two ontologies may be conceptually different in the sense that one ontology is based on its objects, similar to the set theoretic based ontology, and the other one is based on its morphism, like a more processual and dynamic ontology. What are objects in one ontology are morphisms in the other one. This maybe a clue for a translation between both. This translation could be done by, again, a category theory, which is based more on objects or more on morphisms. Obviously, we would establish with this procedure some of the well known infinite regresses of meta-language constructions.

With the help of the diamond strategies we can ask for a "common ground" outside of the dichotomy of category theoretic objects and morphisms. To characterize the position of each formal upper ontology we look for a situation in which there are neither objects nor morphisms, where the whole dichotomy is rejected. This place of emptiness of objects and morphisms is accessible as kenogrammatics, that is, as the kenogrammatics of the play of objects and morphisms.

Kenogrammatic systems are not meta-languages but in some sense proto-inscriptual grammars.

What is wrong with kilts?

This happened recently in the funny conflict of taxonomic notions and cultures between Scotland and the EU. Kilts are skirts, skirts are connected to female, male is connected to trousers, therefore Kilts are female clothes. What to do? Introduce exceptions. In a few turns the ontology consists of thousands of exceptions and some simple general classificatory rules will be left. The other necessary strategy is to ban the object. Therefore nearly all sorts of Camembert cheese have to disappear. This madness happens automatically if we take distinctions like male/female and skirts/trouser as substantial and not as functional and depending on contexts. And how could the European taxonomy run together with one of the many Asian taxonomies? Taxonomy and ontology without ethnology is behind globalization movements.



Is this not exactly the situation of XML? XML tries to be a general language not subsuming the thousands real world languages of the Internet but enabling and supporting this diversity.

But how can this be done if XML is not more than a simple tree?

Web Semantics: Science, Services and Agents on the World Wide Web

This interdisciplinary journal focuses on research at the intersection of three major research areas: semantic web, agent technology and grid computing. We call this interdisciplinary field Web semantics. Web semantics investigates and develops the standards, ontology's, protocols and technology that contribute to the development of a knowledge-intensive and intelligent service Web. This is often referred to as the second generation of the Web.

Background

The data in computers exists in a bewildering variety of mutually incompatible forms and ever more intense efforts are needed to smooth the process of data integration. The most important such efforts lie in database standardization achieved through the construction of benchmark taxonomies into which all the classification systems pertinent to a given domain would need to be translated only once. Benchmark taxonomies can ensure that all databases calibrated in their terms would be automatically compatible with each other.

'Ontology' is the name given by information scientists to the construction of such benchmark taxonomies. This name was chosen in reflection of the fact that in building such taxonomies one is confronted by issues with which philosophical ontologists have grappled since Aristotle's day, issues which have once again moved into the center of contemporary philosophy under the heading 'analytic metaphysics.'

Information systems ontology has implications beyond the domain of data integration. Its methods are used for purposes of information retrieval and extraction from large corporations and libraries (for example of medical or scientific literature). These methods are currently being applied to the problems of navigation on the Internet in work on the so-called Semantic Web. They are used as a basis for work on natural language processing and automatic translation, in enterprise integration, and, most significantly, as a means of integrating the results of inquiries in neighboring scientific fields – for example when inquiries in computational chemistry or structural biology need to be cross-calibrated with the results of inquiries at higher (for example medical or epidemiological) levels of granularity, as for example in the work of the Gene Ontology Consortium .

http://ontology.buffalo.edu/proto-ifo/

Afortunadamente, la situación es hoy muy diferente, gracias a los trabajos pioneros de tres caballeros. Gothard Gunther, un filósofo, ahora profesor en la Universidad de Hamburgo, que desarrolló el más fascinante sistema lógico de valores múltiples [Gunther 1976], muy diferente de los de Tarsky, Quine, Turquette y otros. Lars Lofgren, un especialista en lógica de Lund, Suecia, que introdujo la noción de 'autología',1 es decir, de los conceptos que pueden ser aplicados a sí mismos y que, en algunos casos, se necesitan a sí mismos para existir. Me ocuparé de estos puntos en un momento. Finalmente, Francisco Varela, que está sentado aquí mismo y que, como ustedes saben, expandió el cálculo de indicaciones de G. Spencer-Brown transformándolo en el cálculo de la autoindicación [Varela 1975].

http://ladb.unm.edu/econ/content/cuadeco/1997/january/principios.html

Mr Latifs Laundrette

Many Sorted Logic: Frequently one has a pile of clothes with many different sorts of washing instructions (different temperatures or spin speeds) but not enough of any type to make a full load. Use of Many Sorted Logic will enable all these clothes to be washed together in a single universe (washing machine) whilst preserving the integrity of the clothes.

http://www.aisb.org.uk/hacker/1998.html

Ontology, the new obsession

1 On the General Ontological Foundations of Conceptual Modeling

Diagramm 10 **Aristotelian Hierarchy**

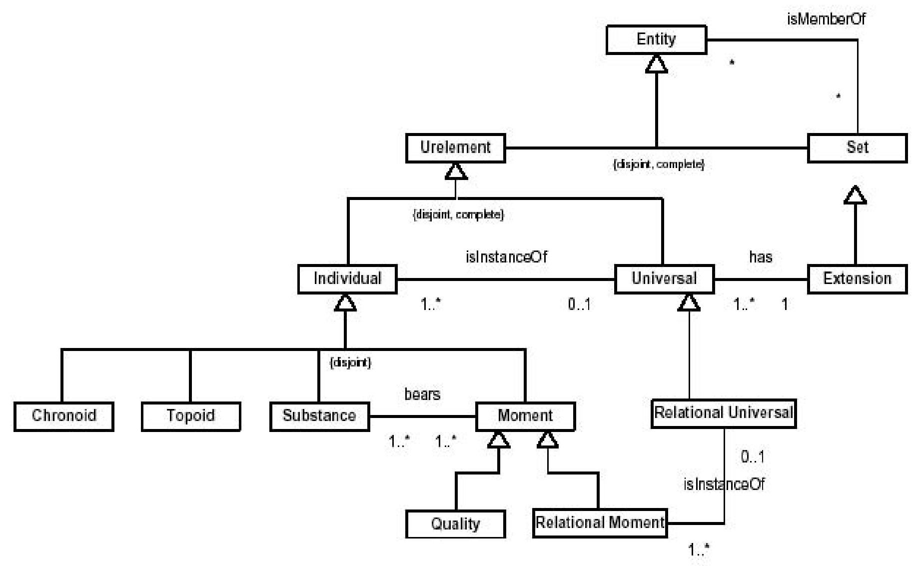
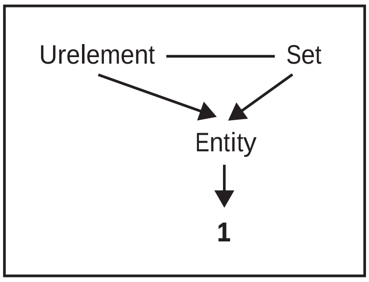


Fig. 1. A UML class diagram describing the basic concepts of GOL.

2 Urelements and Sets

One of the basic distinctions of GOL is the distinction between *urelements* and *sets*. We assume the existence of both urelements and sets in the world and presuppose that both the impure sets and the pure sets constructed over the urelements belong to the world. This implies, in particular, that the world is closed under all set-theoretical constructions. Urelements are entities which are not sets. They form an ultimative layer of entities without any set-theoretical structure in their build-up. Neither the membership relation nor the subset relation can unfold the internal structure of urelements.
In GOL, urelements are classified into two main categories: *individuals* and *universals*. There is no urelement being both an individual and a universal.

Conceptual graph of the basic triple (Entity, Urelement, Set) and its uniqueness 1.



Comments
"We assume the existence of both urelements and sets in the world" in doing this, do "we" belong to this world or not?
"This implies, in particular, that the world is closed under all set-theoretical constructions." Maybe we can live with that. But didn't we not just learned that, to develop a non-onto-theo-logical ontology, we should questioning the very presupposition of classical ontology, namely its presupposed "world". Today, it is not nonsensical to ask "Which world do you mean?" There is surely one world which is build up of Ur-Elements and Sets, but what´s about the other worlds? And what´s between these worlds? And what happens if we cannot resist to clone this very concept of Ur-Elements, too?
"Ur-Elements", are they not Kant´s Ding an sich-type monsters?
What is your Urelement is my "chronoid", why not?
In the world of Ur-Elements there is no liveliness and metamorphosis. All changes in this world concept are based on Ur-Elements, which are stable and eternal.

Why do we need set theory to build ontologies? With this decision we are loosing the chances of a much more flexible modeling say by category theory and combinatory logic. Not to speak of the possibilities opened by polycontextural logics and its first order ontologies.

3 Formal Ontology and First Order Logic, revisted

The new, post-analytic movements towards a reformulation of ontology goes back to Brentano, Meinong and a restricted reading of Husserl and is restoring an old discussion about the relationship between ontology and logics which went lost during the success of formal logic and later by the dominance of computer science paradigms. This discussion is extensively documented in the German literature of the 50th.

Gotthard Gunther, again, was a lonely voice, in America and Germany, to empathize the importance of the connection between ontology and formal logic after the early discussion disappeared from the academia. But in contrast to the new neo-Aristotelian movement, Gunther was able to connect his work to another, still not recognized movement of ontology, the transcendental ontology of Husserl, called phenomenology and the deconstructive efforts to surpass the limits of classical ontology by Martin Heidegger, as a radical non-Aristotelian ontology, called polycontextural theory going hand in hand with an equal non-Aristotelian logic. Not surprisingly Gunthers work was intrinsically connected with attempts to formalize Hegels dialectics and to develop a *“Cybernetic Theory of Living Systems”* at the BCL.

His ontology is therefore not “conservative” and “descriptive” but “constructive” and “revolutionary” thematizing not so much what just is, as given or even natural, but what has to be done, the artificial, and what is primordially interwoven with time, the ontology of living tissues, natural and artificial, and beyond.

> The present paper outlines a formalisation of elementary formal ontology. In contradistinction to a *material* ontology, *formal* ontology is concerned, not with the specification of the constituents (individuals, properties and relations) in a particular domain or region of the world, but with the axiomatisation of the most general, pervading categories that partition and shape reality as a whole.
>
> As Barry Smith has pointed out, the use of the qualifier ”*formal*” is liable to give rise to a fundamental misunderstanding: formal ontology is not merely the application of formal-logical methods to the study of metaphysics.
> Rather, the very success of mathematical logic has led to a “running together of the formal and formal logical”, and ultimately to a confusion of ontology with logic and with the study of the structure and semantics of artificial languages, at least as far as much philosophy in the analytic tradition is concerned.
> Only fairly recently, in an influential collection of studies in the philosophy of Brentano, Husserl and their followers was there triggered a revival of a scientific metaphysics in the Aristotelian tradition that is not a mere appendix to predicate logic and set theory.
> Indeed, the *formal/material* distinction has a wider range than just the specialist area of mathematical logic; it reflects the general opposition between form and matter in the realm of things as well as in the realm of truths. Just as formal logic studies the abstract relations between propositions, so formal ontology is concerned with the formal relations between entities.
> Formal-ontological *constants* are like formal-logical ones insofar as their meaning can be characterised purely in terms of operations and transformation rules. Formal *relations* (such as parthood, dependence, but also identity and instantiation) are not mediated by ties (accidents, moments) of any sort, in contrast to material relations (such as “being a parent of”, “being the moon of”, and so on), but hold directly of their relata. Formal properties and relations can therefore be instantiated by objects in all material domains or spheres of being.
>
> That is why formal ontology as the study of formal categories can justifiably be claimed to be the most general possible theory about the world.

Thus it should not come as a surprise that formal ontology is *realist* rather than *conceptualist*, inasmuch as it is an inquiry into the general features, the real aspects of the denizens of the world out there, and not into the basic characteristics of the conceptual framework which we happen to be equipped with as members of the human species or a particular ethnic group.

Formal ontology is *conservative* or “descriptive” instead of revolutionary or “revisionary”, insofar it takes - salva consistentia - our everyday ways of speaking about the world at face value as the most detailed and corroborated description of reality available, but proceeds to theoretical revisions of so-called commonsense if required for the sake of coherence and, above all, scientific adequacy. p. 2-3

Formalised Elementary Formal Ontology, p. 2-3
ISIB-CNR Internal Report 3/2002
Padova, Italy, June 2002
Luc Schneider, MSc, MA

4 A Four-Category Ontology
4.1 Universals and Particulars
Like Lowe ([70], pp. 203-209) and Smith ([81], p.291, & [117]), I adopt a four-category ontology based on Chapter 2 of Aristotle’s *Categories* ([3], 1a, 20 ?), which classifies possibilia according to whether they are:
1. said of or attributed to a subject or not, i.e. universals and particulars,
and
2. inhering in a subject or not, i.e. accidents and substances.
ibd. p. 36

4 Contributions to the Axiomatic Foundation of Upper-Level Ontologies

Wolfgang Degen, Heinrich Herre

Unary basic symbols.

$Ur(x)$	(urelement)	$Set(x)$	(set)	$Ind(x)$	(individual)
$Univ(x)$	(universal)	$Mom(x)$	(moment)	$Subst(x)$	(substantoid)
$Sit(x)$	(situoid)	$Top(x)$	(topoid)	$Chron(x)$	(chronoid)

Symbols for binary and ternary basic relations.

$\in$	(membership)	$::$	(instantiation)	$:\succ$	(inherence)
$<$	(part-of)	$:\mu(x,y,z)$	(rel. part-of)	$:\sqsubset$	(framing)
$: ass(x,y)$	(y ass. to y)	$:\rhd$	(is contained in)	$: occ(x,y)$	(x occupies y)
$ext(x,y)$	(is extension)				

An ontological signature ? is determined by a set S of symbols used to denote sets (in particular extensional relations), by a set U of symbols used to denote universals, and by a set K of symbols used to denote individuals. An ontological signature is summarized by a tuple ? = (S ,U;K).

The syntax of the language $GOL(\Sigma)$ is defined by the set of all expressions containing the atomic formulas and closed with respect to the application of the logical functors $\vee, \wedge, \rightarrow, \neg, \leftrightarrow$, and the quantifiers $\forall, \exists$. We use untyped variables $x, y, z, \ldots$; terms are variables or elements from $\mathsf{U} \cup \mathsf{S} \cup \mathsf{K}$; r, s, s_i, t denote terms. Among the atomic formulas are expressions of the following form: $r = s$, $t \in s$, $r :\succ \langle s_1, \ldots, s_n \rangle$, $v :: u$, $t{:} \sqsubset s$, $t{:} \rhd s$.

Axioms of Basic Ontology

(a) Sort and Existence Axioms

1. $\exists x(Set(x))$,
2. $\exists x(Ur(x))$
3. $\forall x(Set(x) \vee Ur(x))$
4. $\neg\exists x(Set(x) \wedge Ur(x))$
5. $\forall x(Ur(x) \leftrightarrow Ind(x) \vee Univ(x))$
6. $\neg\exists x(Ind(x) \wedge Univ(x))$
7. $\forall xy(x \in y \rightarrow Set(y) \wedge (Set(x) \vee Ur(x)))$

(b) Instantiation

D. $ext(x,y) =_{df} Univ(x) \wedge Set(y) \wedge \forall u(u :: x \longleftrightarrow u \in y)$.

1. $\forall xy(x :: y \rightarrow Ind(x) \wedge Univ(y))$
2. $\forall x(Univ(x) \rightarrow \exists y(Set(y) \wedge \forall u(u \in y \longleftrightarrow u :: x)))$

(c) Axioms about sets

1. $\forall uv\exists x(Set(x) \wedge x = \{u, v\})$
2. $\{\phi^{Set} \mid \phi \in ZF\}$, where ϕ^{Set} is the relativization of the formula ϕ to the basic symbol $Set(x)$.[3]

KIF adopts a version of the Neumann-Bernays-Gödel set theory, GOL assumes ZF set theory.

6 Conclusions

The development of an axiomatized and well-established upper-level ontology is an important step towards a foundation for the science of Formal Ontology in Information Systems. Every domain-specific ontology must use as a framework some upper-level ontology which describes the most general, domain-independent categories of reality. For this purpose it is important to understand what an upper-level category means, and we proposed some conditions that every upper- level ontology should satisfy. The development of a well-founded upper-level ontology is a difficult task that requires a cooperative effort to make signicant progress.

Formal GOL, referring to the ontology of Aristotle seems to be specially conservative and seems to have no connection to the new trends of digitalism and computionalism. Also, it lacks an understanding and application of Category Theory as a description and construction language.

4.1 Formal GOL and the nature of Digital Metaphysics

Eric Steinhart

"More precisely, programs are ordering of abstract transformations of abstract states of affairs. Their executions are series of concrete transformations of concrete states of affairs, that is, histories. The set of all executions of a program is a nature. Programs have truth-values, and a program is true of a thing exactly to the extend that its nature is coextensive with the nature of the thing."

4.2 Formal GOL and the Metaphor of Cellular Computation

Ali Mohammed"
Computationalism

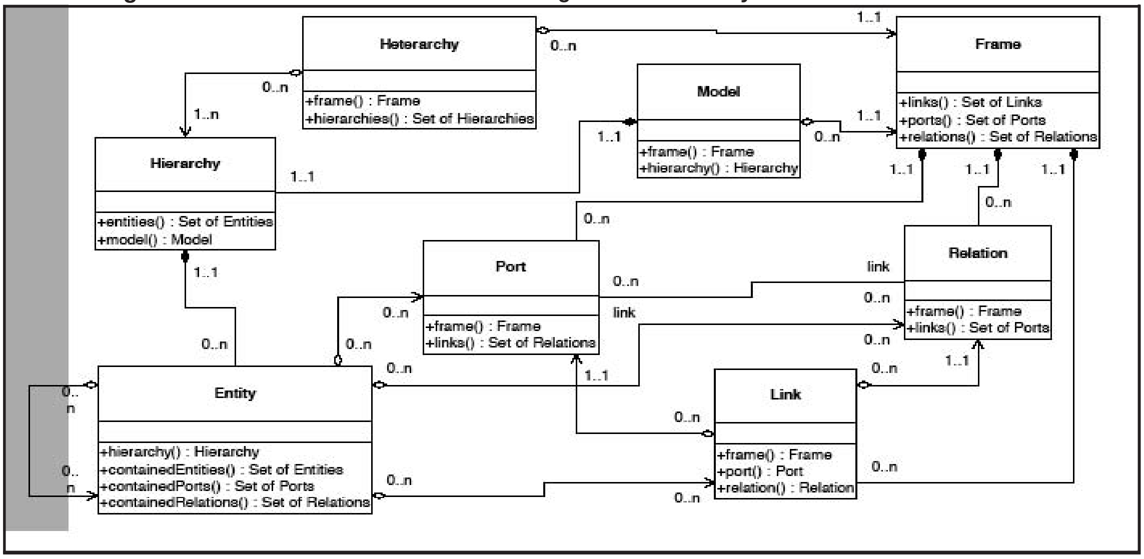
Nature as a CAM

Heterarchies, another obsession

1 Orthogonalizing the Issues

Edward A. Lee, UC Berkeley

Diagramm 11 **UML Diagram of Heterarchy**



Heterarchy
Hierarchy, Frame
Model
Port, Relation
Link
Entity

????
Polycontexturality
Mono-contexture, Proemiality
Type of Proemiality
Type of Metamorphosis, Relations: Order-, Exchange-, Coincidence
Transjunction
Objectionality
Port:: loci of mediation

1.1 Distribution of hierarchical ontologies on heterarchies

1.1.1 UML diagrams of UML diagrams

Heterarchization of the hierarchy ontology UML diagram in respect to the heterarchy UML diagram.

1.1.2 Disseminated formalisms

The same procedure can be applied to the heterarchization of the formal hierarchical ontologies. There whole formalism has to be distributed, including the specific ontological and the general logical definitions.

The kernel of GOL

Dissemination of the kernel of GOL

Interactivity, metamorphosis and simultaneity of different GOLs

1.1.3 Towards poly-GOL

Typology
Algebraic GOL
Co-Algebraic GOL
Metamorphic GOL; Proemiality of algebraic and co-algebraic GOL
Kenomic GOL

1.1.4 Gunther´s Hierarchy of First Order Ontologies

Gunther is developing his First Order Ontologies (FOO) an the basis of the distinction between logic and ontology using two further distinctions “affirmation/negation” and “designation/non-designation” in many-valued logical systems.

> “We shall define an *ontology* as a structural system in which the distinction between designating and non-designating values is inapplicable, and witch is determined by nothing else but the number of values available. In an ontology all values designate. However, if values permit a division between designation and non-designation, the system in question may be considered a *logic*.”, Gunther 1968, p. 37/p.149

By a first order ontology he understands a *“theory of Being (ontos on) in contrast to the plurality of second order ontologies referring to the plurality of classes of existing objects.”* These second order ontologies are referred in philosophy as *regional* ontology (Regionalontologie), in contrast to fundamental ontology or simply ontology.

This many-valued ontology allows the additional distinctions of mono-, dia- and poly-thematic ontologies and reflectional mappings, with and without repetitive redundancy, of the ontologies in the logical systems.

> “No self-reference is possible unless a system acquires a certain degree of freedom. But any system is only free insofar as it is capable of *interpreting* its environment and *choose* for the regulation of its own behavior between different interpretations.” p. 44/p.156

As we can see, the term “self-reference” is not understood as in the tradition of the famous *Circulus Creativus* of Heinz von Foerster, both at the BCL, or the re-entry figure of George Spencer Brown, but in the tradition of complex transcendental logics as introduced by Kant, Hegel, Schelling and further developed also by the Soviet cyberticians (Levebvre). On the other hand, the second order circular interpretation of self-reference is not excluded, it is a quite special case of the complex reflectional mapping process of a living system reduced to a cognitive system.

Gunthers approach to self-referentiality in the framework of polycontexturality involves simultaneously cognitive and volitive procedures. It is not enough to make the statement *“Living systems are cognitive systems, and living as a process is a process of cognition.”* Maturana, p.13, 1970. Excactly, because we learn nowhere anything in the texts of Second Order Cybernetics about the ontology and the logics of the “as”-operator of this statement concerning with system and process, living and cognition. As far it is an interesting, and at its time, a provocative statement, but it is still “magic” –and not operational. Why not?

> “However, there is a fundamental distinction between the idea of a self-referential universe as it was conceived in a former mythical philosophy of nature, as, for example, in Fechner´s “Weltseele”, or, if we want to go back to the most ancient Scriptures of mankind, as in the saying of the Chhandogya Upanishad “Self is all this”, and the idea of self-referentiality as we conceive it here. In the mystical philosophy of nature it was assumed that the universe was self-referential as a whole–because no distinction was made between auto-referentiality and self-referentiality. This led, if a living system was considered to be a (complete or incomplete) structural replica of the Universe, automatically to the holistic interpretation of an organism. In contra-distinction to this tradition we maintain, however, that, althoug the universe as a whole may be considered to be auto-referential, it can have the property of *self*-reference only in preferred ontological locations of suitably high complexity structure.” Gunther, Natural Numbers, p. 32/33; p. 250/251

What to do with all that for a theory of semantics for a Semantic Web?

The Internet is not given, its elements are not entities; the Internet has to be read and its elements have to be interpreted. Interpretation involves freedom to chose a thematization, a perspective of cognition, it involves not only an observer but hermeneutical procedures. Otherwise we understand by the Internet a system of being to be studied and classified by means of ontology in the very sense, also modernized and formalized, by the Aristotle-Leibniz tradition.

The project *Semantic Web* is a challenge for a formalized and operative hermeneutics. Set-theoretical and mereological ontology is mapping only an extremely static and one-sided hierarchical aspect of the “living” tissue of the Web.

A multitude of interacting hierarchies is a question of cognition and volition interpreting the textures of the Web.

Translations from one language to another are not based on a common natural ur-language, but on the co-creative interplay between different languages, natural or artificial.

Ontology in the sense of GOL is “subjectless”. It is a theory of being excluding self-referentiality by definition. Therefore it is a monolitical theory of what is, of objectivity without any freedom of interpretability. Again, this is very useful for subjectless domains, but useless, if not dangerous, in all senses of the word, for worlds including subjects. Today it seems to be quite tricky to find such a subjectless world. Especially if we are forced to ask who is producing this ontology of a subjectless world and even our robots are asking for more "subjectivity". Ontology as *"the most general possible theory about the world"* is fundamentally incomplete. It is incomplete on a semiotical level., incompleteness of ontology and incompleteness of logic, and an a graphematical (grammatological) level, it is not only kenogrammatically incomplete but blind for its own kenogrammatics.To insist on a *realist* point of view to build a general ontology in contrast to a *conceptualist* understanding of ontology allowing some interpretability of the world is a decision which can not be justified easily using scientific and philosophical arguments. At least this decision is not part of the “new” formal ontology. At this point we are confronted with questions of Power and epistemological fundamentalism.

Dynamic Semantic Web

Ontologies: Their Glory and the new bottlenecks they create.
The bottlenecks of the current web techology create
• problems in searching information,
• problems in extracting information,
• problems in maintaining information, and
• problems in generating information. Dieter Fensel

The advent of Web services, and the Semantic Web described by domain ontologies, highlight the bottleneck to their growth: ontology mapping, merging, and integration.
Stephen L. Reed and Douglas B. Lenat, Mapping Ontologies into Cyc

The Dynamic Semantic Web has to deal with the *dynamics* of the Web.

The Web is at a first glance at least distributed, dynamic, massive and an open world (Heflin, Hendler).

What is the Semantic Web? It is "a vision of the future in which the "web of links" is replaced by a "web of meaning" where the meaning is machine readable.

To introduce a web of meaning, *ontologies* appears as the main concepts and tools.

Therefore, the first job of DSW is to develop a dynamics of ontologies.

1 SHOE: Dynamic ontologies on the Web

"Dynamic ontologies on the Web" is the title of an approach by the authors of SHOE.

The dynamics of SHOE works with the constructs of ontology definition, modularization, revision and versioning with the help of the techniques "ID", "USE", "RENAME", etc. as methods of Evolution and Integration of ontologies.

All these concepts are realized and have their semantics in the framework of Hierarchy ruled by FOL.

Problems

Introduction, Navigation, Negotiation and Integration are restricted to hierarchical Unification.

2 Polycontextural Dynamics

DSW can not be realized by restricting it to this kind of ontological dynamics. In contrast to the mono-contextural approach of SHOE, DSW has to be realized in the framework of Heterarchy of polycontextural logics and ontologies.

How can we map ontologies onto Heterarchies?

A first but useful explication of the concept Heterarchy is given by the UML heterarchy diagram.

2.1 Heterarchies

Hierarchies are distributed and mediated by the rules of heterarchy.

Each hierarchy contains ontologies in the classical sense.

2.2 Proemial relationship

The mechanism of the interplay between different ontology is realized by the proemial relationship.

2.3 Poly-Semiotics

Signs in ontologies

signs relative to objects

signs relative to signs

signs relative to users

user: modeller, conceptionalist, instance etc.

Interaction between semiotics based on the immanent difference of "subjectivity" of the users between I- and Thou-subjectivity.

This leads to a post-Peircean semiotics of chiastic nature.

3 Short comparition of SHOE and DSW

Is it possible to develop a Semantic Web with its ontology and logics without having to forget and to deny everything we learned from philosophy, linguistics, logics, semiotics, grammatology and AI in the last century?

3.1 Multiple inheritance

Multiple inheritance can easily be modelled in SHOE:
<DEF-CATEGORY NAME="Chair" ISA="AdministrativeStaff Professor">

Here Chair has 2 parent concepts: AdministrativeStaff and Professor.

As long as there are no contradictions this construction is working. But there are no guaranties to avoid contradictions by means of multiple inheritance, as we know from all sorts of conceptual modellings. Simply change the definition of the organisation and the constellation is producing a conflict and later a contradiction.

3.2 Ambiguity and polysemy

All concepts in an ontology have to be disambiguated.
In SHOE, again, this is easily done by renaming.

<DEF-RENAME > CHAIR in furniture-ont to Seat
<DEF-RENAME > CHAIR in academy-ont to AcademyHead

As long as we live in a very small world this strategy will work. But it stops to work immediately if we accept the dimensions of the Web. The process of renaming runs into a non-stopping procedure.

3.3 Chiastic polycontextural modelling

Also because the renaming procedure to avoid polysemy and ambiguity is not surviving the dynamics of a Dynamic Semantic Web the chiastic modelling is introduced as another more dynamic way of modelling the situation of polysemy and ambiguity.

Instead of domesticating the foreigner ontology into the home ontology by renaming the disturbing concepts a poly-contextural modelling is accepting the new ontology as such but has to offer a mechanism which allows to deal with the new double face situation of accepting and of mediating both ontologies. The process of mediating ontologies accepts the ambiguity between the concepts but rejects its logical conflicts and contradictions because now ambiguity is distributed over two ontological contextures ruled by two logical systems.

In accordance with the constructiviste point of view of conceptualizing as a semiotic process, in contrast to the neo-Aristotelian fundamentalist position of GOL, terms, objects, concepts have to be understood as relative to their use (Wittgenstein, Derrida) and not as pre-given entities of the world (universe).

4 Architectonic Parallelism of DSW

Navigation

Negotiation

Interactivity

Complexity

Reflectionality

5 Dynamics in the Semantic Web Context

There are many attempts to bring more dynamics into the Web. Some answers from the authorities: W3C, DARPA, MIT, ETH Zürich and McLuhan Institute

5.1 Dynamic Ontologies (Heflin, Hendler)

http://www.cs.umd.edu/projects/plus/SHOE/pubs/#aaai2000

The Web is dynamic.
The Web is massive.
The Web is an open world.
in: Towards The Semantic Web: Knowledge Representation In A Dynamic, Distributed Environment, Heflin 2001

Ontology Mapping and Translation

Will the inevitable proliferation of ontologies really solve the semantic interoperability problem? The answer is clearly no. The widespread adoption of ontologies only gets us half-way to semantic interoperability nirvana by forcing the use of explicit semantics. The other major challenge is mapping from one agent's ontology to another agent's ontology. The approaches to solve this problem range from static manually created ontology mappings to dynamic ondemand agent-based negotiation of ontology mappings.
in: Hendler, Semantic Web Technologies for Aerospace

Diagramm 12 **Dynamic Ontologies on the Web**

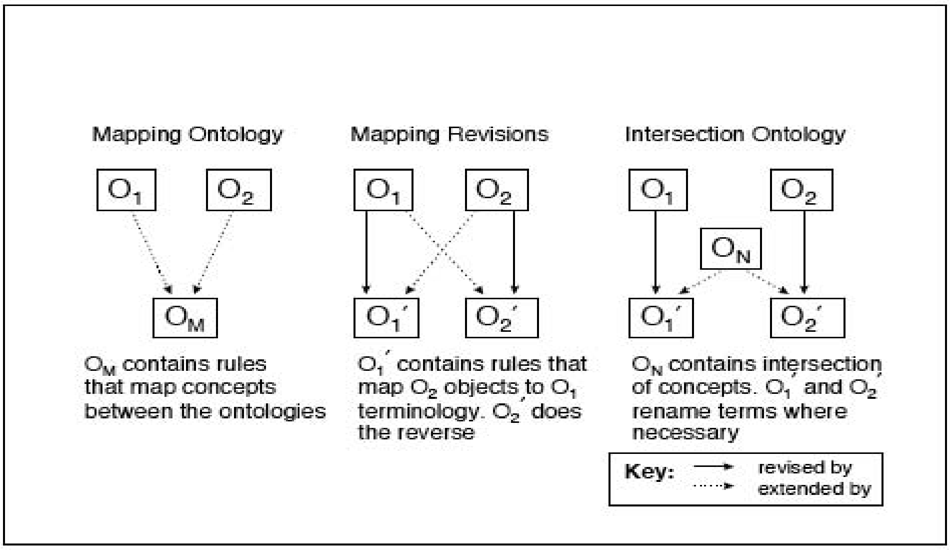


Figure 2. Integration Methods

5.2 Water: Static and Dynamic Semantics of the Web

http://web.media.mit.edu/~lieber/Lieberary/Dynamic-Semantics/Dynamic-Semantics.pdf

Less concern has been given to dynamic semantics of the Web, which is equally important. Dynamic semantics have to do with the creation of content, actions which may be guided by

- *User-initiated interface actions*
- *Time*
- *Users' personal profiles*
- *Data on a server*

and other conditions.

5.3 Cultural dynamic Web

http://semanticweb2002.aifb.uni-karlsruhe.de/proceedings/Position/veltmann.pdf

Towards a Semantic Web for Culture and Challenges for a Semantic Web
Kim H. Veltman, McLuhan Institute, Maastricht

***Logic** is, of course, an excellent starting point. Tim Berners-Lee has a conviction, which can be traced back to early history of Oxford from which he comes, that logic is a way to separating the wheat of truth from the chaff of idle claims. Logic is universally applicable: it reflects the scientific spirit. It represents the dimension concerning which there ought, in theory, to be no debate.*

5.4 Dynamic Semantic Web

In contrast to the precedent approaches the PCL based contribution to a Semantic Web and its dynamics is not accepting the limitations of expression, computation and interactivity forced by logic and its logical systems.

Peter Wegner has clearly analyzed the reason of the failure of the Japanese 5th Generation project: its believe in logics and its logic based programming languages, like Prolog. We have not to accept all the thesis about the change of paradigm in computer science proposed by Wegner, but I agree fully with his analysis of the role of logic. But again, Wegner and his school is not able to think about changing logics, instead he proposes some more empirical concepts to develop his intuition of paradigm change based on interactivity.

It is not necessary to repeat history again and again.

5.5 Dynamics with Modularity

Farshad Hakimpour, Andreas Geppert, Ontologies: an Approach to Resolve Semantic Heterogeneity in Databases

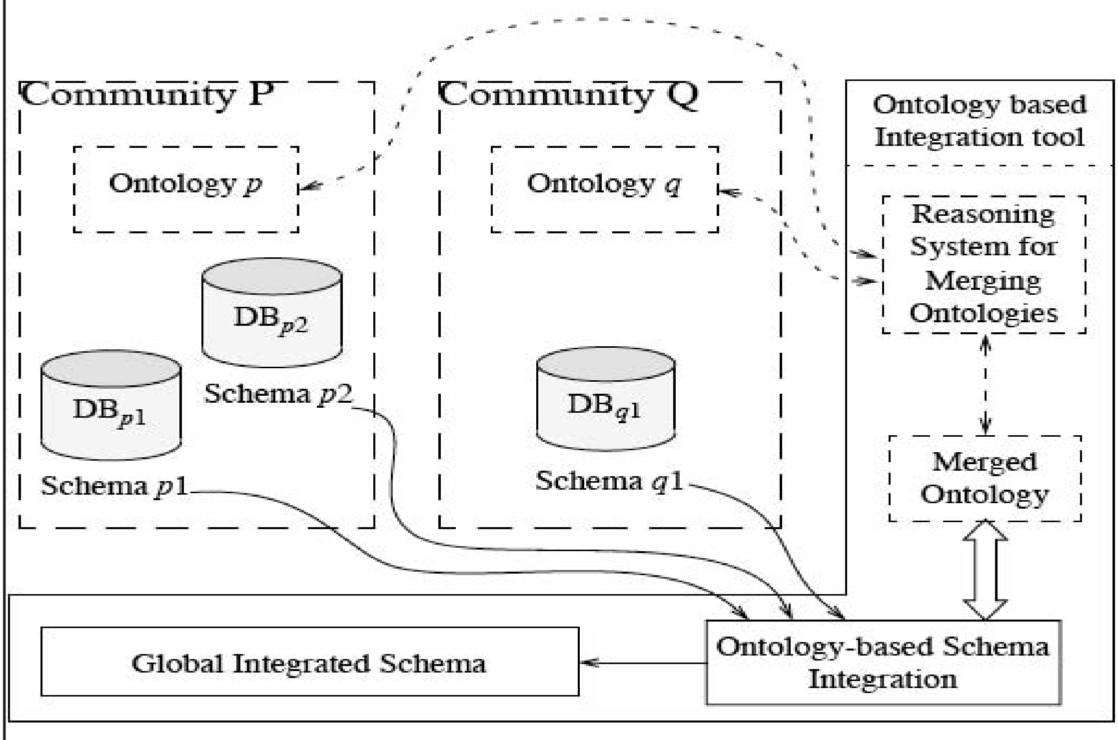


FIGURE 5. Global schema generation based on a common ontology produced by integration of domain ontologies.

This ontology dynamics is based on a constructivite epistemology not naively presuposing data systems. Different communities with different ontologies are introduced.

This Global schema of ontology integration is not telling us what happens with the presupposition of the difference of the ontologies p and q, namely their different Community P and Q.

It maybe of no special problem to integrate DBp1 and DBp2, simply because they are objects of the same community P. What happens to Community Q after merging ontology q with ontology p via merging schema p1, p2 with schema q1?

In this way of thinking, not many possibilities are open: Community P may disappear, or Community Q or a new super-community R will be constructed.

Merging companies, fusions of organizations, always have to deal with this problem. It seems to be an everyday problem, but there are no global solutions in sight.

Diagramm 13

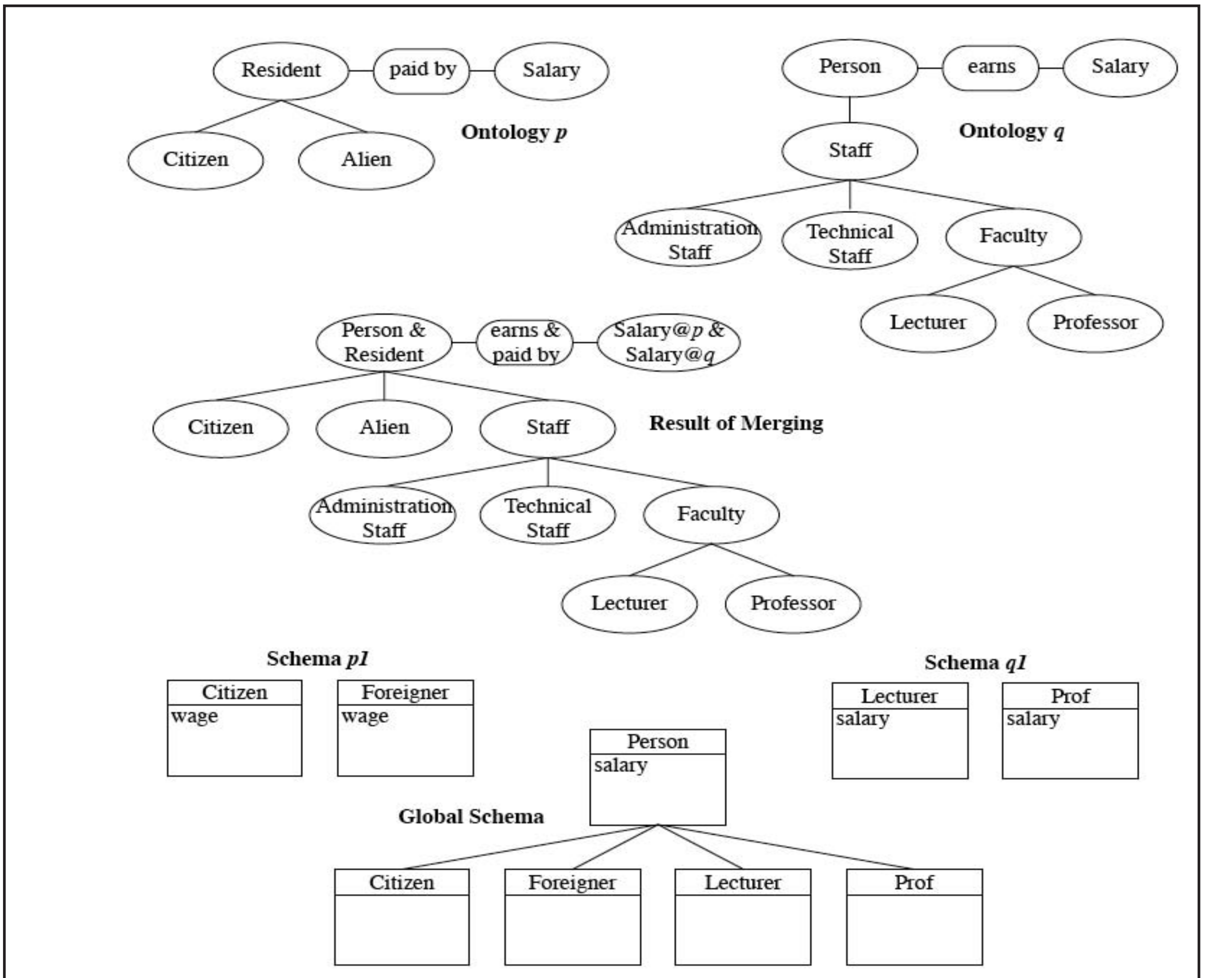


FIGURE 7. Result of finding two equal concepts and creating a class Person in global schema.

A nice vizualitation of a merging is given by robert lee
ai.kaist.ac.kr/~jkim/cs570-2003/ lecture-tp/SemanticWeb.ppt

Diagramm 14

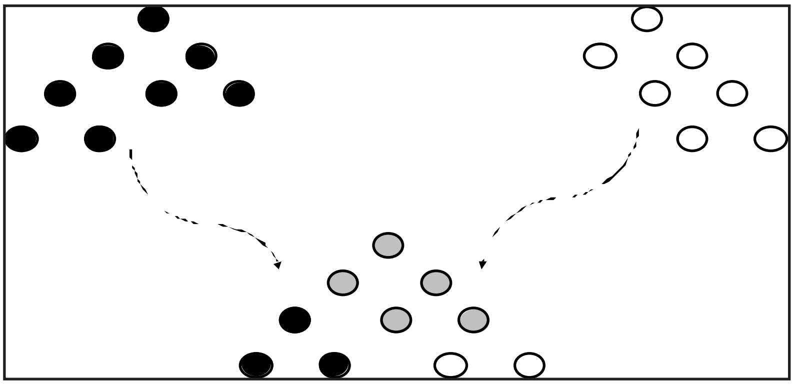


Diagramm 15

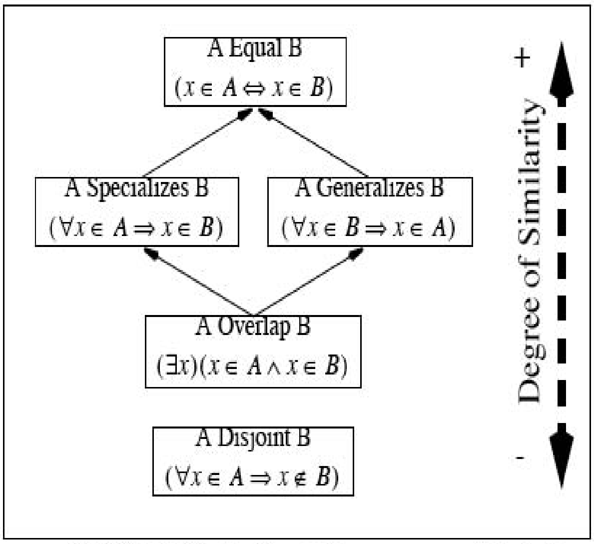


FIGURE 6. Similarity relations among ontological definitions

Obviously, for this scheme of *Degrees of Similarities of Ontologies*, everything Joseph Goguen mentioned abaut classic semiotics is true in an even more strict sense for ontologies.

Semiotics and Ontologies

Semiotics, as the general theory of signs, would seem a natural place to seek a general HCI framework. However

(1) semiotics has not developed in a precise mathematical style, and hence does not lend itself well to engineering applications;

(2) it has mostly considered single signs or systems of signs (e.g., a novel, or a film), but not representations of signs from one system by signs from another, as is needed for studying interfaces;

(3) it has not addressed dynamic signs, such as arise in user interaction; and

(4) it has not paid much attention to social issues such as arise in cooperative work.

A new project to address such problems has so far developed precise algebraic definitions for sign systems and their representations, and a calculus of representation providing laws for operations that combine representations as well as precise ways to compare the quality of representations. Case studies have considered browsable proof displays, scientific visualization, natural language metaphor, blending, and humor, while social foundations are grounded in ideas from ethnomethodology.

Joseph Goguen, Algebraic Semiotics and User Interface Design, 2000

http://www.isr.uci.edu/events/dist-speakers00-01/goguen00.html

Dynamics in Ontologies and Polysemy

1 Dynamic Ontologies in SHOE

SHOE is a well established approach to the Semantic Web emphasizing dynamics of ontologies.

http://www.cs.umd.edu/projects/plus/SHOE/

The dissertation of Henflin gives us a perfect introduction.

http://www.cs.umd.edu/projects/plus/SHOE/pubs/#heflin-thesis

Diagramm 16 **Ontology Scheme**

Ontology
USE-Ontology DEF-Relation DEF-Category DEF-Inference DEF-Rename

Generally, an ontology is a tupel O = (V, A)
V: Vocabulary
A: Axioms

Dynamics of SHOE-Ontologies are given by the operations of

Ont-Building
Ont-Revising
Ont-Versioning
Ont-Perspectiving

Short: Ont-Dyn = {Build, Rev, Vers, Persp}

The dynamics Ont-Dyn don't change the general definition of ontology. The operation of Ont-Dyn is closed, that is, Ont-Dyn(Ont-Dyn(Ont)) = Ont

Ont-Dyn(Ont-Dyn) = Ont-Dyn

The USE-Ontology operation is the key for modularity in SHOE. USE is building ontologies out of other ontologies. That is, ontologies are understood as modules.

Ontology = (Module$_0$, Module$_1$, ..., Module$_n$)

Module$_0$ contains the general base-ontology

The USE-ontology operation is producing a vertical and hierarchic chain of ontology extensions.

Ont-Dynamics: Ont ——> Ont: O^i —> O^i : O^i_1, O^i_2 —> O^i_{1+2}

Linear Modularization: O^i —> O^i: O^i_{1+2} —> O^i_1, O^i_2

Examples of modules (ontologies)

A SHOE Module can be any ontology which is not a base ontology and which is fulfilling the syntactic definition of an ontology.

http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html

general-ont = (Web-Res, Agent, PhysObject, Event, Location, Address, Activity)

document-ont = (Document, unpublished, published)

university-ont = (Faculty, Student, University, Department)

agents-ont = (sequentiell, parallel)

Ontology Dependencies (SHOE)

Below is a tree showing the dependency of ordering of the most recent versions of each ontology.

Base Ontology, v. 1.0
- Dublin Core Ontology, v. 1.0
- General Ontology, v. 1.0
  - Beer Ontology, v. 1.0
  - Commerce Ontology, v.1.0
  - Document Ontology, v. 1.0
    - University Ontology, v. 1.0
      - Computer Science Department Ontology, v. 1.1
  - Personal Ontology, v. 1.0
- Measurement Ontology, v. 1.0
  - Commerce Ontology, v.1.0
- TSE Ontology, v. 1.0

http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html

1.1 Dissemination of Ontologies, a more formal description

Polycontextural logics enable to add a new operation to extend ontologies. The horizontal operation of mediation MED is used to add ontological Modules not vertically like the USE operation but horizontally and therefore is producing a heterarchic organisation of the ontological modules.

Diagramm 17

USE(USE(USE))

Ontology1
USE-Ontology DEF-Relation DEF-Category DEF-Inference DEF-Rename
Ontology2
USE-Ontology DEF-Relation DEF-Category DEF-Inference DEF-Rename
Ontology3
USE-Ontology DEF-Relation DEF-Category DEF-Inference DEF-Rename

Diagramm 18

MED (ont1, ont2, ont3) = ont $^{(3)}$

Ontology1	Ontology2	Ontology3
USE-Ontology DEF-Relation DEF-Category DEF-Inference DEF-Rename	USE-Ontology DEF-Relation DEF-Category DEF-Inference DEF-Rename	USE-Ontology DEF-Relation DEF-Category DEF-Inference DEF-Rename

MED (ont1, ont2, ont3) = ont $^{(3)}$

MED(USE) /= USE(MED)

The mediation of USE ontologies is not the same as the use of mediated ontologies.

MED (Ont) /= Ont

The mediation of ontologies is surpassing the definition or type of the given ontologies.

The interplay of USE and MED defines the ontology grid in its vertical (iterative) and in its horizontal (accretive) dimensions. The grid is produced by the operation DISS (dissemination) which is the interplay of USE and MED, or in other words, the interplay between hierarchy (HIER) and heterarchy (HET).

$DISS_{accretive}$ (ONT) = MED(USE(ONT)) and

$DISS_{iterative}$ (ONT) = USE(MED(ONT))

DISS(ONT) = $DISS_{iterative}$ $DISS_{accretive}$ (ONT) = GRID (ONT)

Mediated ontologies are opening up the possibility for metamorphic changes of the basic categories of the ontologies involved in the interaction.The most basic change surely is the exchange between a base ontology, Mod_0, and a core ontology, say Mod_i. What is basic and primary in one ontology can be simultaneously secondary in another neighbor ontology.

This type of ontology-change is ruled by the proemial operator (chiasm) PR.

PR(Mod0, Modi, Ont1, Ont2)

Example: Disseminating basic concepts

Basic concepts like time, numbers, truth-function are defined in a base-ontology, which is by definition not to be transformed by any operations of Ont-Dynamics.

Veltman who is engaged to enrich the current trends of the Semantic Web toward a much more cultural and historical Semantic Web. He is criticizing SHOE of having implemented only the western model of calender. The real problem seems not be to add different cultural modules of chronology, topography and languages etc., but who to add them. If they are added vertically, in the sense of an iterative hierachical addition of modules, nothing has changed at all.

Only in the case of horizontal organisation of the basic ontologies a simultaneous multi-cultural and multi-lingual use can be processed and interaction between the different world views can be realized without restrictions by a ultimate upper ontology of what kind ever.

My thesis is, not the content but the very structure of the whole ontology is under question. If the modules of whatever content are added vertically, we stay in the western-centred paradigm of thinking. If we allow horizontal organization of the ontologies we are leaving this empire of hierarchical power to a heterarchical world of chiastic interplay of world views.

1.2 Computational complexity of hierarchy and heterarchy

(This is only a very first appraoch to the topic of complexity!!!)

Compl (USE(Ont1...Ontn) > Compl (MED(Ont1...Ontn))

Compl (HIER) > Compl (HET)

The tree of Ont1 may contain 2^m knots, and Ont2 may contain 2^n,

Compl(HIER(Ont1, Ont2)) = 2^{n+m}

Compl(HET(Ont1, Ont2)) = $2^m + 2^n$

This gives the number of knots for isolated parallel mediated ontologies Ont1 and Ont2. Additionally to this we have to calculate the number of interactions between Ont1 and Ont2.

2 Polysemy: Ontology Extension with the procedure rename

An interesting case of combining ontology modules together arise if the ontologies contains equal terms. In contrast to simple multiple inherence the situation of polysemy is introduced.

Remember:

> ***The Web is distributed.*** *One of the driving factors in the proliferation of the Web is the freedom from a centralized authority. However, since the Web is the product of many individuals, the lack of central control presents many challenges for reasoning with its information. First, different communities will use different vocabularies, resulting in problems of synonymy (when two different words have the same meaning) and polysemy (when the same word is used with different meanings).*

> *One of the hardest problems in any integration effort is mapping between different representations of the same concepts – the problem of integrating DTDs is no different. One difficulty is identifying and mapping differences in naming conventions. As with natural language, XML DTDs have the problems of polysemy and synonymy. (12)*

> *Recall that the Web is a decentralized system and its resources are autonomous. As a result, different content providers are free to assign their own meanings to each nonlogical symbol, thus it is likely that multiple meanings will be assigned to many symbols. Different axiomatizations for the same symbolsmay result fromthe polysemy of certain words, poor modeling, or even malicious attempts to break the logic. (23)*

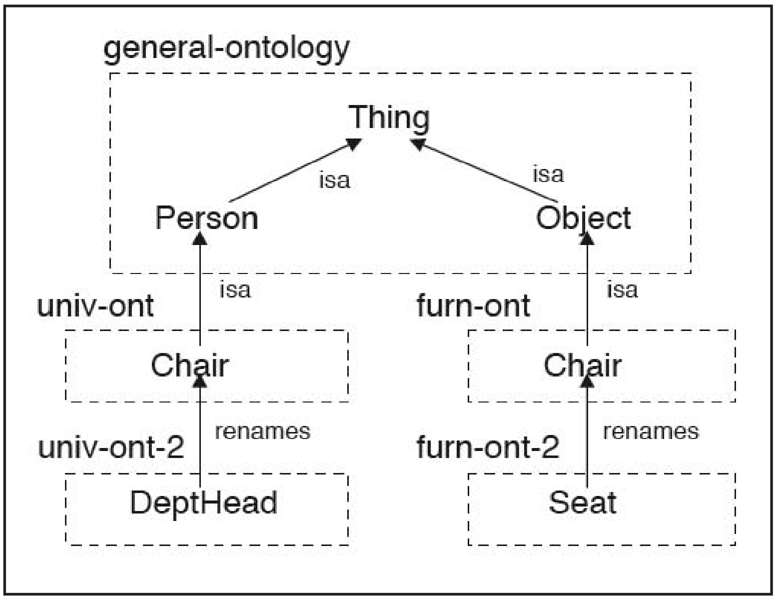
The main principle of ontology is demanding for disambiguating the polysemy of the used term. The simplest and historically oldest method to do this is given by *renaming* the terms. This is working perfectly in a very small world. But as we have learned, not only the weather system is massive, complex, open worlded, but also our WWW.

It is probably not very difficult to find, even if restrict ourselves to the english language, hundreds of different meanings of a term, here in the example of "chair". Therefore the renaming procedure can easily explode to a massive and complex topic in itself, destroying the aim of the simple and innocent procedure of renaming.

> *The problems of synonymy and polysemy can be handled by the extension mechanism and use of axioms. An axiom of the form P1(x1; : : : ; xn) $ P2(x1; : : :; xn) can be used to state that two predicates are equivalent. With this idiom, ontologies can create aliases for terms, so that domainspecific vocabularies can be used.*

> *For example, in Figure 3.1, the termDeptHead in OU2 means the same thing as Chair in OU due to an axiom in OU2. Although this solves the problem of synonymy of terms, the same terms can still be used with different meanings in different ontologies.*

Diagramm 19 **Figure 3.2**



There are many open questions. How does it fit together to have an ontological relation "isa" and an obviously linguistic operation "rename"? To bring the modules furn-ont and furn-ont2 and also univ-ont and univ-ont2 together we need at least a mediation third module, which is reflecting the terminology of both. But this linguistic ontology would produce itself similar possibilities of polysemy.

Polysemy means:

A=C and
B=C and
A /=C

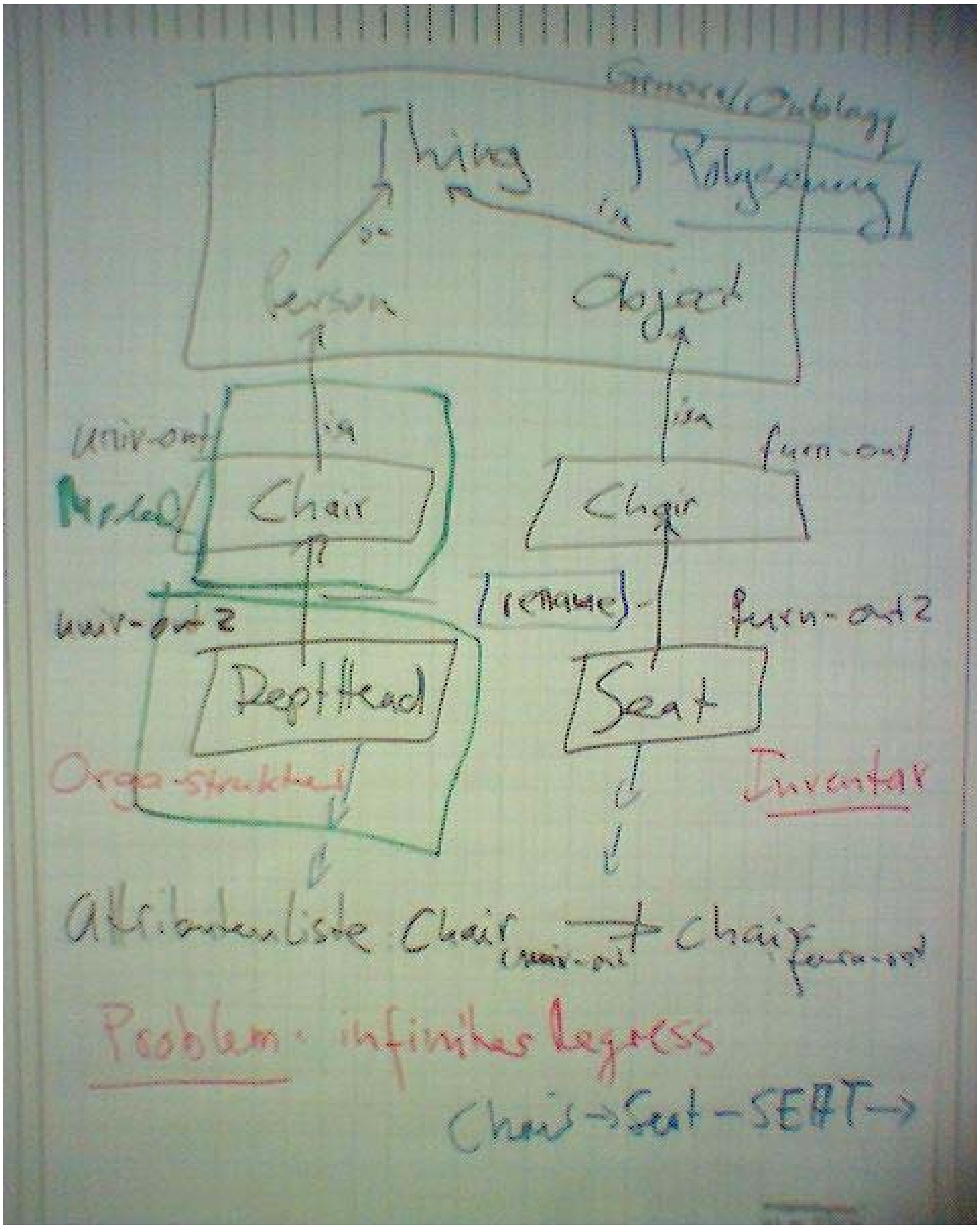
Do it again

There is no reason to not to start the game of polysemy again with the term Seat as furniture and Seat as seat, e.g. position, in the hierarchy of a department. And we can disambiguate this polysemy again with the help of the term Chair. A seat as department is a chair and a seat as furniture is a chair. And now we can turn around as often as we want...

Extension of ontologies by renaming is not violating the principle of verticality, that is hierarchy. Therefore, the tree is growing and with it its computational complexity.

It becomes obvious that the procedure of renaming is part of the broader activity of *negotiation*. Without a proper mechanism of solving the problems of renaming the amount of not machine-assisted negotiation is growing in a contra-productive way, conflicting the very aims of the Semantic Web to support machine-readable semantic information processing.

Diagramm 20


Thing
Person
Object
isa
univ-ont
Chair
furn-ont
Chair
rename
furn-ont2
DeptHead
Seat
Inventar
Problem

http://www.mindswap.org/cgi-bin/2002/searchdamlont.pl

ONTOLOGYHYPER-
DAMLDUMP
ONTConvert to OWLColorn3HIT ON
1NSO-ont[H] [H] [D]OWLcolor viewn3 viewChairmanOfTheJointChiefsOfStaff
2NSO-ont[H] [H] [D]OWLcolor viewn3 viewViceChairmanOfTheJointChiefsOfStaff
3UNSPSC[H] [H] [D]OWLcolor viewn3 viewBedpans-or-commode-chairs-for-people-with-disabilities
4UNSPSC[H] [H] [D]OWLcolor viewn3 viewCamping-chairs-or-stools
5UNSPSC[H] [H] [D]OWLcolor viewn3 viewChair-lifts-or-chair-transporters,-for-people-with-disabilities
6UNSPSC[H] [H] [D]OWLcolor viewn3 viewChairs
7UNSPSC[H] [H] [D]OWLcolor viewn3 viewCoxit-or-arthrodesis-chairs-for-people-with-disabilities
8UNSPSC[H] [H] [D]OWLcolor viewn3 viewMechanized-chairs-to-assist-with-sitting-or-standing-for-people-with
9UNSPSC[H] [H] [D]OWLcolor viewn3 viewPatio-chairs
10UNSPSC[H] [H] [D]OWLcolor viewn3 viewRestaurant-chairs
11UNSPSC[H] [H] [D]OWLcolor viewn3 viewVibrating-chairs-for-training-deaf-people
12UNSPSC[H] [H] [D]OWLcolor viewn3 viewWheelchair-accessories
13UNSPSC[H] [H] [D]OWLcolor viewn3 viewWheelchair-lifting-platforms
14UNSPSC[H] [H] [D]OWLcolor viewn3 viewWheelchair-ramps
15UNSPSC[H] [H] [D]OWLcolor viewn3 viewWheelchairs
16cs1[H] [H] [D]OWLcolor viewn3 viewChair
17cs1[H] [H] [D]OWLcolor viewn3 viewChair
18cyc-transportation[H] [H] [D]OWLcolor viewn3 viewElectricWheelchair
19cyc-transportation[H] [H] [D]OWLcolor viewn3 viewWheelchair
20univ1[H] [H] [D]OWLcolor viewn3 viewChair

20 hits in 186 ontology files

Douglas B. Lenat

The success of the Semantic Web hinges on solving two key problems:

(1) enabling novice users to create semantic markup easily, and

(2) developing tools that can harvest the semantically rich but ontologically inconsistent web that will result.

To solve the first problem, it is important that any novice be able to author a web page effortlessly, with full semantic markup, using any ontology he understands. The Semantic Web must allow novices to construct their own individual or specialized-local ontologies, without imposing the need for them to learn about or integrate with an overarching, globally consistent, master ontology.

The resulting Web will be rich in semantics, but poor in ontological consistency. Once end-users are empowered by the Semantic Web to create their own ontologies, there will be an urgent need to interrelate those ontologies in a useful way. The key to harvesting this new semantic information will be the creation of the Semantic Web-aware agents that can cope with a diversity of meanings and inconsistencies across local ontologies. These agents will need the capability to interpret, understand, elaborate, and translate among the many heterogeneous local ontologies that will populate the the Semantic Web.

http://www.cyc.com/cyc/cycrandd/areasofrandd_dir/sw

These agents will not only *"need the capability to interpret, understand, elaborate, and translate .."* but they also have to be non-human agents, that is programs. What´s difficult to master for human beings should be a fine job for our new agents. It seems that the unsolved problems of AI are emerging again in a new setting.

3 Polycontextural modelling of polysemy

The Internet is a giant semiotic system. Sowa

Polycontextural modelling can be made more transparent if we don´t forget that the concept of ontology is only a very reduced case of general semiotics. (I leave it for further reflections to abandon also semiotics in favor of polycontexturality.)

Exposing a polycontextural modelling of polysemy I am forced to use semiotic distinctions not available in the Semantic Web language SHOE.

3.1 Semiotic Diagram

Remember Charles Sanders Peirce:

> *A sign, or representamen, is something which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign. That sign which it creates I call the interpretant of the first sign. The sign stands for something, its object. It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the ground of the representamen. (CP 2.228)*

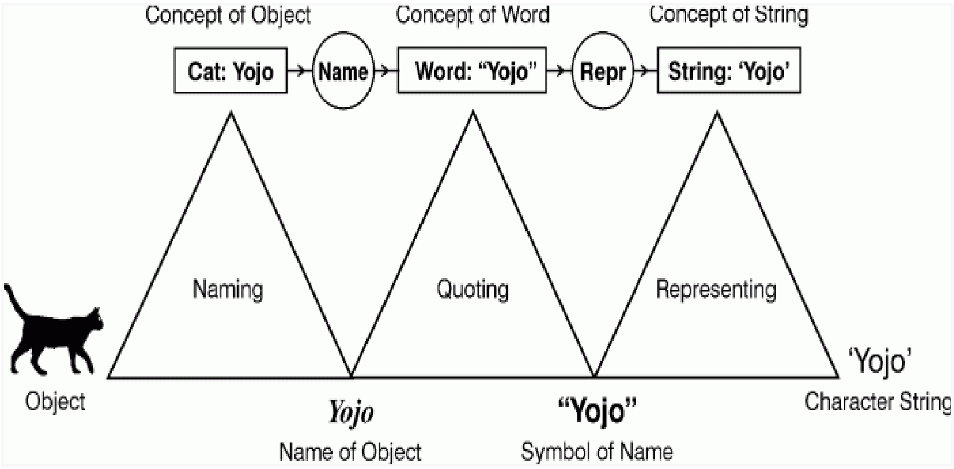
Sowa:

> *Many of the ontologies for web objects ignore physical objects, processes, people, and their intentions.*
>
> *A typical example is SHOE (Simple HTML Ontology Extensions), which has only four basic categories: String, Number, Date, and Truth (Heflin et al. 1999).*
>
> *Those four categories, which are needed to describe the syntax of web data, cannot by themselves describe the semantics. Strings contain characters that represent statements that describe the world; numbers count and measure things; dates are time units tied to the rotation of the earth; and truth is a metalanguage term about the correspondence between a statement and the world. Those categories can only be defined in terms of the world, the people in the world, and the languages people use to talk about the world. Without such definitions, the categories are meaningless tags that confer no meaning upon the data they are attached to.*

Ontology, Metadata, and Semiotics
John F. Sowa

http://users.bestweb.net/%7Esowa/peirce/ontometa.htm

Diagramm 21



A nice semiotic picture of our world of semantic knowledge. It is surely better than the lack of any semiotic knowledge.

Diagramm 22

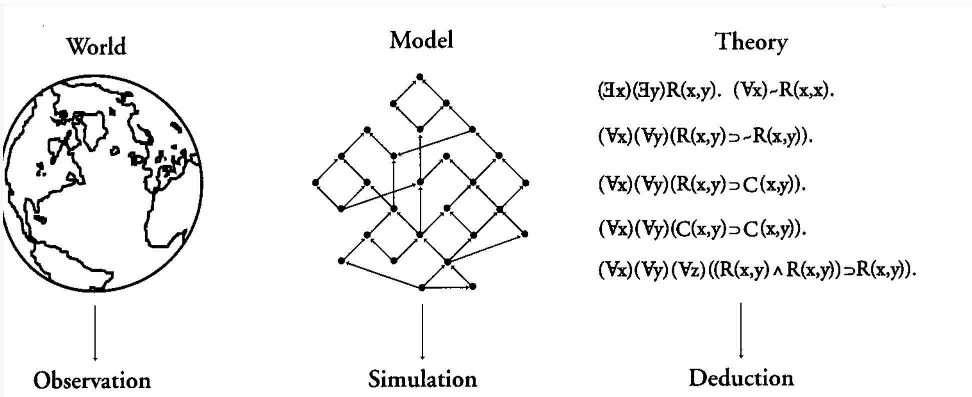
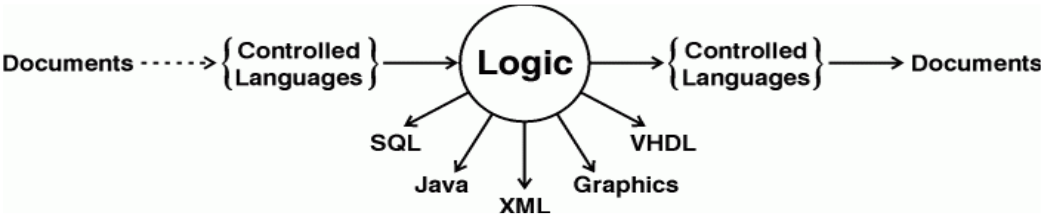


FIGURE 6.13 Three sources of knowledge about the world

Pure logic is ontologically neutral

It makes no presuppositions about what exists or may exist in any domain or any language for talking about the domain. To represent knowledge about a specific domain, it must be supplemented with an ontology that defines the categories of things in that domain and the terms that people use to talk about them. The ontology defines the words of a natural language, the predicates of predicate calculus, the concept and relation types of conceptual graphs, the classes of an object-oriented language, or the tables and fields of a relational database. Sowa

Diagramm 23



Everyone who has studied polycontextural logics know that logic isn't as neutral as it is believed by the community of logicians and computer scientists. At least, logic is presupposing a special type of formality to be accessible to formalization, and this formality as such can turn out as logics restricting content. But it is crucial to understand this neutrality statement because it describes exactly the situation as it is established in contemporary (western) thinking.

Ask for other opinions and paradigms Charles S. Peirce or Gotthard Gunther.

3.2 Reflectional semiotic modelling of polysemy

A reflectional analysis of polysemy is an analysis of the semiotic actions or behaviors of agents which is leading to the phenomenon of polysemy and its possible conflicts with other semiotic or logical principles. Therefore, such an analysis is more complex, because it has to describe the situation intrisically, that is from the inside and not from the outside from the position of an external observer.

Mono-contextural introduction of "isa":
S1: Chair is part of a furniture ontology
S2: Chair is part of a department ontology
S3: Chair is part of a vocabulary

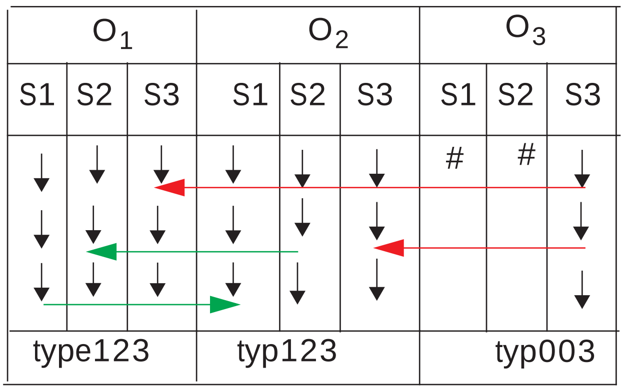
Poly-contexturally we have to distinguish the situations "isa as":
O1S1: Chair as such, that is, as an object "Chair"
O2S2: Chair as such, that is, as a person "Chair".
O3S3: Chair as such, that is, as the token "Chair"

Here, "as such" means, that the ontologies Person, Object and Vocabulary can be studied and developed for their own, independent of their interactivity to each other but mediated in the constellation of their poly-contexturality, that is, their distribution over 3 loci.

Voc O3S3 in Furn O1S3 : The token "Chair" as used to denote the object "Chair"
VocO3S3 in Dept O2S3 : The token "Chair" as used to denote the person "Chair"

Chair O2S2 in Dept O1S2 : The object Chair as used in the person ontology Dept
Chair O1S1 in Furn O2S1 : The person Chair as used in the object ontology Furn

Diagramm 24



Reflectional situations

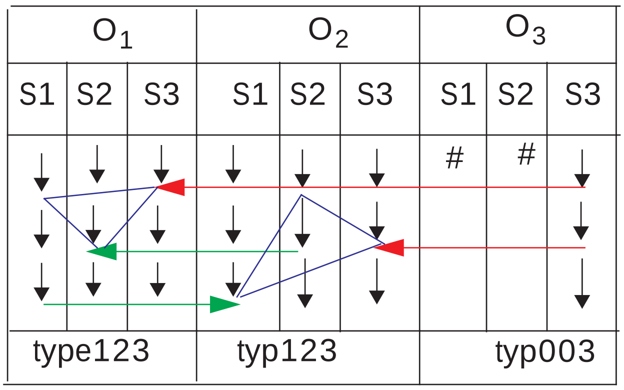
Chair O2S2 in Dept O1S2:
System O1S1 has in its own domain space for a mirroring of O2S2. This space for placing the mirroring of O2S2 is the reflectional capacity realized by the architectonic differentiation of system O1. In other words, O1 is able to realize the distinction between its own data and the data received by an interacting agent. Data are therefore differentiated by their source, e.g. their functionality, and not only by their content.

Chair O1S1 in Furn O2S1:
System O2S1 has in its own domain space for a mirroring of O1S1.

Some exclusions

Some more fascinating possibilities, which are excluded in this construction:
The word Chair as a DeptChair: (empty chair): Voc O3S3 in Dept O2S2
The word Chair as a Furn: (decoration): Voc O3S3 in Furn O1S1
The DeptChair as an object Chair (in a game): Dept O2S2 in Furn O1S1
The DeptChair as a token Chair (as a symbol): Dept O2S2 in Voc O3S3
The FurnChair as a person Chair (Breschnijew) :Furn O1S1 in Dept O2S2

Diagramm 25



A (re)solution of the problem

The solution of the (new) problem is in the (old) problem which the (new) problem is the (old) solution.

The department Dept for itself has no conflict with polysemy. This conflict between Dept and Furn is mediated by the Voc. That is, the Person of the Dept as Chair are persons and nothing else.

The furniture Furn for itself has no conflict with polysemy. This conflict between Furn and Dept is mediated by the Voc. That is, the Chairs as objects of the Furn are chairs and nothing else.

The vocabulary Voc for itself has no conflict with with polysemy between Dept and Furn.

The meaning of the polysemic situation is realised by

Meaning of (O3S3) = interaction of (O1S3, O2S3)

The conditions for a conflict arises excactly between

O1 (S1,2,3) and O2 (S1,2,3) mediated by O3S3 as visualized by the blue triangles.

Both Furn and Dept are using Voc and both are using the string Chair. Both are different and are mapping the Voc differently relative to their position, thus the Voc has to be distributed over different places according to its use or functionality. The Voc used by Furn is in another functionality than the Voc used by Dept.

Until now we have not yet produced a contradiction but only a description of the situation of polysemy, that is, the necessary conditions for a possible ontological contradiction.

A user-oriented or behavioral-oriented approach to the modelling of polysemy has to ask "For whom is there a conflict?". Therefore we have additionally to the semantic and syntactic modelling of the situation to introduce some pragmatic instances. In our example this can be the user of a Query which is answering in a contradictional manner.

Query´s contradiction

Now we have to deal with the contextures: (Query, Voc, Furn, Dept).

In the classic situation the Query answers with a logical conjunction of Chair as Person and Chair as a Department member, which are logically excluding each other and therefore producing for the user a contradictorily answer. Logic comes into the play also for the polycontextural modelling, but here conjunctions too, are distributed over different contextures. And therefore, a contradiction occurs only if we map the complex situation all together onto a single contexture. If we give up all the introduced ontological distinctions of polycontexturality and reducing therefore our ontologies to a single mono-contextural ontology we saved our famous contradiction again. But now, this contradiction is a product of a well established mechanism of reduction. And sometimes it isn't wrong to have it at our disposition.

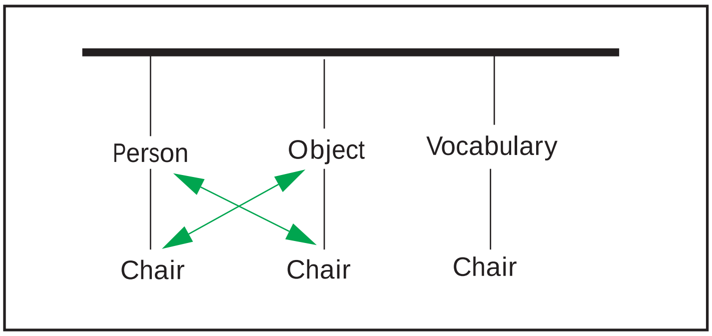
Extension by mediation

The procedure of renaming can now be understood as an accretive ontology extension, using another additional ontology, by the procedure MED-ontology.

To change from Chair as a furniture to Seat and from Chair as Dept to DeptHead is not only a linguistic procedure of renaming in the vocabulary it is also the use of two other ontologies in which these terms are common.

From the point of view of the new ontologies the conflict between Furn and Chair becomes obvious and transparent as a linguistic conflict of using a Voc. Only from the point of view of DeptHead and Seat the conflict appears as a conflict of synonymy. From the positions of Chair as Furn and Chair as Dept their is only a conflict per se. Without the possibility of an insight into its structure and kind of the conflict and therefore there is also no chance for a solution of the conflict.

Diagramm 26



Chiastic situation of the polysemy example:

Person becomes Object and Object becomes Person both relative to their common Vocabulary, that is the word "Chair".

4 Some Polylogical Modelling of Polysemy

To each ontology we have a corresponding logic (or logical system).

Ont ——> Logic

Med(Ont1, Ont2, Ont3) = Ont(3) ——> MED(Logic1, Logic2, Logic3) = Logic(3)

A contexture is the common framework of a logic and its corresponding ontology.

Conjunctive connection of ontological modules A, B, C, D in each contexture:

L_1: A and B and C and D

L_2: A and B and C and D

L_3: A and B and C and D

$L^{(3)}$: $A^{(3)}$ and and and $B^{(3)}$ and and and $C^{(3)}$ and and and $D^{(3)}$

The binary case for short: $L^{(3)}$: (A and B); (A and B); (A and B)

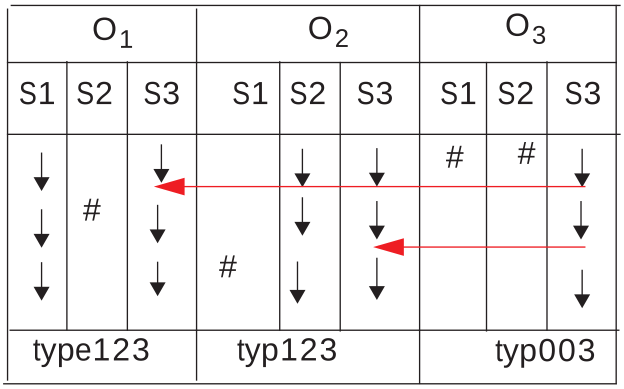
As we see, the possible places for reflecting the neighbor systems are empty, marked with "#" in the case of the monoform junctional distribution.

Diagramm 27

O_1			O_2			O_3		
S1	S2	S3	S1	S2	S3	S1	S2	S3
↓ ↓ ↓	#	#	#	↓ ↓ ↓	#	#	#	↓ ↓ ↓
type100			typ020			typ003		

This corresponds to the purely parallel situation of the ontologies as such without any interaction at all. But nevertheless, these logics are distributed over three places and mediated together in the architectonics of the logical frame $L^{(3)}$.

Diagramm 28



Additionally to the intra-contextural realizations of conjunctive chains we observe a first interaction from the logical system to its neighbor systems. Again, this interaction is not overriding its neighbors but is offered by the neighbors logical space to succeed realization. In other words, the neighbor systems are mirroring, that is, reflecting the interactivity of the logic system L3 in a place or locus of their own systems.

How is this realized? Also no conjunction or disjunction or other intra-logical operation is able to leave its place, we are not lost in the cage of mono-contexturality, because by construction, logical operations which are crossing the borders of their systems are accessible, this is the family of transjunctions.

A transjunction has a continuation simultaneously in its own and in its neighbor systems.

For short, we have in $L^{(3)}$: (A and B; A and B; A trans B)

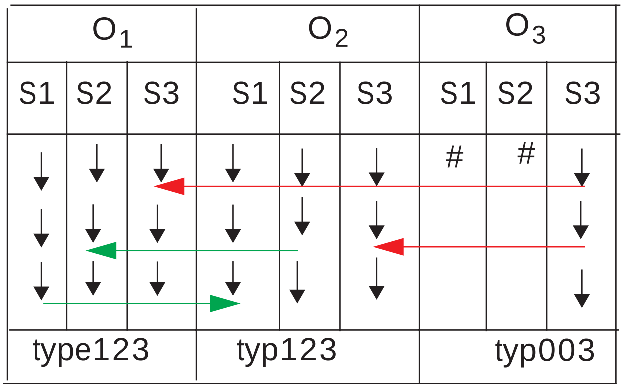
In L_3 the transjunction is crossing to logic L_2 simultaneously to logic L_1 and staying with other parts of the formula in its own logic L_3.

It is easily to see, that the classical conflicts of multiple inheritance would be produced if the mapping would not be transjunctional and reflectional but a simple mapping onto the systems as such, that is, mapping of O3S3 onto O1S1 and O3S3 onto O2S2.

The same argumentation is used for the logical operator “implication” and works in the same sense also for meta-logical constructions like the inference rule(s).

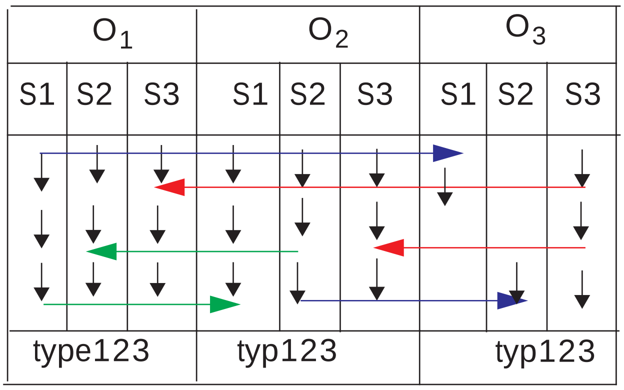
Therefore inferencing in poly-contextural systems is architectonically parallel.

Diagramm 29



$L^{(3)}$: (A trans B); (A trans B); (A trans B)

Diagramm 30



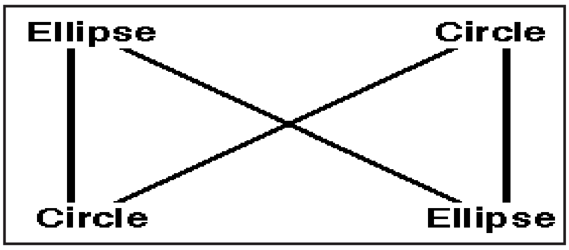
Even more interesting interactions are possible with the introduction of transjunctional mappings from O1S1 to O3S1 and from O2S2 to O3S2.

In these cases, reflectionality enters the domain of the vocabulary Voc. The Voc , again, is not only a collection of facts which exist per se in a dictionary. A vocabulary exists in being used. Therefore the other system s are influencing the system of the vocabulary. The difference is, that these lexical influences are not yet incorporated by the vocabulary in the sense of O3S3. That the reason way they occur in the reflectional ebvironment of Voc as reflecting and accepting the interactive influence of Furn and Dept to the domain of Voc.

4.1 Inconsistency, Contradiction and Polysemy

Building, Sharing, and Merging Ontologies, John F. Sowa

Figure 14 shows a "bowtie" inconsistency that sometimes arises in the process of aligning two ontologies.



On the left of Figure 14, Circle is represented as a subtype of Ellipse, since a circle can be considered a special case of an ellipse in which both axes are equal. On the right is a representation that is sometimes used in object-oriented programming languages: Ellipse is considered a subclass of Circle, since it has more complex methods. If both ontologies were merged, the resulting hierarchy would have an inconsistency. To resolve such inconsistencies, some definitions must be changed, or some of the types must be relabeled. In most graphics systems, the mathematical definition of Circle as a subtype of Ellipse is preferred because it supports more general transformations.

http://users.bestweb.net/~sowa/ontology/ontoshar.htm#Formal

For whom are this two positions a contradiction? Where does the inconsistency appear? Obviously both positions are clean in themselves. The inconsistency or logical contradiction occurs only by the mixing both and mapping them into a third general common position. What happens? The merging produces a new object which involves both different positions and at the same time denies the autonomy of those positions.

Again, for the case of managing a small household, the strategy of subordination maybe accepted for the one or other short termed practical reasons. But, by whom? For more official, and serious solutions, the idea of resolving by the device *"To resolve such inconsistencies, some definitions must be changed, or some of the types must be relabeled."* is not a proof of profound thinking and knowledge about practicability.

4.1.1 From merging to mediating interactivity

From an actional point of view in contrast to an entity ontology standpoint it is more apprpriate to consider the process of merging as a process of conflict resolution. This type of modelling is reasonable only if we accept the relevance of the two different point of views, if both positions have their own reason to exist. Otherwise it would only be a question of terminology and adjustments (renaming, relabelling).

The above example of a "bowtie inconsistency" can easily modelled as a chiastic interaction between two different positions offering at least a conceptual description of the situation as introduced.

Chiasm (Ellipse, Circle, Pos1, Pos2):
OrdRel(Ellipse1, Circle1)
OrdRel(Circle2, Ellipse2)
ExchRel(Ellipse1, Circle2)
ExchRel(Circle1, Ellipse2)
CoincRel(Ellipse1, Ellipse2)
CoincRel(Circle1, Circle2)

To model the full picture of the chiastic situation we can move to the Diamond Strategies.

4.1.2 Diamond strategies and merging inconsistent ontologies

A framework of the distribution of places needed to merge inconsistent or dual ontologies is given by the Diamond Strategies.

Position: a given ontology.

Opposition: the dual ontology to the positioned ontology, short, the contradicting ontology.

Neither-Nor: the position which is neither one nor the other ontology, but in respect to this two ontologies. It is the place of the rejection of both ontologies. Positively, it is the empty place which is common to both in respect of rejecting the ontologies.

Both-And: the position which gives place for both, the first and the second ontology at once. At this place, the position as well as the opposition is accepted, that is, the contradiction between both ontologies is accepted as such.

I hope it becomes slowly clear that the diamond strategies are not at all identical with the tetra-lemma of Buddhist philosophy despite some analogy in the wording.

Rejection of an alternative and acceptance of an inconsistency has nothing to do with negation or set theoretic union of concepts. One of the main differences is that the tetra-lemma is not reflectional at all. It is a good starting point but only as a configuration about the world as it is without including any observational reflectionality.

It is obvious too, that the acceptance of inconsistency is not understood in the sense of para-consistent logics. Nevertheless, it is interesting for other reasons to deal inconsistencies in a para-consistent setting.

Today it shouldn't be a technical problem to represent complementary objects at once on a screen or where ever.

The discipline which would have to deal with such complementary objects and their theories would be called *"Dynamic Diagrammatics"* as a further development of the Peirceian based Diagrammatics.

Politics of examples

Examples and metaphors are not as harmless as it seems to be.

Some more realistic examples instead of innocent circles and ellipses, chairs, penguins and kilts etc. should be introduced. A simple example of renaming is globally introduced by Bush´s doctrine of pre-emptive war.

Logic of execution:
Human beings, animals
allowed to be killed, not allowed to be killed
Friedensfighter, terrorists

The Christian problem of executing humans in a non-war situation is pluntely solved by Bush and Sharon with the not at all rhethorical decision, that terrorist are animals.

In the more theological terminology, animals are replaced by the evil, because animals too are creatures of God.

5 Polycontextural modelling of multiple inheritance

Ontologies differ in how they handle the case of inheriting multiple properties.
Robert Lee

There are no problems neither with polysemy nor with multiple inheritance if you chose your examples carefully and then run away after you have been paid.

The multiple inheritance of CHAIR being a AdminStaff member and a Faculty member as Professor in the example of SHOE is surely innocent of any logical violations, leading to contradictions. SHOE is even excluding logical negations to avoid contradictions. But this is not the situation a Semantic Web designer should be concerned about.

It is simply bad propaganda and contra-productive advise if I have to read in different Web Semantic papers that they have solved the multiple inheritance problem properly.

Let´s have a short look at the scenario.

Several proposals have arisen for thesauri interchange formats based on either RDF or DAML+Oil. The major problems with these is that either they cannot accommodate the multiple inheritance common in many multilingual thesauri or that the semantics of thesauri in the ISO standards are not as precise as these languages require. The links in thesauri hierarchies define the top term in the hierarchy, and the broader or narrower coverage of terms down the hierarchy. There are also links between hierarchies to show equivalence in different languages, or similar meaning in the same language.

http://www.ercim.org/publication/Ercim_News/enw51/wilson.html

A more optimistic view is here. You simply have to do it before the game.

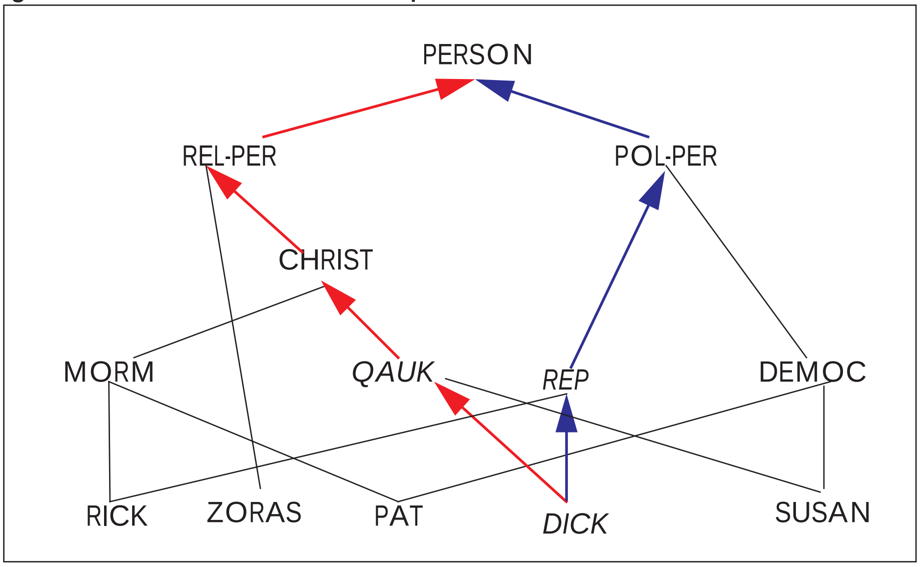
Very flexible ways of combination, such as multiple inheritance, can be specified for types in simple ways. Since agreement on supplied and required interfaces is all that is needed for the exchange of data in a distributed environment types already provide the glue for many useful applications.

The Quaker Example

Most Quaker are Pacifists
Most Republicans are non-Pacifists
Dick is a Quaker
Dick is a Republican.

Query: Is Dick a Pacifist?

Diagramm 31 **Multiple Inheritance**



There are many serious attempts to deal with multiple inheritance in the AI literature (Lokenda Shastri: *Semantic Networks: An Evidential Formalization and its Connectionist Realization*, Pitman London 1988)

It is not the place here to discuss Shastri´s solution. What we can learn is the introduction of different *relevance* criteria and *multiple views* on a token. It is only a simple step further to combine multiple views with multiple contextures and introducing irreducible polysemy into the very concept of "person".

Therefore, Dick has multiple personal identity, one as a Religious Person (REL-PER) and one as a Political Person (POL-PER).

With the introduction od POL-PER and REL-PER the simple question "Is Dick a Pacifist?" is wrongly placed and not well-formed because the particle "as" giving his perspective and role is excluded.

We have to ask "Is Dick *as* a POL-PER a pacifist?" and "Is Dick *as* a REL-PER a Pacifist?" And additionally, which is a very different question, we can ask "How is Dick as Dick, which is neither a political nor a religious person, dealing with his two positions of being a POL-PER and a REL-PER?" And here, we would have to consider the relations of interactions between the different ontologies.

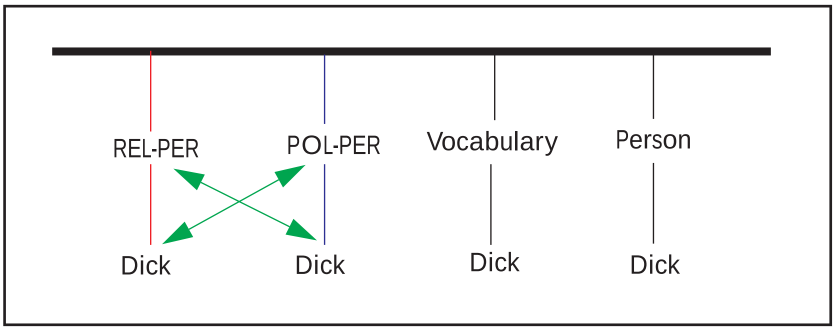
Only if we are reducing the two perspectives and eliminating the as-category, we are reconstructing the contradictions of this multiple inheritance situation. This maybe well known, but because of the lack of a logic which is genuinely dealing with different and mediated perspectives, like polycontextural logic, the implementation of the complex conceptual modeling is lost for mono-contexturality.

I can not go into the details here, but obviously, the polycontextural approach of modelling the multiple inheritance situation has to separate and then to mediate the ontologies REL-PER and POL-PER in a heterarchical interacting poly-ontology.

Obviously, the SHOE trick for multiple inheritance we have learnt before with Chair=(Dept, AdminStaff) doesn't work anymore. Because Dick as (REP, QUAK) is producing a contradiction by definition. By the way, the same can happen with the Chair example, we simply have to change the rules of the organization to a more strict regime.

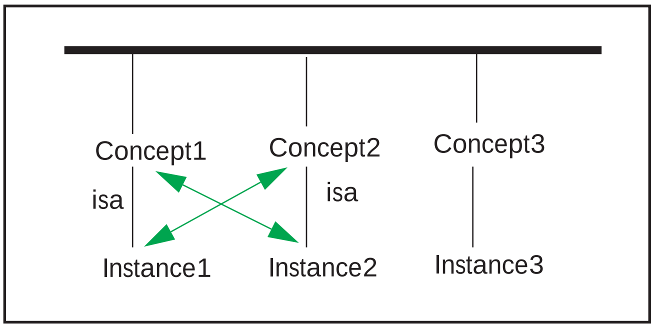
Again, a hint is given by the following chiastic metapattern diagram.

Diagramm 32



I added to the list of
Quaker
Penguine
Whale
Oistrich
Fleuve
etc.
the very neglected case of Kilts.

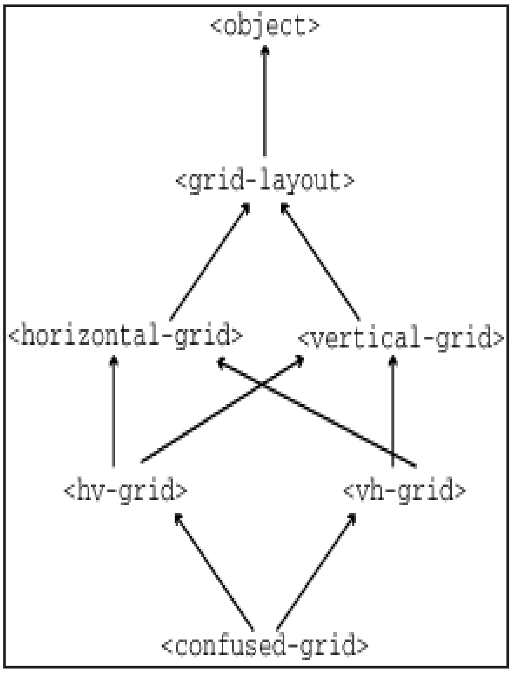
Diagramm 33



Kilt as an instance of female clothes proposed by the Eurpean Administration, that is as a skirt and therefore female, is surely in contradiction to the Scottish definition of Kilts. A chiastic resolution of this crucial conflict has simply to understand that Kilts are Instances of a very different Concept2. It doesn't mean that the Instance1 "Kilt" becomes itself a Concept2, but that the contradiction in system1 with Instance1 and Kilt gives reasons to a *switch* to system2 with concept2 as maybe "folklore" and Kilt as an Instance2 of Concept2. But both systems are as mediated systems not isolated.

Is the Grid of the Dynamic Semantic Web a confused Grid?

Why linearizations? *In a class-based object-oriented language, objects are instances of classes. The properties of an object - what slots or instance variables it has, which methods are applicable to it - are determined by its class. A new class is defined as the subclass of some pre-existing classes (its superclasses - in a single-inheritance language, only one direct superclass is allowed), and it inherits the properties of the superclasses, unless those properties are overridden in the new class. Typically, circular superclass relationships are prohibited, so a hierarchy (or heterarchy, in the case of multiple inheritance) of classes may be modeled as a directed acyclic graph with ordered edges. Nodes correspond to classes, and edges point to superclasses.*



It is possible that an inheritance graph is inconsistent under a given linearization mechanism. This means that the linearization is over-constrained and thus does not exist for the given inheritance structure. An example of an inconsistent inheritance relationship appears in example 1c. <confused-grid> is inconsistent because it attempts to create a linearization that has <horizontal-grid> before <vertical-grid>, because it subclasses <hv-grid>, and <vertical-grid> before <horizontal-grid>, because it subclasses <vh-grid>. Clearly, both of these constraints cannot be obeyed in the same class.

Kim Barrett et al, A Monotonic Superclass Linearization for Dylan

http://www.webcom.com/haahr/dylan/linearization-oopsla96.html

```
define class <grid-layout> (<object>) É end;
 define class <horizontal-grid> (<grid-layout>) É end;
 define class <vertical-grid> (<grid-layout>) É end;
 define class <hv-grid> (<horizontal-grid>, <vertical-grid>) É end;
 define method starting-edge (grid :: <horizontal-grid>)
   #"left"
 end method starting-edge;
 define method starting-edge (grid :: <vertical-grid>)
   #"top"
 end method starting-edge;
```
Example 1a: A simple use of multiple inheritance

```
 define class <vh-grid> (<vertical-grid>, <horizontal-grid>) É end;
```
Example 1b: Reversing classes in the linearization

```
 define class <confused-grid> (<hv-grid>, <vh-grid>) É end;
```
Example 1c: An inconsistent class definition

6 Query, questions and decisions

"Only undecidable questions have to be decided by man" ? HvF

As long as our queries are answering our questions with only non-ambiguous, non-polysemous statements, we are dealing with a very reduced case of semantics. It is semantics reduced to a machine-readable and machine-understandable situation, therefore there is no need for cognitive reflectional decisions.

If i am asking for the earliest flight to Frankfurt/M and the answer is "6.30h", then i have to accept it as the answer to my question. And nothing has to be interpreted, understood or decided. (Except, that the flight is much to early for my rituals.)

Semantics as a reflectional system is not dealing primarily with facts but with meanings. Meanings are at least reflectional multi-leveled, or as we know from Second-order Cybernetics, second-order concepts. That is concepts of concepts (of facts).

What is the purpose of a query system? A query system has to support and to assist decision-making for humans and as far as possible also for machines.

It seems reasonable to make a distinction between machine- and human-decidable decisions. Machine decidable decisions are on the level of dis-ambiguous dis-ambigue meanings, that is zero-level or 1-level meaning.

ambiguous ambigue
dis-ambiguous ambigue
ambiguous dis-ambigue
dis-ambiguous dis-ambigue

To make it easier, a simpler correlation to polysemy is possible by one-to-one, many-to-one, one-to-many and many-to-many relations. All well known in rhethorics and linguistics since Aristotle.

From Metapattern to Ontoprise

1 Parallelism in Polycontextural Logic

Additionally to the well known OR- and AND-parallelism, polylogical systems offer two main extensions to the *logical* modeling and implementation of parallelism. First the distribution of the classical situation over several contextures and second, the trans-contextural distributions ruled by the different transjunctional operators. The distribution over several contextures corresponds to a concurrent parallelism where the different processes are independent but structured by the grid of distribution. The trans-contextural parallelism corresponds to a parallelism with logical interactions between different contextures.

> "The tree corresponding to the search for a solution to a question seems open to various kinds of parallelism. The most obvious technique, called OR parallelism, allows processes to search disjunctive subtrees in parallel, reporting back to the parent node the result(s) of the search.
> The advantage of OR parallelism is that the searches are completely independent of each other and may execute concurrently (except that both may share access to a common data base storing facts and rules). The process performing the search of one subtree does not communicate with processes searching other subtrees." Michael J. Quinn, 212, 1987

Prolog is based not only on its logic, used as an inference machine, but also on its semantics or ontology, realized as a data base. Therefore the process of parallelising has to deal with a deconstructive dis-weaving of the data base´s ontology.

1.1 Strategies towards a polycontextural parallelism in Prolog

Like in the case above, where the number systems had to be cloned, in the Prolog case, the data base has to be decomposed into disjunct parts. These separated conceptual parts, or conceptual subsystems, have to be distributed over different contextures in a mediated polycontexturality.

Additionally the Prolog parallelism which is based on OR- and AND-parallelism has to be mapped into distributed logics, that is, into a polylogical system.

The Prolog example allows to explain in more a plausible way the decomposition or cloning of the common universe of discourse, that is, the data base of facts, into different subsystems. And secondly it is easier to introduce parallelism based on polycontextural logic than on arithmetics and combinatory logics.

Polycontextural logic is not widely known but more accessible than combinatory poly-logic and poly-arithmetics, which I am just introducing. Additionally there exists since 1992 a working implementation of a tablex proof system of an interesting subsystem of polycontectural logics in ML, running on Unix systems like NeXT.

1.1.1 An intermediate step with Metapattern

As an intermediate step in the shift of conceptualization from a hierarchical to a heterarchical way of concept building it maybe helpful to use the strategy of metapattern (Wisse). Metapatterns are used as an new modeling strategy for complex informational systems. Metapatterns are not involved in changing the basic assumptions of programming languages or even their logic as with the PCL approach.

Metapatterns could be helpful to move the process of parallelisation from the OR- and AND-level, that is, from the logical level to the deeper level of the data base, with its facts and rules, shared by the classical parallelism.

> She can relax on a fixed object orientation because — the metapattern determines that — situation and object are relative concepts (Wisse 2001). A particular situation is also object in another, higher-level situation. Likewise, an object can act as situation in which another, lower-level object resides. Situation, then, is a recursive function of object and relationship. Wisse

Hierarchy or chiasm?

> It is this concept of situation that characteristically sets the metapattern apart from traditional object orientation (and provides it with advantages over OO; Wisse 2001). Compared to an object that (only) exists absolutely, an object believed to exist in a multitude a different situations can unambiguously be modeled – to be equiped – with corresponding behavioral multiplicity. Wisse 2001

> The radical conclusion from the orientation at situational behavior is that an object's identification is behaviorally meaningless. The modeler does not have to explicitly include something like an original signature in all her models. Essentially a privileged situation may implied. It serves the only purpose of guaranteeing sameness or, its equivalent, persistent identity across (other) situations. Being a situation in its own right, when included in a model it is represented by a seperate context. Made explicit or not, its role is to authenticate an object's identity in other situations by establishing the signature in other contexts.

Identity as a network of nodes

> Traditional object orientation assigns identity at the level of overall objects. Context orientation replaces this view of singular objects with that of plusrality within the object; the object always nneds a context to uniquely identify the relevant part of an overall object, which is what identifying nodes regulate. When behaviors are identical, no distinction between contexts is necessary.

1.2 Deconstruction of a typical PROLOG example

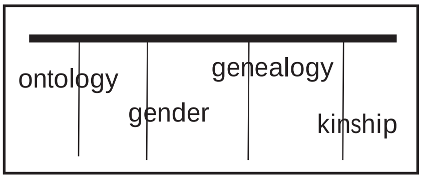
The classical prolog example to prove an “*aunt*”-relationship can be decomposed from its hierarchical ontology into different situations mapped into different contextures and visualized in the metapattern.

kinship: married / not-married, in-law, aunt

gender: male, female

genealogy: parent, sibling

ontology: different / not-different



It is also possible that there is some overdetermination because *parent* and *sibling* could also be part of *kinship.*

In Prolog all the facts belong to one ontology or to one semantic general domain or universe. All the rules are based on this mono-contextural ontology and on the corresponding logical operators AND and OR of the again, mono-contextural logic. Everything therefore is linearized and homogenized to a global or universal domain. This, if corresponding fairly with the real world situation is of great practicality and efficiency in both direction, in the case of the formal system, Prolog, and in the case of its data base.

But often, if not always, real world applications are much more complex than this. Even the fairly classical example is presupposing all sorts of facts which are not mentioned in the definition and which would belong to a different real world situation.

I don’t criticize this kinship model. It is doing its job to explain in a first step Prolog perfectly. Again, I am using this example for deconstructive reasons, that is for introducing the PCL way of thinking. This is, again a form, I guess, of legitimate abuse of classical models.

Instead of linearizing the above separated contextures *kinship, gender, genealogy, ontology* into one universal domain, for the example here represented by *kinship*, the polycontextural modeling is asking for an interweaving and mediating of these different contextures together to a complex poly-contexturality.

Compared to the original mono-contextural modeling this is involving much more complicated mechanisms than it is necessary in the classical case.

Why should we model a simple situation with highly complex tools into a complex model if we can solve the problem with much simpler tools? Simply because the classical approach lacks any flexibility of modeling a complex world. The truth is, that the simple approach needs an enormous amount of highly complicated strategies to homogenize its domains to make it accessible for its formal languages.

To decompose the basic classical ontology into different disjunct domains is a well known procedure and should not be confused with the decomposition, or de-sedimentation of an ontology in the PCL case. In PCL the domains are not simply disjunct and embraced by the general ontology but interwoven in a complex mechanism of interactions.

1.2.1 Polylogical modeling of the metapattern

The metapattern approach has helped to dissolve the hierarchical conception of the "aunt"-relation into different aspects.

In Prolog, the aunt-relation is defined as follows:

ant(x,y):= female(x), sibling(x,z), parent(z,y).

additionally the rule for sibling is:

sibling(x,y):= parent(z,x), parent(z,y), (x/==y).

The aunt-function is fullfilled and is true, if all components which are connected by the conjunction *et* (AND) are true.

true(aunt(x,y) iff (true(female(x)) et true(sibling(x,z)) et true(parent(z,y)))

Metapattern distribute the AND (or: et) over different heterarchical places but gives no formalism to handle this distribution. Polylogics is also distributing these conjucn-tions but in transforming them at the same time into operators of mediation. Polylogics is shortly defined as a distribution and mediation of classical logics.

ant(x,y) := female(x) § sibling(x,z) § parent(z,y)

sibling(x,y):= parent(z,x) § parent(z,y) § (x/==y)

Therefore the polylogical truth-function is transformed to:

aunt(x,y) eTrue ==> $aunt^{(3)}$e(x,y) e (T_1,T_2,T_3)

The metapattern of parts of the formulas can be transformed into the diagram.

S1	female(x)			
S2		sibling(x,z)		
S3			parent(z,y)	
S4				aunt(x,y)

How to read the transformation?

In Prolog, each term as such has an identical meaning. If the variable x is denoted with “mary” and mary is female, then the relation or attribute female(mary) is true. Also the variables x, y, z,... are identical. Obviously no “x” will be read as an “y”; we don´t make a "x" for a "u".

In polylogic the situations are happily a little bit more flexible. The variables are flex-ible to occur as variables in different systems. The variable “x” can occur as the vari-able x in system S1, that is the variable x can occur *as* variable x1.

In the same sense the denotation “mary” can occur as female or as sibling or as parent or as something else. Mary *as* Mary, again something else, maybe a secret.

Our model suggest the following reading:

x as female: x1 and mary as female: mary1

x as sibling: x2 mary as sibling: mary2

z as sibling: z2 stuart as sibling: stuart2.

y as parent: y3 kathleen as parent: kathleen3

z as parent: z3 edward as parent: z3

The result: aunt(mary,kathleen).

x as aunt: x4 mary as aunt: mary4

y as -aunt: y4 kathleen as beeing in relation to her aunt: kathleen4

Also the simultaneity for "mary" of being female and sibling, which is ruled in the Prolog model by the conjunction “et”, is realized in the polylogical model, obviously by the mediation rule “§”.

This example is very simple because the elements of the partition are simple, there are no composed formulas included. Insofar there is no need to involve polycontextural negations, junctions and transjunctions. Only the operator of mediation "§" between distributed attributes and relations are involved.

Only if we freeze the scenario to a static ontological system all the flexibility of the as-function, not to confuse with the as-if-function, can boil down to the well known non-flexible structure. But to allow a flexible ontology with x as x1, as x2, etc. or mary as female, as sibling, etc. allows to change ontology and to be ready for new situations without starting the system from scratch. It is easy to freeze complexity, but there are no known rules how to make a frozen and dead systems alive. Maybe that’s the reason why artificial life is nevertheless so hard.

1.2.2 Prolog´s ontology

Prolog refers as it has to do as a programming language based on First Order Logic (FOL) on attributes, relations between attributes and inference rules etc. and not on behaviors and contexts.

To be a *parent* is classically an attribute of a person, described as a relation to other persons, in PCL this attribute becomes a *behavior*, maybe of a person, in a complex situation. To be parents is not necessary connected with the attribute to be *married*, to be a *sibling* has not to be restricted to have the same parents, to be married has not to involve different *gender*, and so on. And even that a person is different to another person, or that the person is identical to itself is not as natural as it seems to be. All these presumptions are reasonable, and are corresponding to possible real world models only if all the possible ambiguities and over-determinations are ruled out in favor to a very special model of kinship.

The solution to this situation of complexity is not so much to enlarge the given ontology and to introduce the new differences and attributes to cope with the new situation. Because this strategy is based on the exact same ontological presuppositions and is therefore only repeating the old scenario again.

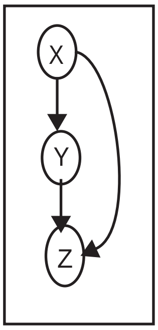
In the framework of PCL mechanism are offered for a great flexibility in interlocking and interweaving different points of view, situations, and modeling.

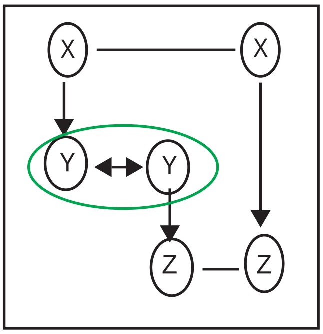
The decomposition of an universal domain into its different components is not only introducing a conceptual advantage for the process of modeling but also on a computational level a new form of parallelism is introduced.

The whole manoeuvre is quite similar to what I proposed as a proemial relation between sorts and universes in many-sorted first order logics.

1.2.3 The devil is in the detail

Polycontexturality is not starting somewhere in a complexity, it is virulent at the very beginning of the basic definition of relationships.





Y as child of X and Y as the father of Z has to be mediated, synchronized, realized. Only in a stable hierarchical ontology this relationship of Y as “child of” and “father of” is automatically connected. And therefore “father of father” can be equal to “grandfather” and realized by a conjunction of the two relations, father(X,Y) et father(Y, Z) eq grandfather(X, Z).

In a polycontextural setting this identity of Y, as child and as father, can not be presupposed but has to be established in a possible context. Y as child and Y as father has to be brought together in a way that the transitivity can hold. It is easily possible that the transitivity is broken for some reasons and that it has to be re-established. The reason why the transitivity can be broken lies in the poly-contextural assumption that a entity or a relation is not a simple identity but involved in a cluster or an intersection of a multitude of possible contextures. Only for restricted and regulated situations a complex situation can be reasonably reduced to a mono-contextural one in which transitivity holds unrestricted. Therefore, identity can not be presupposed it has to be realized from case to case.

Because of the relative autonomy of both relations in a complex kinship system, we can calculate and study them simultaneously, realizing some elementary parallelism. This is obviously not possible in a strict biological interpretation of the father-child-relation. There we have to accept the hierarchical dependencies of the relations. But again, we have to be aware that this is the case only because we restrict the setting to a mono-contextural case. In contrast, real world social relations are always highly complex.

Therefore we have two options, the mono- and the polycontextural. The advantage of the later one is flexibility, the advantage of the first one is stability. Both have there weakness, flexibility is risky and dangerous, stability is restricting and killing.

2 Ontological transitions

2.1 From Types to behaviors

Identity as a network of nodes

Traditional object orientation assigns identity at the level of overall objects. Context orientation replaces this view of singular objects with that of plurality within the object; the object always neds a context to uniquely identify the relevant part of an overall object, which is what identifying nodes regulate. When behaviors are identical, no distinction between contexts is necessary.

From OO: super-level (type: person) —> sub-level(type: national), (type(foreigner) to

metapattern: (nationalship: person), (foreignship: person), (personship: person).

The class hierarchy of the OO model is transformed to a heterarchical model of behaviors, that is simultaneously ruling contexts.

2.2 From behaviors to interactivity

Behaviors, realized as in situations and contexts comes in plurality.

But metapattern doesn´t offer much mechanism of navigation between simultaneous contexts. What we get is the notion of a pointer, *"pointer information objects"*. They are supporting navigation from one context to another. But these pointers don´t give a hint how they could be implemented.

Metapattern points to the relevance of points of view.

From Context(type / instance) to Contextures(context(type / instance))

2.3 From objects to objectionality

2.4 The hidden rules: logic and interferencing

In contrast to the *modelling* aspect emphasized by the metapattern approach, from the point of view of *implementation* of the conceptual models we have to consider the underlying logics of the informational system, here ontologies for the Semantic web.

With this turn we are enabled to show the overwhelming advantage of the PCL approach over the classical modelling and implementing standards. It is the polycontectural, that is the polylogical apparatus which is framing the implementation of the deconstucted ontologies with the help of the metapattern. Without a polylogical implementation, the metapattern is an important modelling device but gives no guidelines for its real world implementation. This can by realized by polylogical funded data base logics.

Data base logics, as F-logic, are grounded on First Order Logics (FOL).

Normally, the user of say OntoEdit, is not involved in the questions of implementations. But to give the OntoEdit more flexibility, the user is offered a "General Axiom" plugin which allows her to define and edit axioms.

To check your new axioms an inferencing plugin is offered.

> Inferencing
> The inferencing plugin can be used to test the ontology and its axioms. In the text field on the upper right you can type queries to query the data model. These queries have to be in F-Logic syntax.

Obviously, the new rules added by the user are only useful if they correspond to FOL.

> F-Logic Tutorial, ontoprise GmbH
> Based upon a given object base (which can be considered as a set a facts), rules offer the possibility to derive new information, i.e., to extend the object base intensionally. Rules encode generic information of the form: Whenever the precondition is satisfied, the conclusion also is. The precondition is called rule body and is formed by an arbitrary logical formula consisting of P- or F-molecules, which are combined by OR, NOT, AND, <-, -> and <->. A -> B in the body is an abbreviation for NOT A OR B, A <- B is an abbreviation for NOT B OR A and <-> is an abbreviation for (A->B) AND (B<- A). Variables in the rule body may be quantified either existentially or universally. The conclusion, the rule head, is a conjunction of P- and F-molecules. Syntactically the rule head is separated from the rule body by the symbol <- and every rule ends with a dot. Non-ground rules use variables for passing information between subgoals and to the head. Every variable in the head of the rule must also occur in a positive F-Atom in the body of the rule. Assume an object base defining the methods father and mother for some persons, e.g., the set of facts given in Example 2.1.
> The rules in Example 7.1 compute the transitive closure of these methods and define a new method ancestor:
> FORALL X,Y X[ancestor->>Y] <- X[father->Y].
> FORALL X,Y X[ancestor->>Y] <- X[mother->Y].
> FORALL X,Y,Z X[ancestor->>Y] <- X[father->Z] AND Z[ancestor->>Y].
> FORALL X,Y,Z X[ancestor->>Y] <- X[mother->Z] AND Z[ancestor->>Y].
> man::person.
> woman::person.
> 8.2. Queries
> A query can be considered as a special kind of rule with empty head. The following query asks about all female ancestors of Jacob:
> FORALL Y <- jacob[ancestor->>Y:woman].
>
> The answer to a query consists of all variable bindings such that the corresponding ground instance of the rule body is true in the object base.

2.5 From Information to Knowledge

Is the logic of data, information and records the same as the logic of knowledge? And further, is logic enough for representing knowledge?

I don´t want to go into the interesting discussions about the relationship of logic and knowledge representation languages as developed by the AI researchers long ago. What has to be mentioned is that in their different approaches they all introduced some two-level languages of object-level and meta-level theories.

To give a further motivation to introduce a poly-contextural view of data-base systems it maybe helpful to use the difference between logic of data and logic of knowledge.

The logic of data is quite strict, and well established by the classical systems of logic. Data are strictly non-ambiguous, they maybe not precise, but there is no need for hermeneutical interpretation. Data are in this sense facts. There linguistic model is the name. Facts have names and names are unambiguos, they name an entity. If someone, a person, is called “Meyer”, he is not called in the same sense “Mueller”. If a data-base consists of data as facts, the rules of logic apply without any restrictions. It is therefore natural to mix these data systems with a hierarchical concept system and to represent them as trees with a single root. The basic names of the Web are URIs, they are based in numbers, and these don´t need any hermeneutics.

But the situation can be considered in a radical different way. If the data-base consists not so much of data as facts but of data as concepts, there is no need to accept the hierarchical system of the classical solution.

If a person is called “Mueller”, it´s about facts. If we deal with “persons” it´s not about facts it´s about concepts. Concepts and categories can be understood by the ontological model of names. This is the Aristotelian way. But this is, as we have learned in contemporary philosophy long ago, not the only way. It is a very restricted and obsolete position. Unfortunately it is what we learn from the ontologies of the Semantic Web.

The knowledge about facts is different from the knowledge about concepts. The knowledge about concepts involves some meta-language knowledge which belongs to another logical level than object-language knowledge.

The hierarchic architecture of concepts, as introduced by Aristotle and Porphyr, is a possible but not a necessary solution. It is oriented by object-knowledge. With this approach concepts are produced by abstraction over data sets. Objects, data, records, etc. are first. They have their identity defined on their object-level. There is no change of identity for objects. They are what they are. In this case, concepts are used to produce knowledge about objects and not knowledge about concepts.

Polycontexturality, like the metapattern approach, takes a different strategy. Objects are objects only in relationship to contexts. More adequate, objects are understood by their behavior. Therefore, an abstract object without any behavior, independent of contexts doesn’t exist; it is a nil object.

Therefore, classical objects, like data, have a one-level behavior, they exist by being named. They are the result of the process of naming.

Semiotically we are making a shift from the dualistic to a trichotomic semiotics, and further to a chiastic graphematics.

What are the objects of the Semantic Web?

While formalizing the principles governing physical objects or events is (quite) straightforward, intuition comes to odds when an ontology needs to be extended with non-physical objects, such as social institutions, organizations, plans, regulations, narratives, mental contents, schedules, parameters, diagnoses, etc. In fact, important fields of investigation have negated an ontological primitiveness to non-physical objects [7], because they are taken to have meaning only in combination with some other entity, i.e. their intended meaning results from a statement. For example, a norm, a plan, or a social role are to be represented as a (set of) statement(s), not as concepts. This position is documented by the almost exclusive attention dedicated by many important theoretical frameworks (BDI agent model, theory of trust, situation calculus, formal context analysis), to states of affairs, facts, beliefs, viewpoints, contexts, whose logical representation is set at the level of theories or models, not at the level of concepts or relations

Sowa ??

Interactions in a meanigful world

1 Queries, question-answering systems

Questions are not innocent. There are no neutral questions. This is obviously true for human communication. But it is naive to think that questions to information systems are excluded from this constellation.

Data mining, elicitation and collection of explicit or implicit information, that is pregiven implicit or explicit answers to well-formed questions from a query system.

2 Diamond based interrogative systems

Questions which are not restricted to information about facts are including aspects of relevance, significance, context dependendness and other criteria of meaningful answers.

A simple scheme to support meaningful questions is given by the Diamond Strategies I introduced long ago.

3 Evocative communications

> William Olander in 1987: "Clough has developed yet another hybrid—a painting which is simultaneously genuine and artificial, cultural and natural, full and empty, without resorting, overtly at least, to the ideological apparatuses of late modernism."4 and Clough characterizes as: "transformation, inflection, turbulence; a very particular vibrating cosmic tension; weave of force; harmonics of intentionality; subliminal erotics of creation; spontaneity, evocativity; meaning as desire and fear in smoky arabesque; rippling quench; refracting enigmatic shimmer; the lethal chop of value; subtle ofity of itness; dancing with tradition, accepting, rejecting and relentless execution; the power in the compulsion to create as a measure of the ultimacy of humanness, depth of drama; a pulsing overlay, overlap, palimpsest, wave upon wave to come again & again & again..."
>
> —Nancy Whipple Grinnell, Curator, Newport Art Museum

Evocative questioning is beyond elicitation and installation (suggestion) and is opening up in a co-creative interplay new answers to new questions, new horizons of questioning.

How are we questioning an object which is characterized by highly hybrid, full of ambiguity and surprising paradoxes? Obviously it can not be done in the same way as we ask for a vacuum cleaner.

On Deconstructing the Hype

This little exercise of deconstruction follows the simple scheme of the DiamondStrategies. All 4 positions of a context, affirmation, negation, neither-nor and both-at-once, have to be considered. All in the same strength of the argument. Here, in deconstructing the hype, that is the position, or positioning of the semantic web and similar, the dynamic semantic web, it will be at first restricted only to the process of rejecting, dualizing, reflecting the not reflected preconditions of the position and not involving the 2 positions of full rejection (neither-nor) and full acceptance (both-at-once).

1 The hype of the distributed, decentralized and open Web

At the beginning of our study we learnt that the Web is at least distributed, decentralized and an open world.

> ***The Web is distributed.*** *One of the driving factors in the proliferation of the Web is the freedom from a centralized authority.*
>
> *However, since the Web is the product of many individuals, the lack of central control presents many challenges for reasoning with its information.*
>
> *First, different communities will use different vocabularies, resulting in problems of synonymy (when two different words have the same meaning) and polysemy (when the same word is used with different meanings).*

There is no reason to deny this description at least as a starting point. Remember, the description of the weather system sounds very similar. But all these emphases of the openness and decentralized distributedness of the Web is describing not much more than the very surface structure of the Web. It emphasizes the use of the Web by its users not the definition and structure, that is, the functioning of the Web. There are no surprises at all if we discover that the structure of the Web is strictly centralized, hierarchic, non-distributed and totally based on the principle of identity of all its basic concepts. The functioning of the Web is defined by its strict dependence on a “centralized authority”.

If we ask about the conditions of the functioning of the Web we are quickly aimed at its reality in the well known arsenal of identity, trees, centrality and hierarchy.

Why? Because the definition of the Web is entirely based on its identification numbers. Without our URIs, DNSs etc. nothing at all is working. And what else are our URIs then centralized, identified, hierarchically organized numbers administrated by a central authority?

Again, all this is governed by the principle of identity.

“We should stress that the resources in RDF must be identified by resource IDs, which are URIs with optional anchor ID.” (Daconta, p. 89)

What is emerging behind the big hype is a new and still hidden demand for a more radical centralized control of the Web than its control by URIs. The control of the use, that is of the *content* of the Web. Not on its ideological level, this is anyway done by the governments, but structurally as a control over the possibilities of the use of all these different taxonomies, ontologies and logics. And all that in the name of diversity and decentralization.

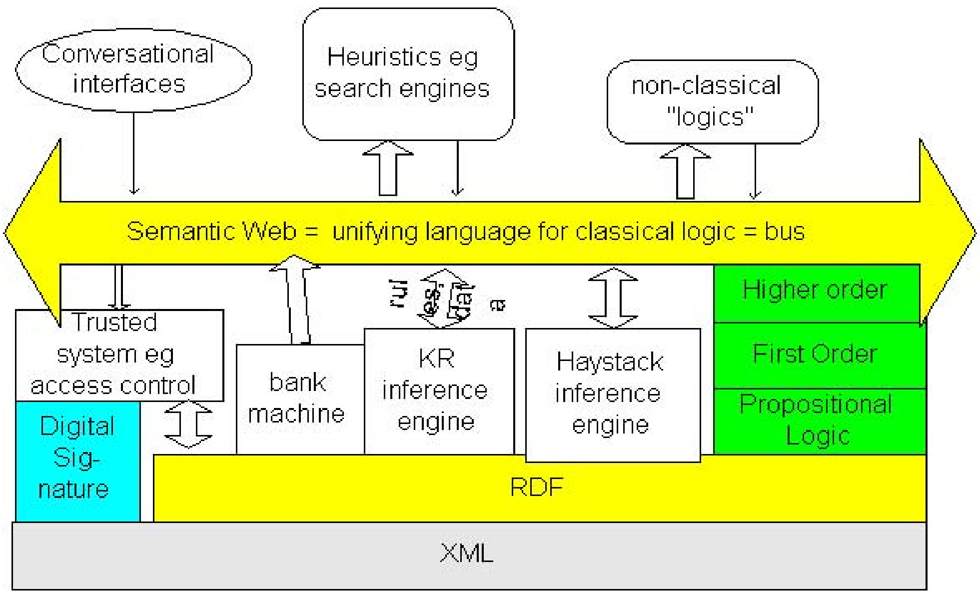
All the fuss about the freedom of the (Semantic) Web boils down to at least two strictly centralized organizational and definitorial conditions: URI and GOL.

It is not my intention to deny the massive complexity of the Web and the growing Semantic Web on its surface structure. Again, remember:

> The World Wide Web currently links a heterogeneous distributed decentralized set of systems.
> Some of these systems use relatively simple and straightforward manipulation of well-characterized data, such as an access control system. Others, such as search engines, use wildly heuristic manipulations to reach less clearly justified but often extremely useful conclusions.
>
> In order to achieve its potential, the Semantic Web must provide a common interchange language bridging these diverse systems.

http://www.w3.org/2000/01/sw/DevelopmentProposal

Nevertheless, it is important not to confuse the fundamental difference of deep-structure and surface-structure of the Semantic Web. This fundamental difference of deep/surface-structure is used in polycontextural logic not as a metaphysical but as on operational distinction. And all the Semantic Web "cakes" are confirming it.
Here is another one from the W3C, its hidden cards, Unicode and URI, are shown in another game. Unicode and URI are the deepest layer of the Semantic Web Cake.



Beyond the layer of Unicode and URI we have to add their arithmetical and code theoretical layers. The Semantic Web Cake is accepting the role of logic, down to its propositional logic, but is not mentioning arithmetics. As we have seen in Derrida´s Machines, arithmetics and its natural numbers are pre-given and natural. There is not much to add. There are many possible open questions with Unicode and URI, but not with its common arithmetics.
The open question which comes back to my proposal is “Why should the deep structure of the Web be questioned?”. At least, it is working. A simple answer, it is not enough. There are to many problems open which cannot be solved properly in the

framework of the existing paradigm.

2 Conflicts between diversity and centralization of ontologies

Our media philosophers are still fantasizing about the virtuality of the Web and the new Global Brain and bodiless decentralized sex, but there is no worry, the authority of the URI is controlling the game from the very beginning. And now we are going a step further, still not remarked by the critical media studies, and have to deal with a much more sophisticated attempt to the centralization and control of the Web by the GOL. Without a General Ontology Language there is no Semantic Web at all. GOL maybe made explicit or may remain in the background, as a new cyber-unconsciousness like the URIs, but it is ruling together with the Unicode and URIs the whole game.

> The development of an axiomatized and well-established upper-level ontology is an important step towards a foundation for the science of Formal Ontology in Information Systems.
>
> Every domain-specific ontology must use as a framework some upper-level ontology which describes the most general, domain-independent categories of reality.
>
> For this purpose it is important to understand what an upper-level category means, and we proposed some conditions that every upper- level ontology should satisfy.
>
> The development of a well-founded upper-level ontology is a difficult task that requires a cooperative effort to make significant progress.

Why do we have to make such a drama about say, polysemy, if the Semantic Web is really in any sense decentralized etc.?

Our global village is dealing with the same, and simple problems, of the old Greek marketplace of discussions, all waiting for a great generalist, Aristotle, to make an end of the semantic chaos by introducing his GOL and Logic.

There is no surprise that the GOL of the Semantic Web is proud to be Aristotelian, it doesn´t change much to be more progressive with Whitehead , Bunge, Kripke or Montague.

All that is not working without conflicts. As we know from Guarani and probably also from the long history of western philosophical, logical and ontological thinking.

> Two different contexts relating respectively to species and environment point of view. With such different interpretations of a term, we can reasonably expect different search and indexing results. Nevertheless, our approach to information integration and ontology building is not that of creating a homogeneous system in the sense of a reduced freedom of interpretation, but in the sense of **navigating alternative interpretations**, querying alternative systems, and conceiving alternative contexts of use.
>
> To do this, we require a comprehensive set of ontologies that are designed in a way that admits the existence of **many possible pathways** among concepts **under a common conceptual framework**.
>
> This framework should reuse domain-independent components, be flexible enough, and be focused on the main reasoning schemes for the domain at hand. **Domain-independent, upper ontologies** characterise all the general notions needed to talk about economics, biological species, fish production techniques; for example: parts, agents, attribute, aggregates, activities, plans, devices, species, regions of space or time, etc. (emphasis, r.k.)

http://www.loa-cnr.it/Publications.html

The conflict between the desire and necessity to *"navigate alternative interpretations"* and the need of *"domain-independent upper ontologies"* is obvious and not easy to deal. Its virulence is quickly stopped by the acceptance of GOL, responsible for the definition of such simple things like *"parts, agents, attribute, aggregates, activities, plans, devices, species, regions of space or time"*.

As we know there are significantly different approaches to ontology
entity ontology (substantialism)
process ontology (functionalism)
system ontology (system theory)
structure ontology (structuralism)
difference ontology (deconstructivism)
and many more. Especially, there is also thinking and being beyond ontology.

It will turn out that the general theory is not so much an ontology GOL but a theory of translating and mediating different ontologies, first order as well second-order ontologies. A Dynamic Semantic Web would add to the translations some mechanisms of transformation and metamorphosis.
Its main candidate is well known too: category theory, the ultimate theory of translation.

3 Trees, Hierarchies and Homogeneity

The general language of the Semantic Web is XML. But what is XML? Short: a tree. The same is true for the other languages like RDF.
As developed in Derrida´s Machines the main structure of formal thinking is natural. Everything has an origin and is embedded in a tree. Natural deduction systems, natural number systems and also the limits of this paradigm of thinking is natural. And this is also the way the Semantic Web is organized. XML is a tree. The tree is natural and universal.
Again.

> *As Natural as 0,1,2*
> *Philip Wadler. Evans and Sutherland Distinguished Lecture, University of Utah, 20 November 2002.*
> *"Whether a visitor comes from another place, another planet, or another plane of being we can be sure that he, she, or it will count just as we do: though their symbols vary, the numbers are* universal. *The history of logic and computing suggests a programming language that is equally* natural. *The language, called lambda calculus, is in exact correspondence with a formulation of the laws of reason, called* natural deduction. *Lambda calculus and natural deduction were devised, independently of each other, around 1930, just before the development of the first stored program computer. Yet the correspondence between them was not recognized until decades later, and not published until 1980. Today, languages based on lambda calculus have a few thousand users. Tomorrow, reliable use of the Internet may depend on languages with logical foundations. "*

But the Semantic Web is artificial, and nobody until now has given a proof that the nature of artificiality is of the same nature as the concept of nature in all these natural deductions, natural numbers et al. Even to make such a distinction between natural and artificial is considered as obsolet and cranky by the academia.

4 Structuration: Dynamics and Structures

What have we learnt on our trip around the fascinating perspectives and problems of a Dynamic Semantic Web?

It is all about dynamics and structures. This brings us back to the central topics of DERRIDA´S MACHINES: Interactivity between structures and dynamics, that is, to the interplay of algebras and co-algebras, ruled by category theory and surpassed by the diamond strategies leading to polycontexturality and kenogrammatics.

We arrive back to terms like translation, metamorphosis, polycontexturality, kenogrammatics, algebra and co-algebra, swinging types of algebras and co-algebras, etc.

A new effort has to be undertaken to collect the concepts, problems and methods of the Semantic Web into a more general and formal framework.

Not surprisingly, the main topic of the Semantic Web is translation, in other words a "interchange language". Translation of taxonomies, ontologies and logics. Translation as interaction, merging and transforming different domains, points of view, contexts. The most general approach to translation is given by the methods of category theory and semiotic morphisms (Goguen) not yet applied by the Semantic Web community. In this sense, translation is conservative, keeping the linguistic categories, tectonics and topoi together, that is, saving the meanings during the process of translation.

It seems to be obvious, that the languages of translation, mediation and metamorphosis are not languages of a general ontology as containing the *"most general, domain-independent categories of reality"* but languages which are neutral to ontologies, describing what happens between ontologies. There purpose is not intra-ontological but inter-ontological, mediating ontologies and not functioning themselves as ontologies.

Dynamics is not only covered by conservative interchange but interwoven in permanent transformations ruled by the play of metamorphosis. Metamorphosis can be understood as an unrestricted interplay of categories disseminated in a polycontextural framework. Metamorphosis is not only preserving but subverting meanings in the process of interactivity. Translation is interchange, metamorphosis is creation of new meanings.

The behavior of the Semantic Web is best modelled in terms of an interplay of algebras and co-algebras in the general framework of category theory. But this is as I have shown enough only a very first step in modeling the interactivity of autonomous systems. This means, that I reject the idea of modeling the structural dynamics / dynamical structure by category theoretical morphisms only.

Interactivity comes with reflectionality, architectonics and positionality. These topics have to enter the game to design a more dynamic Semantic Web as it is considered by the very simple and conservative procedures of merging and integrating ontologies and creating contextual concept spaces.

5 Problems with semantics?

Do we introduce semantics with the addition of a second dimension of syntax to the well known syntax of XML? Is a double syntax enough to establish semantics?

Questions of this kind are very old and goes back to the 1930th when symbolic (mathematical) logic was looking for semantic foundations. It develops in a long chain of names like Tarski, Scholz, Hasenjaeger to model theory and from there to mathematical linguistics with Montague and producing all sorts of criticism, one from formalism with the claim that formal semantics is in itself nothing else than a second syntactic formal system (Curry) and from pragmatism, explaining that even semantics is not enough and has to be developed from a dialogical (Lorenzen) or game theoretic approach (Hintikka).

To introduce semantics into a formal system is not an easy thing if we start with syntax then adding semantics and pragmatics to it, repeating the classical "semiotic cake" of Morris. This is the well known historical way of doing things, it's structure is obviously hierarchic. it should be mentioned that the Morris approach is more a popularization of the genuine concepts of Charles Sander Peirce than a further development of semiotics. Peirceian semiotics is not a hierarchic system of syntax, semantics and pragmatics, but an irreducible triadic-trichotomic design of semiotics. There is no Peirceian cake. The advantage of Morris' cake is its hierarchical order which is compatible to a classical formal logic understanding. The Peirceian trichotomy is strictly heterarchic, demanding for a non-hierarchic concept of logic and mathematics (Peircian tricotomic mathematics) which is still very hard to be developed.

The opposite of hierarchy is heterarchy. To deconstruct this hierarchical way of introducing semantics we have to propose a heterarchical structure of semiotics, parallelizing the chain of syntax, semantic, pragmatic and what ever to heterarchical structure. But this is even less easy done than the classical approach. And further more, a heterarchical approach is not simply parallelizing the aspects of semiosis but is involved into a dynamic metamorphosis of these aspects. Semantics is not simply semantics per se, from another point of view it is equally functioning as a syntactic or pragmatic aspect of the whole process of semiosis.

6 Problems with inferencing?

SHOE Ontology Example "CS Department"

```
<HTML>
 <HEAD>

 <!-- Here we indicate that this document is conformant with SHOE 1.0 -->

 <META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">


 <TITLE> Our CS Ontology </TITLE>
 </HEAD>
 <BODY>

 <!-- Here we declare the ontology's name and version -->

 <ONTOLOGY ID="cs-dept-ontology" VERSION="1.0">

 <!-- Here we declare that we're borrowing from another ontology -->

 <USE-ONTOLOGY ID="base-ontology" VERSION="1.0" PREFIX="base"
           URL="http://www.cs.umd.edu/projects/plus/SHOE/base.html">

 <!-- Here we lay out our category hierarchy -->

 <DEF-CATEGORY NAME="Organization" ISA="base.SHOEEntity">
 <DEF-CATEGORY NAME="Person" ISA="base.SHOEEntity">
 <DEF-CATEGORY NAME="Publication" ISA="base.SHOEEntity">

 <DEF-CATEGORY NAME="ResearchGroup" ISA="Organization">
 <DEF-CATEGORY NAME="Department" ISA="Organization">
 <DEF-CATEGORY NAME="Worker" ISA="Person">
 <DEF-CATEGORY NAME="Faculty" ISA="Worker">
 <DEF-CATEGORY NAME="Assistant" ISA="Worker">
 <DEF-CATEGORY NAME="AdministrativeStaff" ISA="Worker">
 <DEF-CATEGORY NAME="Student" ISA="Person">
 <DEF-CATEGORY NAME="PostDoc" ISA="Faculty">
 <DEF-CATEGORY NAME="Lecturer" ISA="Faculty">
 <DEF-CATEGORY NAME="Professor" ISA="Faculty">
 <DEF-CATEGORY NAME="ResearchAssistant" ISA="Assistant">
 <DEF-CATEGORY NAME="TeachingAssistant" ISA="Assistant">
 <DEF-CATEGORY NAME="GraduateStudent" ISA="Student">
 <DEF-CATEGORY NAME="UndergraduateStudent" ISA="Student">
 <DEF-CATEGORY NAME="Secretary" ISA="AdministrativeStaff">
 <DEF-CATEGORY NAME="Chair" ISA="AdministrativeStaff Professor">

 <!-- And now we lay out our relationships between categories -->
```

```
 <DEF-RELATION NAME="advisor">
    <DEF-ARG POS="1" TYPE="Student">
    <DEF-ARG POS="2" TYPE="Professor">
 </DEF-RELATION>

 <DEF-RELATION NAME="member">
    <DEF-ARG POS="1" TYPE="Organization">
    <DEF-ARG POS="2" TYPE="Person">
 </DEF-RELATION>

 <DEF-RELATION NAME="publicationAuthor">
    <DEF-ARG POS="1" TYPE="Publication">
    <DEF-ARG POS="2" TYPE="Person">
 </DEF-RELATION>

 <!-- Lastly, we lay out our other relationships -->

 <DEF-RELATION NAME="publicationDate">
    <DEF-ARG POS="1" TYPE="Publication">
    <DEF-ARG POS="2" TYPE=".DATE">
 </DEF-RELATION>

 <DEF-RELATION NAME="age">
    <DEF-ARG POS="1" TYPE="Person">
    <DEF-ARG POS="2" TYPE=".NUMBER">
 </DEF-RELATION>

 <DEF-RELATION NAME="name">
    <DEF-ARG POS="1" TYPE="base.SHOEEntity">
    <DEF-ARG POS="2" TYPE=".STRING">
 </DEF-RELATION>

 <DEF-RELATION NAME="tenured">
    <DEF-ARG POS="1" TYPE="Professor">
    <DEF-ARG POS="2" TYPE=".TRUTH">
 </DEF-RELATION>

</ONTOLOGY>
 </BODY>
 </HTML>

<DEF-INFERENCE DESCRIPTION="Transitivity of Suborganizations">
<INF-IF>
<RELATION NAME="subOrganization">
<ARG POS="FROM" VALUE="x" USAGE="VAR">
<ARG POS="TO" VALUE="y" USAGE="VAR">
</RELATION>
```

```
<RELATION NAME="subOrganization">
<ARG POS="FROM" VALUE="y" USAGE="VAR">
<ARG POS="TO" VALUE="z" USAGE="VAR">
</RELATION>
</INF-IF>

<INF-THEN>
<RELATION NAME="subOrganization">
<ARG POS="FROM" VALUE="x" USAGE="VAR">
<ARG POS="TO" VALUE="z" USAGE="VAR">
</RELATION>
</INF-THEN>
</DEF-INFERENCE>
```

7 Modularity

Modules can be added to a ontology by the <USE-ONTOLOGY> operation and adjusted with <DEF-RENAME>.

Modules are added conjunctively or disjunctively, that is hierarchically, to the ontology tree or lattice with a general ontology at its root.

The dynamics of the Dynamic Ontologies (Heflin, Hendler) are restricted to their hierarchical and mono-contextural order.

SHOE Semantics

Revisioning
Versioning

CNLPA-Ontology Modelling

1 CNLPA-ONTOLOGY-object

This type of modelling the CNLPA-ontology is focussed on its classes called *objects* in contrast to a later more *process*-oriented modelling proposed in the CNLPA-ontology-process. A step further towards a *polycontextural* modelling is introduced by the CNLPA-ontology-polylogic.

All three approaches are designed along the lines of the SHOE-CS Department-ontology.

```
<HTML>
 <HEAD>

 <!-- Here we indicate that this document is conformant with SHOE 1.0 -->

 <META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">

 <TITLE> Our CNLPA Ontology-object </TITLE>
 </HEAD>
 <BODY>

 <!-- Here we declare the ontology's name and version -->

 <ONTOLOGY ID=cnlpa-ontology-object" VERSION="1.0">

 <!-- Here we declare that we're borrowing from another ontology -->

 <USE-ONTOLOGY ID="base-ontology" VERSION="1.0" PREFIX="base"
              URL="http://www.cs.umd.edu/projects/plus/SHOE/base.html">

 <!-- Here we lay out our category hierarchy -->

 <DEF-CATEGORY NAME="Organization" ISA="base.SHOEEntity">
 <DEF-CATEGORY NAME="Person" ISA="base.SHOEEntity">
 <DEF-CATEGORY NAME="Publication" ISA="base.SHOEEntity">
 <DEF-CATEGORY NAME="Seminars" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Cooperations" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Buildings" ISA="base.SHOEEntity">

<DEF-CATEGORY NAME="ResearchGroup" ISA="Organization">
 <DEF-CATEGORY NAME="Department" ISA="Organization">

 <DEF-CATEGORY NAME="Worker" ISA="Person">
 <DEF-CATEGORY NAME="Faculty" ISA="Worker">
 <DEF-CATEGORY NAME="Trainer" ISA="Worker">
 <DEF-CATEGORY NAME="AdministrativeStaff" ISA="Worker">
 <DEF-CATEGORY NAME="Teilnehmer" ISA="Person">

 <DEF-CATEGORY NAME="Secretary" ISA="AdministrativeStaff">
 <DEF-CATEGORY NAME="Chair" ISA="AdministrativeStaff Trainer">
```

```
<DEF-CATEGORY NAME="Owner" ISA="CHAIR">

<DEF-CATEGORY NAME="unpublishPublication" ISA="Publication">
<DEF-CATEGORY NAME="publishPublication" ISA="Publication">

<DEF-CATEGORY NAME="printPublication" ISA="Publication">
 <DEF-CATEGORY NAME="audioPublication" ISA="Publication">
 <DEF-CATEGORY NAME="videoPublication" ISA="Publication">
 <DEF-CATEGORY NAME="webPublication" ISA="Publication">
<DEF-CATEGORY NAME="researchPublication" ISA="Publication">

<DEF-CATEGORY NAME="Office" ISA="Building">
<DEF-CATEGORY NAME="Academy" ISA="Building">
```

RULES:

```
 <!-- And now we lay out our relationships between categories -->


 <DEF-RELATION NAME="advisor">
    <DEF-ARG POS="1" TYPE="regularTeilnehmer">
    <DEF-ARG POS="2" TYPE="Trainer">
 </DEF-RELATION>

 <DEF-RELATION NAME="ausbilder">
    <DEF-ARG POS="1" TYPE="Teilnehmer">
    <DEF-ARG POS="2" TYPE="Faculty">
 </DEF-RELATION>


 <DEF-RELATION NAME="member">
    <DEF-ARG POS="1" TYPE="Organization">
    <DEF-ARG POS="2" TYPE="Person">
 </DEF-RELATION>

 <DEF-RELATION NAME="publicationAuthor">
    <DEF-ARG POS="1" TYPE="Publication">
    <DEF-ARG POS="2" TYPE="Person">
 </DEF-RELATION>

 <!-- Lastly, we lay out our other relationships -->

 <DEF-RELATION NAME="publicationDate">
    <DEF-ARG POS="1" TYPE="Publication">
    <DEF-ARG POS="2" TYPE=".DATE">
 </DEF-RELATION>
```

```
<DEF-RELATION NAME="age">
    <DEF-ARG POS="1" TYPE="Person">
    <DEF-ARG POS="2" TYPE=".NUMBER">
</DEF-RELATION>

<DEF-RELATION NAME="name">
    <DEF-ARG POS="1" TYPE="base.SHOEEntity">
    <DEF-ARG POS="2" TYPE=".STRING">
</DEF-RELATION>


addressStreet(Address, .STRING)
  addressCity(Address, .STRING)
  addressState(Address, .STRING)
  addressZip(Address, .STRING)


</ONTOLOGY>
 </BODY>
 </HTML>
```

Constants

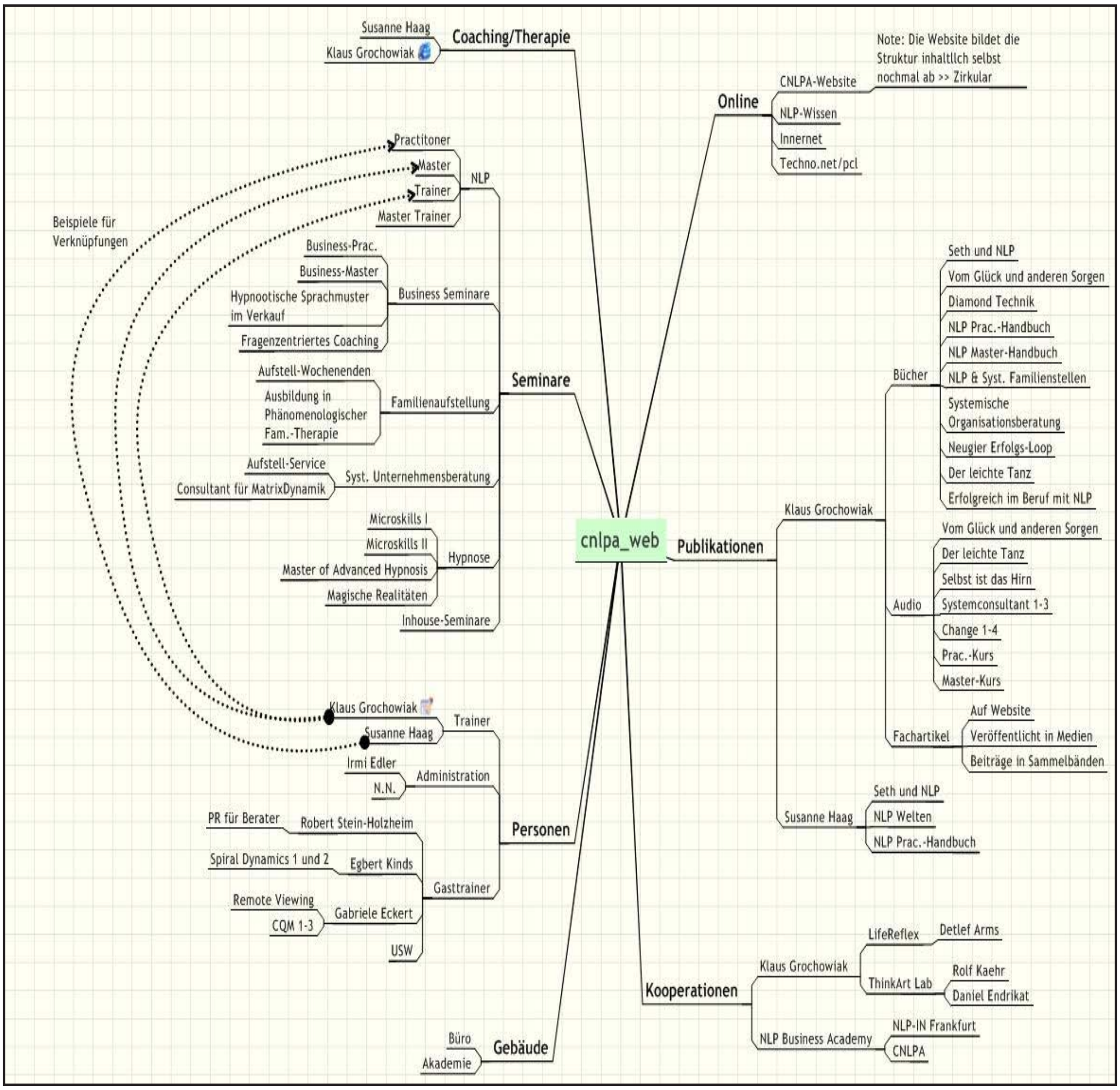
Constants are used to identify instances that may be commonly used with an ontology. In this section, each constant is grouped under its category.

Gender:
- Male
- Female

FACTS:
PERSONS: {Klaus, Susanne, Irmi, Robert, Egbert, ...}
Publication-TITLES
Publication-Dates
Seminar-Titles
Seminar-Dates
Seminar-Locations
Name-of-Cooperations-Partners

etc.

CNLPA-ontology as a Mindmap (Grochowiak, Stein)



2 CNLPA-ONTOLOGY-process

This modelling approach is focussing on the activities, behaviors, short *processes* of the organization CNLPA. Therefore, the base-ontology of SHOE has to be changed and augmented with processual categories allowing categoreis like Activity, Duration, Location and Personship as processes.

<HTML>
<HEAD>
Main Classes of Mindmap:

<DEF-CATEGORY NAME="Organization" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Person" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="**Publication**" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="**Seminars**" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="**Cooperations**" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="**Buildings**" ISA="base.SHOEEntity">

The Mindmap modelling is presupposing facts and relations about chronological dates, linguistic strings and numbers etc. which are modelled in the base-ontology

<USE-ONTOLOGY ID="base-ontology" VERSION="1.0" PREFIX="base"
URL="http://www.cs.umd.edu/projects/plus/SHOE/base.html">

<!-- Here we indicate that this document is conformant with SHOE 1.0 -->

<META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">

<TITLE> **Our CNLPA Ontology-process** </TITLE>
</HEAD>
<BODY>

<!-- Here we declare the ontology's name and version -->

<ONTOLOGY ID=**cnlpa-ontology-process**" VERSION="1.0">

<!-- Here we declare that we're borrowing from another ontology -->

<USE-ONTOLOGY ID="base-ontology" VERSION="1.0" PREFIX="base"
URL="http://www.cs.umd.edu/projects/plus/SHOE/base.html">

<!-- Here we lay out our category hierarchy -->

<DEF-CATEGORY NAME="**Activity**" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="**Duration**" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="**Locating**" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="**Personship**" ISA="base.SHOEEntity">

<DEF-CATEGORY NAME="**Publishing**" ISA="Activity">
<DEF-CATEGORY NAME="**Teaching**" ISA="Activity">
<DEF-CATEGORY NAME="**Cooperating**" ISA="Activity">

<DEF-CATEGORY NAME="**Advising**" ISA="Activity">
<DEF-CATEGORY NAME="**Coaching**" ISA="Activity">
<DEF-CATEGORY NAME="**Administrating**" ISA="Activity">
<DEF-CATEGORY NAME="**Ownership**" ISA="Activity">
<DEF-CATEGORY NAME="**Membership**" ISA="Activity">

<DEF-CATEGORY NAME="**Person**" ISA="**Personship**">

<DEF-CATEGORY NAME="**Housing**" ISA="Location">

<DEF-CATEGORY NAME="**permanent**" ISA="Duration">
<DEF-CATEGORY NAME="**temporary**" ISA="Duration">
<DEF-CATEGORY NAME="**weekend**" ISA="Duration">
<DEF-CATEGORY NAME="**term**" ISA="Duration">

<DEF-CATEGORY NAME="**Cooperation**" ISA="Activity">

<DEF-CATEGORY NAME="**Zertificating**" ISA="Advising">

<DEF-CATEGORY NAME="**Publishing**" ISA="Person">
<DEF-CATEGORY NAME="**unpublishedPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**publishPublication**" ISA="**Publishing**">

<DEF-CATEGORY NAME="**printPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**audioPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**videoPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**webPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**researchPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**webPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**Books**" ISA="**printPublication**">
<DEF-CATEGORY NAME="**Article**" ISA="**printPublication**">
<DEF-CATEGORY NAME="**Video**" ISA="**videoPublication**">
<DEF-CATEGORY NAME="**Audio**" ISA="**audioPublication**">
<DEF-CATEGORY NAME="**Webpage**" ISA="**webPublication**">

<DEF-CATEGORY NAME="**Seminars**" ISA="**Teaching**">
<DEF-CATEGORY NAME="**Busines-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**Hypnosis-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**NLP-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**Family-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**In-House-Seminar**" ISA="**Seminar**">

<DEF-CATEGORY NAME="**Office**" ISA="Housing">
<DEF-CATEGORY NAME="**Office**" ISA="permanent">
<DEF-CATEGORY NAME="**Academy**" ISA="Housing">
<DEF-CATEGORY NAME="**Academy**" ISA="permanent">
<DEF-CATEGORY NAME="**Seminar-House**" ISA="Housing">
<DEF-CATEGORY NAME="**Seminar-House**" ISA="temporary">

<DEF-CATEGORY NAME="**Client**" ISA="**Coaching**">
<DEF-CATEGORY NAME="**Student**" ISA="**Teaching**">
<DEF-CATEGORY NAME="grad**Student**" ISA="**Zertificating**">
<DEF-CATEGORY NAME="**Trainer**" ISA="**Teaching**">
<DEF-CATEGORY NAME="**Office-Worker**" ISA="**Administrating**">

RULES:
<DEF-RELATION NAME="publicationAuthor">
<DEF-ARG POS="1" TYPE="Publishing">
<DEF-ARG POS="2" TYPE="Person">
</DEF-RELATION>

<DEF-RELATION NAME="advisor">
<DEF-ARG POS="1" TYPE="gradStudent">
<DEF-ARG POS="2" TYPE="Trainer">
</DEF-RELATION>

<DEF-RELATION NAME="member">
<DEF-ARG POS="1" TYPE="Membership">
<DEF-ARG POS="2" TYPE="Person">
</DEF-RELATION>

<DEF-RELATION NAME="age">
<DEF-ARG POS="1" TYPE="Person">
<DEF-ARG POS="2" TYPE=".NUMBER">
</DEF-RELATION>

<DEF-RELATION NAME="name">
<DEF-ARG POS="1" TYPE="base.SHOEEntity">
<DEF-ARG POS="2" TYPE=".STRING">
</DEF-RELATION>

</ONTOLOGY>
</BODY>
</HTML>

FACTS:
PERSON: {Klaus, Susanne, Irmi, Robert, Egbert, ...}
Publication-TITLES
Publication-Dates
Seminar-Titles
Seminar-Dates
Seminar-Locations
Name-of-Cooperations-Partners

etc.

3 CNLPA-ONTOLOGY-metapattern

<- The metapattern approach is a intermediary paradigm between the process and the contextural approach. It will be developed later.->

CNLPA:
ownership
personship
publishing
training
cauching
cooperating
administrating
advising

4 CNLPA-ONTOLOGY-polylogic

<HTML>
<HEAD>

<!-- Here we indicate that this document is conformant with SHOE 1.0 ; which surely is not the case at all!!-->

<META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">

<TITLE> **Our CNLPA Ontology-polylogic** </TITLE>
</HEAD>
<BODY>

<!-- Here we declare the ontology's name and version -->

<**POLY**-ONTOLOGY ID=**cnlpa-ontology-polylogic**" VERSION="1.0">

<!-- Here we declare that we're borrowing from another ontology -->

<USE-ONTOLOGY ID="**contexture00**_base-ontology" VERSION="1.0" PREFIX="base"
URL="http://www.cs.umd.edu/projects/plus/SHOE/base.html">

<!-- Here we declare that we're borrowing from another ontology useful for **personal data**-->

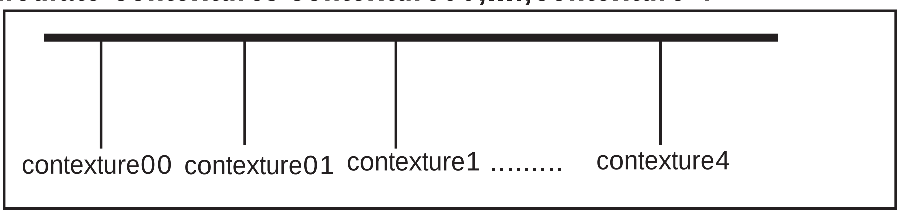
<USE-ONTOLOGY ID="**contexture01**_personal-ont" VERSION="1.0" PREFIX="personal"
URL="http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html#person">

<!-- Here we we lay out our **poly-contextural** ontology -->

<USE-ONTOLOGY ID="**contexture1_activity-ont**" ---> Fictional!!
<USE-ONTOLOGY ID="**contexture2_duration-ont**" ---> Fictional!!
<USE-ONTOLOGY ID="**contexture3_location-ont**" ---> Fictional!!
<USE-ONTOLOGY ID="**contexture4_personship-ont**" ---> Fictional!!

< From object to process modelling, the poly-contextural modelling introduces a new step which goes beyond the process model. The used ontologies are not further organized vertically, but horizontally, building a heterarchic organization. This can only be considered as an analogy to what has to be done and not as polycontextural modelling as such. ->.
< As a consequence the SHOE's base-ontology, contexture00, is no longer in the functionality as a root, but put in parallel together with other neighboring ontologies, called contextures. The terminology <**contexture1_base**.SHOEEntity> is therefore quite fictional. >

<Mediate-Contextures Contexture00,....,Contexture 4>



!!! <!-- Here we lay out our **poly-contextural**-category **HETERARHY** --> **!!!**

<DEF-**CONTEXTURE** NAME="**Activity**" ISA="**contexture1_base**.SHOEEntity">
<DEF-**CONTEXTURE** NAME="**Duration**" ISA="**contexture2_base**.SHOEEntity">
<DEF-**CONTEXTURE** NAME="**Locating**" ISA="**contexture3_base**.SHOEEntity">
<DEF-**CONTEXTURE** NAME="**Personship**" ISA="**contexture4_base**.SHOEEntity">

<!-- Here we lay out our **mono-contextural-**category **HIERARCHY** of each **contexture**-->

<DEF-CATEGORY NAME="**Publishing**" ISA="**contexture1_base**.SHOEEntity">
<DEF-CATEGORY NAME="**Teaching**" ISA="**contexture1_base**.SHOEEntity">
<DEF-CATEGORY NAME="**Cooperating**" ISA="**contexture1_base**.SHOEEntity">
<DEF-CATEGORY NAME="**Advising**" ISA="**contexture1_base**.SHOEEntity">
<DEF-CATEGORY NAME="**Coaching**" ISA="**contexture1_base**.SHOEEntity">
<DEF-CATEGORY NAME="**Administrating**" ISA="**contexture1_base**.SHOEEntity">
<DEF-CATEGORY NAME="**Ownership**" ISA="**contexture1_base**.SHOEEntity">
<DEF-CATEGORY NAME="**Membership**" ISA="**contexture1_base**.SHOEEntity">

<DEF-CATEGORY NAME="**Timing**" ISA="**contexture2_base**.SHOEEntity">

<DEF-CATEGORY NAME="**Housing**" ISA="**contexture3_base**.SHOEEntity">

<DEF-CATEGORY NAME="**Person**" ISA="**contexture4_base**.SHOEEntity">
<DEF-CATEGORY NAME="**male**" ISA="Person">
<DEF-CATEGORY NAME="**female**" ISA="Person">

<DEF-CATEGORY NAME="**permanent**" ISA="Timing">
<DEF-CATEGORY NAME="**temporary**" ISA="Timing">
<DEF-CATEGORY NAME="**weekend**" ISA="Timing">
<DEF-CATEGORY NAME="**term**" ISA="Timing">

< Chair belongs at once to 3 different contoxtures and their ontologies!!
<DEF-CATEGORY NAME="**CHAIR**" ISA="**Ownership**">
<DEF-CATEGORY NAME="**CHAIR**" ISA="**Person**">
<DEF-CATEGORY NAME="**CHAIR**" ISA="**Administrating**">

<DEF-CATEGORY NAME="**Zertificating**" ISA="Advising">
<DEF-CATEGORY NAME="**Practionar**" ISA="Zertificating">
<DEF-CATEGORY NAME="**Master**" ISA="Zertificating">
<DEF-CATEGORY NAME="**Trainer**" ISA="Zertificating">
<DEF-CATEGORY NAME="**MasterTrainer**" ISA="Zertificating">

<DEF-CATEGORY NAME="**Guest-Trainer**" ISA="Teaching">

<DEF-CATEGORY NAME="**Publishing**" ISA="Person">
<DEF-CATEGORY NAME="**unpublishedPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**publishedPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**Advertising**" ISA="**Publishing**">

<DEF-CATEGORY NAME="**printPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**audioPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**videoPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**webPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**researchPublication**" ISA="**Publishing**">
<DEF-CATEGORY NAME="**webPublication**" ISA="**Publishing**">

<DEF-CATEGORY NAME="**Books**" ISA="**printPublication**">
<DEF-CATEGORY NAME="**Article**" ISA="**printPublication**">
<DEF-CATEGORY NAME="**Video**" ISA="**videoPublication**">
<DEF-CATEGORY NAME="**Audio**" ISA="**audioPublication**">
<DEF-CATEGORY NAME="**CNLPA-Web Site**" ISA="**webPublication**">
<DEF-CATEGORY NAME="**NLP Wissen**" ISA="**webPublication**">
<DEF-CATEGORY NAME="**Innernet**" ISA="**webPublication**">
<DEF-CATEGORY NAME="**Techno.net**" ISA="**webPublication**">

<DEF-CATEGORY NAME="**Seminars**" ISA="**Teaching**">
<DEF-CATEGORY NAME="**Busines-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**Managment-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**Hypnosis-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**NLP-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**Family-Seminar**" ISA="**Seminar**">
<DEF-CATEGORY NAME="**In-House-Seminar**" ISA="**Seminar**">

<DEF-CATEGORY NAME="**Office**" ISA="Housing">
<DEF-CATEGORY NAME="**Academy**" ISA="Housing">
<DEF-CATEGORY NAME="**Seminar-House**" ISA="Housing">

<DEF-CATEGORY NAME="**Furniture**" ISA="Housing">
<DEF-CATEGORY NAME="**Chair**" ISA="Furniture">
<DEF-CATEGORY NAME="**Table**" ISA="Furniture">

<DEF-CATEGORY NAME="**Client**" ISA="**Coaching**">
<DEF-CATEGORY NAME="**Student**" ISA="**Teaching**">
<DEF-CATEGORY NAME="**gradStudent**" ISA="**Zertificating**">
<DEF-CATEGORY NAME="**Trainer**" ISA="**Teaching**">
<DEF-CATEGORY NAME="**Office-Worker**" ISA="**Administrating**">
<DEF-CATEGORY NAME="**Cleaner**" ISA="**Administrating**">

<DEF-RENAME >
< specifies a local name for a concept from any extended ontology. >
<DEF-RENAME > CHAIR to Seat
<DEF-RENAME > CHAIR to AcademyHead
<DEF-RENAME > CHAIR to CoffeBar

RELATIONS:

< NOTE: In SHOE it is not possible to specify subsuming categories for a category defined in another ontology.

In contrast to the mono-contextural situation, the poly-ontological apprach has to mediate between different ontologies.

Therefore we can define some conditions in one contexture containing its own logic and also in another contexture containing its own other logic.>

< Intra-contextural relations (based on FOL) >
<DEF-RELATION NAME="publicationAuthor">
<DEF-ARG POS="1" TYPE="Publishing">
<DEF-ARG POS="2" TYPE="Person">
</DEF-RELATION>

<trans-contextural relations (based on poly-contextural logics)>
<This rule has to be transformed into a poly-contextural relation>

<DEF-RELATION NAME="publicationAuthor"> **between Contexture4 and Contexture1**
<DEF-ARG POS="1" TYPE="Publishing"> **of Contexture1**
<DEF-ARG POS="2" TYPE="Person"> **of Contexture4**
</DEF-RELATION>

<DEF-RELATION NAME="age">
<DEF-ARG POS="1" TYPE="Person"> of Contexture4
<DEF-ARG POS="2" TYPE=".NUMBER"> of Contexture0
</DEF-RELATION>

<DEF-RELATION NAME="name">
<DEF-ARG POS="1" TYPE="base.SHOEEntity"> of Contexture0
<DEF-ARG POS="2" TYPE=".STRING"> of Contexture0
</DEF-RELATION>

<DEF-RELATION NAME="tenured">
<DEF-ARG POS="1" TYPE="Trainer">
<DEF-ARG POS="2" TYPE=".TRUTH">
</DEF-RELATION>

<DEF-RELATION NAME="object-age">
<DEF-ARG POS="1" TYPE="Furniture"> of Contexture3
<DEF-ARG POS="2" TYPE=".NUMBER"> of Contexture0
</DEF-RELATION>

</ONTOLOGY>
</BODY>
</HTML>

FACTS in Contextures

person(Grochowiak)

male, female
permanent, temporary
weekend, term

Inference-RULES:

< -Inference rules have to be distributed over different contextures. There is not a single dominating inferenz rule ruling all contextures in the same sense. Distributed inferenz rules can even be differently defined, depending on the intra-contextural structure of the distributed ontology. ->

< - *Mono-contextural case:*>
<DEF-INFERENCE DESCRIPTION="Transitivity of Suborganizations">
<INF-IF>
<RELATION NAME="subOrganization">
<ARG POS="FROM" VALUE="x" USAGE="VAR">
<ARG POS="TO" VALUE="y" USAGE="VAR">
</RELATION>
and
<RELATION NAME="subOrganization">
<ARG POS="FROM" VALUE="y" USAGE="VAR">
<ARG POS="TO" VALUE="z" USAGE="VAR">
</RELATION>
</INF-IF>

< - *Poly-contextural case as an example* ->
poly-<DEF-INFERENCE DESCRIPTION="Transitivity of -sub-----">
poly-<INF-IF>
poly-<RELATION NAME=" sub-------">
Contexture1: <ARG POS="FROM" VALUE="x" USAGE="VAR">
Contexture1 and contexture2: <ARG POS="TO" VALUE="y" USAGE="VAR">
</RELATION>
et.et.et
poly-<RELATION NAME=" sub-----">
Contexture2: <ARG POS="FROM" VALUE="y" USAGE="VAR">
Contexture3: <ARG POS="TO" VALUE="z" USAGE="VAR">
</RELATION>
</INF-IF>

<- But without contradiction it could also be an intransitivity relation- >
<RELATION NAME=" sub-----">
Contexture2: <ARG POS="FROM" VALUE="z" USAGE="VAR">
Contexture3: <ARG POS="TO" VALUE="y" USAGE="VAR">
</RELATION>

QUERIES

Query-types
"mono-contextural search"
polycontextural parallel search

polycontextural multiple parallel search

Cluster questions and question clusters