1

Complex-Adaptive Paradigm

Bootstrap and contextual stigmergy as control for novelty

Mixel Kiemen

MOSI/ECCO Vrije Universiteit Brussel,
Krijgskundestraat 33, B-1160 Brussels Belgium
info@mixel.be

Summary. Complex-Adaptive Paradigm (CAP) will be introduced as a way to understand how novice behavior is the control system for novelty. The paradigm is based on evolution, complex systems and cognition as well as programming modeling and research on innovation. The two main concepts of the paper are bootstrap and contextual stigmergy. The bootstrap is a concept used in programming modeling and can be a technique for novelty. The system we propose will bootstrap syntax and thereby determine its behavior. Based on the original concept of stigmergy[3], we shall introduce a recursive version. The contextual stigmergy is inspired on how a programming language parses and evaluates. We claim to need four contextual stigmergy to create the model: external, internal, directional and adaptational. When the model is processing it behaves constructive, based on two types of learning: behavior mastery and reflection. The two types of learning in a special setting would allow the system to be open for social interaction. Two models, created by authorities in the field, will be examined as a possible control system for novelty. On the individual level this will be the workspace (area in he brain). On the social level we continue the investigation and see how the two types of learning relate to social construtuvism and claim that novelty becomes innovation.

1.1 Programming paradigms and cognition

For those who like to understand control, programming modeling is a wonderful tool. You can create such different programming paradigms requiring such different mind sets to work on. Of course decent programming languages are turning complete, so it seams there is no difference in what language you use. This is however wrong, all paradigms make it easer to understand and create specific structures. [1] For example it is relatively easy to make a logic program where we can create semi-natural language for some simple operations. This

[1] Another and more common reason would be performance, this is however not an issue for the current paper.

has been used by classic AI research. Likewise it is easy to make software architectures using object oriented language. In other words: the paradigm defines the expressivity of a programming language. Related to "expressivity", lets have a look at Simon's quote about computers and thoughts:

> As we succeed in broadening and deepening our knowledge - theoretical and empirical- about computers, we shall discover that in general laws, that what appeared as complexity in the computer program was, to a considerable extent, complexity of the environment to which the program was seeking to adapt its behavior [26, p 22]

Simon talks about computers, but notice a programming language makes controlling computers more easily, so he is exactly pointing out the expressivity of a paradigm. We embed the complexity we try to adapt into a language (logical, object oriented, etc). Interesting is that we use natural language (and scientific paradigms) to understand and express the world in a similar fashion.

As always in science we need the right tool to understand the right environment. We need a programming paradigm dealing with evolution, complexity and cognition. The gap between cognition and programming has a grounding problem: "How do we go from natural to artificial?" We suggest to build a CAP language based on CAS (complex adaptive systems) [8]. The research goal is to understand how a simple behavior can be simply implemented. For example "cleaning a room" has always been seen as a simple behavior, but implementing it in an artificial way never was. We try to create a paradigm where the complex of such an activity is "under the hood" of the language.

1.1.1 Grounding problems

The Chinese room experiment by Searle argues about the intelligence by computer program. In the thought experiment the processor (computer) is replaced by a human and the problem arises that we would not say the human knows chinese (only some arbitrary rules). We agree, in most of the expert systems the "understanding" is not embedded in the program but in the programmer. In contrast to expert systems we investigate novice behavior. To do so we need a good starting point, so we start from where the critic was defined. Let us look at two quotes from the famous paper [23]:

> Strong AI: The computer is not merely a tool in the study of the mind; rather, the appropriately programmed computer really is a mind.
>
> The Brain Simulation Reply: ... I want to digress to note that it is an odd reply for any partisan of artificial intelligence (or functionalism, etc.) to make: I thought the whole idea of strong AI is that we don't need to know how the brain works to know how the mind works
> .

The comment has an historical context, conceptual the strong AI does not need to be related with functionalism. We claim to research strong AI and understanding the brain is essential (but not enough).[2] The research is based on CAS what is seen as connectional model. Splitting the debate in a functional versus connectional paradigm may have relevance for a philosophical debate, but it has been contra productive in understanding "intelligence" from an engineering point of view.[3] While Searle has been seen as a resistor to A.I. we acknowledge his influence for our own approach. We see two other books of him as relevant for our paper. One showing the complex face of rationality [25]. The other is related to social construtuvism and its influence on artifacts [24], a concept we look at in the end of the paper.

The old artificial intelligence (Searle is referring to in the quote) was in its attempt to understanding intelligence naive and took a wrong starting position (expert systems instead of evolutionary). It seams that today many people look at the connections and claim emerging should solve the problem of intelligence. How to engineering "emerging" onto "rationality" seems to be treated equally blind. We can use CAS and autopoietic systems [17, 18] and other concept of the complexity field as basis to conduct research on evolutionary epistemology [4] or evolutionary cybernetics[7]:

> Evolutionary cybernetics can be defined as the study of how the processes of variation and selection give rise to organization. This means, first of all, a study of the dynamics of distinctions, connections, variety, closure and constraint, that is, the fundamental aspects of organized complexity. This will allow us to better understand how systems emerge out of unstructured aggregates of components, and how variation and selection take place at different system levels and between different, co-evolving systems.

The theoretic research on evolutionary cybernetics has not yet moved to engineering applications. A still theoretical but closer to engineering proposal is to see the mind as a virtual machine [27, 28]. We have a similar claim, to be able to adapt to any kind of circumstances a living system [19] would create a general processing machine.[4] Our attempt is to define a paradigm and create

[2] To clarify think about aviation. Understand how birds fly can give us some information, but we need a epistemology of aerodynamics etc. before we can build aviation technology. This will not be different in the our research on novelty. In the last section "innovation" we try to bring clarity on how the relation between science and technology actually create the reality.

[3] Contra productive as it has become a taboo for engineers to research intelligence as a whole, engineers can only research specific parts or so called lower-level cognition. Leading to a fundamental problem for complexity research or higer-level cognition research in A.I.

[4] Notice the difference. We search for a general processing machine, not a Virtual Machine (VM). A VM is of course a general processing machine but the VM claim is more specific and we do not see the a grounded reason for the added feature.

a programming language. We claim that most tools needed to create such a language does already exist: making machine languages has been done before. At least there is enough to start the research. The question is how to make a paradigm for the specific case. We need to construct programming language where the execution of a program behaves constructively.

1.1.2 Complex-adaptive programming

Higher level programming languages are complex systems defined by a syntax and explained by its semantics. The research on memory optimization, computational reflection[16], meta-objects [9] etc. are adaptive systems as well. However we do not see a relation to the complex adaptive paradigm, merely recognize structure that deal with the same phenomena. The question is: "Can we use the structure to push the research on CAP?" What we try to do is use an solution created in a different context (programming modeling) and apply it to our problem on cognition. The technique we propose is a standard (but implicit) technique in innovation. To elaborate the technique, the narrative in Bruno Latour's "Science in action" [12] seams appropriate. The introduction starts with a biochemist researcher who is looking at a 3D-helix structure of DNA. Next the story tracks back two research paths: the research on DNA and the research on computers. Notice that several years before the specific situation the link between organic chemistry and electronic engineering was not present at all. Similar, the relation between intelligence research and programming modeling becomes only clear with the state-of-the-art research in the related fields. Interesting to note is that the programming field did use research on intelligence for there advanced techniques, like the advanced object-oriented "prototyping" [14, 22]. From the A.I to modelling we the agent simulations based on object-oriented programming.The reason why advanced modeling techniques and state-of-the-art cognition may not be so easily combined is the learning cure we need to pay in both fields.

1.2 The model

To start our research on the CAP we have build simulations to get a basic understanding [10]. Prototyping was one of techniques applied, in a next implementation we like to integrate the actor design [1, 2]. However, we first needed to concretizes the theoretical frame, as such the basic implementation was abstracted and linked with cognitive research [11]. With the current paper we build the theory for our paradigm an leave the more fundamental problems in cognition (awareness etc) untouched. We start by look at context in an hierarchical way, where there will be an evaluator for every type of context. Note, in the complex system theory hierarchy means: composed of interrelated subsystems [26, p 87]. In any of this subsystems (contexts) a same kind of evaluation is seen. If we describe the behavior of an evaluator in general it is acting on a specific mark/token (syntax) that is associated with a particular circumstance (semantics), or shorter its acting on stigma.

There exists the concept "stigmergy" that explains us how acting on stigma is achieved in living systems. While previous work on stigmergy has shown how simple instructions can manage a complex body (like a hive) [3], we have a different situation for the evaluator. In case of the evaluator the marks are recursive: it can contains many new nested marks. Still the operation/evaluation of every mark is a simple instruction. We claim the behavior of the evaluator can be seen as the advanced version of stigmergy. It is also evolutionary plausible, just look at the shift form genetical defined stigmas (termites) to a constructive behavior [21] (humans). In case of constructive behavior the stigmergy would be related to previous experience. In this paper the relation between such kind of stigma and our suggested syntax will get explained.

1.2.1 Bootstrapping

We distinguish a constructive behavior and a contextual stigmergy. The constructive behavior needs experience before it can create syntax, but the contextual stigmergy needs syntax before it operates. We have to breakdown this circularity. The process we see involved is a metasystem transition[30], but that does not explain the interplay that leads to syntax. To clarify how we can create something based on the creation itself we look at the concept of bootstrapping as it is used for a specific kind of compiler. We see a relation between bootstrapping and the narrative on innovation used earlier. In case of a bootstrap the transformation is self-inflecting, let us clarify.

We can create a compiler for a programming language without ever needing to write one line of code in the underlying language. The trick is to make a compiler from a higher level language to a lowerlevel language while only staying on the easy higher level language. This is how it goes: create a compiler H $\rightarrow$ L in H, give the compiler itself as input and you will get the same compiler H $\rightarrow$ L in L. We can use this technique to create a better compiler or bootstrap the system from an other language. For example a Scheme program my be working as an interpreter in C++ , creating a "scheme $\rightarrow$ assembler" compiler in Scheme allows you to bootstrap a "scheme $\rightarrow$ assembler" compiler in Assembler and skip the whole C++ part. Notice first you had a compiler, but it was a toy, trivial to the execution process as you still need a working Scheme environment to make it work. By giving the compiler itself as input you shift the context from Scheme to Assembler now the first trivial (toy) compiler became a powerful tool.

In the compiler example contexts are programming languages (Scheme, C++, Assembler). The processing results in a shift from a trivial to a powerful concept. The compiler is processing itself, so the compiler is bootstrapping. The compiler is also constructing itself (cloning), thereby it gives us a notion of how something can come to existence that looks like it came out of the void. Notice that after the process we still have the trivial compiler next to the powerful tool. Also after compiling the process ends, this is not the case with the constructive behavior where we see a more recursive version of bootstrapping. The compiling is resulting in skipping of the underlying language (C++). From a living system point of view we expect the processing to be evolution and the skipping reducing friction.

1.2.2 Primacy of context

Before we can have context we need a separation between inside and outside. The separation gets bootstrapped out of a system that is already complex in relation to what will be separated. For example, before a cell there where complex bio-chemical structures. Before the separation between sensory and motor-control there existed a complex neural network. Notice it is easer to get similar systems once we can have one system capable of cloning a variant of itself. We expect the origin of our model should be explainable by this strategy, so lets get on by explaining the model itself.

Several context stigmergies working as a distributed system. Important is that the experience is stored as "one whole" while the evaluation work on it in there specific context. The feedback, shared via one working memory, has a intriguing side effect. While one evaluator is working in a specific context it is at the same time stimulating triggers for other contextual stigmergies. The feedback is related with one parsing step of the syntax tree. In the step you parse a token and it leads to a more specific expression, this one step is pushed out as feedback. The marks and the way to parse them in CAP are in direct relation to the context of evaluation and the parsing of the stigma is a complex behavior.

As an illustration we look at visual perception, thanks to the many research on it we can see the complex that is in control. We see the processing of vision as part of the external-context evaluation. From more primitive live forms, like fish, it is know that they only have a vague perception, we claim that this is exactly the starting point for human vision as well. It is our sacading eye that will check the boundaries to recognize shape, the whole process is hard coded (like reflex) and goes very fast. Notice how this process goes from evaluating one expression to the other. First you have a general case (light/dark) next you parse the first mark (iterating the boundary between light and dark), leading to shapes.

The internal-context is working on internal stigmas we know the behavior as an association game: the first concept that pops your mind when another is given. For example a shape round may trigger concepts like: ball, balloon, face, etc. As the concept are holistic they contain information of all contexts. For example a face has hierarchy of nose, mound, eyes, etc. All this parts have specific external marks we can parse. So while the concept round may get evaluated by external-context the relation with face is reached by internal-context and the face-marks can be checked by external-context evaluation again. It is the interplay that leads to complex results while the evaluations themselve are straight forward. Just like the parser in a programming language the token need to be parsed to figure out what exact context we are dealing with so the overall behavior is context depended.

The marks and the way of evaluation are very different for each context, while the behavior of the control is similar. External evaluation parses stimuli marks, internal evaluation parses association marks. To get to an intelligent behavior we shall add a directional and adaptational evaluations. The directional works on motivation marks while the adaptational works on change marks. Let us clarify. Take a feeling of hunger, it is first a stimuli measurement (produced by the stomach etc.), it need to be parsed by the external-evaluation to the concept "feeling hunger". Once the concept is triggered the directional-evaluation will react to the stimuli and associations, related to hunger, will be the focus. So the external and internal evaluation will get directed. We can see the directional as a filter on the working memory while the internal and external feed the working memory. The adaptation-context is also adding its specific influence, it will be an observer of the working memory and match the work-flow with pervious experience. When there would be a big difference the adaptational evaluation could influence or even change the directional evaluation. One of your motivations can be to learn, but is some situations, you prefer to neglect it; it is the directional stigmergy that will evaluate the influence of change (related to the motivational marks).

1.2.3 The model in motion

In previous work [11], examples on the learning behavior where given. A summery on how the behavior leads to learning seams appropriate. When the directional evaluation focus on the internal (and adaptational) evaluation we get behavior mastery. Focusing on the external (and adaptational) evaluation results in reflection. For the behavior mastery a virtual-concept gives the direction and physical properties are learned. Examples are learning related to motion control (swimming, biking, painting, etc). With reflection physical properties are abstracted to a virtual concept. Virtual concepts can also be transformed to more abstract concepts. We know reflection as the more classic educational type of learning. When we recursively go from one type of learning to the other we bootstrap concepts. The order of learning is not an issue, but the interaction is important. For example you get information about biking: see people biking (virtual), next you learn to do it (physical). In the other direction by lifting several object you can relate the (physical) object to a (virtual) concept weight. Before the interaction there was no physical or virtual concept and it is the concept itself that will focus the learning so it is a bootstrap.

The step of change are always very small, we will not see the bike example as innovation, but the force behind it may well be the same. Some changes are known to go very slow they pass without us noticing and sometimes we talk about revolution. We know revolution in our human history like the agricultural or the industrial. Still the agricultural changes may be for the individual a stable situation for his whole live. Similar the industrial change is trivial when we look at only one year. The bike example has likewise no tremendous change for the daily activities of a child, but think about how the child change over the years.

We still have a problem to define novelty, when you look before and after a period we do see a phenomena that did not just came out of the void. Separation gets bootstrap out of a complex. We see the similar behavior with the constructive behavior. There need to be a complex in relation to the novelty. What makes us experience it as novel is the shifting of one complex in an other context leading to an escalation of change and the escalation of new concepts can lead to spillover effects.

With the learning outlined we should get back to the relation between stigmas and syntax. We refer to syntax as the result of constructive behavior. By acting experience is create, experience distillates the stigmas as the aspects that can mark a experience. For stable concepts it may be something that change for dynamic concepts it may be something that is stable, it all depends on the specific experience and motivation. Finally by learning the causalities between the marks there is a transformation to a concept. So because of the nature of the learning concepts are related to concepts, some links are strong, some weak.

Explicit knowledge

Untill now we only described one module, the nature of the system forces us to look at the interaction between models and even a meta-system. It is easer to see the model as an agent. We have described before that the information running trough the system should be sean as a holistic concept. To get to explicit knowledge we should see the behavior mastery and reflection in a specific setup of interaction between agents in a semiotic cycle. The perceiver (for spoken interaction the hearer) is undergoing the process and may get new relations out of the process (behavior mastery). The producer (for spoken interaction the speaker) is using his sets of concept to clarify his point (reflection). The semiotic cycle is an interesting case where concepts get related to words and becomes a medium between individuals: the concepts get a new kind of feedback, a social kind. Notice words will again have several contextual-syntax. For the external context it will be the way to produce and process sound (or symbols when its written). The internal context is of course the relation between the word and the concept but there are also relations between similar sounds, similar words, etc. In every respect it is always a complex. On the social level there have been A.I. experiment on agent interacting via language to create common lexicon [29]. It is a challenge to make our model inline with the experiments. On the development the integration of the experiment is an important aspect to shift to the social level. Some caution has to be taken, we can not see the social emerging as a blind process. Explicit knowledge is not created by one level, the social dynamics should not be taken so trivial. We need to investigate how a social implementation of the model exists. It may be more clear if we first go and analyze how we see the theoretical model fit a real situation. We will come back to the "motion of the social model" after outlining the real situations.

1.3 Novelty as a phenomena

We claim to be creating an epistemology to deal with novelty by learning the most generic aspects of programming or: understand how artificial fits the natural. To do so we look how the phenomena (novelty) is controlled in our environment. There are two levels creating novelty, both have there specific influence. At the individual level novelty would be an aspect of intelligence and (more or less) embodied in the brain. At the social level we see the phenomena of innovation, in this case a whole other dynamic is present as will be seen. We shall speculate on two models, created by authorities in the domain. The question if we can falsify the speculation is future research.

1.3.1 Neural workspace

When we ask the question: "What is intelligence and what forces are in control?" We first of all should know science has no clear understanding of it yet. Today we know the control is produced by several systems. There are the hormones that react on behavior like dopamine that is direct related to our reward system or the better know adrenaline to give us an extra boost. On the neural level we have also a dispersion of several systems that add there capacity, like the olfactory as part of the stimuli processing or the amygdala as a primitive processing sector. While this biological level may already be complicated enough we didn't even look at the concept of nurture. So let us focus on the question we try to solve: "Where is the novelty controlled?".

We need to define what novelty is and tern to novelty in knowledge. Other types of novelty can be biochemical like our adaptation of our immunity system when we get infected. The body can change as a whole if we change our behavior (do sport, take a diet etc). Our interest goes to the control system dealing with dynamic information, like memory etc. The region known as "workspace" has as much a theoretical as empirical research on it and several parts of the the cortex are involved. In a recent hypothesis of the global workspace (fig.1) we see a remarkable similarity with our theoretical model. Only one extra entry is recognized, the motor system. The other entries could be related as follows: internal → evaluative; external → perceptual; directional → attentional; adaptive → LTM. We expect the modules to be bootstrapped out a more primitive version. While other parts may get absolute, we can understand the need for an entry to deal with motion in the case of living systems.

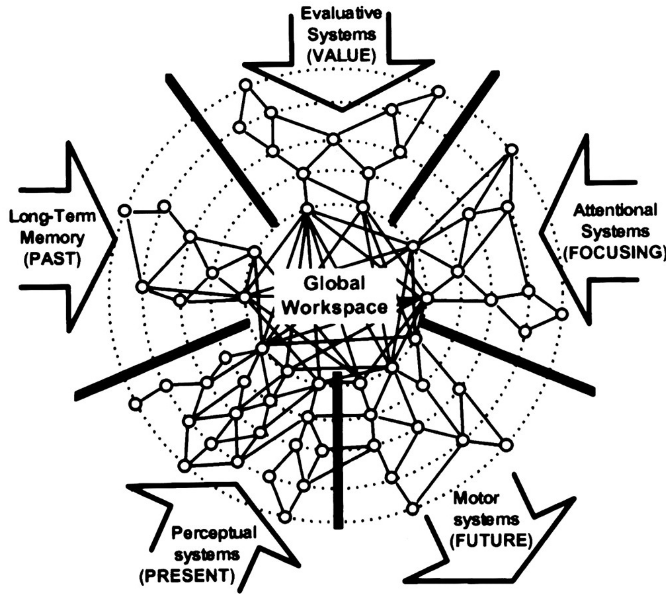

Fig. 1.1. Global workspace [5]

The diagram talks about entries not evaluators, this has to be adressed. First we used the term evaluations as we used the way programming modelers are creating programming languages. However the interpretation of the underlying system may be very different and we expect it to be so in the case of a neural network. In case of neurons the most part of the evaluation may be realized by simultaneous firing of neurons and this can be translated to evaluations to make it more controllable in our epistemology.

1.3.2 Science and technology

To investigate the social lever for novelty we need to look at science and technology. Bruno Latour as an anthropologist on science has came up with a model (fig.2) to understand how science leads to innovation and adaptation of the social context. The details used to construct the model are very depending on the social structure of the west. If we look at the historical analyses of Mokyr about innovation in Chinese at 1400 [20], we understand how the bureaucracy can be the medium for transition. It should be clear the two social context are radical different, however the model, being general enough, should hold for both social contexts. It should not be surprising as the two social context are exactly outlined because of there strong innovation achievement while the model is created to understand how innovation can emerge.

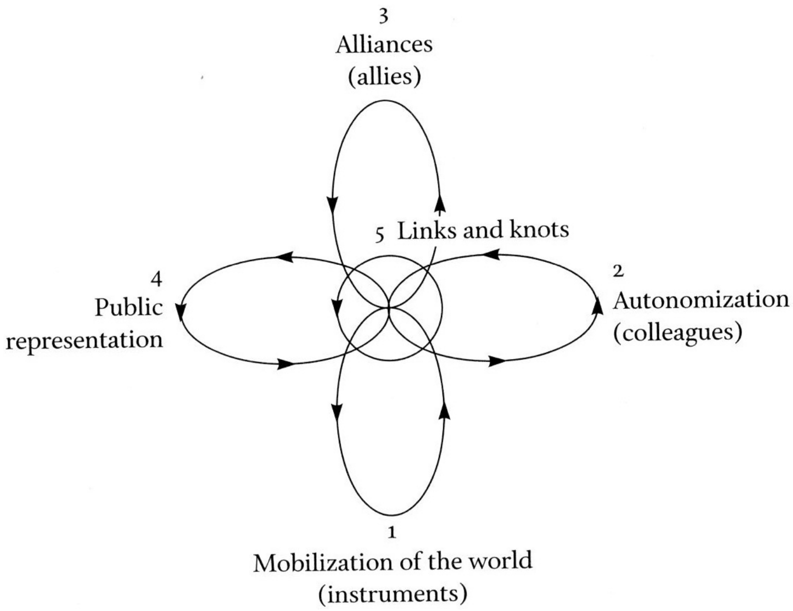

Fig. 1.2. Innovation diagram [13, p 100]

Latour's model (fig.2) contains five loops. When we relate it with our model there still would be four context stigmergies as the "links and knots" is the place for feedback (the working memory). What is called the "Mobilization of the world" is to bring the world to workable environment for the research (tools to measure, frames to apply, etc.), this is the internal-context. The "Alliances" is what provides the system to the required goods (funding, data, raw material, etc.) to produce the research and therefore it is the directional-context. Example could be funding institutes, political powers, venture capitalists etc. The "Autonomization" are the colleagues (peers) who do similar research and use frames to automate and communicate about the research, it is the adaptational-context. The public is the external-context, the institutes (directional) and the peers (adaptational) will get influenced by the public. The internal gets influenced in the same way as on the individual level. The relation between external and internal may be more clear if we look at the model in motion.

Innovation

With the social context we need social constructivism[15] to understand the full cycle of the system. As such we need to find the social version of reflection and behavior mastery. When we look at a quote by Mokyr[20, p 167] on how Gille sees the distinction between science versus technology, we may have found the two types of learning:

> Gille[6, p 1112], who has examined the nexus closely, suggests a distinction based on purpose: science aims at comprehension, whereas technology aim at utilization.

The interplay between science and technology for innovation is all but trivial. We can (as in everything) have innovation in science and innovation in technology, this makes the understanding of the interplay between science and technology for innovation confusing. To be explicit we see the behavior mastery as technology and the science as reflection. The interplay will bootstrap knowledge. As well in Mokyr as in Latours work we always get the fussy origin of a particular knowledge and it seams clear that existing knowledge can be as well a drive as a burden on change. This is way we should look at the behavior as novice behavior, we need to be prepared, but at the same time be open to reject the established knowledge on a subject. While our science (reflection) may come up with interesting theoretical models they will only lead to adaptation of our social context if there is technology created with it. For example in aviation, as long as it was a theory it was open for debate, by building aircrafts (that work) the theory adapt and gets solid. As such the behavior mastery and reflection on a social level do have the same influence as on the individual level, they bootstrap there own reality.

1.4 Conclusion

We have been building a theoretical model as a control system for novelty. The idea is to capture the complexity in the programming language: create a complex-adaptive paradigm. We introduced distributed contextual stigmergy, four in total: external, internal, directional and adaptational. Every context-evaluation deals with a specific set of stigmas and act upon it. The contextual sigmergy is only evaluating existing syntax and enables adaptation in that specific context. The constructive behavior will create the syntax by bootstrapping concepts in its learning process. We could separate learning into two extremes: behavior mastery and reflection. Behavior mastery take up features form its external environment and reflection will build relations between the existing knowledge.

In the process to clarify how the syntax can be created out of actions and stigma we ended up with an explanation how natural language could be part of the syntax. With this jump we go from individual interactions with the world to social interaction with other individuals. Both levels (individual and social) have there influence to build novelty and should not be separated. When we go to practical model we see true complexity. For the individual level we examine the entry to the workspace. The evaluation are in the case of neurons totally different implemented, but the behavior may well be similar to the evaluations.

There are several question that need to get response, like the over all presence of abstracting and its empowerment feature. The bootstrap is exactly a feature we are capable of creating because of the abstraction and notice specializing from a pure (information) modeling point of view is abstracting. Even in the IT field many see abstracting as a luxury and toy only relevant for academic people. It may well be impossible to convince a peer-comity in cognition the relevance to study advanced modelling techniques. A solution may come out of a different angle. While we are investigating novelty the research itself is novel as well. As such we do not only try to clarify but also try to construct. The section on innovation may give use a clue on how to make the theory solid. We need to build technology for our current theoretical model and continue this feedback cycle until the epistemology is grounded or rejected. In the last case we expect to adapt.

References

1. G. Agha. *Actors: A Model of Concurrent Computation in Distributed Systems.* MIT Press Cambridge, MA, 1986.
2. G. Agha and C. Hewitt. Research directions in object-oriented programming. *Actor: Conceptual Foundation fo Concurrent Object-Oriented Programming, in P Wegner and B Shiver eds.*, pages 49–74, 1987.
3. R. Beckers, O.E. Holland, and J.L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. *Proc. A-Life*, 4, 1994.
4. D.T. Campbell. Evolutionary epistemology. *The Philosophy of Karl Popper*, (3):413, 1990.
5. S. Dehaene, M. Kerszberg, and JP. Changeux. A neuronal model of a global workspace in effortful cognitive tasks. *Neurobiology*, 95(24):14529–14534, 1998.
6. Bertrand Gille. *Histories des Technology and Innovation.* Editions Gallimard, Paris, 1978.
7. Francis Heylighen. http://pespmc1.vub.ac.be/EVOLCYB.html, 2000.
8. John H. Holland. *Hidden order: how adaptation builds complexity.* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1995.
9. Gregor Kiczales, Jim des Rivieres, and Daniel Gureasko Bobrow. *The art of the metaobject protocol.* MIT Press, Cambridge, Massachusetts, 1991.
10. Mixel Kiemen. Het appropriatiegedrag. Master's thesis, Vrije Universiteit Brussel, 2003.
11. Mixel Kiemen. A triple loop model to ground higher-level cognition. In *In Proceedings of the 18th European Meeting on Cybernetics and systems Research*, volume 2, pages 369–374, 2006.
12. Bruno Latour. *Science in action: How to follow scientists and engineers through society.* Harvard University Press, Cambridge, MA, 1987.
13. Bruno Latour. *Pandora's hope.* Harvard University Press, Cambridge, MA, 1999.
14. H. Lieberman. Using prototypical objects to implement shared behavior in object oriented systems. In *Proc. OOPSLA-86*, pages 214–223, 1986.
15. Thomas Luckmann and Peter L. Berger. *The Social Construction of Reality. A Treatise in the Sociology of Knowledge.* Garden City, New York, Doubleday, 1966.

16. Pattie Maes. Concepts and experiments in computational reflection. In *OOPSLA '87: Conference proceedings on Object-oriented programming systems, languages and applications*, pages 147–155, New York, NY, USA, 1987. ACM Press.
17. Humberto R. Maturana and Francisco J. Varela. *Autopoiesis and Cognition: the realization of linving*. Dordrecht, Reidel, 1980.
18. Humberto R. Maturana and Francisco J. Varela. *The tree of knowledge*. Shambala, Boston, 1984.
19. James Grier Miller. Living systems. *Actor: Conceptual Foundation fo Concurrent Object-Oriented Programming, in P Wegner and B Shiver eds.*, 1978.
20. Joel Mokyr. *The level of riches*. Oxford university press, New York, Oxford, 1990.
21. Jean Piaget, editor. *Structuralism*. Harper & Row, New York, 1970.
22. E Rosch. *Cognition and Categorization*. Lawerence ERlbaul Hillsdale NJ, 1987.
23. John Searle. Minds, brains and programs. *Behavioral and Brain Sciences*, 3(3):417–457, 1980.
24. John Searle. *The construction of social reality*. Penguin, London, 1996.
25. John Searle. *Rationality in action*. Harvard Universsity Press, Cambridge, 2001.
26. Herbert A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, Massachusetts, first edition, 1969.
27. Aaron Sloman. Semantics in an intelligent control system. *Philosophical Transactions of the Royal Society Series A: Physical Sciences and Engineering*, 349(1689):3–57, 1994.
28. Aaron Sloman and Ron Chrisley. Virtual machines and consciousness. *Journal of Consciousness Studies*, 10(4/5):113–172, 2003.
29. Luc Steels. *The Talking Heads Experiment*. pre-edition, LABORATORIUM Antwerp, 1999.
30. V. Turchin. *The Phenomenon of Science*. Colombia university press, New York, 1977.