Colline - A learning network (intermediate draft)

Johan Tino Frederiksen

10. oktober 2001

Abstract

On the following pages a network of interacting autonomous agents are described. The hypothese behind the design is that the agents will start to specialize into species and show more 'complex behaviour'. The complex behaviour gives rise to persistent structures, that emerge to solve the overall processing task given by an external trainer. During the analysis analogies will be made to the system theory for social systems by Niklas Luhmann, which intends to show the (hypothetical) close relationship between a social system and the agent network.

Chapter 1 Introduction

In this document the analysis and design of a system is described. The purpose of the system is to solve a processing task in the category of pattern recognition.

The design of the system are inspired by the growing interest in how autonomus agents interact, agents being 'entities' ranging from ants to countries, and the rather new concept of autopoiese, which means self-reproduction. Many scientists have in the recent years started to look into these phenomena where e.g. complex behaviour emerge from simple interacting agents.

The system consists of autonomous agents living in a two dimensional matrix, where each agent have well de…ned simple rules for how to act in di¤erent situations and for which agents to interact with. For this reason the system is sometimes referred to as the agent network or simply the network. The behaviour are determined by the agents DNA string, which lie within certain bounds speci…ed by the system designer. The agents that through their interactions with other agents has showed a successfull behaviour, are selected for reproduction. Since all agents has a limited lifetime this will give an evolution pressure towards a system that solves the processing task to the best of the systems ability. The best that the system can do depends on the 'degrees of freedom' speci…ed by the design; if for instance a concept of time is needed to solve the task and this dimension doesn't exist in the system, the system will ofcourse not be able to solve the task.

Real-life terms Although the agents only exists as bits and bytes in a computer memory by the good will of a program, real-life terms like physical, observe, mating, eat, alive, dead and others will be used in the analysis of the network. Provided they describe what actually happens or rather describe what might as well had happened in the network, the analysis power of these terms lies in their clear intuitive meaning to us, and thus demands a minimum of cognitive e¤ort to understand and manipulate, compared to less familiar descriptions.

Analogies Besides using real-life terms, where ever it seems appropriate, the analysis will also use analogies. The phenomena showing signs of self-organization and self-reproduction occurs in a variety of situations, from the cell level to human societies and it is believed that they are more or less manifestations of the same universal rules or principles. Assuming that these universal principles exists, which many observations point to, it is the thesis that the interacting agents of the network described here, are also a manifestation of these principles. To help illuminate a hypothesized principle, one or more of the (believed) manifestations will be used whenever possible as analogies to the manifestation in the network, see …gure 1 for a conceptual illustration.

One of the analogies that are made with a theoretical foundation, is to Niklas Luhmanns theory for social systems. This is done under the assumption that a social system and what happens in the network can be viewed as two manifestations of the same universal principles, where the latter is just a more simple manifestation.

Organization of the report In the following chapter the behaviour in common for all agents are described. Chapter 3 deals with the reproduction of the

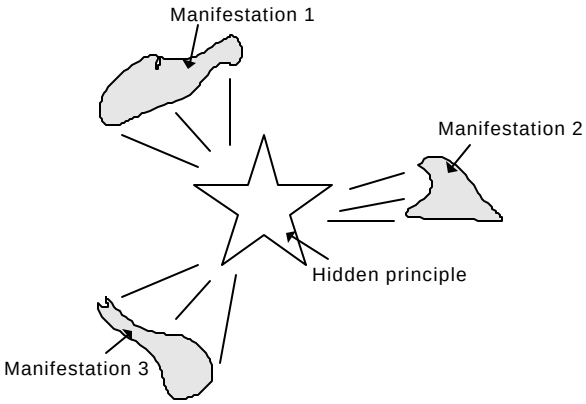

Figure 1 A universal principle or truth only appears as di¤erent manifestations (shadows) to human perception. To understand (or narrow down) the hidden principle at least two manifestations are needed.

agents (and the system elements) and in chapter 4 the concept of 'structures' are treated. Chapter 5 describes the three di¤erent types of agents in more detail but are not up-to-date.

As this document describes a suggestion for a speci…c system, the system will be referenced by the name Colline[1] (sometimes just the system or the network).

[1] The …rst letters in: Collector, inpoder and e¤ector

Chapter 2 Colline agents

(intro)

2.1 Agent primitives

The autonomous agents living in Colline can be characterised by the following attributs and behaviour:

Agent attributs	Agent behaviour
- DNA string	- can observe other agents
- Appearance	- can trade messages
	- can process messages
	- can reproduce

A message is simply a string of N bits, e.g. [10011], where N=5. The four types of behaviours: Observing, trading, processing and reproducing, are in common for all agents, but varies (within system designed bounds) from agent to agent. E.g. some agents have very strict demands to who they observe others are more relaxed, some agents buy large messages some buy small, the processing changes from agent to agent and the amount of …tness transferred to o¤spring varies from agent to agent, just to mention a few examples. Exactly how the agent behaves in di¤erent situations is determined by the DNA string attribute. The appearance attribut can be interpreted as a shorter description for the DNA string and is the only thing that is visible to other agents. The appearance is treated in more detail later.

The DNA string places the agent in a behaviour space, see …gure 1.

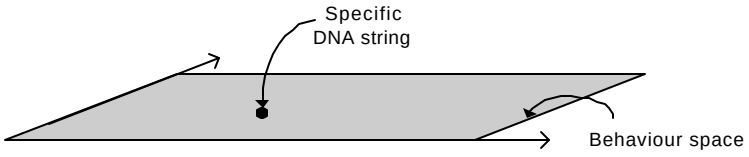

Figure 1 The shaded area represents all possible behaviours for the agents, where each DNA string can be interpreted as a point. (The real behaviour space is 4-dimensional, one for each behaviour type).

Agents whose DNA string cluster together in this behaviour space, have almost the same behaviour, and vice versa.

The four di¤erent basic types of behaviours which span the behaviour space are intended to constitute a primitive set of behaviours. Primitive in the sense that two of the other behaviours could not be combined to form one of the others. Also, with a somewhat fuzzy term, the four of them together constitute a wholeness. To explain this the four di¤erent kinds of technic lego pieces seen on …gure 2 can be used.

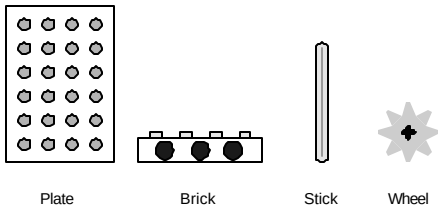

Figure 2 The four di¤erent kinds of lego pieces constitutes a wholeness, that cannot be further divided.

Each kind of the four lego pieces comes in various forms: Plates and bricks have di¤erent sizes, sticks have di¤erent lengths and the wheel comes in di¤erent sizes with or without tooths. This corresponds to the legal variability for the four agent behaviour primitives. The four lego pieces is a primitive set that can be putted together to form a bunch of various machines. Only the imagination sets the limit, i.e. the imagination and that only lego machines can be build from lego pieces. This last statement is known as operational closure (right?): When building additional structures to a lego system, only lego bricks can be used. In the lego universe, wheels from old alarm clocks, screws, nails and wood doesn't exist. If a child (or a childish adult) wants to add something that is not one of the primitives or a combination it is either impossible (or 'messy' as when using tape etc.) or the item will have to be 'disguised' as one of the primitives. As an example, inorder to add an electric engine to a lego system, the engine will have to be disguised as one of the primitives: e.g. use what looks like a stick as the rotor and put what looks like a plate on its upper and lower side. In other words when you want to play with the lego pieces you have to play on their terms, so to speak.

The same is true for the agents in Colline. When you want to add messages from the environment to the network, you have to disguise the entrypoints as agents, and simulate the primitive trading. And when the trainer needs/ requests information from the network, he must simulate the primitives observing and trading. What exists or is important in the world for an agent is similar agents, or what looks like similar agents.

2.1.1 The ghost in the machine

The di¤erence between the lego pieces and the agent behaviour primitives, is that the latter is collected in an entity, the agent, with a will of its own: The agent observes its environment and act according to these observations through simple internal rules. The lego pieces are dependent of external forces (the person putting the pieces together) in a di¤erent way than the agents are dependent on external forces. The agent network are guided towards a solution by a trainer, like a barrel wheel rolling down a hill that is guided by a stick in the hand of a child, whereas lego systems are build using blueprints. The solution strategy for the agent network is based on stage-setting rather than blueprints.

Figure 3 shows the external forces working on the system.

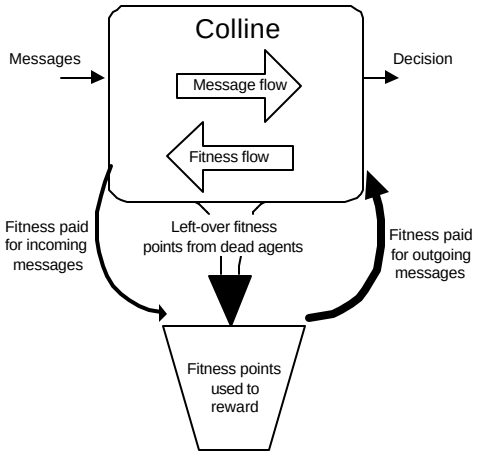

Figure 3 Two streams ‡ows through Colline in opposite directions: The …tness ‡ow and the message ‡ow.

The 'ghost in the machine' that guides the network is the …tness stream. The …tness stream is used by the trainer, like the stick guiding the barrel wheel, to push the network in the right direction. The message ‡ow is tightly connected to the …tness ‡ow, in that it is the ground to which Colline relates the guidance received from the trainer. Using the barrel wheel analogy, the message ‡ow is the hilly landscape that the wheel is rolling in.

2.1.2 System elements

The system elements are by de…nition not the agents in the network: The elements are the transactions that occur between agents in the system. A transaction occurs when an agent sells a message that another agent process and receives …tness points in return. The selling agent causes a change in the buying agent (processing state and …tness) and likewise the buying agent changes the state of the selling agent by increasing its …tness (a kind of "con…rmation"). Therefor a transaction can be interpreted as agent communication, and the network as a whole as a social system of agents communicating

This makes it possible to link the agent network to the social system theory by Niklas Luhmann (NL), thereby taken advantage of his theoretical groundwork. The elements constituting a social system in NL's theory is communication actions (events) between humans. This corresponds to transactions between agents in the agent network. The agents is a part of the environment for the network.

	Social system	Colline
Elements	Communication actions	Transactions
Environment	Psychic systems (people)	Agents and fountains

These de…nitions are untraditional in science and is a part of what NL refer to as an emerging paradigm change in science.

In the following excerpts from Social Systems will be used where it seems appropriate.

The autonomous behaviour of the agents, combined with the guidance from the trainer gives rise to the creation, conditioning and evolution of the system elements constituting Colline, see …gure 4.

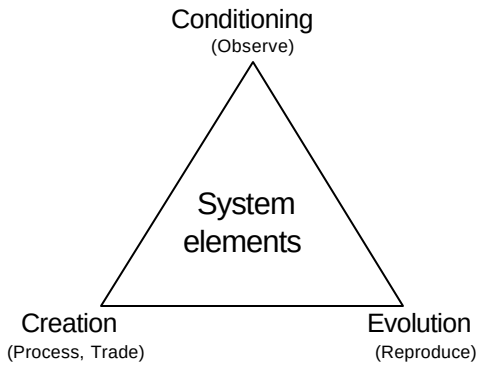

Figure 4 The four primitives are each involved in creating, evolving and conditioning the system elements. The close inter-relationship between the concepts and their 'wholeness' is symbolized by the triangle.

The creation of the elements (transactions) is based on the primitives processing and trading. Evolution of the system elements (and their relations) is based on the primitive reproduction. Conditioning of the system elements are based on the primitive observing. The conditioning is the constraint given on the relations possible among elements. This constraint is innate in the observe primitive, in that an agent only observes agents with a speci…c appearance(s). Luhmann:

> "Out of the relation among elements emerges the centrally important systems-theoretical concept of conditioning. Systems are not merely relations (in the plural!) among elements. The connections among relations must also somehow be regulated. This regulations employs the basic form of conditioning." (NL 1.II.5)

Agents observes agents with a certain appearance, because they know what to expect in some sense (who they observe and so on). This chain of agents observing each other is here what is understood by conditioning (is elaborated in the chapter on structures).

2.2 Three agent sub-classes

Of the four lego pieces on …gure 2 the wheel separates out in that it falls into three classes: with teeth, without teeth and with a slope, see …gure 5.

The di¤erences between the three wheel classes (and classes in general), is of a di¤erent nature than the di¤erence between di¤erent instantiations of the same class: If the characteristics of two instantiations of the same class is mixed together (mating), it forms a third instantiation (the o¤spring) that also lies within the class. Mixing between classes are not legal; it doesn't produce anything meaningfull (e.g. round plates, sticks with holes in them etc.).

In the same way the agent primitive processing separates out from observing, trading, and reproducing, in that it falls into three classes: One where messages are collected, one where messages are searched for a speci…c pattern and one where

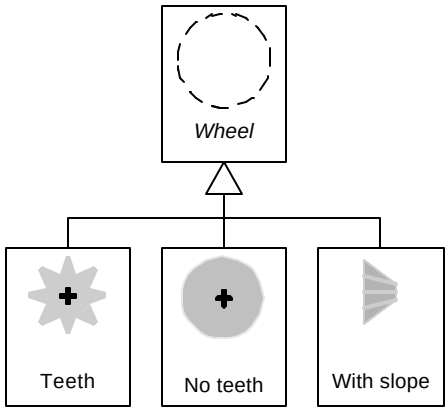

Figure 5 The primitive 'wheel' exists in three di¤erent classes, all derived from the abstract class Wheel.

two messages (agent states) are compared. These three classes for processing give rise to three types of agents: Collectors, inpoders and e¤ectors, only distinguished by the type of processing they do. The splitting into three classes (also referred to as agent types) are illustrated on …gure 6.

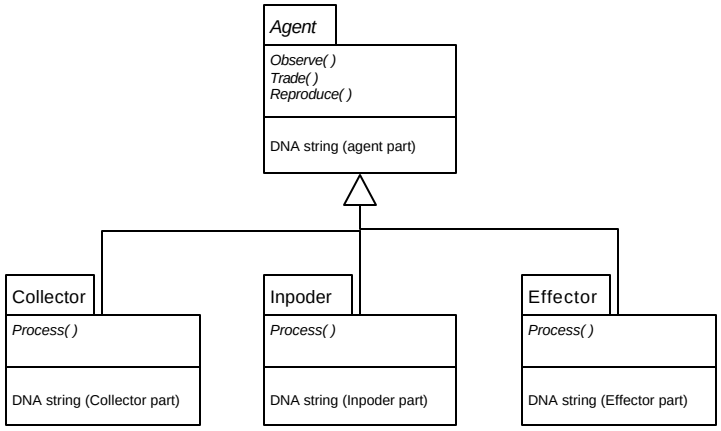

Figure 6 The three agent types inherent the common primitives from the abstract Agent class

The three agent types and their di¤erences are described in more detail in a later chapter, until then they are all treated as agents since the di¤erent processing doesn't matter for the following analysis.

2.3 Agent states

The time in the network is divided into cycles, where each living agent is de…ned to be in one of three states: Observer, Processor or Drifter.

2.3.1 Observer

The observer observes a number of other agents in its neighbourhood. The agent doesn't move like the observed agents don't move. The agents observed are the ones that meet the observers appearance criteria. If one of the observed agents processes a message the observer 'wakes up' and gets the possibility to make a bid for a message.

2.3.2 Processor

The processor is an agent who bought a message in the previous cycle, which is now processed. After processing observers are noti…ed that they can now buy. Trading is done with everyone interested in not-scarce messages (state information from collectors and inpoders) and with the highest bidder for scarce messages (messages are in general treated as scarce resources).

2.3.3 Drifter

The drifter are moving in the network for one of two reasons: (1) The agent is ready-to-reproduce and are therefor looking for a mate (2) The agent lost the agents it observed, because one or more of them moved and is therefor now looking for a spot with enough agents in the neighbourhood for su¢cient input (at least one for collectors and inpoders, at least two for e¤ectors).

The di¤erent states with transition arrows are shown in …gure 7.

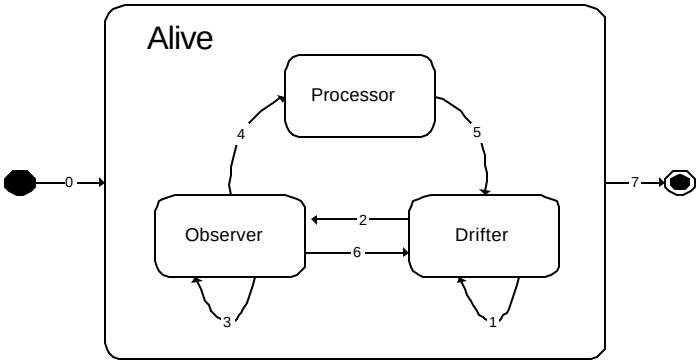

Figure 7 The behavoural states an agent can be in. In each cycle one of the arrow paths are taken.

#	This cycle	Next cycle	Criteria that must be ful…lled
0	Non-exis	Alive	The agent is born, either during startup or because of agents reproducing. The beginning state is drifter.
1	Drifter	Drifter	Drifters that produce o¤spring and drifters that takes a random step keeps the drifter status for next cycle.
2	Drifter	Observer	The agent is not-ready-to-mate and enough agents in neighbourhood matches observe criteria
3	Observer	Observer	The agent still needs one or more messages before processing can occur.
4	Observer	Processor	The agent has bought a message and is ready to process in nect cycle.
5	Processor	Drifter	After processing and trading the agent becomes drifter.
6	Observer	Drifter	When the last supplier to one of the inputs starts drifting, the agent is forced to move itself.
7	Alive	Non-exis	The agent dies because of low …tness or old age.

2.4 Cycle behaviour

In this section the agent activities in one cycle and their coordination is described.

The main loop (called syncronizer in …gures):

1. All processors are initiated in random order (see …gure 8)
2. All drifters are initiated in random order (see …gure 10)
3. Find and delete dead agents, collect their remaining …tness
4. Test whether training session is over, if not increase cycle number and goto 1.

text
text

2.4.0.1 Disintegration of elements

Note on …gure 9 that the change from observer to drifter happens in one cycle for all in‡uenced observers. To give an example of the resulting behaviour, say that agent 1 in …gure 11, cycle n, makes a transaction with another agent causing it to sign up for drifter in cycle (n+1) .

In cycle (n+1) agent 1 is a drifter, but do nothing else than sign up as an observer for cycle (n+2) because it still got the previously observed agent in vicinity (observed agent not shown). However, since the '…x point' of the structure in cycle n, are drifting in cycle (n+1) all agents shown in cycle (n+1) will be drifters. Because only agent 1 in cycle (n+1) has a settled agent (i.e. an observer) in vicinity that match business criteria, all other drifters takes a random step. In cycle (n+2) agent 2 and 3 connect to agent 1, which now again is an observer, the rest takes a random step. In cycle (n+5), i.e. after …ve cycles, the structure will be reestablished.

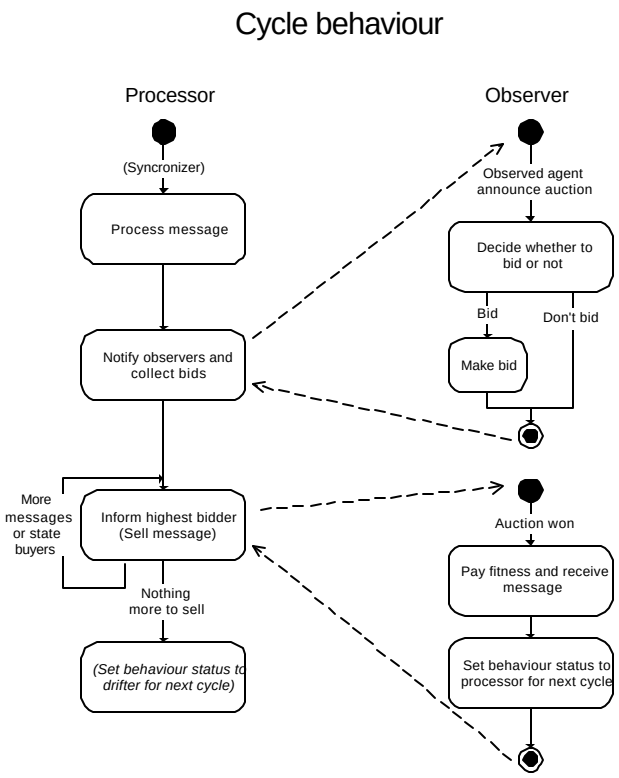

Figure 8 The processors activities are initiated by the main loop, whereas observers activities are syncronized by the processors.

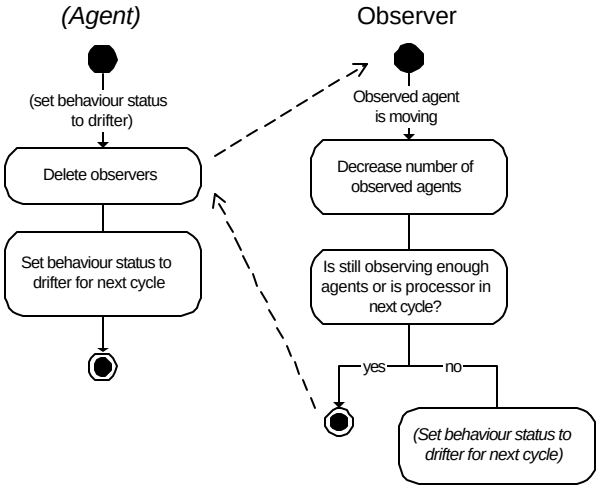

Figure 9 An iterative process is needed when an agent changes status to drifter: All observers stops observing the agent and tests if they need to become drifters themselves.

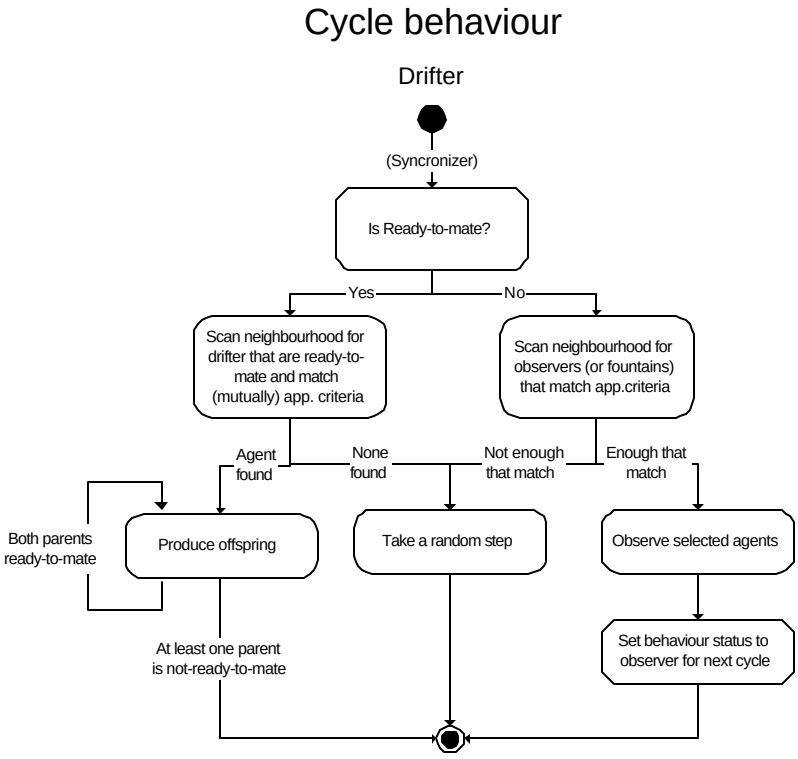

Figure 10 Agents with the status drifters, starts by scanning the neighbourhood for agents to observe or mate with, if none is found a random step is taken. Note that a drifter that mate keeps the drifter status for next cycle, as the next step after mating is …nding someone to observe (do business with).

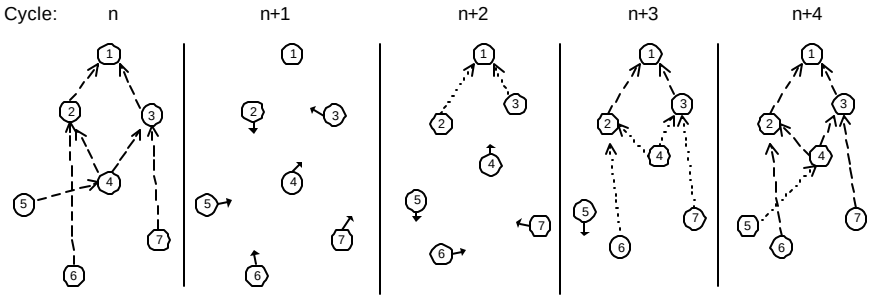

Figure 11 The behaviour of agents when connected to a single '…x point' (agent 1) that starts drifting. (The dotted arrows illustrates that the agent will start observing in next cycle.)

As one cycle is the smallest time-step in the system, the dissolving and following re-establishing of the structure can be thought of as a sudden 'shaking' or 'unsettlement' in the network, like ice-crystals brie‡y getting liquid and then returning to frozen form. In the brief shaking the structure could reassemble into something di¤erent, as some agents might through their random steps get out of vicinity from the agent they previously observed, or new agents from dissolvements elsewhere in the network might enter the scenario and start observing. These sudden 'shakes' helps to ensure that enough agents are drifting, ready to step in when another agent needs buyers. Luhmann:

> "The theory of temporalization's most impressive consequence is that a new interdependence of the disintegration and reproduction of elements results. Systems with temporalized complexity depend on constant disintegration. Continuous disintegration creates, as it were, a place and a need for succeeding elements; it is a necessary, contributing cause of reproduction. Moreover, it supplies freely available material as a result of disintegration, for example, a labile chemical or physical combinatory capacity." [NL 1.III(5)]

The supplied material is in this case drifting agents. Note, however, that this shaking only occurs for agents without redundancy in business connections: If an agent that only need one input observes two or more agents, one of the observed agents becoming a drifter will not cause a change of its own observer status, see …gure 9. It will stay where it is, but eventually all observed agents will dissappear and it will have to move or at least scan the neighbourhood again.

2.5 Training

As a training example lets consider a network that classi…es the input into two di¤erent categories: A and not-A. Before the training starts, the designer chooses which agents that represents category A and not-A. These selected agents are for convenience referred to as decision-agents (sometimes just agents when the context is given). The decision-agents are chosen randomly (however always the type e¤ector) e.g. every agent that matches the appearance [111* **** ****] represents category A and every agent that matches [000* **** ****] represents not-A. Note that the number of decision-agents in a given group changes over time. All agents that does not match any of the two appearance criteria (the mainpart presumeably) belong to the 'inner part'/ processing part of the network.

Say messages representing category A is fed to the network. The network pays a certain amount of …tness for these messages. These …tness points are added to the …tness points that leaves the network because of agents dying, and are divided into two equal piles: One to reward category A decision-agents and one for non-A decision-agents. In this way all …tness points that leaves the network re-enters it again, which is illustrated on …gure 3 on page 5.

After all agents have determined their states, the trainer looks at the selected decision-agents, say 5 for A and 4 for not-A, as shown on …gure 12.

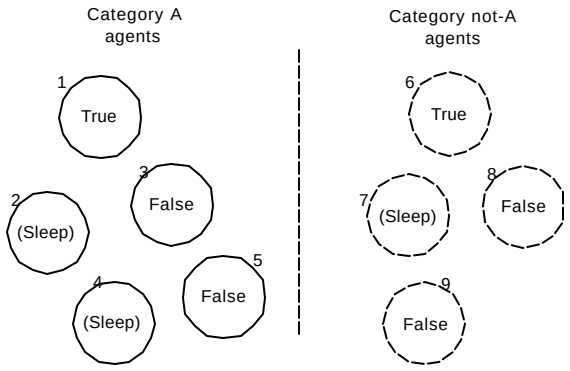

Figure 12 A total of nine decision agents are shown. Five to decide the …rst category and four the second. (The positions shown doesn't (necessarily) resemble physical position in the network).

For the category A decision-agents, only agent 1 gives the correct answer and is therefor rewarded with all the …tness points meant for category A agents. Agent 3 and 5 are punished for their wrong decision by not getting paid, thereby lowering their …tness with what-ever they paid to determine their state. Agent 2 and 4 doesn't change their …tness through this training, but if they keep being inactive/sleeping they will eventually die without producing o¤spring. In the same way agent 8 and 9 are rewarded for making the correct decision for category not-A; they each get 50% of the category not-A …tness points. Agent 6 and 7 are not paid.

The number of decision-agents that decides for each category should probably be chosen higher than 4 or 5 (e.g. 15-20). The number of agents in a group should be able to stay fairly stable, since few agents means that each get a bigger piece of the reward-cake and thus reproduces more and vice versa. If this turns out not to be the case (e.g. during startup of the network) some kind of surveilance of the category agents is needed, such that when the number of agents in a speci…c group gets below a certain lower threshold number (e.g. 10) actions are taken to increase this number. These actions, if necessary, could be to give extra …tness points to the remainding agents (increase reproduction), widening the appearance range for agents in the category or arti…cially adding agents to the group.

2.5.1 Structures emerge

Hypothesis: After a certain period of training the underlying information-processing structure, that causes a correct response, are dominating, thereby ensuring that a majority of the agents for each category gives the correct response.

By 'underlying information-processing structure' is meant the sequence of agents buying and selling messages that led to a speci…c decision-agent giving a speci…c response. Correct responding decision-agents increase their …tness, which 'ripple down the chain' because a certain percentage is always used for buying. In this way succesfull structures are rewarded. Wrong responding decision-agents cannot deliver …tness to their supliers, causing these structures to wither away. More e¤ective structures increase their …tness more than less e¤ective structures in the same category, where e¤ectiveness is simply the number of agents needed

to produce the answer (there don't seem to be any other measure for e¤ectiveness in this case). The structure with many agents will have to divide the cake (…tness from trainer) to more agents than the more e¤ective structure, thus the agents in the e¤ective structure will gain …tness faster than their competitors and through higher biddings start to dominate that territory.

The algorithm used for dividing the …tness, is the bucket-brigade algorithm proposed by John Holland.

2.5.1.1 Money market analogy

The money economy can be used as an analogy to explain the underlying ideas, where …tness is money, agents are companies, messages are goods and the trainer is the …nal consumer. The companies build up structures, i.e. buying and selling of goods, adding value to the goods all the way to the display window. If the consumers like what they see, they buy the goods. The money that the selling company earns, not only bene…ts that company, but also the suppliers to the company and so on. Likewise if the …nal consumer does not like the end product, not only the end product company will su¤er but all the suppliers as well, decreasing their potential for reproduction.

Using the economy analogy it is also clear that a more e¢cient structure will increase …tness more than a less e¤ective structure, where e¤ectiveness here is the number of agents (companies) needed to produce a given answer in a given category (a certain endproduct). If twenty agents (companies) is needed to produce a certain category-answer (certain product), the cake will have to be divided to more agents, than the structure producing the same with only ten agents.

2.5.1.2 Majority-response principle

When the point is reached, where the majority of decision-agents for each category responds correctly, an external trainer is no longer needed; the system can train itself. This is done by considering the majority response to be the correct answer, which is then used for rewarding agents. The majority principle will in this example give responses of two bits [00], [01], [10] or [11], where only [01] and [10] are legal responses since the input pattern must belong to either A or not-A, and can't be both.

The majority response principle allows the decision agents to err once in a while without it a¤ects the performance of the network. In fact a certain error percentage is desired, since the environment can change. If the system is to adapt to changes in the environment where a new response is needed, e.g. (for a more complex network) the response [1 1 1 0 0] instead of [1 1 1 0 1], it must keep a 'small door open' for the bit #5 to be 1 instead of 0. The 'small door open' in this case is the bit #5 agents that made a wrong decision in the old environment, which then will be rewarded (heavily since they are only few) in the new environment. When a new response is needed the system will probably have to be supervised again for a while, but not necessarily: e.g. in speech recognition applications the user can act as a trainer by expressing that the response was wrong.

Chapter 3 Reproduction

The way Colline evolves, to meet the demands of the external trainer, is by reproducing the transactions. The transactions are reproduced by having enough agents and the right kind of agents in the environment, which is to be achieved through only succesfull agents reproducing. The agents reproduce by cross-over mating in which the DNA's is crossed over to …nd the o¤spring DNA.

The criteria for when an agent can reproduce is that its …tness is above a certain threshold, which is the …tness it was born with times a certain factor (e.g. 1.5, 2.0 or 3.0 * birth …tness). This state is referred to as ready-to-mate, alternative not-ready-to-mate.

3.1 Species

In a network that uses cross-over mating (with mutual appearance acceptance), a 1-bit mating appearance gives the possibility for 3 di¤erent species: One where agents with appearance [0] mate with appearance [0] agents, one where [0] mate with [1] (and vice versa) and one where [1] mates with [1]. Each of these three mating connections will cause the genepool to divide into three species (or 'branches'). No more than three species for each agent type is possible with a 1 bit appearance, but less than three is possible because of extinction. Figure 1 shows a hypothetical snapshot of a group of agents in the network, with a 1-bit mating appearance.

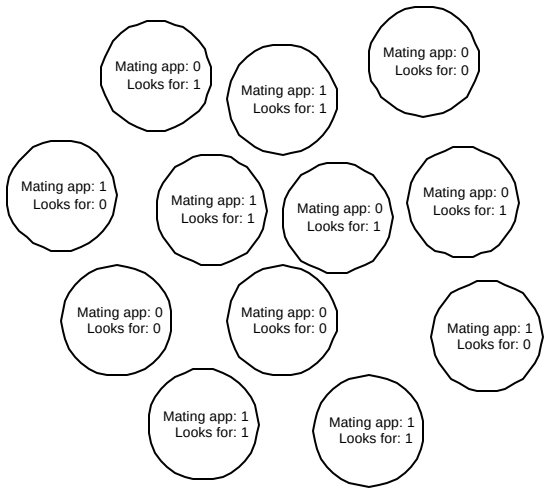

Figure 1 A snapshot of a group of agents in the network showing their physical position, with highlighted info about mating appearance and mating criteria.

Figure 2 illustrates conceptually how they will divide into three distinct species.

The groups of agents on …gure 2 can be viewed as hills in the genepool landscape, a high hill visualises a highly specialized group of agents, i.e. the DNA values has clustered around speci…c values in the behaviour space.

For various reasons (stability, ‡exibility, adaptation and inbreeding-protection) it is probably desirable to allow spreading of genes within species. For this purpose agents with less strict demands for who they mate with are introduced. The less strict demand are represented by '*' (don't-cares) in the accepted mating appearance. These agents are referred to as smear-agents. With a 1-bit mating appearance two types of smear-agents are possible as illustrated on …gure 3.

Figure 3 can also be illustrated more compact, in the following table, using the notation (<mating appearance A> , <mating appearance B>) or (<mating appearance B> , <mating appearance A>) to reference the species where A mate with B. Smear-agents are referenced by their mating appearance.

nSpecies Smear agents n	(0,0)	(1,1)	(0,1)
[0]	X		X
[1]		X	X

The number of smear agents working on a species can be seen by the number of 'X's in the respective column, and likewise the number of species a certain group of smear agents work on is the X's in the respective row. As is seen from the table (and …gure 3) the species (0,1) are smeared by two groups of agents whereas (0,0) and (1,1) are only smeared by one group of agents, which indicate that the species (0,0) and (1,1) will specialize more rapidly than the species (0,1), which is illustrated on …gure 4.

For a 2-bit mating appearance there exist up to 10 di¤erent species, with the following smearing[1]:

n Species Smear **n**	(00,00)	(01,01)	(10,10)	(11,11)	(00,01)	(00,10)	(00,11)	(01,10)	(01,11)	(10,11)
[00]	X				X	X	X			
[01]		X			X			X	X	
[10]			X			X		X		X
[11]				X			X		X	X

Again it is seen that the species evolving from agents who mate with agents of similar appearance are smeared less than agents mating with agents with another mating appearance. This observation …ts well with the type of reproduction we know from real life (we don't hear alot of sheeps through the wall at night, although it happens). These less-smeared species are referred to as clean species.

Assuming that the 'picky' agents, i.e. agents with no '*'s in their accepted mating appearance (non smear-agents), are distributed evenly in the population[2] the chance that mating will occur when two picky agents meet can be calculated: For agents with a 1-bit mating appearance, the chance that agent A accepts the mating appearance of agent B is .5 and likewise the chance that agent B accept A's appearance is .5, which gives a mating probability of 0.5*0.5 = 0.25. A picky agent with a 1-bit mating appearance will then on the average need to meet 4

[1] Note that with a two-bit mating appearance, …ve kinds of smear-agents exists for each of the groups [00], [01], [10] and [11]: Namely the ones looking for [0*], [1*], [*0], [*1] and [**], i.e. a total of 20 di¤erent smear-agents.

[2] At time 0 this will be the case, after a random startup

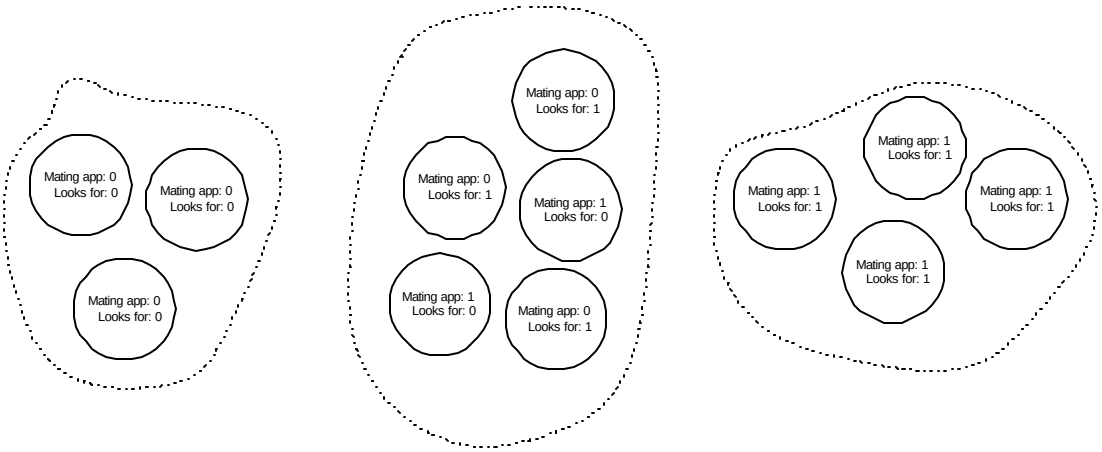

Figure 2 From a reproduction viewpoint the agents divide into three groups, with no exchanging of genes.

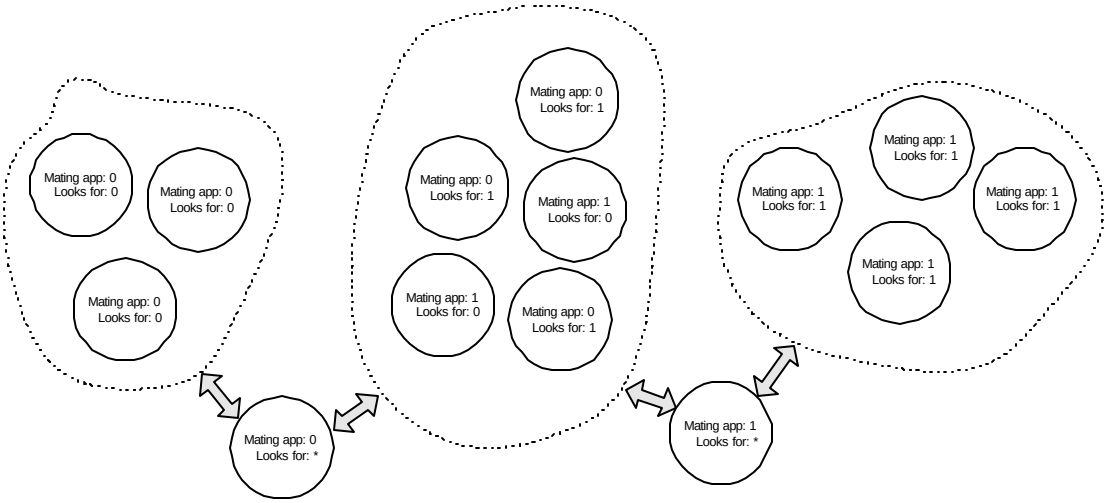

Figure 3 The species are 'smeared' by the two types of smear-agents, i.e. genes are exchanged with another species.

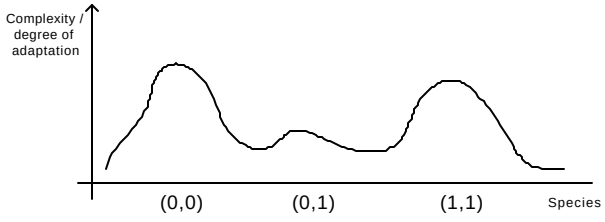

Figure 4 Conceptual illustration of the e¤ect of smearing

other picky agents before mating occurs. The following table show number of species and the chance of mating per meeting, for 1 to 8 bit mating appearances[3]:

# bits in mating app.	# of species	# of clean species[4]	Picky mating chance
1	3	2	1 / 4
2	10	4	1 / 16
3	36	8	1 / 64
4	136	16	1 / 256
5	528	32	1 / 1024
6	2080	64	1 / 4096
7	8256	128	1 / 16384
8	32896	256	1 / 65536

The values in the table are the maximum/ minimum values. Depending on how many agents the network are initiated with and the chosen number of bits in the mating appearance, a number of species will become extinct, thus increasing the chance that a picky agent mate with another picky agent. But in any case the picky agent will often meet a smear-agent instead of a picky-agent, which is both good and bad seen from the agents point of view, good because it reproduced its genes, bad because it did it by mating with an agent which werent as specialized as itself, so most likely the o¤spring won't 'climp any higher up the gene hill'. However, some picky agents will mate with each other, giving resistance to the 'smearing e¤ect'.

3.1.0.3 Subsystems

Hypothesis: Agents from specialized species will start cooperating with each other in a consistent way, thereby forming organizations that is referred to as meta-agents (or subsystems). Seen on a meta-agent level, some of the meta-agent species will eventually start to show a more and more consistent and specialized behaviour, which …nally result in a fully trained network that shows consistent behaviour. The hypothesis is illustrated on …gure 5.

As a metaphor a football team can be used, where the individual players are the agents. The defense, midd-…eld and attack are meta-agents and …nally the whole team is equavalent to the network. In a situation where completely untrained players are used (start-up of network), defense, midd-…eld and attack does not exist. Only 11 confused players running around on the …eld, which corresponds to the three graphs on …gure 5 being almost complete ‡at . Then after a trainer has been working with the team for a while, a defense, midd-…eld and attack slowly starts building up and …nally, if the training is succesfull, a team has emerged, and the trainer can 'let go' and settle with biting nails on the sideline.

[3] The (cumbersome) method I have found for calculating the number of species, S, from the # of bits N is: M=(2^N), S=M+(M-1)+(M-2)...+2+1. A method without the iteration surely exist but I didn't manage to …nd it, something like: (2^N)^2 - ???

[4] The number of clean species is also the # of species each smear-agent group work on, i.e. the more clean species that exists in the network the more species also exchanges genes.

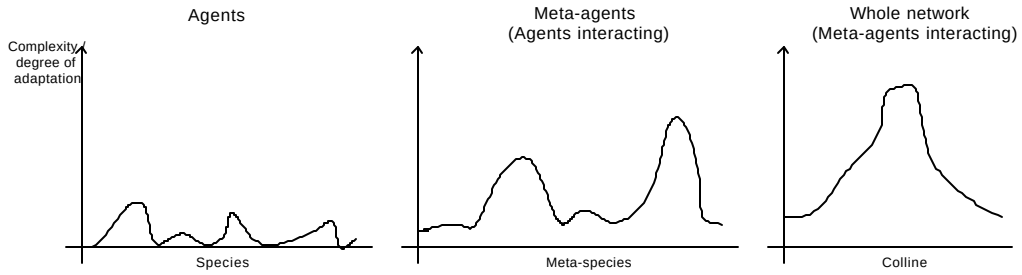

Figure 5 Conceptual illustration of how the co-evolution between agents and meta-agents can be interpreted. The graphs are highly correlated (hypothesis) such that complexity on a higher level only occur when complexity on a lower level has 'settled in'. (Colline actually means small hill in english, so hopefully it will live up to its name!)

Chapter 4 Structures

intro

4.1 The purpose of appearance

The appearance is used by other agents to determine whether or not they would like to do business (observe) or mate with the agent. The appearance of an agent is not a part of the DNA for the agent, it is rather a shorter description for the DNA string, i.e. a shorter description for the agent, see …gure 1.

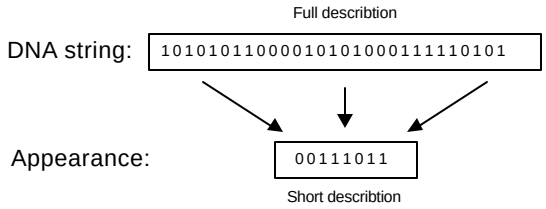

Figure 1 The appearance is a shorter describtion of an agent.

When o¤spring is produced by crossing over the parents DNA, the new appearance that the o¤spring gets, is likewise a crossover of the parents appearance. In this way the appearance will eventually correlate with the DNA string. When the network has just started up the appearance is not correlated with the DNA string, i.e. connections will be made randomly, but soon the succesfull agents who reproduce more will dominate and a correlation between the DNA string and the appearance will start to develop.

4.1.1 Business appearance

The appearance of an agent is divided into an appearance that other agents use for mating selection (referenced mating appearance) and an appearance used for observing (referenced business appearance). The mating appearance was treated in the previous chapter, here the business appearance is treated.

The upperlimit for the business appearance length is the same number of bits as the DNA string counts, this is equavalent to a total description of the agent, in which case the DNA string itself obviously could be used as an excellent 'description' instead of a correlated appearance. This would most likely overwhelm the observing agents with too much information. The observing agent has the opportunity to consider absolutely everything when choosing who to observe, including details about which agents the agent in question observes, how big a percentage it use for bidding, etc.. In real life it would correspond to animals using information about blood pressure, weight, length of tale and so on, when choosing who to interact with or that children in a kindergarden take everything into account when deciding who they want to play with ('do business with').

This is not how it works in real life, (which is a good indication that the strategy is unsuccesfull) where a shorter description of the animal is enough, e.g.

a special scent or bright feather colors etc.. and the kid in the kindergarden wants to play with the kid who has the best toy, regardless of body-odor etc..

4.1.1.1 Limiting relations

Just like the mating appearance is a limitation of the possibilities for reproduction among agents, so is the business appearance a limitation of the 'business' connections possible among the agents. A connection here is simply one agent observing another, with the purpose of buying messages from this agent. The business appearance can also be viewed as a limitation of the possibilities for reproduction on a meta-agent level. On this level the reproduction is not of the agents them-selves, but of the transactions the agent is responsible for.

If there were no limitation of which agents could observe which agents, it would be completely random (within system constraints) what type of information processing would happen next. When this randomness is limited structures build up in the system, Luhmann:

> "Thus structure, whatever else it may be, consists in how permissable relations are constrained within the system." NL 8.II

Constraining permissable relations corresponds to limiting randomness and is in this case the concept of conditioning, which is represented by the business appearance of agents.

.

An example of a meta-agent is shown on …gure 2, where the business appearance is 1-bit long.

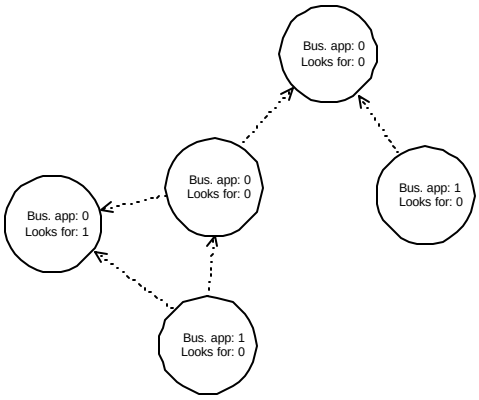

Figure 2 A snapshot of a group of agents in the network and how they observe each other (the dashed arrow). The business criteria is shown for each agent.

The business appearance for an agent and the business appearence it looks for, is referred to as the business criteria for that agent.

For an observer the structure/ meta-agent on …gure 2 will process incoming messages in some way. After a little while one or more of the agents will either leave or die, and be replaced by other agents. If these replacing agents have very di¤erent business criterias than their predecessors, the agents will organize di¤erently and thus create a di¤erent structure. Or if the replacing agents process information in a very di¤erent way than their predecessors this will also cause

observers to label the new group 'not the same'. In this case one would hardly talk of a meta-agent at all, or at least the …rst meta-agent only existed in a glimpse; it didn't manage to reproduce it-self. Luhmann:

> "...only by a stucturing that constrains can a system acquire enough 'internal guidance' to make self-reproduction possible. From every element, speci…c other (not just any other) elements must be accessible, and this must be so due to speci…c qualities of the elements that stem from their own accessibility." (NL 8.II)

The accessibility of an element is here the type of transaction that would follow the present transaction, which is determined from the business appearance of the buying agent.

The reproducibility of the meta-agent is a somewhat fuzzy concept. Say a meta-agent consists of the transactions produced by 100 agents, then one agent leaves and is replaced by another with the same business criteria but it doesn't perform quite the same type of information processsing (e.g. collects messages of maximum 15 bits instead of maximum 20 bits). This change would probably not be noticeable, i.e. wouldn't in‡uence the behaviour of the meta-agent in any signi…cant way. Most observers would consider the identity of the meta-agent to be the same. But what if a few agents are replaced by others with di¤erent business criteria, causing a change in the structure? Depending on how big the change is, it becomes harder to give a clear 'yes' or a clear 'no' to the identity question. The same question can be raised for other self-organizing systems: E.g. for human beings where every single cell in the body are replaced regularly and for the football team playing with 3 new players because of injuries.

4.2 Structures as expectation structures

For convenience the concept of expectation is ascribed to agents[1]: The agent who buys and processes a message expects that buyers exists for the processed message (if buyers doesn't exist there was no reason to buy any messages seen from a …tness point of view). These buyers again expect buyers for their messages and so on until the ultimate buyer and source of …tness, i.e. the trainer, decides if the expectations of the …nal agent should be ful…lled. The expectation goes the other direction as well, since a potential buyer will not observe an agent unless it is expected that the agent will sell a message that is fruitfull to process.

This means that a transaction (element) only occurs in Colline because it is expected that other transactions (elements) will connect to it.

In this sense the structures that develop in Colline are expectation structures, which follows Luhmanns theory:

[1] Expectation is another real-life term that is used, even though the agent ofcourse doesn't expect anything. However, because expectations is something that might as well occur in the system, it can be used for analysis purposes.

> "Event/structure theory and theory of expectation come together in the thesis that the structures of social systems consist in expectations, that they are structures of expectation, and that there are no other structural possibilities for social systems, because social systems temporalize their elements as action-events." (NL 8.V)

Regarding the response of Colline to the trainer, the concept of expectation reduces the complexity of the training, to either ful…llment or disappointment of the trainers expectation (for each category, see …gure 12 page 13). The trainer can simply ignore the wrong deciding agents and reward all others; there is no need for an explanation why the answer was wrong. Luhmann:

> "Signi…cantly, the formation of expectations reveals deviance to be disturbance without requiring one to know why. (...) The formation of expectation equalizes a multiplicity of highly heterogeneous occurences under the common denominator of disappointing an expectation and thereby indicates lines of action." (NL 8.V)

The disturbance here is the decision-agents giving the wrong answer in the categories, and the lines of action taken is simply not giving them any …tness points.

According to Luhmann there is two possibilities to how one can react to disappointments:

> "One is almost forced to react to disappointment. One can do so by adapting the expectation to the disappointment (learning) or conversely, by retaining the expectation despite the disappointment and insisting on behaviour conforming to the expectation." (NL 8.V)

These two choices for reaction: Learning and stubbornness, is both used in the network. Learning happens from the trainer rewarding only the agents giving the correct answer, thereby he can be said to adapt his expectations: He expects the agents in a certain category to give the correct answer (afterall he selected only a small group for each category) but then if it turns out not to be the case, he decides to not reward the agent. The adaptation that the trainer is capable of requires the knowledge to distinguish right from wrong answers (or good from evil one might add) a knowledge that the agents do not have. The agents react to a dissappointment by being stubborn, convinced that what they do are always right. They don't change a single bit of their behaviour from birth to grave

The two choices of reaction to dissappointed expectations, divides expectations into two, the ones that are willing to learn when dissappointed and the stubborn ones that are not:

> "Expectations that are willing to learn are stylized as cognitions. One is ready to change them if reality reveals other, unanticipated aspects. (...) By contrast we will call expectations not disposed toward learning norms." (NL 8.XII)

In these terms the expectations of the agents in Colline, are all normative expectations. Only the expectations of the external trainer is cognitive expectations. Note that it is not the expecting agent, which either learn or don't learn.

If this was so, it would be the trainer (viewed as the …nal agent) that would learn from the training, which of course is wrong. Rather it is the social system, here Colline, that learns. By using the trainer as a guide, the Colline network gets an answer to the question: Which transactions should a given transaction connect to? The answer is given in rewarding the expectation structures, i.e. the sequence of transactions, that lead to a correct true/ false decision.

Many di¤erent expectation structures will be tried out but only the successfull ones will sustain . The succesfull structures are those that create a stream of …tness points from the external environment back to the external environment. The stream of …tness points can be interpreted as the blood of Colline, which gives nutrition to the agents leading the stream.

4.2.0.1.1 Choosing the business criteria So how many bits should the business appearance be? What is enough 'internal guidance'? The straigh forward answer is: The business criteria gives enough 'internal guidance' if it causes leaving agents to be replaced with agents that don't change the structure and the transactions (much).

The upper limit for the number of di¤erent business appearances that is needed is the number of di¤erent species that exists (more)

Chapter 5 Agent types

intro

CHAPTER NOT UPDATE

5.0.0.1.2 DNA determined variables common for all types of agents A_L_{max} : The maximum lifetime measured in number of cycles

A_App : The appearance of the agent, a string of bits.

A_App_{mate} : The accepted appearance used for cross-over mating, a string of bits where only some are active the rest are don't-cares.

A_Bid: The percentage of …tness used for bidding.

A_Mate_{thr} : A …tness threshold value. When this value is reached the agent starts reproducing (until the …tness is below half the birth …tness).

A_O¤ : The percentage of …tness transferred to o¤spring (…xed in …rst implementation).

5.1 Fountains

Fountains are the entry points in the network from the external environment. The fountains are stationary and dead, so to speak, and hence are not a type of agents. As the name implies they simply pour in messages which can be processed (but not necessarily) by the agents in the network.

5.2 Collectors

The collector agent buys messages from, i.e. observes, either fountains (the external environment), e¤ectors or inpoders. The messages that the collector buys are concatenated to one long message, hence the name collector.

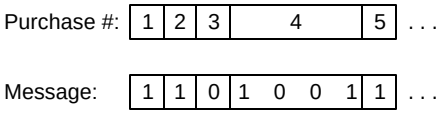

Figure 1

The concatenated message is sold to inpoders.

Everytime the collectors message grows by a chunk, the observing inpoders gets the chance to make a bid for the message. The collector has an upper limit for the size of the message its building. If the upper limit is reached and the message is still not sold the collector discards the message and starts over. To determine its state the collector have a message size threshold value: When the size of the message it is gathering is below the threshold value, the processing state is false, other wise the processing state is true. The processing state can be used

by inpoders when buying messages. A state of true indicates that the message is 'large', otherwise 'small'. The message size indication through the collector state is in the beginning very loose, but as the inpoder 'keys in' to observing only certain type of collectors (with 'keyed in' size thresholds) the collector-state information gets more and more valuable. The term 'key in' can be described using the size threshold as an example. At a given spot in the network there will be some bene…tial range for C_N_{thr} for the collector at that spot. Bene…tial meaning that the …tness of the buyers increase. If the threshold C_N_{thr} is outside this range, by de…nition the …tness of the buyers decrease, which means lower income for the collector and ultimately another collector will step in with another C_N_{thr} value. This process will cause the C_N_{the} value to key in to a bene…tial value.

The more time the more information and the better foundation for making a correct decision. But in…nite time is not available, therefore a tradeo¤ between amount of information (message size) and time must be made. This trade o¤ is represented by C_N_{thr} in the collector. The collector is the only agent which can be said to have a (simple) perception of time, where time is measured in number of purchased message chunks. The state false indicates to potential buyers that 'the message you can buy is small, I suggest you wait a little while longer for more information' and correspondingly the state true indicates 'the message is big, come on buy before I have to discard the information'.

5.2.0.2 DNA determined collector variables:

C_N_{max}: The maximum allowed number of bits in the concatenated message. If this value is reached or exceeded and the message is still not sold, the message is discarded.

C_N_{thr} : The threshold value that separates 'small' messages from 'large' messages. The agent state is false if the message is below C_N_{thr} , otherwise true.

C_App_{obs} : The appearance criteria that an agent must full…ll inorder to be observed by this one.

C_Proc: Only used when the collector observe inpoders. If C_Proc is true the collector only buys processed messages, otherwise only unprocessed messages.

5.3 Inpoder

The inpoder buys messages from collectors and examines this message from left to right, looking for the bit pattern it 'eats'. If the pattern is in the message the state becomes true, otherwise false. As an example consider an inpoder that eats the bit pattern [1 1 0 *] . The inpoder buys the message [1 0 1 1 1 0 0 0 1 1], and after processing the message, the state will be true since bit #4-7 matches the search pattern. This bit string is now erased (eaten) from the message and the remains are separeted into two messages: A processed (searched) [1 0 1] and an unprocessed (unsearched) message [0 1 1] which can be sold to observing collectors.

The job of the inpoder is to reduce the amount of information ‡owing in the network, i.e. eat messages. In the example four bits is transformed to one bit (the

processing state). In this manner messages from the environment are transformed to a state pattern, like the one shown on …gure 12.

Say the message from the environment is the following four vectors:

"Ã !Ã !Ã !Ã !#
1 6 7 3
7 6 3 4

binary representation

"Ã !Ã !Ã !Ã !#
001 110 111 011
111 110 011 100

aligning vertical

MSB_1	0	1	1	0
-	0	1	1	1
LSB_1	1	0	1	1
MSB_2	1	1	0	1
-	1	1	1	0
LSB_2	1	0	1	0

There is several possibilities how this information can be fed to the network. One method is to feed each horisontal line to its own fountain, either bit by bit or in larger chunks. Another possibility is to use only one fountain receiving chunks of 6 bits (horisontal line in table), representing one vector. Whatever is chosen, information will be lost because of the (potential) *'s or don't-cares in the chunks that inpoders eat. Thus the incoming message might be interpreted like this by the network:

MSB_1	0	1	*	0
-	0	1	1	1
LSB_1	*	*	*	1
MSB_2	*	1	0	1
-	1	*	1	*
LSB_2	1	0	1	*

which (strictly mathematical) could arise from the following input vectors:

"Ã !Ã !Ã !Ã !#
0;1 6;7 2;3;6;7 3
3;7 4;6 3 4;5;6;7

In practise the potential patterns should (I think) however be limited by the network, since there will/ must be correlation by two following input vectors (more)...

5.3.0.3 DNA determined variables for inpoders:

I_App_{obs} : The appearance criteria that collectors must fullfull to be observed.

I_FOOD : The bitstring that the inpoder looks for in messages, e.g. [1 * 11 * 0]

I_CS: Boolean value that speci…es whether the inpoder cares about the state of the selling collector

I_PS: The preferred state from selling collectors.

5.4 E¤ectors

E¤ectors performs logic opererations (OR, AND, XOR etc.) and uses the state of other agents as input. The output of the e¤ector is considered as a 1-bit message, which can be sold to other e¤ectors or collectors. The output can also be viewed as the processing state of the e¤ector, but it can only be sold once. The logic operations is performed by summing the weighted inputs (which is 0 or 1). If this sum is below a threshold value the output is 0 otherwise 1 (similar to perceptrons).

The e¤ector has two or three inputs. Three inputs is necessary for operations like XOR and XNOR. The e¤ector could also be designed with only two inputs and an internal value which decides which operation to perform (maybe better than the perceptron design).

The agents that decided the response of the system are e¤ectors.

5.4.0.4 DNA determined values for e¤ectors:

E_3in : Boolean value that speci…es whether the e¤ector has two or three inputs

E_App_{obs1} : The appearance for input 1 supliers

E_App_{obs2} : The appearance for input 2 supliers

E_App_{obs3} : The appearance for input 3 supliers

E_W_1 : Weight assigned to input 1

E_W_2 : Weight assigned to input 2

E_W_3 : Weight assigned to input 3

E_Thr : The threshold value used for computing the output

NOTES:

the symbolic generalization is big in the system to begin with, but respeci…es and gets more and more speci…c, e.g. observes [***1] to begin with and later [*011]

If the decision-agent pays a high percentage to suppliers, it will have to have a high accuracy to survive

The primitive behaviours observe, trade, process and reproduce can be found again on the system level in a more complex form.