

University of Groningen

Self-organising processes of task allocation

Zoethout, K.

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2006

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Zoethout, K. (2006). *Self-organising processes of task allocation: a multi-agent simulation study*. [Thesis fully internal (DIV), University of Groningen]. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Chapter 3

Formalisation and Verification³

Abstract

In this chapter multi-agent simulation is applied to explore how people organise themselves when they have to perform a task. The multi agent model that we used is based on the formalisation of psychological and organisational theories. Three experiments are presented in which multi-agent simulation is being used to study processes of self-organisation. This chapter is structured as follows. First, we describe how expertise differences and coordination time affect the duration of the task allocation process. Second, we demonstrate how task variety and coordination time are related. Finally, we depict the relation between boredom, performance and task allocation.

3.1 Introduction

How do people organise themselves when they have to perform a task? Before attempting to answer this question, let us start with an observation we made in the Dutch Postal Company about students who had signed up for holiday-jobs. The work they had to do was quite easy. Every day the incoming mail had to be sorted, collected, put together into bundles, and thrown into mailbags. These bags were loaded into transport vans that distributed the mail to the different locations. Although all students were initially assigned to all the different tasks, they had some freedom in interchanging them among each other. And that was precisely what happened. Sorting mail all day appeared to be boring, just as collecting it and putting it into bundles. Therefore, after a period of sorting, most of the holiday-workers left their sorting closets and started collecting and bundling the mail. This switching of tasks eliminated the boredom caused by the sorting work, and so the workers kept on bundling until either there was no more mail left to bundle or they became bored again. Then they returned to their sorting closets and the whole process started all over again. However, not only boredom determined the process of switching tasks. While performing either the sorting task or the bundling task, the workers experienced a certain degree of improvement in their skills. As opposed to the boredom effect, this degree of improvement caused the workers to stick to their tasks for a little longer before they

³ This chapter is published as: Zoethout, K., Jager., W, & Molleman, E., (2006), Formalizing Self-Organizing Processes of Task Allocation, *Simulation Modelling Theory and Practice*, 14, 342-359, special issue on simulating organisational processes

decided to start a new one. Furthermore, some workers appeared to be better sorters whereas others turned out to be better bundlers. This also affected the process. All in all, the workers somehow managed all by themselves to find a balance between improvement of skills and monotony of the task.

This example shows three findings: first of all, the allocation process is determined by the expertise and the motivation of the individual workers. Second, since some tasks are more boring or difficult than others, the allocation process may also be determined by task characteristics. Finally, rotating tasks does not necessarily need to be implemented by an external designer but may emerge from the behaviour of the individual workers as well. This latter finding is related to two approaches in management science: the top-down approach and the bottom-up approach. The top-down approach redesigns organisations in order to meet the altered demands of their environments on the basis of empirical research and organisational theories (Daft, 2004). However, due to the advent of the information age and the use of computational models, the past years the bottom-up approach, which makes use of computer simulation, has become increasingly popular (Carley, Prietula, and Lin, 1998; Carley, 2002; see also Axelrod & Cohen, 2000). Moreover, this approach does not only seem to be a scientifically sound method to study the processes we described in our example above, it might be the only way. In particular cases computer simulation is indeed the only way of studying a certain phenomenon (Vallacher & Nowak, 1994; Harrison, 2002).

This chapter deals with the study of self-organising processes similar to those described in the example. These processes are self-organising because the workers themselves re-allocate the tasks rather than some system, i.e. manager or designer, dictating this re-allocation. We modelled these processes by means of a multi-agent system, i.e., a system consisting of simple agents, workers in our case, who are able to influence each other. We will start this chapter by proposing a theoretical framework that describes the properties of the agents and the self-organising processes of task allocation. Then we will describe how this framework is formalised into the computer simulation program WORKMATE I. The theoretical framework and the formalisation of WORKMATE I will be the main issue of this chapter. To test the model and the dynamics it describes, we conducted three experiments. These experiments were very simple and the results are quite evident, because they mainly show what the model predicts. However, the importance of these experiments lies in the insight in the dynamics that the model generates. The empirical relation is less emphasised.

We conducted three experiments. First, we tested how expertise differences and coordination time affect the duration of the task-allocation process. This relation shows the dynamics of specialisation. Second, we tested how task variety affects coordination time. This relation indicates the dynamics regarding specialisation and task changes. Finally, we tested the relation between boredom, performance and job rotation. This relation shows the dynamics of motivational changes and task allocation.

3.2 A Theoretical Framework

To understand how a system organises itself when it has to perform a task, we will first distinguish between a task and the system that has to perform the task. Then, we will describe how changes in the task will affect the behaviour of the system.

3.2.1 The task

Despite all the different definitions and perceptions regarding tasks, it is quite clear that a task requires one or more skills to be performed. These skills are related to task actions, i.e., the parts which a task consists of. Theories on tasks indicate that a task can be split up into task actions. Tasks can be related to skills in a large number of ways (Hunt, 1976; Weick, 1979; Tschan & von Cranach, 1996). This chapter focuses on the effects of task variety on self-organisational processes by using computer simulation. To do so, the task concept has to be formalised in such a way that it can be subjected to a definition of task variety as well as a way of splitting the task into task actions. This has led us to use the following definition of a task:

A task consists of actions in such a way that for every action exactly one skill is required to perform this action.

This definition can be related to the description of Wood on task variety, which he calls dynamic complexity. Dynamic complexity refers to the task that is changing over time (Wood, 1986; see also Tschan & von Cranach, 1996) and is considered one of the most important dimensions of task complexity. In Figure 5, the actions which the task consists of are depicted by a grey box. The component of task variety (e.g. dynamic complexity) indicates the variety of actions. Every action activates the skill that is necessary to perform the action. In Figure 5 the activated skills are represented by the grey dots. High task variety implies a high degree of changes in the task over time. For example, at t1, the task consists of action 2,3, and 4, which activate skills 2,3, and 4. In case of a high task variety at t2, the task consists of actions 4,5, and 6, whereas in case of a low task variety, the task consists of actions 3,4, and 5.

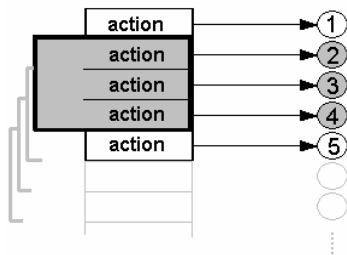


Figure 5: task variety

For instance, at t1, the task is ‘running an apple orchard’. This task consists of three actions: plucking the apples, putting them in a box, and selling them. However, after the apples have been plucked, it appears that some buyers prefer apple-juice to apples.

This implies that at t_2 , the task of running an apple orchard remains, but the actions have changed to: putting the apples in a box, juicing some of them, and selling the juice.

An important characteristic of the actions which a task consists of with respect to task performance is repetitivity (Hackman & Oldham, 1980). Taylor-like organisations where workers have to perform tasks of a highly repetitive nature generate highly specialised, but very de-motivated workers. The repetitivity of the actions can be described by cutting the task into cycles. We define a cycle as the smallest part of a task that still requires the same skills to be performed as are needed for the whole task. For example, a single cycle of running the apple orchard would consist of: plucking a single apple, putting a single apple in the box, and selling it.

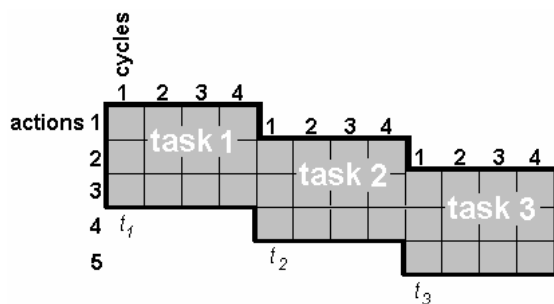


Figure 6: Tasks consisting of actions and cycles

Schematically, we can describe a task as a matrix of actions and cycles (see Figure 6). Splitting up a task into cycles may seem a somewhat artificial approach. Moreover, the number of cycles which the different actions consist of differs for each action. Nevertheless, this schematic description is a useful way of dealing with the different aspects of tasks.

3.2.2 A task performing system

Since the discovery of the ‘Hawthorne effect’ in the 1940s, scholars became more focused at social-psychological components of group performance (Scott, 1981). This led to numerous psychological and sociological studies on group processes in organisational settings (e.g. Thompson, 1967; Hulin & Blood, 1968; Steiner, 1972; Hunt, 1976; Hackman & Oldham, 1980, for an elaborate overview, see Arrow, McGrath, & Berdahl, 2000). Wilke and Meertens state that the most important components that determine group performance are: expertise, motivation, and coordination costs (Wilke & Meertens, 1994; see also Steiner, 1972). Expertise and motivation are individual components that can be considered as characteristics of skills: agents perform some skills better or worse than others (expertise) and prefer some skills more or less than others (motivation). Coordination costs are considered to depend on social interaction. We will describe these components by using a multi-agent model (Zoethout, Jager, & Molleman, 2004). A multi-agent model can describe

complex behaviour at the macro level by using a set of simple interacting agents at the micro-level (see also Gilbert & Troitzsch, 1999). This way of modelling may clarify how task characteristics, individual properties and social interaction may affect the processes of task allocation.

3.2.2.1 Agent properties

According to our description an agent is a simple model of a human being with properties that are necessary to perform tasks. The individual properties are represented as a set of skills, each skill consisting of two variable components: expertise and motivation. Analogous to human memory, skills can be passive or active (Newell, Rosenbloom, & Laird, 1989). The long-term memory store (LTM) consists of a large set of passive skills. Once an agent starts to perform a task, a subset of these skills corresponding to the actions which the task consists of, is activated. The set of active skills corresponds to the short-term memory store (STM). This means that the motivation of the agent is a function of the motivation of the active skills and the expertise of the agent is a function of the expertise of the active skills.

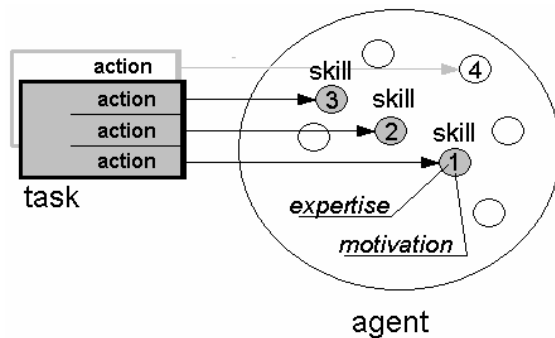


Figure 7: the agent

Figure 7 summarises what we just described. The task, which is indicated by the grey box, activates the skills that are necessary to perform it, for instance, (1) plucking apples, (2) putting them in a box, and (3) selling them. As a result of this activation these skills go from LTM (white dots) to STM (grey dots). Now let us assume that the task changes in time, i.e. there is task variety, for example because some buyers want apple juice. In Figure 3, the box behind the grey box represents the 'new task'. This task also consists of three actions, corresponding to skill (2): putting apples in a box, (3): selling them, and (4): juicing them. This means that as soon as the task changes, skill (1): plucking apples, goes from STM to LTM and skill (4): juicing the apples, goes from LTM to STM. Both the LTM and the STM are subject to different processes regarding changes in expertise and motivation.

3.2.2.2 Interaction between agents

To every skill applies that the values of the expertise and motivation determine whether or not that skill will be used to perform the corresponding action. We can represent this

choice by using thresholds. If the expertise exceeds its threshold a skill is ‘good enough’ and if the motivation exceeds its threshold a skill is ‘nice enough’. If a skill is both good enough and nice enough, the agent can decide to actually use it. If neither expertise nor motivation is high enough, the agent decides not to use it. Because at this point, this decision is solely based on individual properties and has not been affected by social interaction, we will call this decision initial choice. Because this choice is made for every skill, the initial choice is a representation of what every agent would like to contribute to a task individually.

On the basis of the initial choice the allocation process starts. This implies that the agents themselves allocate the tasks, which means that the final allocation of the tasks is carried out by means of a self-organising process. Self-organisation refers to the process in a system leading to the emergence of global order within this system without the presence of another system dictating this order (Dalenoot, 1989; Heylighen, 1997). The process of allocation takes place by mutual influence on the basis of initial choice. This influence causes the agents to try to reach a complementary situation in which every agent has to perform different actions (Kerckhoff & Davis, 1962). This means that in our model every action can only be performed by one agent at the same time, and that every action which a task consists of has to be performed. We have based the process of influencing on a neural network analogy by using excitatory and inhibitory connections. Excitatory connections increase the tension of the node that is affected and inhibitory connections decrease its tension. The choice whether or not to use a skill can be represented by two nodes, an I-node and a You-node of which only one is activated. An activated I-node excites the You-nodes of other agents and inhibits the I-nodes of other agents:

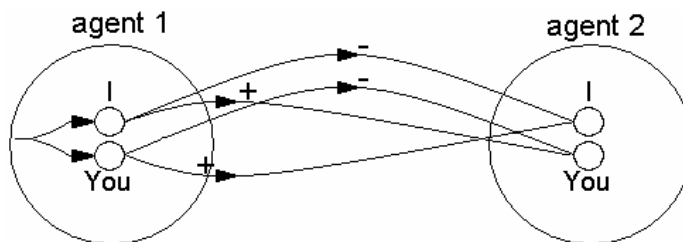


Figure 8: Excitation (+) and inhibition (-) of two agents

Figure 8 only shows the influence of agent 1 on agent 2. Agent 1 wants to put the apples in a box and therefore tries to influence agent 2 not to do so. The same happens the other way around. Both agents influence each other simultaneously until a complementary state is reached, for instance, agent 1 puts the apples in a box and agent 2 sells them. This is called the final allocation. As soon as this state is reached, the unused skills will become passive again and as a result return to the LTM. At this stage the tasks have been allocated and the agents can start performing them.

3.2.2.3 *Learning processes and performance*

During the process of performing their tasks, within the agents two potential changes may occur: changes in expertise and changes in motivation. Changes in expertise refer to processes of individual learning and forgetting. Individual learning refers to improving a skill by using it and forgetting refers to deteriorating a skill by not using it (Nembhard, 2000). This means that the expertise regarding the skills of the agents will change during the performance of the task. Motivation may also change during the performance of the task, for instance due to effects of boredom, especially in the case of tasks that are highly repetitive (Hulin & Blood, 1968; Hackman & Oldham, 1980). Moreover, the motivation of the agent not only changes during the task performance but also as a consequence of the discrepancy between the initial choice and the final allocation. The motivation of the agent is a function of the motivation regarding the active skills. If there is a discrepancy between the initial choice and the final allocation, the active skills change. Therefore, the motivation of the agent changes. If there is no discrepancy, the agent simply ‘gets what he wants’ and therefore his motivation is maintained (see also Hirst, 1988).

Learning can be considered as a process of increasing success in a fixed environment (Krippendorf, 1986). This definition not only applies to increase in expertise, but also refers to the increase in motivation, since motivation and performance are positively correlated (Hackman and Oldham, 1980). In a setting consisting of three or more agents, agents may be able to choose their co-worker. This indicates that their mutual relations may change as a result of task performance or task allocation. If these changes lead to higher performance, we may speak of learning as well. Kitts et al. call this type of learning *structural learning* (Kitts, Macy, & Flache, 1999). The term *structural* refers to changes in the social-network-structure, which is based on Zoethout’s notion of applying learning principles in neural networks to describe structural changes of social networks (Zoethout, 1994; Zoethout, Jager, & Molleman, 2004).

Since we stated that learning refers to a process of increasing success, we have to define what success really is. A system that performs successfully can be considered as a system of high performance. Expertise, motivation, and coordination are the most important components that determine performance (Wilke & Meertens, 1994). This means that a highly performing system should score high on expertise and motivation, and low on coordination costs. In our model, the concept of coordination costs refers to the time it takes to allocate a task. The total performance time is defined as the sum of the performance time of the single agents plus the coordination time.

3.3 Formalisation and Description of WORKMATE

WORKMATE I is a deterministic discrete event-based simulation program for simulation self-organising processes of task allocation, which we developed in DELPHI6. The event is the task that triggers the allocation processes of a multi-agent system. Once the task has been allocated, this system performs it and stops until another event starts. Discrete means that all events are treated separately. Discrete event simulation is often used for queuing models, but is also appropriate for a larger

class of problems related to social and management science (Gilbert & Troitzsch, 1999). In this section we will describe the formalisation of the theoretical framework as depicted in the former section, based on a setting consisting of two agents. The description will follow the actual process, starting with the task and finishing with its performance.

3.3.1 The task

A task can be manipulated by using four parameters: the number of cycles, which refers to the number of times that the same skills are needed, the number of actions, which refers to the minimal number of skills that the system needs to perform the task, the variety of actions, which refers to the extent to which a task changes over time. A variety of 1 means that every next task differs 1 action from the former task. The variety value cannot exceed the number of actions that a task consists of because a variety equal to the number of actions already indicates that every new task is completely different from the former. The fourth parameter is the task unit duration. A task unit is defined as a single cell within the 2-dimensional matrix of actions and cycles (see Figure 6). The task duration is the product of task unit duration, the number of actions and the number of cycles. Task duration is defined as the minimal required time to perform a task. The significance of this parameter will be explained later on in this section.

3.3.2 Initial choice

Once the parameters of the task and the agents have been set, we can start the system. First, all the actions which the task consists of activate the skills of the agents that are necessary to perform the task. Activation of skills means that for every skill the value of expertise and motivation goes from LTM to STM where they are compared to the thresholds according to the following decision rule:

IF Expertise > ExpertiseThreshold AND Motivation > MotivationThreshold THEN I DO ELSE YOU DO

This decision rule means that an agent wants to perform a particular action (I DO) if both the expertise and the motivation related to the skill to perform that action are sufficient. The logical reverse of this rule is that, if either the expertise or the motivation is insufficient, the agent does not want to perform that particular action (YOU DO).

The decision whether or not to perform an action is depicted by means of two nodes, an I-node and a You-node, each having values between 0 and 1. The values of these nodes are functions of expertise, motivation, their thresholds, their maxima, and a parameter λ that determines the balance of expertise and motivation. The initial choice depends on these nodes according to the rule that I DO means that the value of the I-node is bigger than the value of the You-node, and You DO means the other way around.

The I- and You-nodes are calculated as follows: expertise and motivation have values (x) between 0 and a maximum with a threshold somewhere in between. A value of x between 0 and the threshold refers to the You-node and a value between the threshold

and the maximum refers to the I-node. The I-node will be 1 if the x is maximal, and will approach 0 if the x reaches the threshold. The You-node will be 1 if x is 0, and will approach 1 if x reaches the threshold. Therefore, for the I-node, the height of the threshold is subtracted from x . For the You-node, x is subtracted from the threshold. Then, these values are divided by their maxima to make up the I - and You -values between 0 and 1.

We can distinguish three situations. In the first situation there is insufficient expertise, which will consequently lead to a YOU DO choice. In this situation, no matter how high the motivation is, the skill will not be influenced since the agent cannot perform that particular action anyway. The values of I and You can simply be described as:

$$I = 0 \quad (1a)$$

$$You = 1 \quad (1b)$$

The second situation refers to expertise and motivation both exceeding their thresholds, which leads to the initial choice of I DO. In this situation the I-node is a function of the expertise (e) and motivation (m), their thresholds (th_e , th_m), their maxima (e_{max} , m_{max}), and a parameter λ [0,1] that indicates the balance between expertise and motivation.

$$I = \lambda * \frac{e - th_e}{e_{max} - th_e} + (1 - \lambda) * \frac{m - th_m}{m_{max} - th_m} \quad (2a)$$

$$You = 0 \quad (2b)$$

The third situation refers to sufficient expertise and insufficient motivation. According to the decision rule, this situation would lead to an initial choice of YOU DO, whereas the I-node =0 and the You-node are determined by motivation, their thresholds and the balance parameter λ [0,1]. However, this situation may not provide a sound basis to start the allocation process. If we look at the initial choice of the other agent, again, we may distinguish three situations: insufficient expertise leading to YOU DO, sufficient expertise and motivation leading to I DO, and sufficient expertise and insufficient motivation. In the first and second situation the choice is clear. In the third situation both agents have insufficient motivation and sufficient expertise. It will not be plausible to determine the choice of these agents solely by their motivation, as rule 1 suggests. Therefore, we have defined the I-node as a function of expertise rather than labelling it zero, and the You-node as a function of motivation:

$$I = \lambda * \frac{e - th_e}{e_{\max} - th_e} \quad (3a)$$

$$You = (1 - \lambda) * \frac{th_m - m}{th_m} \quad (3b)$$

3.3.3 Excitation and inhibition

Figure 8 indicates the way in which an agent influences another agent by means of excitatory and inhibitory connections. As a result of this influence, the values of the *I*- and *You* nodes change. These changes depend on the values of the *I*- and *You* nodes of both agents: the higher the value of the sending agent, the higher the potential influence. This potential influence is limited by the values of the *I*- and *You* node of the receiving agent. Excitation will decrease as the values approach their maxima and inhibition will decrease as the values approach 0. Therefore, excitation can be described as follows:

$$y_1 = y_0 + x_0(1 - y_0) \quad (4a)$$

y_0 represents the old value of the receiving node, y_1 the new value, and x_0 the old value of the ‘sending’ node. x_0 is multiplied by $(1 - y_0)$ to make sure that the value of y_1 does not exceed the maximum value of 1. Inhibition can be described as:

$$y_1 = y_0 - x_0 y_0 \quad (4b)$$

x_0 is multiplied by y_0 to make sure that the value of y_0 does not exceed the minimum of 0.

The agents simultaneously influence each other. If two nodes with the exact same value influence each other, it is likely that nothing will happen. This means that the difference between two nodes should be included into the excitatory as well as the inhibitory functions:

$$Diff_{II} = Abs(I_1 - I_2) \quad (4c)$$

$$Diff_{YouYou} = Abs(You_1 - You_2) \quad (4d)$$

I_1 , I_2 , You_1 , and You_2 correspond to Figure 8. $Diff_I$ equals the absolute value of $(I_1 - I_2)$ and $Diff_{YouYou}$ equals the absolute value of $(You_1 - You_2)$. We combine the equations (4a) to (4d) to describe all excitatory and inhibitory connections as shown in Figure 8:

$$I_2 := I_2 - \iota I_1 I_2 Diff_I \quad I_1 \text{ inhibits } I_2 \quad (5a)$$

$$You_2 := You_2 - \iota You_1 You_2 Diff_{YouYou} \quad You_1 \text{ inhibits } You_2 \quad (5b)$$

$$You_2 := You_2 + \varepsilon (1 - You_2) I_1 Diff_I \quad I_1 \text{ excites } You_2 \quad (5c)$$

$$I_2 := I_2 + \varepsilon You_1 (1 - I_2) Diff_{YouYou} \quad You_1 \text{ excites } I_2 \quad (5d)$$

ε and ι $[0,1]$ represent parameters that can set the height of the excitation (ε) and the inhibition (ι). Next, we will give an example how the influence actually works. In this example, we will use maximal excitation and inhibition, which means that $\varepsilon = 1$ and $\iota = 1$.

An example: $I_1 = 0.6$, $You_1 = 0$, $I_2 = 0.2$, $You_2 = 0$. We see that both agents have an initial choice of I DO but $I_1 > I_2$, which means that agent 2 will change his choice from I to You. We will now describe the first step of the allocation process.

$$I_1 \text{ inhibits } I_2: \quad I_2 = 0.2 - 0.6 * 0.2 * 0.4 * 0.1 = 0.152$$

$$I_1 \text{ excites } You_2: \quad You_2 = 0 + (1-0) * 0.6 * 0.4 = 0.24$$

$$I_2 \text{ inhibits } I_1: \quad I_1 = 0.6 - 0.2 * 0.6 * 0.4 = 0.552$$

$$I_2 \text{ excites } You_1: \quad You_1 = 0 + (1-0) * 0.2 * 0.4 = 0.08$$

So, $I_1 = 0.552$, $You_1 = 0.08$, $I_2 = 0.152$, and $You_2 = 0.24$. This means that the initial choice of agent 1 is still the same. Agent 2's choice has been changed into You DO because $You_2 > I_2$.

These allocation processes take place for every action. The time it takes to allocate an action is called allocation time. According to the formulas (5a)-(5d), the allocation time is determined by two components. First, we mention the values of the sending and receiving nodes. As the I -and You nodes of the agents are more similar, the allocation time increases. Second, the allocation time is determined by the values of the excitation and inhibition parameters, ε and ι . As the value of these parameters increase, so will the allocation time. This explains the use of these parameters. If the allocation time of the different actions is too low, we cannot distinguish between them. If the allocation time is too high, the simulations will take too much time.

The total allocation time is the sum of the allocation time of the different actions which a task consists of. Since we stated that the coordination costs refer to the time it takes to

allocate a task, we define the total allocation time as the coordination time, t_{coord} , which can be expressed as:

$$t_{coord} = \sum_{i=1}^n t_i \quad (6)$$

whereas t_i is the allocation time of action i .

3.3.4 Learning

After the task has been allocated, the agents start performing it. During the performance processes, learning and forgetting as well as boredom and motivation will occur. An important characteristic of most learning curves is that they reach a maximum asymptotically (Nembhard, 2000). Therefore, we define learning by means of: the relation between expertise at a certain time (t), expertise in the future ($t+1$), the maximum expertise, and a parameter that determines the learning speed:

$$e_{(t+1)} = e_t + \lambda \frac{e_{max} - e_t}{e_{max}} \quad (7)$$

e being expertise, t being the current time step, $(t+1)$ being the next time step, e_{max} being the maximum expertise, and λ being the parameter value between 0 and 1 that indicates the learning speed. In the program, parameter λ is called *learning speed*. Not only learning curves can be described as being asymptotic with a maximum, motivation curves can be described by using the same characteristics. This means that in case of boredom recovery we use the same formula for both learning and motivation. In the latter case parameter λ is replaced by a parameter that is called rest rate, which indicates the motivational recovery from boredom.

Studies on learning and forgetting suggest that the slope of the second is in fact the inverse of the slope of the first (Nembhard, 2000). Therefore, forgetting can be represented as the inverse of formula (7):

$$e_{(t+1)} = \frac{(e_t - \mu)e_{max}}{e_{max} - \mu} \quad (8)$$

μ being a parameter value between 0 and 1. In the program, parameter μ is called *forget speed*. Just as formula (7) expresses both learning speed and the rest rate, formula (8) expresses both forget speed and the boredom rate. Boredom rate refers to the decrease

in motivation as a result of boredom. In this case, parameter μ is replaced by a boredom rate parameter.

3.3.5 Performance

The performance of the system is a function of expertise, motivation and coordination costs (Wilke & Meertens, 1994). The concept of coordination costs is defined as the time that it takes to allocate a task. Expertise and motivation are components of the skills of the agents. We define both expertise and motivation in terms of the time it takes to perform a task: the higher the degree of expertise or motivation, the sooner the task is finished. Furthermore, we define a *minimal* time to complete an action, t_{action} , which is equal to the actual time it takes to perform the action at a *maximal* rate of expertise and motivation. The actual performance time of a single agent, $t_{perf.}$, can therefore be expressed as:

$$t_{perf.} = \sum_{i=1}^n \frac{t_{action_i}}{\lambda \frac{e_i}{e_{max}} + (1-\lambda) \frac{m_i}{m_{max}}} \quad (9)$$

In the program, the agents perform the actions simultaneously. This means that the time it takes to perform the total task, $T_{perf.}$, is determined by the slowest agent and the coordination time:

$$T_{perf.} = \max(t_{perf1}, t_{perf2}, \dots, t_{perfk}, \dots, t_{perfn}) + t_{coordination} \quad (10)$$

3.4 Results

In this chapter, for two reasons we are especially interested in testing the dynamics of WORKMATE I. First, by testing its dynamics, we will observe if the agents behave according to the model we described in the former section. The results will show whether our model has correctly been implemented and whether it does not show any unpredictable behaviour. The second reason is related to the comprehensibility of the system. WORKMATE I is able to generate complex behaviour due to the deterministic interaction of simple elements, the agents. We will not be able to describe the complex behaviour of task allocation if we cannot build our studies on insights in the dynamics of the system. In that sense, the experiments we describe here only serve as a basis for more sophisticated experiments on the relation between tasks and groups.

The dynamics which WORKMATE I is able to generate refer to processes of learning and forgetting, boredom, and changes as a result of task variety. We are especially interested in the way in which these processes are related to task allocation and task

performance. On this basis, we have conducted experiments to answer three questions. Our first research question is:

What is the relation between specialisation and coordination time?

On the basis of the model, we hypothesise that coordination time decreases as specialisation increases. We tested two conditions, small differences and large differences, and compared the performance time. Our second research question refers to the relation of task variety and coordination time:

What is the relation between task variety and coordination time?

Since the agents will specialise less whenever the task variety increases, we hypothesise that coordination time increases as task variety increases. We tested three conditions, no task variety, moderate task variety, and high task variety. The third question depicts the relation of boredom, task allocation, and task performance:

What are the consequences of boredom effects with respect to specialisation, task allocation, and task performance?

The consequences of boredom effects are more difficult to hypothesise. The relation between motivation and task performance is evident but the relation between motivation and task allocation is more difficult to predict. We tested three conditions, a low degree of boredom and recovery, a moderate degree of boredom and recovery, and a high degree of boredom and recovery.

We conducted all three experiments under the following constraints:

The task unit duration is 10^4

The system consists of two agents

The maximum of both motivation and expertise is set at 25

The motivation and expertise thresholds are set at 10

Expertise and motivation are equally balanced, i.e. $\lambda=0.5$

Within each skill, expertise and motivation have the same initial values

The height of the excitation (ϵ) and inhibition (ι) is set at 0.1^5

3.4.1 Experiment 1: expertise differences and coordination time

Since the allocation process ends at the point when the agents have become complementary with respect to the performance of the task, we expect the coordination time to correlate positively with the resemblance of the agents. Here resemblance

⁴ Early testing indicated that this value would give a proper balance between performance time of the individual agents and coordination time.

⁵ Early testing indicated that this value would lead to a proper spread of the interaction time

means: making the same initial choice, i.e. values of the I-and You nodes, with respect to one or more actions to be performed. This would imply that two identical agents are not able to allocate a task. Instead they would generate an infinite coordination time. Because this would not be a realistic assumption we defined an upper limit of the coordination time. Of course this upper limit should not be too low because if so it would disturb the processes. It should not be too high either because that would not be realistic. After some testing, we decided to set the upper limit at 1000 time steps. We tested two conditions. In the first condition we used agents with small differences in expertise. In the second condition the agents showed large differences. These conditions were tested under the following constraints:

Each agent has two skills

The agents have to perform one task consisting of two actions and 50 cycles

The task variety is 0 (no task variety)

The boredom rate is 0 (no boredom)

The learning speed is set at 100, the forget speed is set at 50

The agent has two skills because that is all we need to describe the process. The absence of task variety is trivial because the agents only have to perform one task. However, it is indicated here to make sure that task variety does not affect the process.

Table 1 shows the initial expertise of both agents in both conditions:

Table 1: Small and large expertise differences

Low	Agent 1	Agent 2	High	Agent 1	Agent 2
Exp. skill 1	16	14	Exp. skill 1	20	12
Exp. skill 2	14	16	Exp. skill 2	12	20

3.4.1.1 Results

The results confirmed our expectations as derived from the conceptual model. In the first condition, the agents started off quite similarly, with an expertise difference of 2. But with every cycle the difference increased due to learning and forgetting effects. Figure 9a and 9b show only the active skills, i.e. skills that are necessary to perform the task. In Figure 9a, we see that agent 1 specialises in skill 1 (the upper curve, indicated by the dots) while forgetting skill 2 (the lower curve, indicated by the crosses).

Self-Organising Processes of Task Allocation

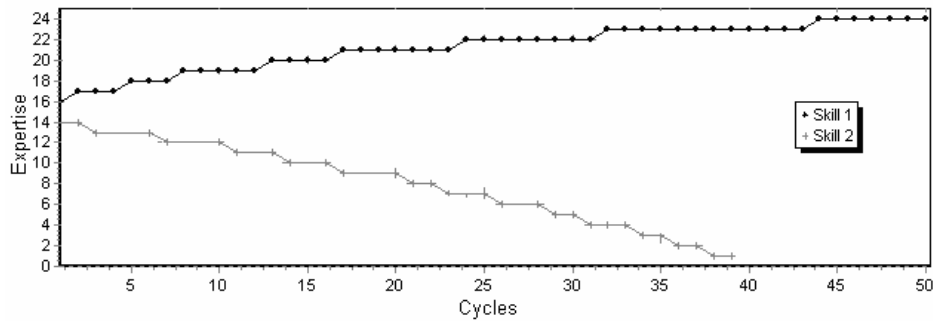


Figure 9a: First condition: small differences: expertise of agent 1

Figure 9b shows the same situation in reverse: agent 2 only uses skill 2 and therefore specialises in performing this skill. He does not use skill 1 and as a result loses it. Hence, agent 1 uses skill 1 and agent 2 uses skill 2.

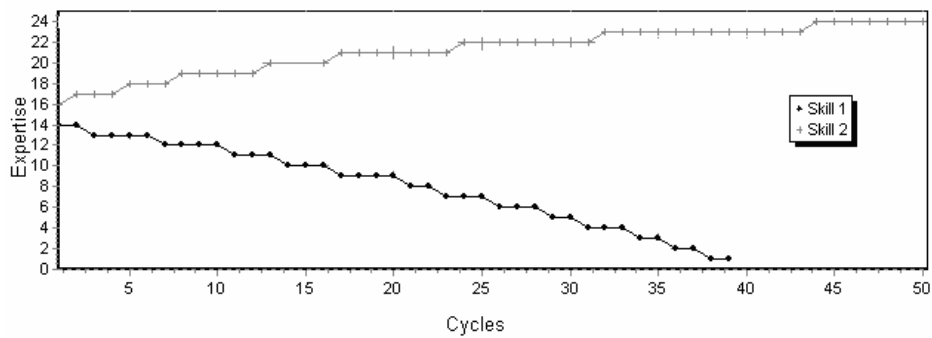


Figure 9b: First condition: small differences: expertise of agent 2

Because of this specialisation, the coordination time gradually decreases, since coordination time is negatively correlated to agent differences. Figure 9c shows this decrease. It shows four curves of which the curve of agent 2 is covered by the curve of agent 1. The curve of agent 1 covers the curve of agent 2. These curves indicate the performance time of the single agents, which slightly decreases due to the increase in expertise. The crossed curve shows the coordination time, and the triangled curve shows the performance time of the entire system, i.e. the performance time of both agents + the coordination time. We see that the coordination time starts at 43, and then decreases until it reaches 0 during the 17th cycle. From that point on we see that the performance time of the whole system is only determined by the performance time of the individual agents.

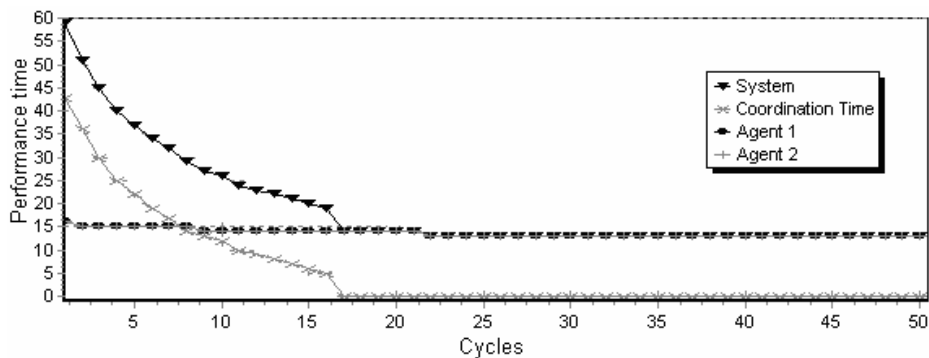


Figure 9c: First condition: performance

The second condition shows the same process starting with different values (see Table 1). As in the first condition, the agents specialise. The main difference with the other condition is the initial difference of the agents (8 vs. 2). The larger difference in the second condition does not affect the specialisation processes with respect to the expertise of the agent. As in figure 9a and 9b, the curves of skill 1 and skill 2 of agent 1 are identical to the curves of skill 2 and skill 1 of agent 2. Therefore, we will only show the expertise of agent 1:

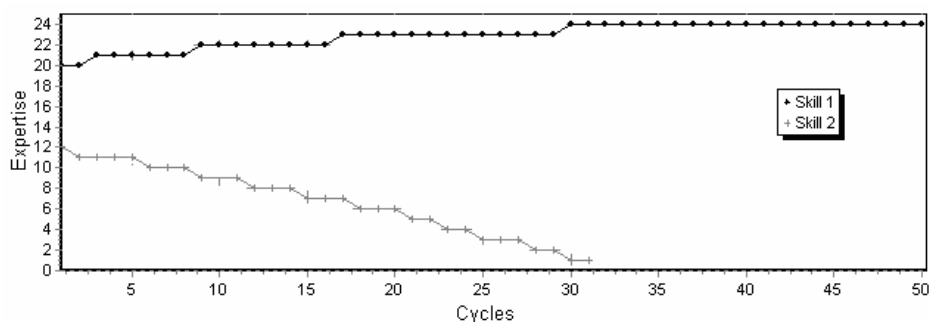


Figure 10a: Second condition: large differences: expertise of agent 1

Furthermore, the larger difference in the second condition leads to a decrease in the interaction time that is much larger than it was in the first condition. The coordination time starts much lower (4) than in the first condition (43) and has turned to 0 in the 9th cycle.

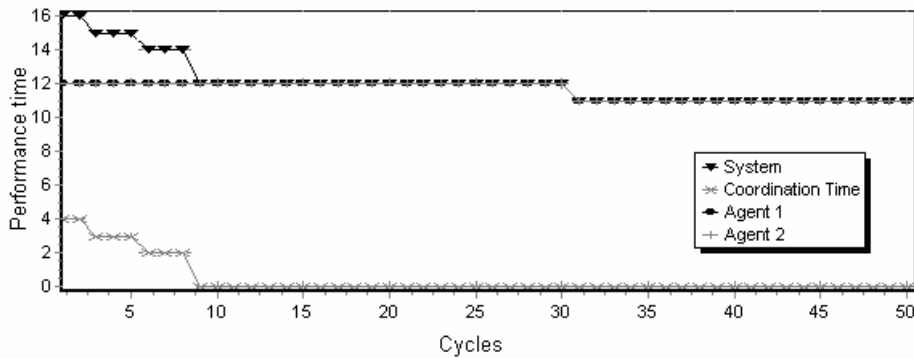


Figure 10b: Performance in the second condition

3.4.1.2 Conclusions

These results confirm that WORKMATE1 generates dynamics that are in accordance with the expectations as derived from the conceptual model. No matter what the initial conditions are, under the constraints we described at the beginning of this section, the best becomes better and the worst becomes worse. This process indicates that the system will always end up in a stable state in which both agents specialise themselves in one or more skills. But again, this only happens in systems where there is no boredom, and in which the same skills are used over and over again.

3.4.2 Experiment 2: task variety, coordination time, and specialisation

As task variety increases, we expect the agents to specialise less. Furthermore, less specialisation implies more coordination time because of the inverse relation between coordination time and specialisation. We tested three conditions: no task variety, moderate task variety (2), and high task variety (4). These conditions were tested under the following constraints:

The agents have to perform 10 tasks consisting of 4 actions and 5 cycles

Every agent has 40 skills

The boredom rate is 0 (no boredom)

The learning speed is set at 100, the forget speed is set at 5

The agents require minimally 40 skills to be able to perform 10 tasks consisting of 4 actions in the condition with a high task variety. High task variety is defined as maximal task variety, i.e. every new task only consists of new actions. We lowered the forget speed to 5 to prevent situations in which the agents forget their skills to perform the last tasks. Table 2 indicates that in all three conditions the agents started off with the same level of expertise.

Table 2: initial expertise of both agents in all three conditions

Skill	Agent 1	Agent 2
$1 + 8*k$	18	17
$2 + 8*k$	15	16
$3 + 8*k$	14	13
$4 + 8*k$	11	12
$5 + 8*k$	12	11
$6 + 8*k$	13	14
$7 + 8*k$	16	15
$8 + 8*k$	17	18

$x + 8*k$ means that the value of skill x equals the value of skill $x + 8*k$, with $k \in [0,4]$.

3.4.2.1 Results

The first condition of no task variety leads to results that are quite similar to those of our first experiment. Since the expertise differences are 1 instead of 2, the coordination time starts at a higher level and decreases more slowly, but both slopes show the same process (see Figure 11a:).

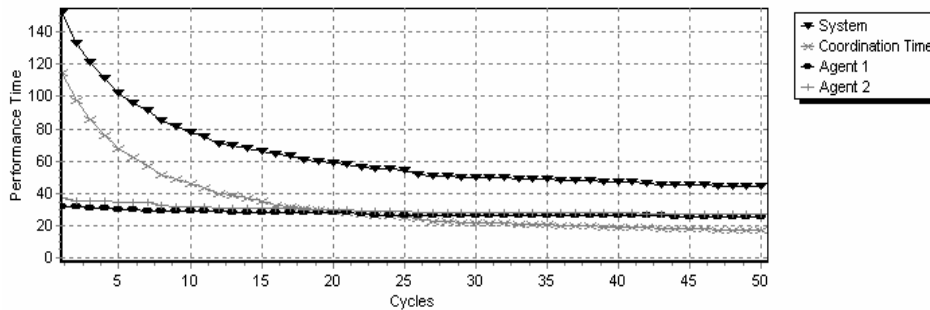


Figure 11a: Performance with no task variety

As depicted in Figure 11b, in the second condition, we observe a decrease in the interaction time while the task is being performed. As soon as the task changes, the interaction time suddenly increases because the emerging new task has to be allocated. Furthermore, we see the performance time of the single agents decrease slightly, due to the specialisation effect within a single task. As soon as the agents start performing a new task, their performance time increases because they are not specialised yet in the new skills that are needed to perform this task. The peaks as shown in Figure 11b are somewhat arbitrary, which can be explained by the level of task variety. Moderate task variety implies that for every new task, only half of the actions change. The remaining half is subject to the specialisation effects we described in the first experiment. This

means that specialisation effects within a task will affect the allocation processes of the next task.

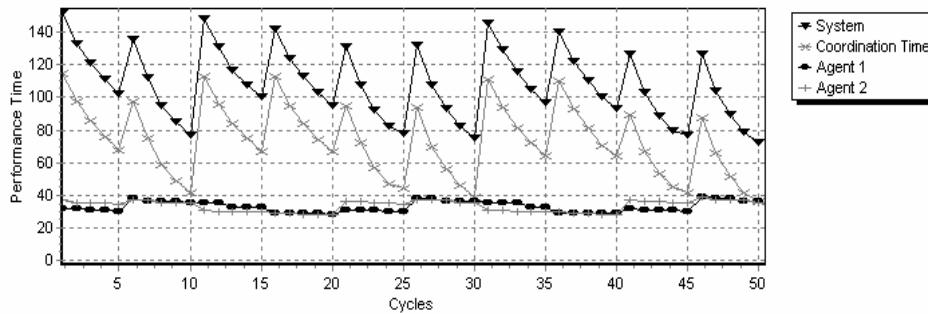


Figure 11b: Performance with moderate task variety

The third condition of high task variety shows that the peaks are quite equal, because the specialisation effect within a single task now does not affect the processes of the next task (see Figure 11c).

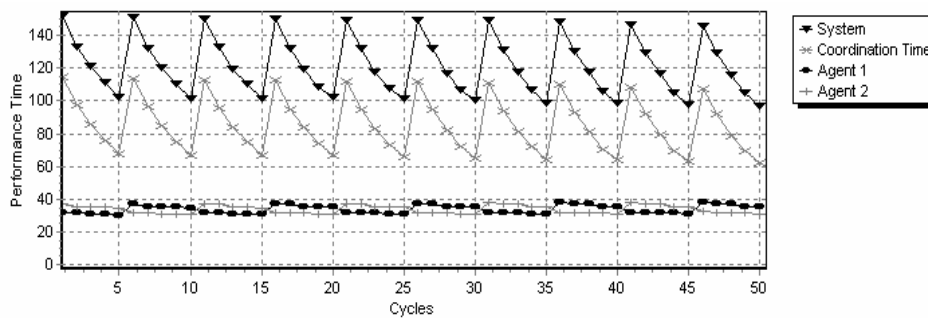


Figure 11c: Performance with high task variety

3.4.2.2 Conclusions

In accordance with our expectations, the results show that in a setting with task variety the coordination time increases straight after the change in the task and decreases until the next change. The decrease is caused by the specialisation effect as described in the first experiment. The increase results from the time it takes to allocate the new actions. Furthermore, we see that in conditions with low and moderate task variety, there is some overlap between old actions and new actions, resulting in an erratic performance time. In the condition where there is a high task variety with no overlap, every change implies a task that is completely new, generating a performance time that is quite regular.

3.4.3 Experiment 3: boredom

Boredom implies a decrease in motivation, which leads to a decrease in task performance. Furthermore, a change in motivation may lead to a change in the initial choice, which could lead to a change in the allocation of the task. We tested three conditions, a high degree of boredom and recovery (100), a moderate degree of boredom and recovery (50), and a low degree of boredom and recovery (10). We tested the effects of boredom under the following constraints:

The agents have to perform tasks consisting of 100 cycles of 2 actions

Every agent has 2 skills

The learning speed and the forget speed are 0, i.e. no learning and forgetting

In the first setting, the initial values of the agents were set as follows:

Table 3: initial values of agents 1 and 2

	Agent 1	Agent 2
Expertise skill 1	<i>17</i>	<i>13</i>
Motivation skill 1	<i>17</i>	<i>13</i>
Expertise skill 2	<i>13</i>	<i>17</i>
Motivation skill 2	<i>13</i>	<i>17</i>

3.4.3.1 Results

Figure 12a shows the results of the first condition. We see a curve consisting of two phases. The first phase indicates the first 18 cycles. We see an increase in the coordination time because of the decrease in the motivation of both agents. The second phase starts during the 19th cycle, where the coordination time suddenly drops. Apparently the agents switched actions! After the switch, the coordination time increases again because of the motivational decrease of both agents and then the agents switch again. This continuous switching of actions demonstrates that a job rotation schedule emerges under this condition.

Self-Organising Processes of Task Allocation

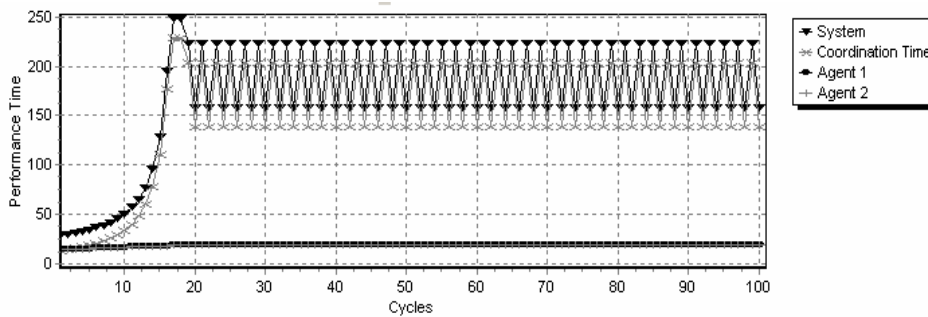


Figure 12a: boredom and action rotation: $boredom/recovery=100$

Furthermore, in Figure 12b we see that, apart from the decrease in the first phase, the motivation of the agents does not further decrease in the second phase.

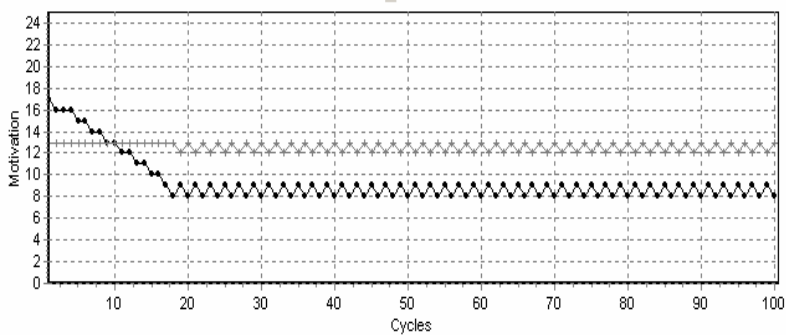


Figure 12b: motivation of agent 1

We only depicted the motivation of agent 1 because his motivation for skills 1 and 2 is equal to the motivation of agent 2 for skills 2 and 1.

The next condition, moderate boredom and recovery, shows that the first phase now ends in the 36th cycle. Because of the lower boredom rate, it takes longer before the agents reach a state in which they start switching actions. Furthermore, we see a fluctuation of the coordination time, indicating the same rotation process as in the former condition. We see that the fluctuations are somewhat smaller, i.e. between 190 and 240 instead of 150 and 240 as in the first condition. This can be explained by the lower boredom and recovery rates: because the processes of boredom and recovery are much slower, after the switch the initial choices of the agents remain more similar, resulting in higher coordination times.

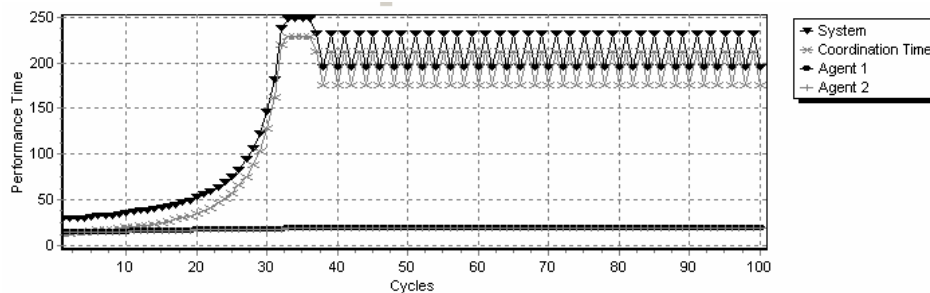


Figure 12c: boredom and action rotation: boredom/recovery=50

The third condition, a boredom/recovery rate of 10, indicates that the first phase shows an increase in the interaction time which is so small that the second phase of switching actions does not even occur in a task consisting of 100 cycles.

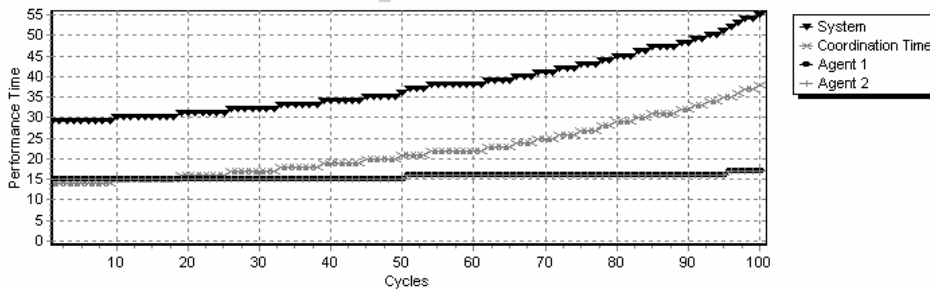


Figure 12d: boredom and action rotation: boredom/recovery=10

3.4.3.2 Conclusions

The results show that boredom affects the performance time in two subsequent phases. In the first phase, we see an increase in the coordination time because of the decrease in the motivation of both agents. When the motivational decrease causes the initial choice to change, the second phase emerges, in which the agents rotate their actions. When we lower the boredom and recovery rates, it takes more time before the second phase starts because a lower degree of boredom implies less motivational decrease, which diminishes the effect on the change in the initial choice.

3.5 Discussion

Not only do the results show that WORKMATE I generates dynamics in accordance with the expectations as derived from the conceptual model, they also reveal interesting behaviour, such as the emergence of job rotation. This type of behaviour is particularly interesting because it can be observed in daily life (Van den Beukel, 2003). This implies that WORKMATE I is able to produce emergent behaviour that shows a strong resemblance with daily-life phenomena. This resemblance does not apply to all the

behaviour which the results show. For instance, the negative correlation between coordination time and agent differences is plausible, but an increase in coordination time as part of the rotation process every time the action shifts is not. People invent standard rules to shift actions, which the agents in WORKMATE I cannot do. Furthermore, the model describes the task allocation of two agents. A setting consisting of more agents would be more realistic and could result in processes that are far more complex than the ones we have described here so far. Moreover, the allocation process we described is only based on expertise and motivation, which are individual components. It does not involve any social components. The relation between task duration and motivation is not involved either although this relation might influence the allocation process (see also Csickszentmihalyi, 1992). Therefore, an important subject for further research lies in the plausibility and the realism of the model (Zoethout et al., 2004). This does not only refer to the agents and their interaction but applies to the task as well. This study merely concerns dynamic complexity, but as the examples of the apple orchard suggest, interdependence is also an important task characteristic (Wood, 1986; Thompson, 1967). Although many scholars are in favour of the KISS-principle: Keep It Simple, Stupid, the principle of EROS: “Enhanced Realism Of Simulation”, as Conte stated in 1997 at the First International Conference on Computer Simulation and the Social Sciences (ICCS&SS), may well be used as long as we add complexity in a stepwise manner and only after a full understanding of the dynamics of the simpler model (Harrison, 2002). Nevertheless, we should be aware of what is called Bonini’s paradox: the more realistic and detailed one’s model, the more the model resembles the modelled organisation, including resemblance in the directions of incomprehensibility and indescribability (Starbuck, 1976, cited in Weick, 1979). Another subject for further research could be the relation of simulation experiments with empirical data. Although simulation may be the best method of studying self-organising processes of task allocation, we still wish to validate the results by comparing them with daily-life situations.

WORKMATE 1 may be useful in different scientific areas. Within the area of management science, it serves as a tool for studying self-organising social processes of task allocation. Moreover, computer simulation is a common method within the area of Operations Management (Law & Kelton, 2000). Within this area, physical infrastructures are modelled, but psychological theories are less emphasised. Applying WORKMATE 1 results in interesting outcomes regarding the simulation of production processes with agents, which are psychologically more realistic. Within the area of social sciences, the theoretical framework may contribute to the integration of the amount of fragmented theories and models (Jager, 2000; see also Vallacher & Nowak, 1994).

The field of computational social and organisational science is growing rapidly (Carley, 1998). Applications arising from this field lead to new perspectives and new approaches, such as for example, complexity theory. These approaches and applications will find their way into science and society, generating new ways of thinking as well as new combinations of existing views.