AN INTRODUCTION TO PARADIGM

SASKIA C. VAN DER MADE-POTUIJT AND ARIE DE BRUIN

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2002-09-LIS
Publication	January 2002
Number of pages	19
Email address corresponding author	vandermade@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Faculteit Bedrijfskunde Erasmus Universiteit Rotterdam P.O. Box 1738 3000 DR Rotterdam, The Netherlands Phone: +31 10 408 1182 Fax: +31 10 408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

ERASMUS **R**ESEARCH **I**NSTITUTE OF **M**ANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

BIBLIOGRAPHIC DATA AND CLASSIFICATIONS		
Abstract	By using Paradigm, it is possible to model cooperating processes and to make the communication between these processes very clear. This report gives a formal description of this modeling method using state-transition diagrams in order to model processes and homomorphismes and interleavings in order to model the cooperation and synchronization of the processes involved. Paradigm has been used succesfully as the modeling language of Socca, a software proces modeling method.	
Library of Congress Classification (LCC)	5001-6182	Business
	5201-5982	Business Science
	HF 5691	Business Mathematics
Journal of Economic Literature (JEL)	M	Business Administration and Business Economics
	M 11 R 4	Production Management Transportation Systems
	C 88	Computer Programs, Other Computer Software
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K 240 B	Logistics Information Systems Management
	240 S	Computer programming
Gemeenschappelijke Onderwerpsontsluiting (GOO)		
Classification GOO	85.00	Bedrijfskunde, Organisatiekunde: algemeen
	85.34 85.20	Logistiek management Bestuurlijke informatie, informatieverzorging
	54.10	Theoretische informatica
Keywords GOO	Bedrijfskunde / Bedrijfseconomie	
	Bedrijfsprocessen, logistiek, management informatiesystemen	
	Coordinatie, processen, parallelle gegevensverwerking, programmeertechnieken, modellen (vorm)	
Free keywords	coordination, communication, modeling, processes, state-transition diagrams	

An Introduction to Paradigm

Saskia C. van der Made-Potuijt
Erasmus University Rotterdam
vandermade@few.eur.nl

Arie de Bruin
Erasmus University Rotterdam
adebruin@few.eur.nl

15 Januari 2002

Abstract

By using Paradigm, it is possible to model cooperating processes and to make the communication between these processes very clear. This report gives a formal description of this modeling method using state-transition diagrams in order to model processes and homomorphism's and interleavings in order to model the cooperation and synchronization of the processes involved. Paradigm has been used successfully as the modeling language of Socca, a software process modeling method.

1 Introduction

In this report the modeling method for parallel behaviour, Paradigm, will be introduced. The name is an abbreviation for PARallelism, its Analysis, Design and Implementation by a General Method. The Paradigm method has been developed at the Leiden University, Department of Computer Science by Dr. Luuk Groenewegen in the period 1986-1991 [Groe]. In this original description of Groenewegen the method uses parallel decision processes for modeling parallel phenomena. The description we present is simplified and is based on STD's .

By using Paradigm, we are able to model cooperating processes and the synchronization of these processes. The processes in the model will be modeled by means of STD's. Some of the most important extra notions in Paradigm are subprocess, traps and employee process. We use homomorphisms and interleavings in order to synchronize the processes. The main advantage of Paradigm is that it has the notion of subprocess, so we do not have to synchronize on all the states, only on subsets of the states.

We will introduce the concepts of Paradigm by defining them, by discussing them very briefly and by illustrating them by means of a simple example.

2 State Transition Diagrams and Behaviour

We define an STD's as a 5-tuple.

Definition 1 *A* State Transition Diagram (STD) *is a 5-tuple* $\mathcal{S} = (S, A, T, \sigma, \tau)$*. Here S is called the set of states, or the state space; A is called the set of actions; $T \subset S \times A \times S$ is the set of transitions. $\sigma \subset S$ is the set of starting states and $\tau \subset S$ is the set of final states.*

We will follow the usual graphical representation of an STD as a directed graph where the nodes visualize the states and where the directed edges visualize the transitions. The direction of an edge is from the state being the first coordinate of the transition towards the state being the transition's third coordinate, and the labels of the edges are the actions, being the transition's second coordinate. Although it is essential that this is a relation we will use a functional notation: we will write $T(s,a)= s'$ instead of $(s,a,s') \in T$. As it may happen, σ and τ can be chosen such that $\sigma = \tau = S$ implying that each state may occur as starting state and also as final state.

A part of the real world that we want to model is called a *process*. A process can be modeled by means of an STD, assuming that the behaviour of the process in the course of time closely resembles the behaviour of the STD. To that aim the behaviour of an STD is defined as follows.

Definition 2 *Given an STD* $\mathcal{S}=(S,A,T,\sigma,\tau)$ *and a starting state* $s_0 \in \sigma$, *a* behaviour of $\mathcal{S}$, starting in s_0 *is a sequence* $s_0, a_0, s_1, a_1, s_2, a_2, \ldots$ s_{n-1}, a_{n-1}, s_n, *such that* $s_i \in S$, $a_i \in A$ *and* $(s_i, a_i, s_{i+1}) \in T$ *with* $0 \leq i \leq n-1$. *In addition, any infinite sequence* $s_0, a_0, s_1, a_1, \ldots$ *such that* $s_i \in S$, $a_i \in A$ *and* $(s_i, a_i, s_{i+1}) \in T$ *with* $i \in N_o$ *is called a* behaviour of $\mathcal{S}$ starting in s_0 *too. A finite behaviour* $s_0, a_0, s_1, \ldots s_n$ *is called* properly terminating *if* $s_n \in \tau$.

When the STD and the starting state are clear from the context, we just speak about a behaviour. For a fixed starting state, an STD actually represents the set of the behaviours starting in this starting state. In fact an STD represents a set of behaviours: the union of the sets of behaviours starting in one of the starting states belonging to σ.

We assume that a transition of a process takes a certain amount of time. Furthermore, a process is in one state for a certain period. Therefore a state of an STD, can be described as a transition from this state to itself, but we will not represent such a transition explicitly. The reason for assuming that a process is in a state for a certain period is that there is some activity in a state. However we assume that this activity will not involve activity as a result of communication with other processes. These types of activities can only be modeled by means of transitions.

The definitions will be illustrated with the following example. We will describe a meeting of n people with one chairman. The participating people will be called *Member*, the chairman will be called *Chairman*. The behaviours of these people are represented by the graphs in Figure 1.

The Member process can be modelled as an STD $\mathcal{S}=(S,A,T,\sigma,\tau)$. For each Member the set of states, S, consists of the two states 1 and 2.

1 The Member is not speaking.

2 The Member is speaking.

There are two actions for each Member, the Member starts speaking and the Member stops speaking. There are also two transitions for each Member: $1 \rightarrow 2$ and $2 \rightarrow 1$.

1→2 The Member starts speaking.

2→1 The Member stops speaking.

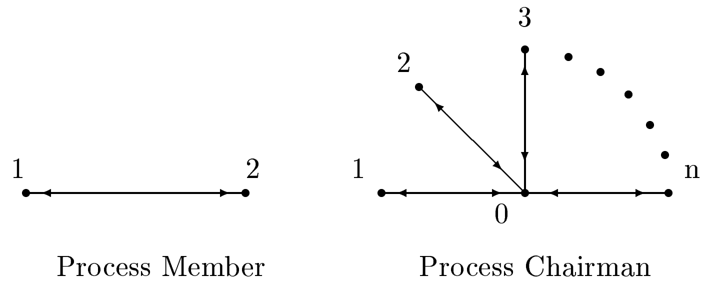

Figure 1: A Meeting

Actions can be considered as the labels of the transitions. There is one starting state, state 1 in which the Member is not allowed to speak and this is also the final state. The behaviour of a Member can be described by a sequence like 1, 1→2, 2, 2→1, 1, 1→2, 2 ... ending in 1.

The Chairman can be modelled as an STD $\mathcal{S}=(S, A, T, \sigma, \tau)$. For the Chairman the set of states, S, consists of $n+1$ states: 0, 1 ... n.

0 The Chairman is allowing no one to speak.

i The Chairman is allowing Member(i) to speak.

Between the states there are $2n$ transitions.

0→i The Chairman gives Member(i) permission to speak.

i→0 The Chairman withdraws Member(i)'s right to speak.

There are as many transitions as there are actions. There is one starting state, state 0 in which the Chairman does not allow any Member to speak and this is also the final state.

The behaviour of the Chairman can be described by a sequence like 0, 0→2, 2, 2→0, 0, 0→5, 5, 5→0, 0, 0→1, 1, ... ending in 0.

3 Subprocesses and traps

Dependencies between processes imply that processes communicate with each other. In general when a process P^1 is dependent on the behaviour of another process P^2 and no communication has yet taken place, P^1 will be restricted in its behaviour from a certain point on, until the communication has taken place. In our example the behaviour of a Member will be restricted by the behaviour of the Chairman, e.g. in the sense that a Member can only take the transition nonspeaking → speaking if the Chairman allows him/her to do so. In Paradigm the restrictions of behaviour are modelled by means of *subprocesses* and *traps*.

Definition 3 *A* subprocess *of an STD* $\mathcal{S}=(S, A, T, \sigma, \tau)$ *is an STD* $\mathcal{S}'=(S', A', T', \sigma', \tau')$ *such that* $S' \subset S$*,* $A' \subset A$ *and* $T' \subset T$*.*

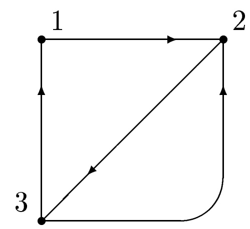

Figure 2: An Active Member.

So a subprocess is a restriction of an STD to a, usually smaller, set of states and to a usually smaller set of actions. Notice that from definition 1.1 we can infer that $T' \subset T \cap (S' \times A' \times S')$.

It is the task of a modeller to choose a subprocess such that it represents the relevant restriction on the behaviours of an STD, as it is imposed by some other process.

The definition of subprocess will be illustrated with an extension of the example of the Meeting. This second model features a refinement of the Member process from the previous situation, we do not have to extent the model of the Chairman yet. The new process will be called the Active Member process.

For an Active Member the set of states, S, consists of three states

1 The Active Member is neither speaking nor expressing a wish to speak.

2 The Active Member is raising his/her finger, indicating he/she wants to say something.

3 The Active Member is speaking.

There are four actions and these actions correspond with four transitions:

1→2 The Active Member raises his/her finger.

2→3 The Active Member starts speaking.

3→2 The Active Member renounces from speaking (being interrupted by the Chairman), but he or she immediately raises his/her finger.

3→1 The Active Member stops speaking, either on his/her own free will or being interrupted by the Chairman and does not raises his/her finger again.

There is one starting state, state 1 and this is also the final state. The process is reflected Figure 2.

The Active Member Process can be divided for instance into three subprocesses, I, II and III that are represented by their corresponding graphs in Figure 3. The rectangles in the graphs are *traps*, this notion will be defined in the next definition.

I The Active Member has no permission to speak.

II The Active Member has permission to start speaking and to speak. Furthermore an Active Member that is allowed to speak is also allowed to stop speaking by him/herself

III The Active Member's permission to speak is withdrawn. If the Active Member still wants to say something and therefore raises his/her finger the Active Member enters state 2 takes, otherwise he/she ends in state 1.

After the Chairman has given an Active Member the right to speak, the Active Member is in state 3 of subprocess II, the Chairman decides whether to allow this Active Member to stop speaking by him/herself or to withdraw the right to speak from this Active Member. In the first case the Active Member will enter subprocess state 1 of subprocess III, in the second case he/she enters state 2 of subprocess III.

It is easily seen that a subprocess actually is a restriction of the original process' full behaviour. In Paradigm we can model processes in which such a behaviour restriction is (temporarely) prescribed by some other process.

The next Paradigm notion to discuss is the trap.

Definition 4 *A* trap *of a subprocess* $\mathcal{S}'{=}(S', A', T', \sigma', \tau')$ *is a nonempty set of states* $D \subset S'$ *such that* $(z, a, s) \in T'$ *for* $z \in D$ *implies* $s \in D$*. If two different traps* D *and* E *of the same subprocess have the property* $D \subset E$ *these traps are called* nested*;* D *is called the* inner *trap of the two, and* E *is called the* outer *trap of the two. If* $D = S'$ *then the trap is called* trivial*, otherwise the trap is called* nontrivial

So a trap is a part of a subprocess' state space that cannot be left given the behavioural restriction of this subprocess. Therefore entering a trap serves as a criterion for marking an irrevocable step within the subprocess' behaviour. By entering a trap a subprocess expresses the whish to enter another subprocess.

It is the task of a modeller to choose a trap such that entering it marks a relevant point of no return in subprocess' behaviour, to be indicated by some message sent by this subprocess. In fact the message will be sent by the STD of which this subprocess is a restriction. Note that the point of no return is relative to the subprocess only; as soon as the STD starts behaving according to the restriction of a different subprocess, the trap can be left according to the new behaviour restriction.

In our example we have chosen the trap of subprocess I is the set {2}. In this trap the Active Member has raised his/her finger, marking its readiness to start speaking. The set {1,3} is a trap of subprocess II, as in these states the Active Member is allowed speak, to stop speaking by him/herself and to have stopped speaking. The set {1} is an innertrap trap of subprocess II, as in this state the Active Member has stopped speaking and will not be allowed to speak again unless he/she raises his/her finger again. The sets {1} and {2} are the traps of subprocess III, the meaning of these traps have been described already as

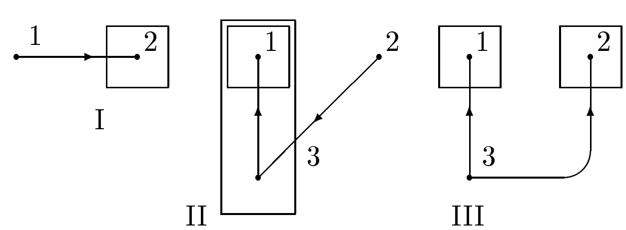

Figure 3: Subprocesses of an Active Member

traps of the subprocesses I and II. It is also possible that a subprocess has more than one state in a trap but this is not the case in our example.

Definition 5 *Let* $\mathcal{S}^i$=$(S^i, A^i, T^i, \sigma^i, \tau^i)$ *and* $\mathcal{S}^j$=$(S^j, A^j, T^j, \sigma^j, \tau^j)$ *be two subprocesses of the same STD. A trap D of* $\mathcal{S}^i$ *is called a* connecting trap *from* $\mathcal{S}^i$ *to* $\mathcal{S}^j$ *if* $D \subset S^i \cap S^j$*. Such a connecting trap D will often be written as a triple (*$\mathcal{S}^i$*, D,* $\mathcal{S}^j$*).* $\mathcal{S}^i$ *is called the* source of the connecting trap D, $\mathcal{S}^j$ *is called the* destination of the connecting trap D.

In our example we have the following connecting traps

(I,{2},II), (II,{1},I), (II,{1,3},III), (III,{1},I), (III,{2},II).

4 Employee interfaces

Definition 6 *Let* $\mathcal{S}$=(S, A, T, σ, τ) *be an STD and let* $\mathcal{P}$= $\{\mathcal{S}^i\}$ *be a set of subprocesses of* $\mathcal{S}$*, with* $\mathcal{S}^i = (S^i, A^i, T^i, \sigma^i, \tau^i)$*,* $1 \leq i \leq n$*,* $n \in N$ *then the set* $\mathcal{P}$ *is called a* partition of $\mathcal{S}$ *if it satisfies the following properties:*

1. *each state from S occurs in at least one* S^i*;*
2. *each action from A occurs in at least one* A^i*,*
3. *each transition from T occurs in at least one* T^i*.*

Notice that the third property implies the second. In our example is $\mathcal{P}$={I, II, III} a partition of the process Active Member.

Definition 7 *Let* $\mathcal{S}$=(S, A, T, σ, τ) *be an STD with partition* $\mathcal{P} = \{\mathcal{S}^i\}$*. A set* $\mathcal{C}$ *is called a* trapstructure of $\mathcal{S}$ and $\mathcal{P}$ *iff each element of* $\mathcal{C}$ *is of the form (*$\mathcal{S}^i$*, D,* $\mathcal{S}^j$*) where* $\mathcal{S}^i$*,* $\mathcal{S}^j \in \mathcal{P}$ *and D is a connecting trap (*$\mathcal{S}^i$*, D,* $\mathcal{S}^j$*) from* $\mathcal{S}^i$ *to* $\mathcal{S}^j$*.*

A trapstructure of the Active Member process and partition $\mathcal{P}$ is the collection of connecting traps as we gave it in the previous paragraph.

$\mathcal{C}$= (I,{2},II), (II,{1},I), (II,{1,3},III), (III,{1},I), (III,{2},II).

In this case, the trapstructure is the same as the collection connecting traps, described in the previous paragraph. A trapstructure is always associated with a partition, a collection connecting traps not necessarily. Notice that a subset of a trapstructure is also a trapstructure.

Definition 8 *Let* $\mathcal{S}$=(S, A, T, σ, τ) *be an STD with partition* $\mathcal{P}$=$\{\mathcal{S}^i\}$ *and trapstructure* $\mathcal{C}$= *{(*$\mathcal{S}^i$*, D,* $\mathcal{S}^j$*)}. The* employee interface $\mathcal{E}$ *associated with (*$\mathcal{S}$*,* $\mathcal{P}$*,* $\mathcal{C}$*) is the STD* $\mathcal{S}'$= $(S', A', T', \sigma', \tau')$ *with* S' *being the set of subprocesses belonging to the partition* $\mathcal{P}$*, with* A' *being the set of traps D belonging to the trapstructure* $\mathcal{C}$ *and* T'= $\mathcal{C}$*. The set of starting states* σ' *is a set of subprocesses* $\mathcal{S}^i$ *that contain at least one starting state of* $\mathcal{S}$*. The set of final states* τ' *is a set of subprocesses* $\mathcal{S}^i$ *that contain at least one final state of* $\mathcal{S}$*. The process* $\mathcal{S}$=(S, A, T, σ, τ) *to wich a employee interface is associated is called the corresponding* employee process.

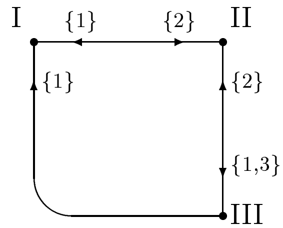

Figure 4: The employee interface of an Active Member

We will illustrate this concept also with our example of an Active Member. Let $\mathcal{S}$=(S, A, T, σ, τ) be the process of the Active Member with partition $\mathcal{P}$={I, II, III} and trapstructure $\mathcal{C}$ as described above. The *employee interface* $\mathcal{E}$ associated with ($\mathcal{S}$, $\mathcal{P}$, $\mathcal{C}$) is the STD $\mathcal{S}'$= $(S', A', T', \sigma', \tau')$ with S' being the set of subprocesses belonging to the partition, so {I, II, III} and with A' being the set of traps D belonging to the trapstructure $\mathcal{C}$ and T'= $\mathcal{C}$. We choose as the collection of starting states {I}, this is a subprocess that contains the only starting state of the process Active Member. As the set of final states we also choose the set {I}.

Figure 4 shows the employee interface associated with the (Active Member, $\mathcal{P}$, $\mathcal{C}$). It shows clearly the transitions that are possible between the subprocesses of an Active Member. All the transitions are labelled with the corresponding connecting trap belonging to the trapstructure $\mathcal{C}$.

Definition 9 *Let* $\mathcal{S}$=(S, A, T, σ, τ) *be an STD with partition* $\mathcal{P}$=$\{\mathcal{S}^i\}$ *and trapstructure* $\mathcal{C}$= $\{(\mathcal{S}^i,\ D,\ \mathcal{S}^j)\}$. *Let* $\mathcal{E}$ *be the employee interface associated with (*$\mathcal{S}$, $\mathcal{P}$, $\mathcal{C}$*). Then a* global behaviour *of the STD* $\mathcal{S}$ *is a behaviour of the associated employee interface* $\mathcal{E}$

So the global behaviour of an Active Member is a sequence $\mathcal{S}^0$, D^0, $\mathcal{S}^1$, D^1, $\mathcal{S}^2$, D^2 ... where for $i \in N_o$ each $\mathcal{S}^i$ is a subprocess of the $\mathcal{S}$, where each D^i is a trap from $\mathcal{S}^i$ to $\mathcal{S}^{i+1}$, and where the state space of $\mathcal{S}^0$ contains at least one starting state from σ.

A global behaviour serves as a view on a STD's behaviour from outside that STD; the view is taken, as it were, from a viewpoint this STD is communicating with, at least as far as the communication is concerned with respect to the STD's subprocesses and traps.

From Figure 4 we can construct three different and relevant subsequences of such global behaviour for an Active Member:

1. I, {2}, II, {1}, I
2. I, {2}, II, {1, 3}, III, {1}, I
3. I, {2}, II, {1, 3}, III, {2}, II, {1, 3}, III, ..

The first global behaviour has the following interpretation. An Active Member starts in subprocess I in which he/she is not allowed to speak. Subprocess II can be entered after the Active Member has entered trap {2} of subprocess I, in which he/she has raised his/her finger. In subprocess II the Active Member is allowed to speak and in trap {1,3} the Active Member has permission to speak or to stop speaking. If the Active Member then has finished speaking, he/she has entered the inner trap {1}.

In the second global behaviour an Active Member starts of course also in subprocess I and enters subprocess II after the Active Member has entered trap {2} of subprocess I. If in subprocess II the Active Member is in state 3 of trap {1,3}, the permission to speak can be withdrawn although the Active Member has not finished speaking. This is indicated by the transition to subprocess III. If the Active Member has entered trap {1} in subprocess III which indicates the Active Member does not want to continue to speak, subprocess I will be entered.

The beginning of the third global behaviour is the same as the behaviour of the second on. If in subprocess III the Active Member again raises his/her finger, reflected by entering trap {2} in subprocess III, a transition from subprocess III to subprocess II can take place. After this the Active Member can enter III again and after subprocess III subprocess II can be described again. So in the third global behaviour the subprocesses II and III can alternate for a certain period, and this will end in subprocess I.

From the description it is clear, that another entity is involved that restricts the behaviour of the Active Member. This entity, that can give permission to speak or to withdraw this permission will be a type of chairman. Later we will give a precise description of this chairman process. This will not only make clear how this process controls the behaviour of one Active Member, but also, how it controls the behaviour of several Active Members in one meeting.

5 Homomorphisms and employment complexes

Definition 10 *Let $\mathcal{M}$ be an STD $(S_1, A_1, T_1, \sigma_1, \tau_1)$ and let $\mathcal{N}$ be an STD $(S_2, A_2, T_2, \sigma_2, \tau_2)$. A* homomorphism *from $\mathcal{M}$ to $\mathcal{N}$ is a combination of two mappings*

1. ϕ^s: $S_1 \rightarrow S_2$,
2. ϕ^a: $A_1 \rightarrow A_2$,

such that

1. $\phi^s\ (\sigma_1) \subset \sigma_2$,
2. $\phi^s\ (\tau_1) \subset \tau_2$,
3. *all mappings are total on their domains and*
4. *if $(x, a, y) \in T_1$, then $(\phi^s(x),\ \phi^a(a),\ \phi^s(y)) \in T_2$.*

ϕ is denoted as (ϕ^s, ϕ^a).

In order to illustrate a homomorphism we need to have two STD's. In our example we have an Active Member process and we will now present a process for a related chairman. We will call this process Chairman for One Active Member, abbreviated to COAM. Later we will extend this model to a Chairman for n Active Members.

The process Chairman for One Active Member is reflected in Figure 10. A precise description of this process is given by means of the following states:

A The COAM is not allowing the Active Member to speak.

B The COAM is allowing the Active Member to start speaking.

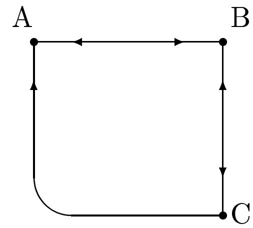

Figure 5: Chairman for One Active Member (COAM)

C The COAM has withdrawn the right to speak from the Active Member.

There are five actions and five transitions:

A $\rightarrow$ **B** The COAM gives the Active Member permission to speak.

B $\rightarrow$ **A** The COAM withdraws the right to speak from the Active Member, he/she was ready.

B $\rightarrow$ **C** The COAM is withdraws the permission to speak from the Active Member.

C $\rightarrow$ **A** The COAM has withdrawn the right to speak from the Active Member and the Active Member has indicated he/she has finished speaking.

C $\rightarrow$ **B** The COAM allows the Active Member again to speak.

The only starting state of process COAM is A, this is also the final state

In order to illustrate the definition of homomorphism, we will give a homomorphisme from the COAM to the employee interface of the Active Member, so not directly to the Active Member. This will in general be the situation: a homomorphisme will be defined from a certain STD to the employee interface of another STD, this employee interface is of course also an STD.

Let $\mathcal{M}$ be the STD $(S_1, A_1, T_1, \sigma_1, \tau_1)$ that represents the COAM and let $\mathcal{N}$ be the *employee interface* $\mathcal{E}$ associated with ($\mathcal{S}$, $\mathcal{P}$, $\mathcal{C}$), where $\mathcal{S}$ is the STD that represents the process Active Member, $\mathcal{P}$={I, II, III}, the partition of the Active Member as we defined it in the previous paragraph and let the trapstructure also be as we defined it there,
$\mathcal{C}$=(I,{2},II), (II,{1},I), (II,{1,3},III), (III,{1},I), (III,{2},II).

As we explained, the employee interface associated with ($\mathcal{S}$, $\mathcal{P}$, $\mathcal{C}$) is the STD $\mathcal{S}'$= $(S', A', T', \sigma', \tau')$ with S' being the set of subprocesses belonging to the partition $\mathcal{P}$ and with A' being the set of traps D belonging to the trapstructure $\mathcal{C}$ and T'= $\mathcal{C}$. The set of starting states σ' is {I}, which is also the only final state. The homomorphism from $\mathcal{M}$ to $\mathcal{N}$ is the combination of two mappings

1. ϕ^s: $S_1 \rightarrow S'$, being

   ϕ^s (A)= I
   ϕ^s (B)= II
   ϕ^s (C)= III

2. ϕ^a: $A_1 \rightarrow A'$, being

ϕ^a(A $\rightarrow$ B)={2}

ϕ^a(B $\rightarrow$ A)={1}

ϕ^a(B $\rightarrow$ C)={1,3}

ϕ^a(C $\rightarrow$ A)={1}

ϕ^a(C $\rightarrow$ B)={2}

Notice that

1. ϕ^s $(\sigma_1) \subset \sigma_2$ as $\phi^s(\sigma_1)=$ ϕ^s(A) $=$ I $\subset \sigma_2$
2. ϕ^s $(\tau_1) \subset \tau_2$, as $\phi^s(\tau_1)=\phi^s$(A)=I $\subset \tau_2$.
3. all mappings are total on their domains and
4. if $(x, a, y) \in T_1$, then $(\phi^s(x),\ \phi^a(a),\ \phi^s(y)) \in T_2$.

This last point means that if (A, A$\rightarrow$B, B)$\in T_1$, then (I,{2},II) $\in T_2$.

A homomorphism can be expressed graphically by a so called *labelling function* that labels each state and transition of the manager process with the values of the homomorphism. To this aim the graph of the left part of the homomorphism is extended with the images of the homomorphism: the images of the states are the indicated subprocesses of the associated employee process and the images of the actions are the traps of the trapstructure. The homomorphism from the COAM to the employee interface of the Active Member is reflected in Figure 6

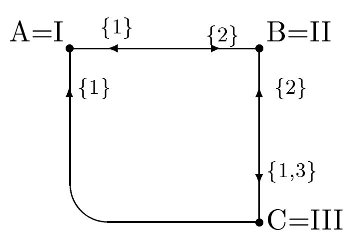

Figure 6: The homomorphisme from COAM to an Active Member

Definition 11 *Let* $\mathcal{S}$=(S, A, T, σ, τ) *be an STD with partition* $\mathcal{P}$={$\mathcal{S}^i$} *and trapstructure* $\mathcal{C}$= {($\mathcal{S}^i$,D, $\mathcal{S}^j$)}. *Let* $\mathcal{E}$ *be the employee interface associated with* $(\mathcal{S}, \mathcal{P}, \mathcal{C})$. *An STD* $\mathcal{M}$= (S, A, T, σ, τ) *is called a* Manager Process *for* $\mathcal{S}$ *if there exist a homomorphism* ϕ *from* $\mathcal{M}$ *to* $\mathcal{E}$.

So the Chairman for One Active Member is a manager process for the Active Member because there is a homomorphism from COAM to the employee interface associated with the Active Member, its partition $\mathcal{P}$ and trapstructure $\mathcal{C}$.

Definition 12 *Let* $\mathcal{S}$=(S, A, T, σ, τ) *be an STD with partition* $\mathcal{P}$ *and trapstructure* $\mathcal{C}$. *Let* $\mathcal{E}$ *be the employee interface associated with* ($\mathcal{S}$, $\mathcal{P}$, $\mathcal{C}$) *and let* $\mathcal{M}$ *be a manager process for* $\mathcal{S}$. *Then the quadruple* $\mathcal{Q}$=($\mathcal{S}$, $\mathcal{E}$, $\mathcal{M}$, ϕ) *is called an* Employment Complex

The quadruple $\mathcal{Q}$=($\mathcal{S}$, $\mathcal{E}$, $\mathcal{M}$, ϕ) with $\mathcal{S}$ being the STD that represents the Active Member, $\mathcal{P}$={I, II, III} the partition of the Active Member as we defined it in the previous paragraph, let $\mathcal{C}$ be also as we defined it there and $\mathcal{E}$ its employee interface associated with ($\mathcal{S}$, $\mathcal{P}$, $\mathcal{C}$). Let $\mathcal{M}$ represent the COAM process, and ϕ defined as above, then quadruple $\mathcal{Q}$=($\mathcal{S}$, $\mathcal{E}$, $\mathcal{M}$, ϕ) is an example of an Employment Complex.

The next two definitions emphasize the strong dependency between the manager and its related employee process in an employment complex. An employee process can only take a transition if this transition is permitted by the manager in the state where this manager now is. Also a manager can only take actions if the related employee is in a state that permits this transition.

Definition 13 *Given an employment complex $\mathcal{Q}$= ($\mathcal{S}$, $\mathcal{E}$, $\mathcal{M}$, ϕ). A transition t=(s_i,a,s_{i+1}) of $\mathcal{S}$, is* permitted by $\mathcal{M}$ in state p *if the transition t=(s_i,a,s_{i+1}) is in subprocess $\phi^s(p)$.*

So a transition t=(s_i,a,s_{i+1})of an Active Member is permitted by the manager process COAM in state p if this transition is in subprocess $\phi^s(p)$. For example the transition $1 \rightarrow 2$ of the Active Member is permitted by COAM in state A as ϕ^A=I and the transition is in subprocess I. This transition is however not permitted by COAM in state B, as ϕ^B=II and transition $1 \rightarrow 2$ is not part of subprocess II. We will say that a manager *prescribes* subprocess S' in state A if ϕ^s(A)=S'. The behaviour of the employee is then restricted to this subprocess.

Definition 14 *Given an employment complex $\mathcal{Q}$=($\mathcal{S}$, $\mathcal{E}$, $\mathcal{M}$, ϕ). A transition t=(s_i,a,s_{i+1}) of $\mathcal{M}$ is* permitted by the employee in state s *if $s \in$ $\phi^a(a)$.*

So, a transition t=(s_i,a,s_{i+1}) of the COAM is permitted by the Active Member process in state s if s $\in \phi^a(a)$. For example the transition $A \rightarrow B$ of the COAM is permitted by the Active Member in state 2 if $2 \in \phi^a$. This is true because ϕ^a ={2}. This transition of the COAM is however not permitted by the Active Member if this active Member is not in this trap. An employee *permits* a transition t=(s_i,a,s_{i+1}) of the manager if the employee is in the trap $\phi^a(a)$.

In Paradigm we want to be able to make models in which one STD is the manager process of several other participating STD's. This is possible if there exist homomorphisms from this manager process to all these participating STD's. The manager process, together with the participating STD's it is a manager process for and the homomorphisms is called an Extended Employment Complex. Here is the precise definition.

Definition 15 *Let $\mathcal{S}_i$=$(S_i, A_i, T_i, \sigma_i, \tau_i)$ be a set of STD's, $1 \leq i \leq n$, $n \in N$. Let each STD $\mathcal{S}_i$ have a partition $\mathcal{P}_i$={${\mathcal{S}_i}^j$} and trapstructure $\mathcal{C}_i$= {(${\mathcal{S}_i}^j$,D_i, ${\mathcal{S}_i}^{j+1}$)}. Let $\mathcal{E}_i$ be the employee interface associated with ($\mathcal{S}_i$, $\mathcal{P}_i$, $\mathcal{C}_i$). An STD $\mathcal{M}$= (S, A, T, σ, τ) is called an* Extended Manager Process *iff it is a Manager Process for all $\mathcal{S}_i$. The $(3n+1)$-tuple $\mathcal{Q}$=($\mathcal{S}_1$, $\mathcal{S}_2$, ..., $\mathcal{S}_n$, $\mathcal{E}_1$, $\mathcal{E}_2$, ..., $\mathcal{E}_n$, $\mathcal{M}$, ϕ_1, ϕ_2,..., ϕ_n) is called an* Extended Employment Complex.

We will illustrate this concept again with the example of the meeting. Until now we had only one Active Member participating in the meeting, which is not a very realistic situation. Now we assume there are n Active Members $\mathcal{S}_i$=$(S_i, A_i, T_i, \sigma_i, \tau_i)$, $1 \leq i \leq n$, $n \in N$, all having the same STD's $\mathcal{S}_i$, having partition $\mathcal{P}_i$= {I_i, II_i, III_i} and trapstructure $\mathcal{C}_i$=(I_i,$\{2\}_i$,II_i), (II_i,$\{1\}_i$,I_i), (II_i,$\{1,3\}_i$,III_i), (III_i,$\{1\}_i$,I_i), (III_i,$\{2\}_i$,II_i).

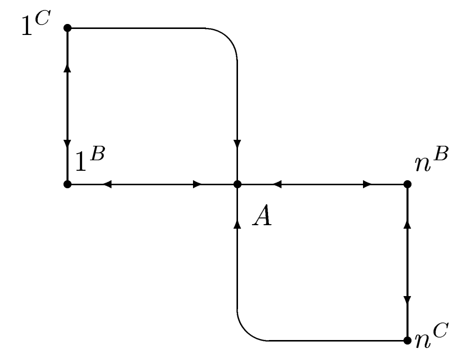

Figure 7: The Chairman for n Active Members

We will give a model for a chairman that is able to coordinate a meeting with several, say n, Active Members. This process will be called Chairman for n Active Members, abbreviated to CNAM

This CNAM process is shown in Figure 7. The left part represents the behaviour of the CNAM towards Active Member(1) and in the right part the behaviour of the CNAM towards Active Member(n) is represented. The state in the center, state A represents the situation were none of the Active Members are allowed to speak. This figure has to be interpreted as being extended with similar parts, each part representing the behaviour towards one Active Member. As the Chairman will only allow one Active Member at the time to speak, say Active Member(k), the CNAM will be in a state belonging to the part of the figure that corresponds to the Active Member(k). All the other Active Members will not be allowed to speak.

Figure 7 is an extension of figure 4 in the sense that it repeats it for all the n Active Members although not all the states are represented in the figure. There are 2 states for all the n Active Members and one central state, so $2n + 1$ different states.

A precise description of the Chairman for n Active Members is given by means of the following states and transitions:

A The CNAM is not allowing any Active Member(i) to speak, $1 \leq i \leq n$

i^B The CNAM is allowing Active Member(i) to start speaking.

i^C The CNAM has withdrawn the right to speak from Active Member(i).

The actions and transitions are:

$A \to i^B$ The CNAM gives only Active Member(i) the permission to speak.

$i^B \to A$ The CNAM withdraws the right to speak from the Active Member(i), he/she was ready.

$i^B \to i^C$ The CNAM withdraws the right to speak from the Active Member(i).

$i^C \to A$ The CNAM has withdrawn the right to speak from Active Member and this Active Member has indicated has indicated he/she has finished speaking.

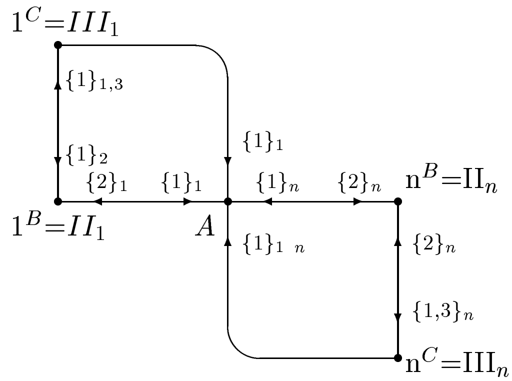

Figure 8: The homomorphisme from CNAM to the n Active Members

$i^C \rightarrow i^B$ The CNAM allows the Active Member(i) again to speak

The starting state of CNAM is A, this is also the final state. We can now define the n homomorphisms from CNAM to the n Active Members.

The homomorphism from CNAM to the n Active Members is the combination of two mappings

1. $\phi_1^s, \phi_2^s, \ldots, \phi_n^s$: $S \rightarrow S_1', S_2', \ldots, S_n'$

   $\phi_1^s, \phi_2^s, \ldots, \phi_n^s$ (A)= $\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_n$

   $\phi_1^s, \phi_2^s, \ldots, \phi_n^s$ (i^B)= $\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{II}_i, \ldots, \mathrm{I}_n$

   $\phi_1^s, \phi_2^s, \ldots, \phi_n^s$ (i^C)= $\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{III}_i, \ldots, \mathrm{I}_n$

2. ϕ^a: $A_1 \rightarrow A_1', A_2', \ldots, A_n'$ being

   $\phi_1^a, \phi_2^a, \ldots, \phi_n^a$ (A $\rightarrow i^B$)=$\{2\}_i$

   $\phi_1^a, \phi_2^a, \ldots, \phi_n^a$ ($i^B \rightarrow A$)=$\{1\}_i$

   $\phi_1^a, \phi_2^a, \ldots, \phi_n^a$ ($i^B \rightarrow i^C$)=$\{1,3\}_i$

   $\phi_1^a, \phi_2^a, \ldots, \phi_n^a$ ($i^C \rightarrow$ A)=$\{1\}_i$

   $\phi_1^a, \phi_2^a, \ldots, \phi_n^a$ ($i^C \rightarrow i^B$)=$\{2\}_i$

The labelling function for our model is reflected in Figure 8. This is an extension of the graph reflected in Figure 7 as now the states are labelled with the subprocesses that are described for the Active Members and the transitions are labelled with the relevant traps.

The interpretation of the labels of the states is as follows:

A The CNAM is not allowing any Active Member(i) to speak, $1 \leq i \leq n$ All Active Members are restricted in their behaviour to subprocess I.

i^B The CNAM is only allowing Active Member(i) to start speaking. For this Active Member subprocess II_i is prescribed, for the other Active Members subprocess I is still prescribed.

i^C The right to speak has been withdrawn from Active Member(i). Subprocess III_i is prescribed for this Active Member(i), subprocess I for the others.

The interpretation of the labels of the transitions is as follows:

$A \to i^B$ The CNAM gives only Active Member(i) the permission to speak. This transition can only be made if Active Member(i) is in trap $\{2\}_i$ of subprocess I_i.

$i^B \to A$ If the Active Member(i) is in trap $\{1\}_i$ of subprocess II_i, a transition to subprocess I_i is possible. The CNAM withdraws the right to speak from the Active Member(i), he/she was ready.

$i^B \to i^C$ The CNAM is withdraws the right to speak from the Active Member(i). This transition can only be made if the Active Member(i) is in the trap $\{1,3\}_i$ of subprocess II_i.

$i^C \to A$ After the Active Member(i) has entered trap $\{1\}_i$ in subprocess III_i, a transition of the Active Member(i) to subprocess I_i is possible. This indicates the Active Member has indicated he/she has finished speaking, the CNAM withdraws the right to speak from this Active Member.

$i^C \to i^B$ If the Active Member(i) is in trap $\{2\}_i$ of subprocess III_i, a transition to subprocess II_i is possible. The CNAM allows the Active Member(i) again to speak.

Although we only gave a description of the meaning of the traps in relation to Active Member(i), it will be clear how this can be extended to the other Active Members. We have given now an example of an Extended Employment Complex.

6 Transition/state mixes and interleavings

Definition 16 *A* transition/state mix α over a number of STD's $\mathcal{S}_1$,..., $\mathcal{S}_n$ starting in $s_{01},...,s_{0n}$ *is a sequence of transitions and states* $r_1, r_2, \ldots, r_k$ *where each* r_i *is a transition or a state in one of the* $\mathcal{S}_i$*'s,* $1 \le i \le n$, $n \in N$.

An example of a transition/state mix over the STD's An Active Member and the Chairman for One Active Member is $\alpha_1 = A$,(A→B),1,A,(2→3),B,(B→C), 3,(B→A),(2→1),2,(B→C)

Definition 17 *Let* $\alpha = r_1, r_2, \ldots, r_k$ *be a transition/state mix over the STD's* $\mathcal{S}_1, \ldots, \mathcal{S}_n$ *starting in* $s_{01},...,s_{0n}$. *A* projection of α over $\mathcal{S}_1$,..., $\mathcal{S}_n$ on $\mathcal{S}_i$, $\alpha \downarrow \mathcal{S}_i$ *is the part of the sequence that consists of all transitions and states from* $\mathcal{S}_i$ *and no transitions or states belonging to other STD's. Let* I *be a set of numbers* i *such that* $1 \le i \le n$, $n \in N$. *A* projection of α over $\mathcal{S}_1$,..., $\mathcal{S}_n$ on $\mathcal{S}_I$, $\alpha{\downarrow}\mathcal{S}_I$ *is the part of the sequence that consists of all transitions and states from* $\mathcal{S}_i$, $i \in I$ *and no transitions or states belonging to other STD's.*

The two projections of the transition/state mix α_1 are

1. $\alpha_1{\downarrow}\mathcal{S}_{AnActiveMember}$= 1,(2→3),3,(2→1)2
2. $\alpha_1{\downarrow}\mathcal{S}_{COAM}$= A,(A→B),A,B,(B→C),(B→A),(B→C)

Definition 18 *Transition/state mix* α *over STD's* $\mathcal{S}_1,...,\mathcal{S}_n$ *starting in* $s_1,...,s_n$ *is* a canonical transition/state mix *iff all projections* $\alpha \downarrow \mathcal{S}_i$ *are behaviour of STD* $\mathcal{S}_i$, $1 \le i \le n$, $n \in N$.

The above given transition/state mix α_1 is not a canonical transition mix, even both projections do not represent behaviour. The next transition mix,
α_2=A,(A$\rightarrow$B),1,(1$\rightarrow$2),B,2,(2$\rightarrow$3),3,(3$\rightarrow$2), (B$\rightarrow$C),C,(C$\rightarrow$B),2,(2$\rightarrow$1)is canonical as its two projections are behaviours of the corresponding STD's.

1. $\alpha_2{\downarrow}\mathcal{S}_{\ AnActiveMember}$= 1,(1$\rightarrow$2),2,(2$\rightarrow$3),3,(3$\rightarrow$2),2,(2$\rightarrow$1)
2. $\alpha_2{\downarrow}\mathcal{S}_{\ COAM}$=A,(A$\rightarrow$B),B,(B$\rightarrow$C),C,(C$\rightarrow$B)

Definition 19 *Let* α= $r_1, \ldots, r_k$ *be a canonical transition/state mix over the STD's* $\mathcal{S}_1, ..., \mathcal{S}_n$ *starting in* $s_{01}, ..., s_{0n}$. *Suppose* r_i *is a transition, being* t_i *The* state of $\mathcal{S}_j$ before the execution of transition t_i *is defined as follows in an inductive way:*

1. *the state of* $\mathcal{S}_j$ *before the execution of* t_1 *is* s_{0j}
2. *the state of* $\mathcal{S}_j$ *before the execution of* t_k *is (* $k > 1$*)*
   (a) *the state of* $\mathcal{S}_j$ *before the execution of* t_k *if* r_{k-1} *is a state of* $\mathcal{S}_j$
   (b) s' *if* r_{k-1}*is a transition* (s', a, s) *belonging to* $\mathcal{S}_j$
   (c) *the state of* $\mathcal{S}_j$ *before* r_{k-1} *if* r_{k-1} *is not a transition or state of* $\mathcal{S}_j$

So in transition mix α_2

1. the state of COAM before the execution of transition (C$\rightarrow$B) is C, due to rule *(2a)*
2. the state of COAM before the execution of transition (2$\rightarrow$3) is B due to rule *(2c)*and*(2a)*
3. the state of COAM before the execution of transition (A$\rightarrow$B) is A due to rule *(1)*
4. the state of the Active Member before the execution of transition (C$\rightarrow$B) is 3 due to rule *(2c)* and *(2b)*
5. the state of the Active Member before the execution of transition (2$\rightarrow$3) is 2 due to rule *(2a)*
6. the state of the Active Member before the execution of transition (1$\rightarrow$2) is 1 due to rule *(1)*

Definition 20 *Given an employment complex* $\mathcal{Q}$=($\mathcal{S}$,$\mathcal{E}$,$\mathcal{M}$,ϕ). *Let* $\alpha = r_1, \ldots, r_k$, *be a canonical transition/state mix over the STD's* $\mathcal{S}$ *and* $\mathcal{M}$ *starting in* s_{01}, s_{0m}. α *is an* interleaving over $\mathcal{S}$ and $\mathcal{M}$ *iff for all transitions* t_k *in* α *the following conditions are satisfied:*

1. *if* t_k *is a transition in* $\mathcal{S}$ *and* p *is the state of the manager* $\mathcal{M}$ *before transition* t_k *then the transition must be permitted by* $\mathcal{M}$ *in state* p
2. *if* t_k *is a transition in* $\mathcal{M}$ *and* s *is the state of the employee* $\mathcal{S}$ *before transition* t_k *then the transition must be permitted by* $\mathcal{S}$ *in state* s.

To illustrate this concept of interleaving we will first show that the canonical transition mix α_2=A,(A$\rightarrow$B),1,(1$\rightarrow$2),B,2,(2$\rightarrow$3),3,(3$\rightarrow$2),(B$\rightarrow$C),C,(C$\rightarrow$B),2,(2$\rightarrow$1) is not an interleaving. To this aim we have to check if all the transitions are permitted. The first transition is already not permitted as (A$\rightarrow$B) is only allowed if the state of the Active Member before this transition, so the only starting state of the Active Member, being state $1 \in \phi^a$(A$\rightarrow$B) but ϕ^a (A→B)= 2. Therefore this transition is not permitted and so α_2 is not an interleaving.

We will now have a closer look at the first global behaviour as we described it in section 1.4. The figure on the next page a parallel execution of the processes COAM and an active member is given. We will see which interleavings of these processes are permitted.

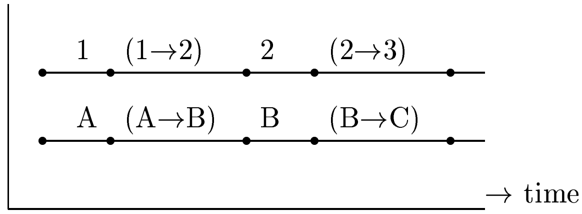

Figure 9: A Parallel description of COAM and An Active Member

In the next explanation we use the abbreviation M for the Chairman and E for the Active Member.

1. Interleaving 1,A,(1→2),(A→B) is not permitted. Transition (1→2) is permitted by M, if M allows this transition in the state before the execution of this transition. The state of M before the transition is state A and ϕ^s(A)=I. So (1→2)$\in I$ transition is permitted. But transition (A→B) is not permitted by E in this interleaving, as the state before the execution of this transition is state 1 and 1 $\notin \phi^a$(A→B)=2, therefore the transition is not permitted.

2. Interleaving 1, A, (A→B), (1→2), is also not permitted. As transition (A→B) is not permitted by E, as the state before the execution of this transition is state 1 and 1 not $\in \phi^a$(A→B)=2.

3. Interleaving 1,A,(1→2),2,(A→B),B,(B→C)is not permitted. Transition (B→C)is permitted by E, if E allows this transition in the state before the execution of this transition. The state of E before the transition however is state 2 and ϕ^s(B→C)={1,3}. In the interleaving there should be a 3 before the transition (B→C).

4. Interleaving 1,A,(1→2),2,(A→B),B,(2→3),3, (B→C) is permitted. Transition (2→3)is permitted by M, if M allows this transition in the state before the execution of this transition. The state of M before the transition is state B, ϕ^s(B)=II and 2→3 in II. Transition (B→C)is permitted by E, if E allows this transition in the state before the execution of this transition. The state of E before the transition is state 3 and ϕ^s(B→C)={1,3}.

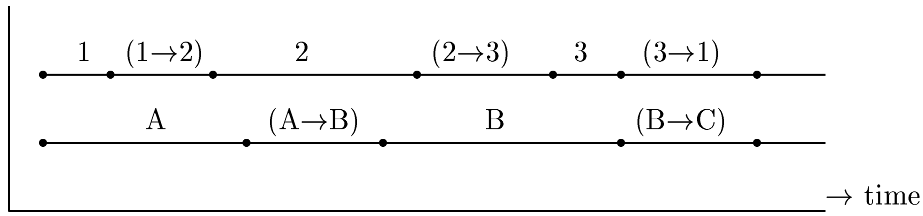

Figure 10: A second parallel description of COAM and An Active Member

This description corresponds to the interleaving 1, A, (1→2),2,(A→B),B,(2→3),3, (B→C),(2→3) that was permitted as we saw above. Two remarks:

1. It makes no difference whether the interleaving starts with 1,A or A,1
2. after state 3 it makes no difference whether first transition (B→C) takes place or transition(2→3). In fact they can happen at the very same moment: Transition(B→C)is allowed as the state of E before this transition is 3 and ϕ^s(B→C)={1,3}. Transition(3→1)is allowed as the state of M before this transition is B, ϕ^s(B)=II and (3→1)∈ II.

References

[BHGK] J.Banerjee, Hong-TaiChou, J.F.Garza, W.Kim, D.Woelk and N. Ballou. *Data model Issues for Object-Oriented Applications. MCC and Hyoung-Joo Kim, 1987* ACM transactions on Office Information Systems 5(1): 3-26.

[Harel] David Harell, *Statecharts: a Visual Formalism for Complex Systems/.* Science of Computer Programming 8 , 1987, 231-274.

[GELG] Engels, G., Groenewegen L.P.J. *SOCCA: Specifications of Coordinated and Co-operative Activities.* In A. Finkelstein,J.Kramer, and B.A. Nuseibah, editors, *Software Process Modelling and Technology/*, pages 71-102. Research Studies Press, Taunton,1994

[GELG] Engels, G., Groenewegen L.P.J. *Object-Oriented modeling: A roadmap.* In A. Finkelstein, editor, *The future of Software Engineering, 22nd International Conference on Software Engineering/*, pages 103-116. ACM Press, 2000

[Groe] L.P.J.Groenewegen, *Parallel Phenomena, a series consisting of technical reports. 1986-1990* Department of Computer Science, University of Leiden.

Publications in the Report Series Research* in Management

ERIM Research Program: "Business Processes, Logistics and Information Systems"

2002

Equivalent Results in Minimax Theory
J.B.G. Frenk, G. Kassay & J. Kolumbán
ERS-2002-08-LIS

An Introduction to Paradigm
Saskia C. van der Made-Potuijt & Arie de Bruin
ERS-2002-09-LIS

Airline Revenue Management: An Overview of OR Techniques 1982-2001
Kevin Pak & Nanda Piersma
ERS-2002-12-LIS

2001

Bankruptcy Prediction with Rough Sets
Jan C. Bioch & Viara Popova
ERS-2001-11-LIS

Neural Networks for Target Selection in Direct Marketing
Rob Potharst, Uzay Kaymak & Wim Pijls
ERS-2001-14-LIS

An Inventory Model with Dependent Product Demands and Returns
Gudrun P. Kiesmüller & Erwin van der Laan
ERS-2001-16-LIS

Weighted Constraints in Fuzzy Optimization
U. Kaymak & J.M. Sousa
ERS-2001-19-LIS

Minimum Vehicle Fleet Size at a Container Terminal
Iris F.A. Vis, René de Koster & Martin W.P. Savelsbergh
ERS-2001-24-LIS

The algorithmic complexity of modular decompostion
Jan C. Bioch
ERS-2001-30-LIS

A Dynamic Approach to Vehicle Scheduling
Dennis Huisman, Richard Freling & Albert Wagelmans
ERS-2001- 35-LIS

---

* A complete overview of the ERIM Report Series Research in Management:
http://www.ers.erim.eur.nl

ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Marketing
F&A Finance and Accounting
STR Strategy and Entrepreneurship

Effective Algorithms for Integrated Scheduling of Handling Equipment at Automated Container Terminals
Patrick J.M. Meersmans & Albert Wagelmans
ERS-2001-36-LIS

Rostering at a Dutch Security Firm
Richard Freling, Nanda Piersma, Albert P.M. Wagelmans & Arjen van de Wetering
ERS-2001-37-LIS

Probabilistic and Statistical Fuzzy Set Foundations of Competitive Exception Learning
J. van den Berg, W.M. van den Bergh, U. Kaymak
ERS-2001-40-LIS

Design of closed loop supply chains: a production and return network for refrigerators
Harold Krikke, Jacqueline Bloemhof-Ruwaard & Luk N. Van Wassenhove
ERS-2001-45-LIS

Dataset of the refrigerator case. Design of closed loop supply chains: a production and return network for refrigerators
Harold Krikke, Jacqueline Bloemhof-Ruwaard & Luk N. Van Wassenhove
ERS-2001-46-LIS

How to organize return handling: an exploratory study with nine retailer warehouses
René de Koster, Majsa van de Vendel, Marisa P. de Brito
ERS-2001-49-LIS

Reverse Logistics Network Structures and Design
Moritz Fleischmann
ERS-2001-52-LIS

What does it mean for an Organisation to be Intelligent? Measuring Intellectual Bandwidth for Value Creation
Sajda Qureshi, Andries van der Vaart, Gijs Kaulingfreeks, Gert-Jan de Vreede, Robert O. Briggs & J. Nunamaker
ERS-2001-54-LIS

Pattern-based Target Selection applied to Fund Raising
Wim Pijls, Rob Potharst & Uzay Kaymak
ERS-2001-56-LIS

A Decision Support System for Crew Planning in Passenger Transportation using a Flexible Branch-and-Price Algorithm
Richard Freling, Ramon M. Lentink & Albert P.M. Wagelmans
ERS-2001-57-LIS

One and Two Way Packaging in the Dairy Sector
Jacqueline Bloemhof, Jo van Nunen, Jurriaan Vroom, Ad van der Linden & Annemarie Kraal
ERS-2001-58-LIS

Design principles for closed loop supply chains: optimizing economic, logistic and environmental performance
Harold Krikke, Costas P. Pappis, Giannis T. Tsoulfas & Jacqueline Bloemhof-Ruwaard
ERS-2001-62-LIS

Dynamic scheduling of handling equipment at automated container terminals
Patrick J.M. Meersmans & Albert P.M. Wagelmans
ERS-2001-69-LIS

Web Auctions in Europe: A detailed analysis of five business-to-consumer auctions
Athanasia Pouloudi, Jochem Paarlberg & Eric van Heck
ERS-2001-76-LIS

Models and Techniques for Hotel Revenue. Management using a Roling Horizon.
Paul Goldman, Richard Freling, Kevin Pak & Nanda Piersma
ERS-2001-80-LIS

2000

A Greedy Heuristic for a Three-Level Multi-Period Single-Sourcing Problem
H. Edwin Romeijn & Dolores Romero Morales
ERS-2000-04-LIS

Integer Constraints for Train Series Connections
Rob A. Zuidwijk & Leo G. Kroon
ERS-2000-05-LIS

Competitive Exception Learning Using Fuzzy Frequency Distribution
W-M. van den Bergh & J. van den Berg
ERS-2000-06-LIS

Models and Algorithms for Integration of Vehicle and Crew Scheduling
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-14-LIS

Managing Knowledge in a Distributed Decision Making Context: The Way Forward for Decision Support Systems
Sajda Qureshi & Vlatka Hlupic
ERS-2000-16-LIS

Adaptiveness in Virtual Teams: Organisational Challenges and Research Direction
Sajda Qureshi & Doug Vogel
ERS-2000-20-LIS

Assessment of Sustainable Development: a Novel Approach using Fuzzy Set Theory
A.M.G. Cornelissen, J. van den Berg, W.J. Koops, M. Grossman & H.M.J. Udo
ERS-2000-23-LIS

Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-31-LIS

An NPV and AC analysis of a stochastic inventory system with joint manufacturing and remanufacturing
Erwin van der Laan
ERS-2000-38-LIS

Generalizing Refinement Operators to Learn Prenex Conjunctive Normal Forms
Shan-Hwei Nienhuys-Cheng, Wim Van Laer, Jan Ramon & Luc De Raedt
ERS-2000-39-LIS

Classification and Target Group Selection bases upon Frequent Patterns
Wim Pijls & Rob Potharst
ERS-2000-40-LIS

Average Costs versus Net Present Value: a Comparison for Multi-Source Inventory Models
Erwin van der Laan & Ruud Teunter
ERS-2000-47-LIS

Fuzzy Modeling of Client Preference in Data-Rich Marketing Environments
Magne Setnes & Uzay Kaymak
ERS-2000-49-LIS

Extended Fuzzy Clustering Algorithms
Uzay Kaymak & Magne Setnes
ERS-2000-51-LIS

Mining frequent itemsets in memory-resident databases
Wim Pijls & Jan C. Bioch
ERS-2000-53-LIS

Crew Scheduling for Netherlands Railways. "Destination: Curstomer"
Leo Kroon & Matteo Fischetti
ERS-2000-56-LIS