I) Introduction

This chapter deals with the description of the network and the most important parameters that should be taken into account in the research of a proper control strategy.
A attempt will be made to give a good understanding of the reasonnings that applied in the building of the production rules.

II) General description of the network

The **BORBECKERMÜLHENBACH** catchment has a area of about 2400 hectares whose 1200 hectares are impervious.
This catchment is situated within a region, where the land elevation continuously sinks, due to intensive coal extraction in the past.

As a result, every sewerage water must be pumped at the outlet of the catchment ,to be thrown away into the Emsher river.
This, as a matter of fact, means that an upper limit for every catchment outflows should be determined to prevent any exceedings of the pumps capacities.

For the **BORBECKERMÜLHENBACH** catchment :
the maximal allowed flow at the outlet is $30m^3/s$ -$31m^3/s$.

As this limit is overrun, during strong events (specially summer storms), it was decided to built at first one and then several retention bassins to store the exceeding flows during the rainfall peak.
4 retention bassins in chain, have actually been planned, along the main steam which are numbered from upstream to downstream B1,B2,B3,B4.

Ret. bassin number	state	capacity (m^3)
B1	planned	16000
B2	planned	60000
B3	already built but does work properly	36000
B4	operational	100000

In the main part of the network, the canalization are opened and there is very little chance that these should overflow , even during the strongest events.

However, there are two important exceptions :

- Just downstream the retention bassin B1 (subcatchment number 16000), there is a covered canalization part which does not allow more than 12m^3/s - 13m^3/s.

- Downstream the retention bassin B2 and upstream the retention bassin B3 (subcatchment number 23000), there is also a covered canalization which does not allow more than 21m^3/s - 22m^3/s.

A few simulations have been previously started, to get an idea of the principal difficulties and limits of the network.
The test rainfall is a constant rainfall over the whole catchment whose intensity is 1.58mm every 5 minutes, during 1 hour.

<u>These trials have shown that these 2 covered parts, can easily be filled up, and the control of the retention bassins should aim to limit the flow charges.</u>

As the simulation model cannot properly simulate the functionning of retention bassin outlets (specially it cannot calculate the outflows corresponding to the position of a gate), it was decided to introduce fictitious pumps into the network, which can be taken into account by the model and which influence the network state,in a similar way as the gates.

Schema of the principle.

1) In reality

upstream	inflow Qin	Retention bassin Bi	outflow Qout	downstream

storage flow
Qsto

2) In the model

upstream	inflow Qin	Pump i	Qout is pumped	downstream

(Qin - Qout) > 0
is stored

Each pump Pi, is refered by a number i, which corresponds to the number of the retention bassin Bi to which it is associated.

Here are given the characteristics of the pumps :

Pump	Type	rate1	rate2	rate3	rate4	
P1	on-line	13	-	-	-	-
P2	on-line	2	7	13	19	
P3	on-line	28	-	-	-	
P4	on-line	31	-	-	-	

Rem : The rates are given in m^3.

III) The variables used in the control strategy research process.

III.1) As stated previously, the aims of the control are :

- not to deliver more than 30-31 m^3/s out of the free outlet.

- to prevent any overflow due to the restricted capacities of the canalizations 16001-17000 (12-13 m^3/s), and 23000-25000 (20-21 m^3/s).

III.2) To reach this goal, the only possibity is to developp some control strategy at the retention bassin B2, that is to say, determine, at each control step, which is the most proper rate for pump P2.
The other retention bassin cannot be controlled, in the really meaning of it, since the pumps have only two possible states (pump off - pump on).

III.3) Typology of the variables which are used in the determination of a strategy.

There are three kinds of variables which take part in the writting of the rules :

- the parameters

- the operators
- the values (qualitative).

This derives from the structures according to which the rules are built (that was described elsewhere).

The parameters are themselves divided into two classes :

- The real parameters , which can take any value in a proper interval of the real continuum, in the limit of the authorized computer precision.

- The step parameters, whose possible values are qualitative , this means that the values which the paramater can be assigned should beforehand be coded too (qualitative values).

The list of every possible parameters, along with the list of every ooperators and qualitative values (step variable) is registred in the data file **'CODDAT'**,with their correspondent code (the integer value which identifies each of them) .

III.4) The **meta-code.**

- Ideally, the choice of an integer to identify a variable should be completly free, totally arbitrarily. This, in fact, is not the case. In other words, there is a meta-code, defined in the design of the system that operates, and necessitates that some rules be applied in the coding.

ABOUT THE PARAMETERS :

1) - The parameters PIWAS, in which i is the number of the pump should be given the code 22I. At least, is it supposed in programm **'NEWPRO'**.

ABOUT THE OPERATORS :

1) - There are operators whose code is not stored in any data file . The codes are directly written in a programm.

The operator ' ->' is coded in programm '**INPRO**' : 9988

The operator '-->' is coded in programm '**INPRO**' : 9999

Both operators bind the 'assumption part' (first part) of a rule to the 'conclusion part' (second part).

Ex : (assumption part) ' ->' (conclusion part).

The difference between them, is that :

by using '-->', the information used to verify the 'assumption part', is erased from the 'Working Memory', if the rule should apply.

- this supposes, that the parameters whose values are verified in the 'assumption part' of the rule , in case this one apply, should not be of utility, in any other rules to be choiced during the same decision process.

by using ' ->', all the informations, about the state of the network, remain in the 'Working Memory'.

- 2) <u>The other operators</u> :

- They intervene in an more elementary level in the building of the rules, namely in the writting of the unities ('words'), which compose the rules, whose muster is (parameter,operator,value).

- These too, are coded by the designer and should not be recoded (assumptions are made in the treatments programms).

The list of the operators is >, >=, =, <=, < which allow comparisons between the value of a given parameter and determined limits.

As the system was built, it is the operator, which also

gives the information about the nature of the parameter (real parameter or step parameter). This means that the list should be doubled :
for real parameters : >r, >=r, =r, <=r, <r
for step parameters : >, >=, =, <=, < .

The operators which correspond to step parameter are obligatory coded the following way :
< is coded 1; <= is coded 2; = is coded 3; >= is coded 4; > is coded 5 (this will, in particular, be assumed in the building of new rules by programm '**NEWPRO**')
Moreover, any operator which operates with step parameters should be coded between 0 and 9 included.

The operators which correspond to real parameter are obligatory coded the following way :
<r is coded 21; <= is coded 22; = is coded 23; >= is coded 24; > is coded 25
Moreover, any operator which operates with real parameters should be coded between 20 and 29 included.

III.5) The free coded parameters (and their corresponding qualitative values) :

Remarque:
A lot of the parameters which are used, are primarily numerical variables. However, they are translated into step variables, for the sake of clarity in the writting of the control rules to be applied at pump **P2**.
This translation occurres also, by the processing of rules (qualification rules) which are assembled in rules blocks, each one corresponding to one parameter.

1) The **pump rates** :
Pi (step variable): pump rate of the pump i.
The corresponding numerical flow rate values are given in the data file '**SONDERBW.NET**'.
Ex : (Pi = 2) means that the pump number i, is set

at rate 2.

2) The **rainfall intensities** measured in the catchment (step variable):
We suppose that 2 rainfall intensity measurements are available;
- the RINT1 parameter : for the upstream part of the catchment
- the **RINT2** parameter : for the downstream part of the catchment.

Here is the table which describes the translation from numerical into qualitative values :

	LEVEL 1	LEVEL 2	LEVEL 3
RINT	0/25	25/50	≥ 50

3) The **water level** in the retention bassins (step variable see Rem1):
HRBi : water level in the retention bassin Bi.

	Number of the corresponding junction (+ ground/top heights)
HRB1	16002 (61.90m/64.70m)
HRB2	22002 (52.95m/56.50m)
HRB3	27000 (47.80m/53.00m)
HRB4	31002 (37.60m/47.20m)

These heights (particularly for retention bassins 1 and 2), are translated into qualitative values that should qualify how much the retention bassin full is.

	LOW	MID	HIGH	OFL
HRB2	52.95/53.80	53.80/55.60	55.60/56.5	≥ 56.5
HRB4	37.60/40.00	40.00/44.80	44.80/47.20	≥ 47.20

Actually, the principle is quite simple :
LOW corresponds to a fullness percentage smaller as 25%.
MID corresponds to a fullness percentage between 25% and 75%
HIGH corresponds to a fullness percentage between 75% and 100%
OFL corresponds to a fullness percentage higher as 100%.

4) The **flow rate** calculated at some junctions of the network (step variable see Rem1):
Q12 : flow calculated at the junction number 12000.

The flow rate value is also translated into a qualitative value (according to a similar translation principle as was explained upper)

	LOW	MID	HIGH	OFL
Q12	0/8	8/24	24/32	32

5) **other variables** .

- Parameters which should be used in case there should be rainfall forecast:

VOR10:
VOR22:
VORAU: (= **FALSE** because no forecasts are actually available).

- Parameters used in the writting of meta-rules
- **PiWAS** : which pump is to be punished.

The corresponding operator is automatically **TOO** . **PiWAS** are two step variables (**LOW** , **HIGH**).
- **STRAF** : the value of the punition (numerical variable)
- **ZURUK** : The number of control step that should be back-tracked in the punition process (numerical variable).

DUMMY:

ENDE: This parameter specifies that the control strategy research process is terminated when its value is '**TRUE**'. Otherwise (value = '**FALSE**'), there is still, at least, one other control block to read.

IV) THE BUILDING OF THE STRATEGY RESEARCH RULES.

Apart from the blocks of "qualitative rules" which are used for translating the numerical values of the parameters into step values, there is actually only one control block : the one which contains the rules to operate pump **P2**.
In this block, there are different kinds of rules which are implemented .
Two great categories should be differentiated :
- the rules describing a "local strategy".
- the rules describing a "global strategy".

- In the local strategy, the determining variables are situated nearest the decision variable, in our case the pump **P2**.
There is actually, only one variable which is verified to determine the decision in pump **P2**: **HRB2** height in the retention bassin **RB2**.
The principle is simple : the higher the water level in the bassin is, the higher the operating rate of pump **P2** should be set in oder to prevent any overflow.

- In the global strategy, there are, on the contrary several art of reasonnings depending on which parameters are considered.

- Some parameters, are first useful because they allow a forecast of the flow rates quantities that will be transported in the critical parts of the network, specially in the retention bassins **RT2** and **RT4**.

Subsequently, can the pump **P2** be operated, in order to prevent any desorder.

- The rainfall intensities **RINT1** (upstream) and **RINT2** (downstream) .
- The flow rates in strategical part of the network, specially upstream **Q12** .

- Secondly, one should not forget that the most important goal is to prevent the flow rate outside the network (junction 32000), to overrun a given limit (30 m^3/s).
In this respect, it is important, to give special attention to the water level in **RT4**, which is the greater bassin (100000m^3) and is particularly aimed at storing the exceeding flow during peak rainfall intensity : the retention bassin, should be operate in such a way that it can have the greater storage capacity, when the peak occurres.
Subsequently, the flow rate in pump **P2** (in direction to **RT4**), should be diminished when the storage capacity, in **RT4**, threatens to be fulfilled, as meanwhile the peak intensity has not yet occurred, or is not terminated.

THE **CONTROL** PROGRAMM

The **CONTROL** programm is the first level programm which manages the whole treatment that is to say, the simulation of the functionning of the network (before, during and after the rainfall) and the research of a well fitting strategy at each control step .

These different tasks are distributed into different subroutines whose most important ones are written hereafter, in their order of apparition.

- **EXTINI** (initialisation of the parameters for the **EXTRAN** simulation model)
- **STEINI** (initialisation of the parameter for the research of strategy module)

- **INDAT**

It reads the data files that describe :

- the rainfall,
- the canalisation network,
- the special structures,
- the coding of every symbols to be used by the strategy research and self-learning modules
- the implementation in machine memory of the production rules .

- **DRYWEA** (determination of the state of the network before the next event occurs - dry weather state -)
- **STEUER** (first strategy determination)
- **HEAD** (?)

TREATMENT LOOP DURING THE SIMULATION OF THE CURRENT EVENT

- **STEUER** (determination of strategy)
- **TRANSX** (simulation : calculation of the values of every state variables)

END OF LOOP

- **HEAD** (?)

- **NEWPRO** (evaluation of the new values to attribute to each production rule (after the summing up of the punition values), and if necessary, building of new rules out of the old defectuous ones (when the sum of these punition values exceeds a given threshold).

- **OUTGES** (display of some statistic results about the last simulation period).

- **REINI** (reinitialisation if another event is to be simulate).

- **OUTPRO** (display of the new set of production rules).

In a more detailed description of the treatments, as they are proceded, one distinguishes the following tasks :

I) Reading the COMMON BLOCKS

- the common block **COMEXT** deals with the simulation programms variables .
- the common block **COMCON** deals with the control strategy programms variables.

II) Initialisation of the variables

- **EXTINI** initializes the required variables for the simulation (EXTRAN model).
- **STEINI** initializes the required variables for the control strategy programms.

III) Reading the data files that describe the network configuration and the rainfall event. This programm is called **INDAT**. It reads the following data files :

- **BORBECK.NET** gives the characteristics of every canalizations (their junctions numbers, their length, their roughness, ...)
- **SONDERBW.NET** gives the specifications of every special structures (storage points, pumps, weirs, free outlets,...)
- **RTEST.REG** gives the rainfall heights that are measured at every step, for each gage (maximal number of gages 3).

IV) Calculating the values of the state variables before the rainfall event (dry weather period state) and the regulation of each gage at this time.

V) Beginning of the treatments loop for the current event. This treatment period includes the rainfall duration plus the following period when the inferred inflows are still to be taken into account.

- Running the control module named **STEUER** ('control')
- Running the simulation module named (**TRANSX**).

End of the treatment loop.

VI) Calculating the water volumes that remain in canalisations after the treatment period.

VII) Evaluating the set of rules that were used and when required building new ones. (**NEWPRO**).

IX) Displaying some statistics (about the rainfall characteristics, or the regulation levels for each control gage).

X) In case other events should be treated (others rainfall events are stored in the rainfall data file), a new initialisation is started (**REINI**), and the pointer is set back for recalculation of the dry weather state.

XI) In case no other events should be simulated, the computer ends the treatment by displaying the new existing (after the rebuilding if it was necessary) set of rules. (subroutine **OUTPRO**).

The programm INDAT ('INPUT DATA'), reads all information required by the simulation programms as well as the control strategy determination programms.

Before reviewing in details each treatment which is achieved during the running of this programm, we shall give the list of the most important subroutines that which called from **INDAT** in their order of appearance.

Treatment of data to be used for the simulation of the network functionning.

- **INREG** (rainfall data)
- **INKAN** (data about the canalisations)
- **INTRA** (data about special structures)

Treatment of data to be used by the strategy research module.

- **INKOD** (coding of all the symbols that shall be used by the strategy research module)
- **INPRO** (coding and implementation of the production rules in computer memory).

I) READING THE FILE **'EINDAT'** (N5).

Rem : The parameter in parenthesis after the name of a file corresponds to the variable which defines to which unit the file is referred. The value of these parameters (Ni) are determined in the initialization programms (**EXTINI**, **STEINI**).

I.1 Reading the names of the should be read files :

1) The file where the results are displayed named in our case **AUSDAT** (N6). This file stores the most important assumptions, characteristics and obtained results of each network simulation and control trial.
2) The file where the canalisation net data are stored named **BORBECK.NET** (N10).

3) The file where the special structures are described named **SONDERBW.NET** (N11).

4) The file where the rainfall intensities are stored named **RTEST.NET** (N20).

5) The file where the already existing set of control rules are stored named **PRDBORB** (N30).

6) The file where the new set of control rules (after evaluation and modifications at the end of the treatment) shall be stored , named **NPRDBORB** (N31).

7) The file identifying every punished rules along with

- the value of the punition
- the values of every parameters which are (or could be) taken into account in writting the rule, at the moment when the punition occurs.

It is named **PAWDAT** (N35).

8) The file containing the codes (translation in integer value) of each symböl which is manipulated by the expert-system and its self-learning facility. It is called **CODDAT** (N38).

9) A file (specialy designed for the Bremen Project), containing the values of the energy consumption peaks for both pumps stations (there are 2 pumps installations in Bremen) . It is called **KONDAT** (N40).

I.2 The determination of the simulation period .

- IDATEZ, THILFZ date of start
- IDATEE, THILFE date of end.

I.3

- NTCYC Number of simulation time steps to be performed, after the end of the rainfall event,
- DELT Duration of one simulation step (s) .

I.4 Values of some control parameters.

- ISINT Duration of a control step (mn) ,
- MEMMAX duration of the memory of the system about its own actions (mn),
- IVMIN : When forecasts are available how long is the prediction

interval (mn) ,

- IPRODU : If simulations with evaluation of rules were already done and a **'PAWDAT'** file subsequently already exists, then IPRODU = 1 . Otherwise IPRODU = 0 .
- ILPMIN
- SVMIN is a threshold value whose overrun starts the generation process of a derived new rule by the self learning module (**NEWPRO**).
- HERVAL : When HERKU = HERVAL this means that the rule was generated by the system itself and not by the designer of the system.

I.5 Value of some simulation parameters.

- JSWTH is a switch parameter (There are two versions of the simulation model).
- KBW : If the value of roughness is calculated according to PRANDTL-COLEBROOK, then KBW=1. Otherwise when calculated according to MANNING, KBW<>1 .
- ITMAX maximal number of iterations in case the calculated results oscillate.
- SURTOL indicates at which condition, the iteration process can be stopped if the maximal number of iterations has not already been run: The error by oscillation is smaller as SURTOL % of the calculated flow.

I.6 Values by default of runoff parameters.

- INEICH If INEICH = 0 then the standard values for each parameter apply (initialized in the programm itself).
- ISOIL: (2)
- WNULL: (10mm)
- DEPG: (5)
- KSTU: (4)
- ALPHAU: (2.3)
- ALPHAV: (11.)
- ABSTRT: (0.7)

- VMULD: (2.5)
- AA: (25. %)
- AE: (85 %)

Rem : The characteristics of the pervious and impervious parts of the catchment area, are the same wherever they take place.

I.7 Parameters controlling the displaying of the results.

- IPRINT (0/1): When IPRINT = 1 results shall be printed during the calculations; else no printings occur during the simulation time.
- NDRUCK (0/1): When NDRUCK = 1 the results shall be printed after the simulation time; else no printings after the simulation time.
- NHPRT: Number of junctions whose results are plotted.
- NQPRT: Number of canalisations whose results are plotted.
- IIOUT:
- ISOUT:
- IPOUT:
- IGOUT:
- IROUT:
- NSTART:
- INTER: When intermediary results are to be printed, these are printed every INTER mn.

I.8 The numbers of each junction whose parameters (water level, flow rate) are plotted, are stored in JPRT(i).

I.9 The numbers of each canalisation whose parameter (flow rate) is plotted, are stored in CPRT(i).

II) OPENING EVERY FILES WHOSE NAMES HAVE BEEN DERTERMINED IN PART I).

III) READING THE RAINFALL DATA : The subroutine '**INREG**' is started.

- BB = 0
- INTEFF = 5.0 mn (measurement step).

- MESANZ = 3 (number of raingages).
- TINDAY = 81001 (Year: 81, day:1).
- TINNEX = 00.00 (Hour:0)

A verification is done, whether the given dates are possible.

IV) READING THE DATA ABOUT THE CANALIZATIONS stored in file **'BORBECK.NET'** (N10) ; the subroutine **'INKAN'** (INPUT KANALNETZ) is started.

For each canalization part, are given :

- NCOND(i) : the number assigned to the ith canalization .
- NJUNC(i,1) and NJUNC(i,2) : the number assigned to its two (upstream and downstream) junctions.
- LEN(i) : the length of the canalization i.
- NKLASS(i) : form of the section

. circular profile NKLASS = 1
. square profile NKLASS = 2
. trapeze profile NKLASS = 9
. special profiles NKLASS = 3,8 ; that are to be specially defined in the file **'SONDERBW.NET'**

- ROUGH(i) : The roughness of the canalizaton i.- standard values are, according to Manning's method 0.014 ; according to Colebrook's method 1.5 -.
- DEEP(i) :
- WIDE(i) :
- AFULL(i) : the whole area of the cross-section of canalization i.

- ZP(i,1) and ZP(i,2) : height of the canalization ground, upstream and downstream the jonction.
- B1(i) and B2(i) : land elevation, upstream and downstream the canalization i.

- B3(i) and B4(i) : initial (dry weather) flows, upstream and downstream the canalization i.

- STHETA(i) : parameter describing the trapeze profile.

- SPHI(i) : parameter describing the trapeze profile.

- FGES : whole catchment area, drained by the ith canalization.
- FVERS : impervious catchment area, drained by canalization i.
- INDGEF : coefficient which indicates the land gradient at each side of the canalization (determines the speed of the runoff process to the canalization).
- IC1 :
- IC2 :
- MESTAT(i) :
- ALPHV :

V) READING THE CHARACTERISTICS OF THE SPECIAL STRUCTURES which are stored in file **SONDERBW.NET** (N11) ; the subroutine **'INTRA'** is started.

V.1) The storage capacities in the network (no more than 20 elements).

The model is so conceived that each retention bassin must be considered as a junction with a storage capacity corresponding to its own volume .

Another case when storage capacity must be introduced at junction points , is when there is a fall (water flow disruption) at this point - in order to prevent too great oscillations in resolution of the differential equations –.

For each junction with a storage capacity, are set the following parameters :

- JSTOEL(i) : Number of the ith element (junction) to which a sto₂age capacity is attributed .
- ATOP(i) : Area of the top of the storage element.
- HTOP(i) : Heigh of the top of the storage element.
- AGRD(i) : Area of the bottom of the storage element.
- HGRD(i) : Height of the bottom of the storage element.
- NSTOEL : number of storage elements.

V.2) The weirs (no more than 5 weirs).
For each weir, are stated :

- NJUNC(n,1) and NJUNC(n,2) : the junction numbers between which the correspovdent weir (special structure number n), stretches.
- KWEIR(i): coefficient to be attributed to the ith weir.
- YCREST(i): Height of the crest of the ith weir.
- YTOP(i): Height of the top of the ith weir.
- WLEN(i): Length of the ith weir.
- COEF(i): another coefficient attributed to the ith weir.

V.3) The pumps (no more than 7 pumps).

As the utilized simulation model is not able to take into account what happens especially at the outlets of retention bassins (gates...), fictitious pumps are introduced, that are supposed to pump more or less the quantities which go out of the bassin.

For each pump, are given :

- NJUNC(n,1) and NJUNC(n,2) : The junction numbers, upstream and downstream, which the corresponding pump (special structure number n) links.
- IPTYP(i) : Type of the ith pump :
- IPTYP=1 : off-line pump
- IPTYP=2 : on-line pump
- IPTYP=5 : pump which simulates the outflow of a retention bassin .

In our case, all the pumps belong to the type 2. The pumps of type 5, are not used, because their regulation rates depend upon the stored quantities in the bassin, which is not acceptable when some other control strategy determination rules should be developped.

- VWELL(i) : starting volume in the bassin - does only apply to type 5-.
- PSVOL(i) : volume of the bassin (to be extracted) - does only apply to type 5.
- PVOL(i,j) : the pump rate - does only apply to type 5.
- PVOL(i,j) : The jth step (pump rate) according to which the ith

pump is regulated (type 2).
- PLEI(i,j) : The amount of energy required to set the pump to the jth rate (type 2).

There are no more than 5 rates available for one pump (type 2), nominally (0,1,2,3,4) - the step 0 corresponds to the state "pump out" -.

V.4) The free outflows (no more than 5)

- JFREE(i) : number of the junction that corresponds to the ith free outflow.

V.5) The special profiles (no more than 6 profile types whose class numbers - NKLASS - are 3-8).

- ANORM(i,j) : Area of each part (under-section) of the cross-section into which it was divided to allow its proper determination.
i= profile number ;
j= 1-26 : the 26 under-section number .
- HRNORM(i,j) : height of each under-section .
- TWNORM(i,j) : width of each under-section.

These values are normalised just after their reading.

V.6) Reading of the initial speed and flow values, pertaining to each canalization when they are available.
Q(n) : initial flow in the nth canalization.
V(n) : initial speed in the nth canalization.
Y(n) : initial water depth in the nth canalization.

VI) THE SETTING OF THE JUNCTION NUMBERS WHERE WATER DEPTH RESULTS ARE AVAILABLE FOR THE STRATEGY RESEARCH MODULE.

- PEGEL(i) : junction number where the ith water depth value, in the network, is available (actually, these values are calculation results) .

VII) THE SETTING OF THE INITIAL PUMP RATES. (?)

VIII) THE READING OF THE CODDING FOR EACH SYMBOLS TO BE UTILIZED BY THE STRATEGY RESEARCH CONTROL ; the subroutine **'INKOD'** is started).

'INKOD' reads the data file **'CODDAT'** (N38) and then the data file **'KONDAT'** (N40).

VIII.1) **'INKOD'** reads in file **'CODDAT'** , the symbols as they are originally named (character string), and their numerical codded equivalents (integer) which alone can be manipulated by the treatment programms .

According to the way the rules are written, the symbols have been divided into three categories :

- the parameters
- the operators
- the values

<u>The production rules are actually, the union of conditions (resp conclusions) whose general pattern is :</u> **(parameter, operator, value)** .

The parameters:
- VARTAB(j) : The jth parameter symbol (character string) .
- VARCOD(j) : The coding of the jth symbol (integer).

The operators:
- OPETAB(j) : The jth operator symbol (character string).
- OPECOD(j) : Its coding (integer)

The values :
- WERTAB(j) : The jth value (character string).
- WERCOD(j) : its coding (integer).

VIII.2) Determination of :

- IPCOD(i) : The coded value of pump i.
- IZCOD : The coded value of ' ZERO'.
- INCOD : The coded value of '   0'.

VIII.3) For each parameter which is registered, the reading of :

- The lower limit of the NRth parameter (in its order of appearance in the coding phase) : UGR (Untergrenzwert) = GRENZ(NR,1) .
- The upper limit of the NRth parameter : OGR (Obergrenzwert) = GRENZ(NR,2).
- (?) The variation direction of the NRth parameter in comparison to the variation direction of the ith pump: VAROUP(NR,i).
When both evoluate in the same direction (increase and decrease in the same time) VAROUP(NR,i)=1, when on the contrary, their variations are inverted (one increases when the other one decreases) VAROUP(NR,i)=-1 .

VIII.4)

The set of rules for strategy research , is divided into blocks which are proceded one after the other.
Especially, the determination of the working rate of a particular pump corresponds to the treatment of a single block of rules.

For each block of rules, 3 pieces of information are given :

- 1) The parameters that should possibly be used in the modifying of old rules to generate new ones. This list of character string symbols is coded and then stored in PALIST(IAB,i,1) where :
- IAB is the number of the block of rules.
- i is the number of the ith parameter in the list corresponding to this block (there can not be more than 16 parameters pro block that is to say i<=16).

- 2) The compared variation rate for each parameter of the list that is stored in PALIST(IAB,i,2) ; (this can only take the value 1 or -1) .

This information is very important to build new conditions (extended set of conditions) in new computer generated rules, because it allows to determine which operator applies; operator >= or operator <= .

- 3) The last parameter allows to determine to which class belongs the corresponding ith parameter. It is stored in PALIST(IAB,i,3)

There are 2 kinds of parameters :

- The parameters which are attributed "numerical values" ; for example water deapth values, rainfall height values - They are supposed to vary continuously -.
- The parameters which are attributed "step values" ; for example the pumps rates - they vary discontinuously -.

The distinction, is introduced particularly, through the use of different operators to designate equality, superiority, inferiority (=, <= , >=), in the writting of the condition (resp action) parts of rules.

The symbol =r coded 3, is the equality operator corresponding to numerical parameters.
The symbol = coded 23, is the equality operator corresponding to step parameters.

- PALIST(IAB,i,3) = 3 => this is a numerical parameter.
- PALIST(IAB,i,3) =23 => this is a step parameter.

VIII.5) At least, **'INKOD'** reads in file **'KONDAT'** (N40) , the 3 highest peaks of energy consumption during a month, for each pump installations (2 in the BREMEN project).

- HWSTH : The mean value of the three highest peaks of energy consumption in the first installation.
- HWH(i) i=1-3 : the 3 peaks in the first installation .
- HWSTK : The mean value of the three highest peaks of energy consumption in the second installation.
- HWK(i) i=1-3 : The 3 peaks in the second installation.

This information was particularly important, in the BREMEN project, where these peaks intervened in the determination of the energy costs (the higher the peaks are, the greater is the financial penalty to be paid, just for being authorised to start the pumps).
In the BOTTROP project, this information is of no use. The pumps are only imaginary pumps (that were introduced only for the sake of simulation).

IX) READING AND IMPLEMENTING THE SET OF PRODUCTION RULES ,IN THE COMPUTER MEMORY ; the programm **'INPRO'** is started.

1) It reads each production rule in file **'PRDBORB'** (N30) - where the initial rules are stored – , then it codes it and implements it in the array PRDMEM(,) (subroutine **'INSPRO'**).
2) If the expert-system has already been tested on other events (IPRODU = 1), the programm reads on file **'PAWDAT'** (N35), the punition values attributed to each production rule (in case they were punished) , and the values of the parameters listed in the corresponding block, at every punition time .

This information is used to determine :
- when a new production rule is to be automatically generated (the total of every punition values for one rule exceeds a given threshold).
- How the condition part of the old rule is to be completed to generate the new condition part (choice of the block parameter which shall be used).

X) SUMMING OF THE ERRORS NUMBER DURING THE READING PROCESS .

These errors may be :
- a bad definition of the network configuration or its special structures; every possible configurations of a network are not allowed by the simulation model. For example :
- a junction can not be bound to more than 8 other junctions,
- a pump can not be bound to more than one canalisation upstream and one downstream ...

This information was particularly important, in the BREMEN project, where these peaks intervened in the determination of the energy costs (the higher the peaks are, the greater is the financial penalty to be paid, just for being authorised to start the pumps).
In the BOTTROP project, this information is of no use. The pumps are only imaginary pumps (that were introduced only for the sake of simulation).

IX) READING AND IMPLEMENTING THE SET OF PRODUCTION RULES ,IN THE COMPUTER MEMORY ; the programm **'INPRO'** is started.

1) It reads each production rule in file **'PRDBORB'** (N30) - where the initial rules are stored – , then it codes it and implements it in the array PRDMEM(,) (subroutine **'INSPRO'**).
2) If the expert-system has already been tested on other events (IPRODU = 1), the programm reads on file **'PAWDAT'** (N35), the punition values attributed to each production rule (in case they were punished) , and the values of the parameters listed in the corresponding block, at every punition time .

This information is used to determine :
- when a new production rule is to be automatically generated (the total of every punition values for one rule exceeds a given threshold).
- How the condition part of the old rule is to be completed to generate the new condition part (choice of the block parameter which shall be used).

X) SUMMING OF THE ERRORS NUMBER DURING THE READING PROCESS .

These errors may be :
- a bad definition of the network configuration or its special structures; every possible configurations of a network are not allowed by the simulation model. For example :
- a junction can not be bound to more than 8 other junctions,
- a pump can not be bound to more than one canalisation upstream and one downstream ...

This information was particularly important, in the BREMEN project, where these peaks intervened in the determination of the energy costs (the higher the peaks are, the greater is the financial penalty to be paid, just for being authorised to start the pumps). In the BOTTROP project, this information is of no use. The pumps are only imaginary pumps (that were introduced only for the sake of simulation).

IX) READING AND IMPLEMENTING THE SET OF PRODUCTION RULES ,IN THE COMPUTER MEMORY ; the programm **'INPRO'** is started.

1) It reads each production rule in file **'PRDBORB'** (N30) - where the initial rules are stored – , then it codes it and implements it in the array PRDMEM(,) (subroutine **'INSPRO'**).
2) If the expert-system has already been tested on other events (IPRODU = 1), the programm reads on file **'PAWDAT'** (N35), the punition values attributed to each production rule (in case they were punished) , and the values of the parameters listed in the corresponding block, at every punition time .

This information is used to determine :
- when a new production rule is to be automatically generated (the total of every punition values for one rule exceeds a given threshold).
- How the condition part of the old rule is to be completed to generate the new condition part (choice of the block parameter which shall be used).

X) SUMMING OF THE ERRORS NUMBER DURING THE READING PROCESS .

These errors may be :
- a bad definition of the network configuration or its special structures; every possible configurations of a network are not allowed by the simulation model. For example :
- a junction can not be bound to more than 8 other junctions,
- a pump can not be bound to more than one canalisation upstream and one downstream ...

This information was particularly important, in the BREMEN project, where these peaks intervened in the determination of the energy costs (the higher the peaks are, the greater is the financial penalty to be paid, just for being authorised to start the pumps).
In the BOTTROP project, this information is of no use. The pumps are only imaginary pumps (that were introduced only for the sake of simulation).

IX) READING AND IMPLEMENTING THE SET OF PRODUCTION RULES ,IN THE COMPUTER MEMORY ; the programm **'INPRO'** is started.

1) It reads each production rule in file **'PRDBORB'** (N30) - where the initial rules are stored – , then it codes it and implements it in the array PRDMEM(,) (subroutine **'INSPRO'**).
2) If the expert-system has already been tested on other events (IPRODU = 1), the programm reads on file **'PAWDAT'** (N35), the punition values attributed to each production rule (in case they were punished) , and the values of the parameters listed in the corresponding block, at every punition time .

This information is used to determine :
- when a new production rule is to be automatically generated (the total of every punition values for one rule exceeds a given threshold).
- How the condition part of the old rule is to be completed to generate the new condition part (choice of the block parameter which shall be used).

X) SUMMING OF THE ERRORS NUMBER DURING THE READING PROCESS .

These errors may be :
- a bad definition of the network configuration or its special structures; every possible configurations of a network are not allowed by the simulation model. For example :
- a junction can not be bound to more than 8 other junctions,
- a pump can not be bound to more than one canalisation upstream and one downstream ...

- some problems in the definition of the symbols .For example:
- use of too many symbols,
- use of unknown symbols that is to say symbols which were not previously coded ...
- some problems in the reading of the production rules. For example:
    - use of unknown symbols,...

If errors are discovered, these ones are displayed and the treatment is aborted.
If not, **'INDAT'** is terminated.
The treatment goes further.

I) INTRODUCTION

- This chapter deals with the different treatments, which are proceded in the module of control strategy research.
Every treatments correspond to a set of programms and subroutines, whose order determines the treatment order. We shall try in this description, to stay as far as possible, near to the programms structure.

II) THE READING OF THE STRATEGY CONTROL RULES

This operation is performed in the programm called **'INPRO'**.
The rules are originally stored in the file **'PRDBORB'** (N30).

- The rules shall be read one after another, in their original form.

- They shall be coded according to the information originally stored in file **'CODDAT'**(N38) - that were read by the computer in programm **'INKOD'**-
As said elsewhere, each rule comprehends unities whose muster is (parameter,operator,value). According to the coding of the operator is the parameter supposed to be a numerical or a qualitative parameter. Only in case the parameter be a qualitative parameter shall the value be coded.
The currently treated rule, is then stored, in the integer array called **PRONEU**(60) – This means that the limit to be inserted in the writting of the rules can only be integer limits –

- The array **PRONEU**(), shall then be stored in the array called **PRDMEM** (Production memory), which contains all the previous already read rules. This particular operation shall be performed by calling the subroutine **'INSPRO(,,)'**

PRDMEM is an array 1000 lines × 12 columns, where every coded production rules are stored, and later read .
- The first 11 columns are utilized to store the coded productions;

- the 12th column contains information to control the reading of the array.

- The other parameters to characterize the rule, determined in programm '**INSPRO**':

Suppose that one is reading the **ILX** th production rule and that the currently read production block is block **IAB** .

- **IAB<=IBI** maximal possible number of blocks, while **ILB** is the total number of blocks which were actually written in the data file).

- The rules are stored in file '**PRDBORB**' block by block.
The first block is the meta-rules block. The meta-rules are not useful in the strategy research module, but only in the learning module. Its limit is defined with the sign '88888'.
The other blocks, are rules block, to be read in the control strategy determination process. Each block termination is defined with the sign '77777'.

- The number of the production block in which this rule is stored is given in **BLOCKP(ILX)** (=IAB).
- The number of the rule in the production block is **NUMERP(ILX)**.
- The number of the line in **PRDMEM** where the beginning of the coded rule is stored, is given in **INDEXP(ILX)**.

The number of the last possible storage line in **PRDMEM** is given in **ILP**.
The number of the actual last storage line is given in **IFP**.
Bothe parameters - **ILP**, **IFP** - allow to check uo whether there stay enough place in **PRDMEM** to store the current rule.
In case there should not be enough place, the less efficient rules are destroyed in **PRDMEM** (till 10% of the total storage in **PRDMEM** capacity, is available for the next rules to be stored).

If the first 11 columns in line **INDEXP(ILX)**, are not sufficient to store the rule, the writting goes on in line **PRDMEM(INDEX(ILX),12)**.
The 12th column, in the last line where the rule is stored, contains 0 - (**PRDMEM(,12)** = 0) -.

Other parameters are valuated :

- **VALUEP(ILX)** : The value attributed to the rule ILX.
- **WIEOFP(ILX)** : The number of times the rule ILX was chosen and utilized to determine a strategy.
- **ALTERP(ILX)** : The age of the rule ILX.
- **HERKUP(ILX)** : The origine of the rule ILX whether it is generated by the machine (**HERKUP()** =1) or by the designer (any other value normally greater than 1 because the greater the value is, the more important the rule is supposed to be).

Rem :
- These parameters shall be read in the data file **'PRDBORB'** (N30), in case the learning process has already been applied (**IPRODU** = 1). Otherwise, these parameters are attributed default values.

The programm checks up whether the currently treated rule ILX, may not previously have been stored in **PRBMEM()**. However, the checking is not perfect, cause it only verifies whether two rules are identically written, when it is also possible that the same rule admits several ways of writting . It suffices to change the order according to which the unities are written, for example :
rule (A B C ——> X Y) <===> rule (B A C ——> X Y).

In case the learning module has already been applied (**IPRODU=1**), another data file should be read, namely **'PAWDAT'** (N35).
This file - Parameter Tabelle Datei -, contains for each rule which was punished - noted **STPRTA(IZPT)** -,
- the number of the meta-rule which is responsible for the

punishment : **BWPRTA(IZPT)**,
- the value of the punition : **STRATA(IZPT)**,
- the value of every parameters which may be taken into account in the building of the rules in the corresponding block : **PAWETA(ISPT,IZPT) ISPT<=ITI1** (=16 **'STEINI'**).

IZPT : the number of punition occurences shall not exceed **ITI2** (=999 **'STEINI'**).

III) THE FULLFILMENT OF THE WORKING MEMORY.

- Each time a strategy is to be determined, the subroutine **'STEUER'** of programm **'CONTROL'** is run.

- The first operation to be performed, is to write the present state of the network in an array,called **WRKMEM()**,(**WORKING MEMORY**), - which acts as the basis of acts for the expert-system -.

WRKMEM() is an integer array 100 lines × 5 columns.

- the first and the last columns are used to control the storing of information and its retrieving.
- the three columns in the middle, contains the actual information about the state of the network according to the following muster .

col 2	col3	col4
parameter	operator	value

- Subsequently, the information corresponding to one rule-unity (also called word = (parameter, operator, value)), is stored in one line of the working memory.
- The first column, contains the number of the previous inserted line in the working memory ,
- the last column the number of the following inserted line.

IWI : (= 100) is the maximal number of lines that the working memory could contain (determined in the programm **STEINI**)

EINIGE ERWÄGUNGEN ÜBER DIE VERSAGEN .

Die Frage lautet : Wie kann noch eine akzeptable Steuerungs-Strategie erfunden werden, wenn Störungen während der Übertragung der den Netzszustand beschreibenden Daten oder wenn Ausfälle in den Steuerungs-Einrichtungen festgestellt werden ?

Es soll ein Unterschied zwischen zwei Fällen eingeführt werden :

a) Der Wert eines Parameters wird nicht zu dem Zentralsteuerungs- System übertragen - Störungen der im Kommunikation mit der Zentrale.

b) Ein Wert-Ungültigkeit-Kennziffer wird übertragen
- Die Kommunikation lauft gut, aber die Werte, die übertragen werden sollen, dürfen nicht in Berücksichtigung eingenommen werden -.

Es gibt vielleicht noch einen anderen Fall, nämlich wenn ein Wert zu der Zentrale als gültig übertragen wird, der aber aus der Kenntnis der physischen Prozesse und der Eigenschaften des Netzes als unmöglich betrachtet werden kann.
- Um solch einen Fall zu berücksichtigen, sollten 'Konzistenzregeln' eingestellt werden-.

Der Unterschied zwischen beiden Fällen ist irrelevant, wenn es um Messungen geht.
Ob sie nicht übertragen worden sind oder ungültig sind, wird die gleiche Abarbeitung durchgeführt.

Der Unterschied gilt nur für Stellglieder (Pumpen, Schieber...), wenn es möglich ist , falls die Kommunikation mit der Zentrale gestört ist, eine unabhängige lokale Strategie für den entsprechenden Stellglied einzuschalten.
Diese lokale Strategie kann die Zentrale annehmen, wenn sie für die anderen Stellglieder eine geltende Strategie ersucht.

I) Es fehlt ein Regenmeßwert.

Zu verschiedenen Teilen des Netzes sind verschiedene Regendaten zugeordnet.

Wenn die Daten von einem Meßgerät fehlen, werden sie von den eines anderen ersetzt.

Die Wahl des ersetzenden Gerätes kann anhand verschiedener Erwägungen (wie zb der Entfernung der Geräte, der Topographie, statistichen Vergleichungen der Messungen...) getroffen werden.

Dieses Wissen kann durch das Schreiben von Ersetzungsregel in das Steuerungssystem eingefügt werden.

Wenn jedoch keine benutzbare Ersatzregendaten zur Verfügung stehen , wird im Augenblick keine explizite Abarbeitung vorgeschlagen.

- Das Steuerungsystem ist fähig, Regeln zu benutzen, deren Bedingungen nicht zu 100% erfüllt sind -.

II) Es fehlt einen Wasserzustand

Im Augenblick, die einzigen gebrauchten Höhewerte, sind die Becken- wasserstandwerte -um die entsprechenden Pumpenleistungen zu bestimmen-.

Höhe im Rückhaltebecken 1 = Höhe am Schacht 31002
Höhe im Rückhaltebecken 2 = Höhe am Schacht 26001

a) Wenn einer dieser Werte nicht gegeben worden ist, kann wie vorher erwähnt, einen Versuch unternommen werden, ihn zu ersetzen.

a.1) Das einfachste Verfahren besteht darin, den für den am nähsten gelegten Schacht geltenden Wert zu betrachten.

Schacht 31002 wird von Schacht 31001 ersetzt.
Schacht 26001 wird von schacht 26000 ersetzt.

a.2) Gibt es andere Möglichkeiten den fehlenden Wert zu ersetzen?

Bemerkung: Die Ersetzung kann auch durch das Einfügen von Regeln eintreten.

b) Man sollte anderseits untersuchen, ob man Steuerungsregeln für die Pumpen schreiben kann, deren Bedingungsteil die entsprechende Beckenhöhe nicht zu berücksichtigen benötigen.

III) Es gibt Schwierigkeiten an den Stellglied-Einrichtungen.

Es gibt mehrere Fälle zu unterscheiden.

a) Der Datenaustausch zwischen der Einrichtung und der Zentrale ist abgebrochen, ohne daß die wirklichen Leistungsfähigkeiten des Gerätes beschränkt worden sind.
Falls eine lokale Steuerung als Sicherheitsmaßnahme vorgeplant wurde, wird das System sie annehmen, um seine Entscheidungen in den anderen Teilen des Einzugsgebietes zu treffen.

b) Die Einrichtung ist nicht mehr steuerbar.
In diesem Fall sind auch mehrere Möglichkeiten zu berücksichtigen.

b.1) Gibt es noch Möglichkeiten das Gerät zu einem bestimmten Leistungszustand festzustellen -einem sogenannten Sicherheits--Zustand-.
Dann wird er von dem Zentralsystem für die weitere Erforschung der Strategie anderswo im Netz angenommen.

b.2) Gibt es keine solche Möglichkeiten, dann wird der letze Zustand des Gerätes als der gegenwärtige festgestellt.

Allgemeine Bemerkung:
Es sollten in diesem Fall Konzistenzprüfungs-Regeln eingestellt werden, um zu verifizieren, ob die entsprechenden Annahme eigentlich mit dem, was den wirklichen Zustand des Netzes

betrifft, übereinstimmen.

LISTE DER PROGRAMME UND IHR ZUSAMMENHANG