3. DUPROL

3.1 Language reference

The interactions involved without transport processes, advection and dispersion, need to be supplied by the quality model development part of the program. These are stored in the process description file *.mod. The resulting set of equations has to be compiled using DUPROL. After compilation a *.mob file is created which can be read by DUFLOW.

3.1.1 Syntax

DUPROL is not case sensitive.

Some rules of syntax:

- An *identifier* (name of a variable) starts with a character and consists of a maximum of six characters. Non alpha-numerical characters are not allowed. The name of variable *d* is reserved for the dispersion coefficient and should not be used.
- *Comments* start with /* and end with */.
- An *include statement* can be used to include parts of the DUPROL code from other files (syntax: #include “<filename>”, for <filename> the name of the file must be substituted)

A model file consists of the parts:

- Declaration section:
  In this part the different variables are defined.
- Compound statement for water courses:
  This part contains the equations describing the processes in the water courses.
- Compound statement for weirs, general structures and culverts:
  The mass transfer of gasses and volatile substances at these structres can be modelled. Although the concept was originally introduced to simulate the enhanced oxygen transfer at weirs, it also can be used to describe the release of volatile substances.
  This part is optional.
  The weirs part starts with the keyword ‘weirs’.

The *.mod file must be closed with an empty line at the bottom, otherwise DUPROL cannot compile the file correctly.

Choose to edit the model definition file if you want to develop the quality model. The program will ask which model definition file is to be edited. (*.mod). The

editor will be activated so the desired process description file can be entered. A shortened example of a model definition file is shown below.

```
/*     Simple Eutrophication Model  EUTROF1.MOD   DUFLOW                  */
/*                                                                         */
/*     Hans Aalderink & Nico Klaver                                        */
/*                                                                         */
/*                   Agricultural University of Wageningen                 */
/*                   Department of Nature Conservation                     */
/*                   Water Quality Management Section                      */
/*                   P.O. BOX 8080                                         */
/*                   6700 DD Wageningen                                    */
/*                   The Netherlands                                       */
/*                                                                         */
/*     September 1992                                                      */

water  A              [ 2.000]       mg-C/l         ;Algal biomass
water  PORG           [ 0.110]       mg-P/l         ;Organic Phosphorus
water  PANORG         [ 0.040]       mg-P/l         ;Inorganic Phosphorus
water  NH4            [ 0.300]       mg-N/l         ;Ammonia
water  NO3            [ 3.000]       mg-N/l         ;Nitrate
water  NORG           [ 0.800]       mg-N/l         ;Organic Nitrogen
water  O2             [10.000]       mg/l           ;Oxygen
water  BOD            [ 5.000]       mg-O2/l        ;BOD-5
water  SS             [ 5.000]       mg/l           ;Suspended Solids
parm   kp             [ 0.005]       mg-P/l         ;Monod constant Phosphorus
parm   kn             [ 0.010]       mg-N/l         ;Monod constant Nitrogen
parm   ealg           [ 0.016]       ug-Chl/l,m     ;Specific extinction chlorophyll
parm   e0             [ 1.000]       1/m            ;Background extinction
..
..
..

xt     sod            [ 1.000]       g-O2/m2.day    ;Sediment Oxygen Demand
xt     i0             [ 10.00]       W/m2           ;Surface Light Intensity
xt     t              [ 20.00]       oC             ;Temperature
xt     resf           [  0.50]       g/m2.day       ;Resuspension flux Suspended Solids
xt     pflux          [  0.00]       g P/m2,day     ;Phosphorus release flux fromsediment
xt     nflux          [  0.00]       g N/m2,day     ;Ammonia release flux from sediment

flow   z              [  2.00]       m              ;Water depth
flow   Q              [  0.10]       m3/day         ;Flow
flow   As             [ 10.00]       m2             ;Cross sectional Area
flow   Width          [1]            m              ;Width of structure
flow   Fallh          [1]            m              ;Fall height

{
       fdpano=1/(1+kpip*SS);
       PORTO=PANORG*fdpano;
       Chla=achlc*A;

       fn=MIN(PORTO/(PORTO+kp),(NH4+NO3)/(NH4+NO3+kn));
       etot=e0+ealg*Chla;
       ister=i0/is;
       fl=2.71*(exp(-1*ister*exp(-1*etot*z))-exp(-ister))/(etot*z);
       ft=tga^(t-20);
       Groei=umax*fn*fl*ft;
       Resp=kres*tra^(t-20)+kdie;
       k1(A)=Groei-Resp;
       ..
       ..

       Ptot=PORG+PANORG+A*apc;
       Nkj=NORG+NH4+anc*A;
       Ntot=Nkj+NO3;
}

Weirs
{
       /* Oxygen saturation is required for the calculation of the
          actual oxygen concentration at the end of the structure */
       O2S=14.652-0.41022*T+0.007991*T*T-0.000077774*T*T*T;
       rt(o2) = max(1,0.866+0.602*fallh-0.107*(q/width)^0.21*z^(-1.7)*fallh^0.06);
}
```

The include statement can be used in the following way:

```
#include "declarations.inc"
{
#include "oxy-nit.mod"

       Ntot=Nkj+NO3;

#include "sediment.mod"
```

```
}
weirs
{
#include "rearation.inc"
}
```

After editing the model file, leave the editor and choose to compile the model file. The program will translate the *.mod file into a *.mob file.

The *.mob file can be divided into a header, a section with declarations and the program section. The header contains the dimensions as defined by the water quality model. The section with declarations defines the substances that are used along with variables and parameters. The program part of the file contains the programming code of the quality model in reversed polish notation (RPN).

3.1.2 Declaration section

In the declaration section all different variables have to be defined. Five types of variables are distinguished.

water Water column state variables. The flow has affect on this type of variable.

bottom Sediment state variables. For these type of variables horizontal transport is omitted. Hence only the processes are calculated. Exchange between the sediment and the overlying water column should be described by the user. The flow has no effect on this type of variable.

xt External variables, which are space and/or time dependent.

parm Parameters, constants and coefficients used in the process equations.

flow Flow variables, supplied by the hydraulic part of the model. These variables differ from the other variables by the fact that the identifiers of these variables are built-in. The following identifiers are available:

Z	Depth of water (m)
Q	Flow (m^3/s)
As	Flow area (m^2)
Ab	Storage area (m^2)
ds	Section direction(degrees - 360^o, measured clockwise from the North)
dt	Quality time step (s)
dx	Half of the length of section (m)
V	Half of the volume of section (m^3)
Wf	Wind velocity (m/s)
Wd	Wind direction (degrees -360^o, measured clockwise from the North)
Fallh	Fall height (m) defined as the distance between the sill level of the structure and the water level down stream of the structure.
Hc	Thickness of the over flowing water
Width	The width of a structure

The latter three flow variables may only be used in the weirs section of the quality model. In the figure below the meaning of the several

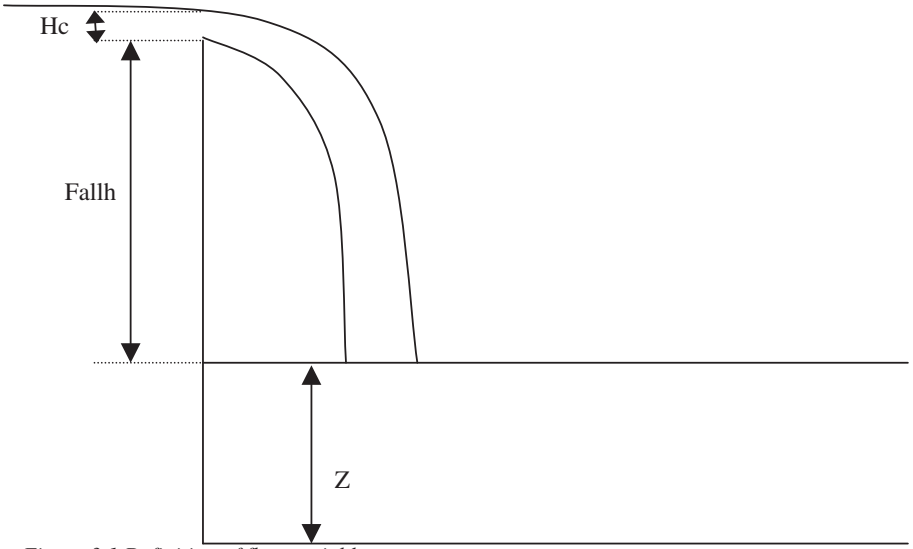

Figure 3.1 Definition of flow variables at structures

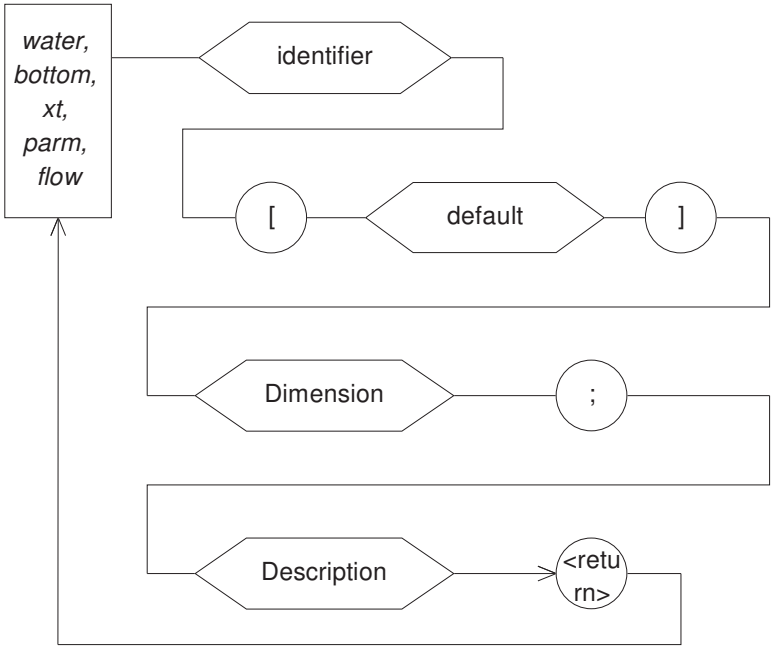

Figure 3.2 Syntax of the declaration section

Default	Default values are used if the user does not supply values within the other menus. For the state variable the default values are used if the user does not enter initial conditions. The default value has to be given in between brackets `[' and `]'.
Dimension	The unit of the variable may consist of a maximum of 10 characters, followed by `;'. Longer names will be truncated. The unit of time in rate constants should always be defined in days.

Description The description of the variable may not exceed a maximum of 40 characters. Longer lines will be truncated.

Below a part of the declaration section is given of the model description file.

```
/* Simple declaration section for explaining DUFLOW         */
/* remarks: No physical meaning              !!!            */
/*                                                          */

water   A      [ 2.000]  ug-C/l       ;Algal biomass
water   O2     [10.000]  mg/l         ;Oxygen
water   BOD    [ 5.000]  mg-O2/l      ;BOD
water   SS     [10.000]  mg/l         ;Suspended solids

parm    kp     [ 0.005]  mg-P/l       ;Monod-constant P algal growth
parm    kn     [ 0.010]  mg-N/l       ;Monod-constant N algal growth
parm    ealg   [ 0.016]  ug-Chl/l,m   ;Specific extinction algae
parm    e0     [ 1.000]  1/m          ;Background extinction of the water

xt      i0     [ 10.00]  W/m2         ;Irradiation
xt      t      [ 20.00]  oC           ;Temperature
xt      resf   [  0.50]  g/m2.dag     ;Resuspension  flux
```

3.1.3 Compound statement

3.1.3.1 Processes in sections

In this part the process descriptions have to be included. This section starts with `{' and should be closed `}'. All arithmetic expressions may be used (see next paragraph). The way the differential equations for the state variables should be entered needs some additional explanation.

For most state variables the kinetic derivative has the following form:

$$\frac{\partial C}{\partial t} = k_1 C + k_0$$

In this equation all first and zero order terms should be separated. For example the following equation:

$$\frac{\partial C}{\partial t} = k_a\left(C_s - C\right) - k_d L$$ should be rearranged like:

$$\frac{\partial C}{\partial t} = -k_a C + k_a C_s - k_d L$$

Internally lumped first and zero order coefficient are used, which should be defined by the user as:

$$k_1(C) = -k_a$$

$$k_0(C) = +k_a C_s - k_d L$$

If the k_1 and k_0 coefficient are not defined they will be set equal to zero.

For non state variables a function identifier is used. The declaration of these type of variables is implicit, which means that they may not be declared in the declaration section.

3.1.3.2 Rearation at structures

3.1.3.3 Compound statements

A compound statement consists of statements. Two types of statements are available:

- formula — See section 3.1.3.4
- if-statement — See section 3.1.3.5

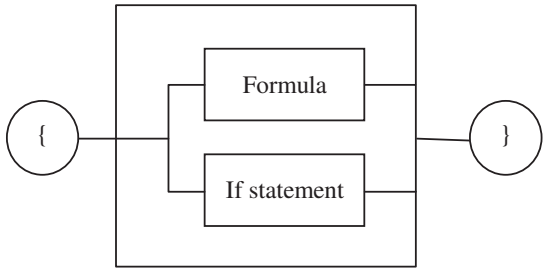

Figure 3.3 Syntax of a compound statement

3.1.3.4 Formula

The general syntax of a formula is shown in

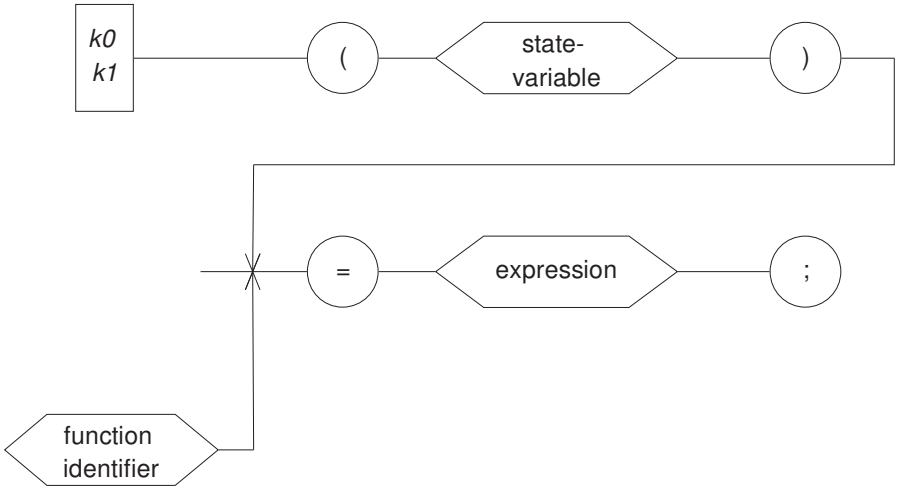

Figure 3.4 Syntax of Formula

k1	First order reaction coefficient.
k0	Zero order reaction coefficient.
state variable	Water or bottom state variable (maximal 6 characters).
function identifier	Identifier of a function (maximal 8 characters).
Expression	Definition of a function. A formula, defining the function identifier at right side of the equation. This formula consists off regular mathematic operators completed with several built-in functions . A function identifier is not allowed to appear in the right side of his own definition. A function identifier must

already have been defined before it can be used in the right side of an equation. The following operators are available (in order of priority):

Arithmetic :	Built -in functions:
()	sin(x)
^ (to invoke)	cos(x)
*	tan(x)
/	exp(x) (e^x)
+	ln(x)
-	log(x) (^{10}log(x))
	abs(x) (| x |)
	min(x1,x 2,..., xn)
	max (x1, x2,... ,xn)
	int(x) (truncate x to integer value)
	Round(x) (round x to integer value)
	asin(x) (arc sine(x))
	acos(x) (arc cosine(x))
	atan(x) (arctangent(x))
	rnd (random value)

3.1.3.5 If-statement

DUPROLcontains a flow-control statement. The syntax of this statement is shown in Figure 3.5.

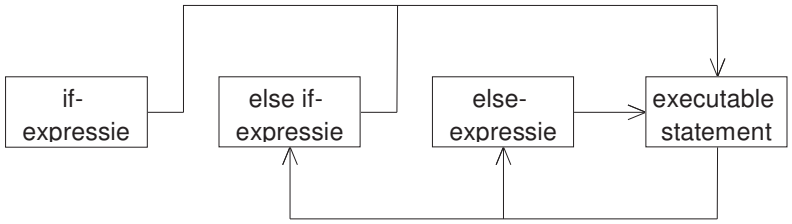

Figure 3.5 *Syntax if statement*

The *if statement* can be applied in several ways. The most common use of the statement will be shown in the following examples.

Example 1

Most basic formulation. If the condition in '(' and ')'is true the statement between the '{' and '}' will be executed.

```
if (NO3==0.0)
{
       pnh4=0.0;
}
```

Example 2

Example 1 can be extended with an alternative executable statement if the condition is false.

```
if (NO3==0.0)
{
       pnh4=0.0;
}
else
{
       pnh4= NO3/NH4*Kmn;
}
```

Example 3

If statements can be applied in several complicated situations, even nesting is allowed.

```
if (NO3==0.0)
{
       pnh4=0.0;
}
else if (NH4==0.0)
{
       pnh4= NH4/NO3*Kmn;
       if ((Kop>=1.0) && (Lap!=0.0))
       {
              Mip=pnh4*Kmn/2+0.001;

       }
}
else
{
       pnh4=NO3/NH4*Kmn;
}
```

General remarks concerning the use of if statements :

- Conditions can be defined using the following relational and logical (in order of priority):
  >
  >=
  <
  <=
  == (is equal to)
  != (unequal to)
  ! (not)
  && (and)
  || (or)
- Also in an if statement every executable statement must be closed with a ';'.
- Between the '{ }' the user can define a block of executable statements (= compound statement).