



ScorePP is a Specific Targeted  
Research Project (STREP)  
funded by the European  
Commission under the Sixth  
Framework Programme

# ScorePP



## Mathematical formulations of pollution generating processes

Deliverable No: D7.1, Date: 07 April 2008, made public 10 May 2010

Dissemination level: PU

*Webbey De Keyser<sup>1</sup>, Veerle Gevaert<sup>1</sup>, Lorenzo Benedetti<sup>1</sup>, Frederik Verdonck<sup>1</sup>, Luca Vezzano<sup>2</sup>, Peter Steen Mikkelsen<sup>2</sup>, Peter A. Vanrolleghem<sup>3</sup>*

<sup>1</sup> Department of Applied Mathematics, Biometrics and Process Control, Ghent University

<sup>2</sup> Department of Environmental Engineering, Technical University of Denmark

<sup>3</sup> Département de génie civil, ModelEAU

**Source Control Options for Reducing Emissions of Priority Pollutants (ScorePP)**

Sixth Framework Programme, Sub-Priority 1.1.6.3, Global Change and Ecosystems

Project no. 037036, [www.scorepp.eu](http://www.scorepp.eu), Duration: 1 October 2006 – 30 March 2010

<b>Deliverable number:</b>	D7.1
<b>Deliverable title:</b>	Mathematical formulations of pollution generating processes
<b>Authors:</b>	Webbey De Keyser Veerle Gevaert Lorenzo Benedetti Frederik Verdonck Luca Vezzano Peter Steen Mikkelsen Peter A. Vanrolleghem
<b>Review and Assessment</b>	Luca Vezzano Peter Steen Mikkelsen
<b>Date submitted to project coordinator:</b>	2008-04-07
<b>Approved by (Work package leader) :</b>	2008-04-07
<b>Approved by (Project Coordinator):</b>	2010-05-10

### Abstract

This task forms the link between the PP release data that were collected in WP 3 and the dynamic models used in WP 7. It provides a software tool in which mathematical and logical functions are implemented to automate the generation of PP release time series, based on the release pattern descriptions in Deliverable 3.3. The generated time series are stored in files that can directly be fed to the integrated dynamic model developed in Task 7.4. The software tool was also extended to handle vectors of dependent components. Like this, the tool can be used to generate time series for generic pollutants (BOD, COD, N, P) next to priority pollutants. Implemented features include stochastic functions, time series generating functions and user interfaces for data editing.

### Acknowledgement

The presented results have been obtained within the framework of the project ScorePP - “Source Control Options for Reducing Emissions of Priority Pollutants”, contract no. 037036, a project coordinated by Institute of Environment & Resources, Technical University of Denmark within the Energy, Environment and Sustainable Development section of the European Community’s Sixth Framework Programme for Research, Technological Development and Demonstration.

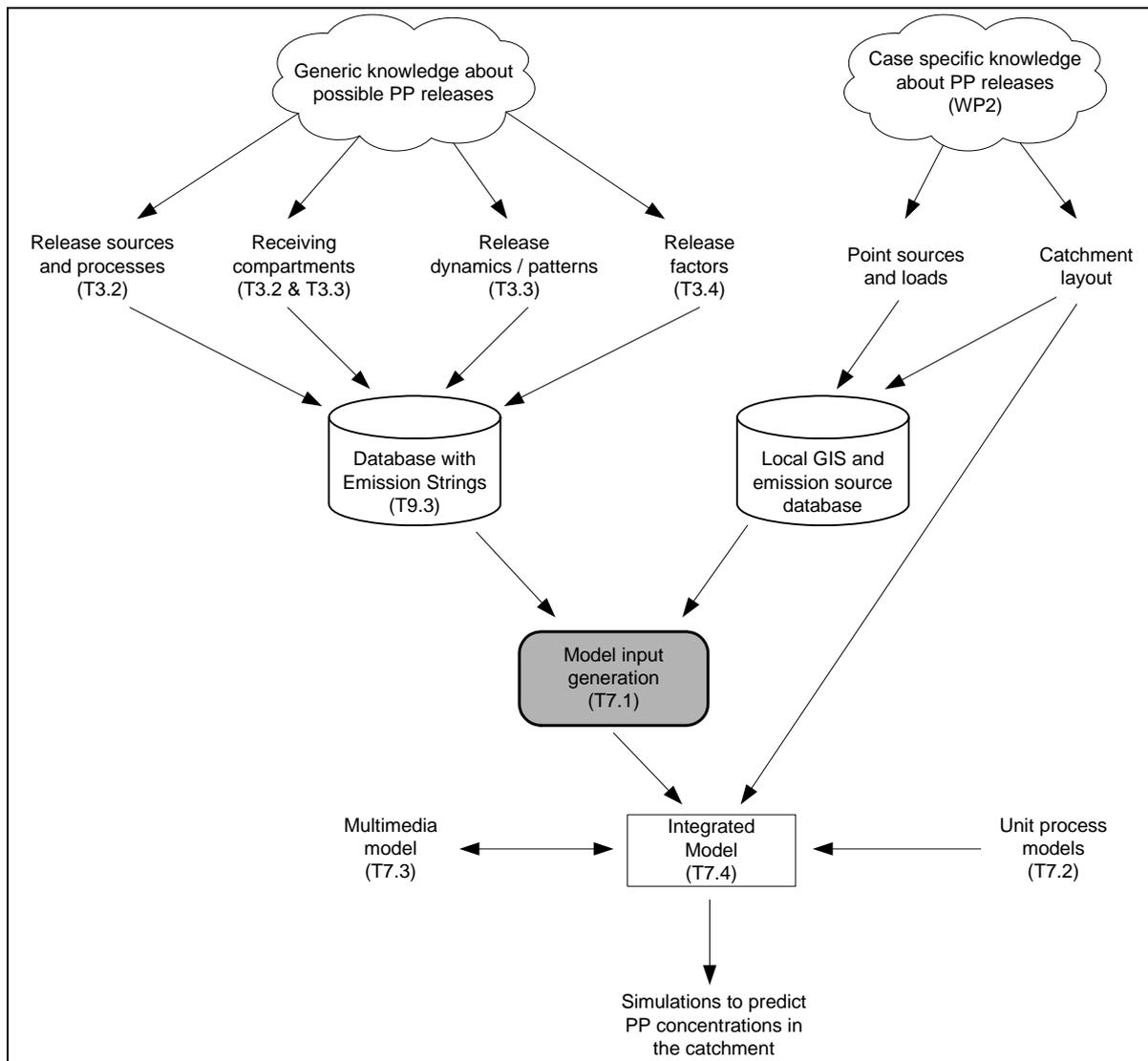
## Table of Contents

1	Introduction .....	2
1.1	The need for mathematical formulation of PP release dynamics.....	2
1.2	Subtasks .....	3
2	Methods.....	4
2.1	Dynamic model input generation – a brief literature review .....	4
2.2	PP release patterns .....	4
2.3	Deterministic and stochastic models.....	7
3	Development of a model input generator .....	9
3.1	Programming environment .....	9
3.2	Final aim of the input generator.....	9
3.3	Implemented functions.....	9
3.3.1	<i>Patterns</i> .....	9
3.3.2	<i>Linear combinations of patterns</i> .....	9
3.3.3	<i>Stochasticity</i> .....	10
3.3.4	<i>Normalization</i> .....	10
3.3.5	<i>Multiple occurrences of a pattern</i> .....	10
3.4	Dynamics of generic pollutants.....	10
4	The creation of PP release time series: an illustrated example.....	12
5	The creation of generic pollutants release time series.....	17
6	Conclusions .....	19
7	References .....	20
	Appendix: Implemented functions for the creation of daily, weekly and yearly patterns.....	21
A.	Daily patterns .....	22
B.	Weekly patterns.....	29
C.	Yearly patterns .....	31

# 1 Introduction

## 1.1 The need for mathematical formulation of PP release dynamics

In this deliverable, the translation of pollution generating process dynamics into “mathematical functions” is described. This is necessary to be able to use in a quantitative way the information gathered in WP 3. Figure 1 shows the relationship between some of this project’s tasks, in order to situate this deliverable (D7.1) in the project context.



**Figure 1. The relations between Task 7.1 and other tasks.**

The final aim is to obtain an integrated model (Task 7.4) that can be used to run simulations on the fate of PPs in a catchment and on the impact of possible source control options. The model is a compilation of unit process models (from Task 7.2) and interacts with the outer environment by a multimedia model (Task 7.3). Either a virtual catchment or a specific case can be modelled. Whatsoever, dynamic model inputs are needed to run simulations. One of these model inputs is a time series of PP releases (loads) to the different model compartments at discrete time steps.

In WP 3, knowledge about priority pollutants properties, sources, release dynamics and release factors was gathered and stored in a database in the form of emission strings (ESs – a practical example of these ESs is shown in Table 2 of this Deliverable). This qualitative information must be translated into generic time series. Then, by combination with case specific quantitative data (yearly loads) on the release of a certain PP by certain emission points or sources, the needed time series can be created.

## 1.2 Subtasks

In this task, several subtasks could be distinguished.

First, the generic release dynamics that were identified and stored as release patterns in a database in WP 3 (see further in §2.2), had to be translated into mathematical functions.

Second, when doing a case study, to each PP release point (or source) a certain release pattern and a certain PP load (e.g.; kg PP released by that source per year) have to be associated. Although this work is case specific, it should also be taken into account in Task 7.1 since it is a necessary step in the creation of dynamic model input files. The default parameters assigned to the release patterns will not be applicable in all cases. Therefore, it should be possible to change their values for case specific PP release point/source.

Third, when all the needed information is present, the time series can be created. Every time series has to be normalised so that the integral is equal to one.

Fourth, for each PP release point, the normalised time series should be multiplied with the point-specific PP load to obtain a point-specific PP release time series. In case this specific load is not available, a default release factor from Task 3.4 should be used to calculate a default load.

And finally, all individual point-source time series are to be summed per compartment and per sub-catchment in order to obtain the total dynamic PP release pattern to a certain model compartment in that sub-catchment.

As is explained further, a stand alone software tool was developed in the framework of this task to automate these steps, so that the generation of a case specific time series can be performed in less time and with less (human) mistakes.

## 2 Methods

### 2.1 Dynamic model input generation – a brief literature review

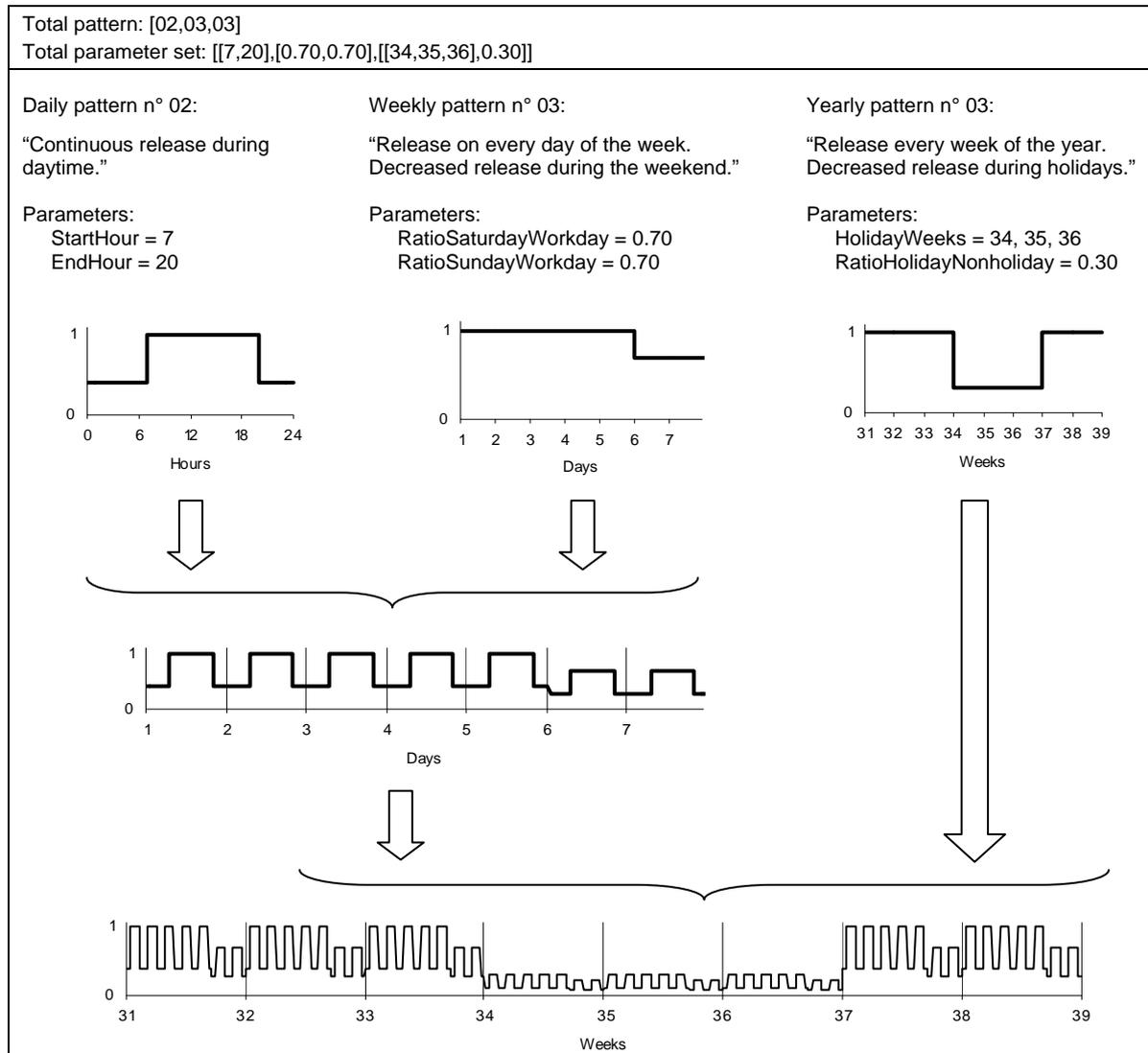
In literature, different techniques can be found to generate a dynamic model input. For the Benchmark Simulation Model N<sup>o</sup>2, a WWTP influent wastewater generator developed by Gernaey *et al.* (2005) and used by Jeppsson *et al.* (2006) included four entities: (1) diurnal phenomena modelled using a second-order harmonic function, (2) seasonal phenomena, (3) weekend effects and (4) holiday effects, both modelled with lower flow rates and pollutant loads. Because this generator directly modelled the treatment plant influent, also infiltration of groundwater into the sewer, urban drainage phenomena, rain and storm weather and first flush events were included in the generator. In Task 7.4 of this project, however, the aim is not to model only the WWTP but the whole catchment. The WWTP influent will therefore be composed in the model itself, by integrating sub-models like the draining catchment, the sewer network, a soil drainage model, etc. A second substantial difference between the BSM influent generator and the one built in this task, is that the BSM influent is meant to reflect an average WWTP influent, focussing on hydrodynamics and on the generic pollutants N, P and COD/BOD, while the input file generator of this project is meant to reproduce the release of priority pollutants by different sources in a catchment, which have different characteristics than the generic pollutants.

Although the aim of this project is different, the idea of including diurnal phenomena, weekend effects, holiday effects and seasonal effects was adopted also in this project to describe the PP release dynamics (see Deliverable 3.3). A very similar approach of multiplying a daily, weekly and yearly pattern can be found in Gernaey *et al.* (2005) to generate a typical household wastewater flow rate pattern.

Another interesting study is the modelling of benzotriazole discharges by household dishwashers in a catchment by Ort (2006). The discharge activity of the individual dishwashers was modelled as block shaped peaks that transform to Gaussian peak shaped pulses when they are transported in the sewer network. The peak broadening (parameter  $\sigma$ ) was calculated as a function of the travel time from a particular household to the main sewer. The occurrence time of the discharge and the initial discharge duration were randomly sampled from certain intervals. Calibration of the model was performed by means of a Monte Carlo experiment. As the discharges only took about 1 to 2 minutes, the temporal resolution used in this study was maximum 1 minute. For the ScorePP project, this resolution would be computationally too intensive. Moreover, such a resolution would not be relevant for the effects that are modelled (PP fate and concentrations in a catchment over a period of several years). Although the time resolution of that study differs from the one in this project, the idea of modelling individual random peaks and then summing all individual time series to obtain a release pattern for a whole group of households was adopted and incorporated into the input file generator.

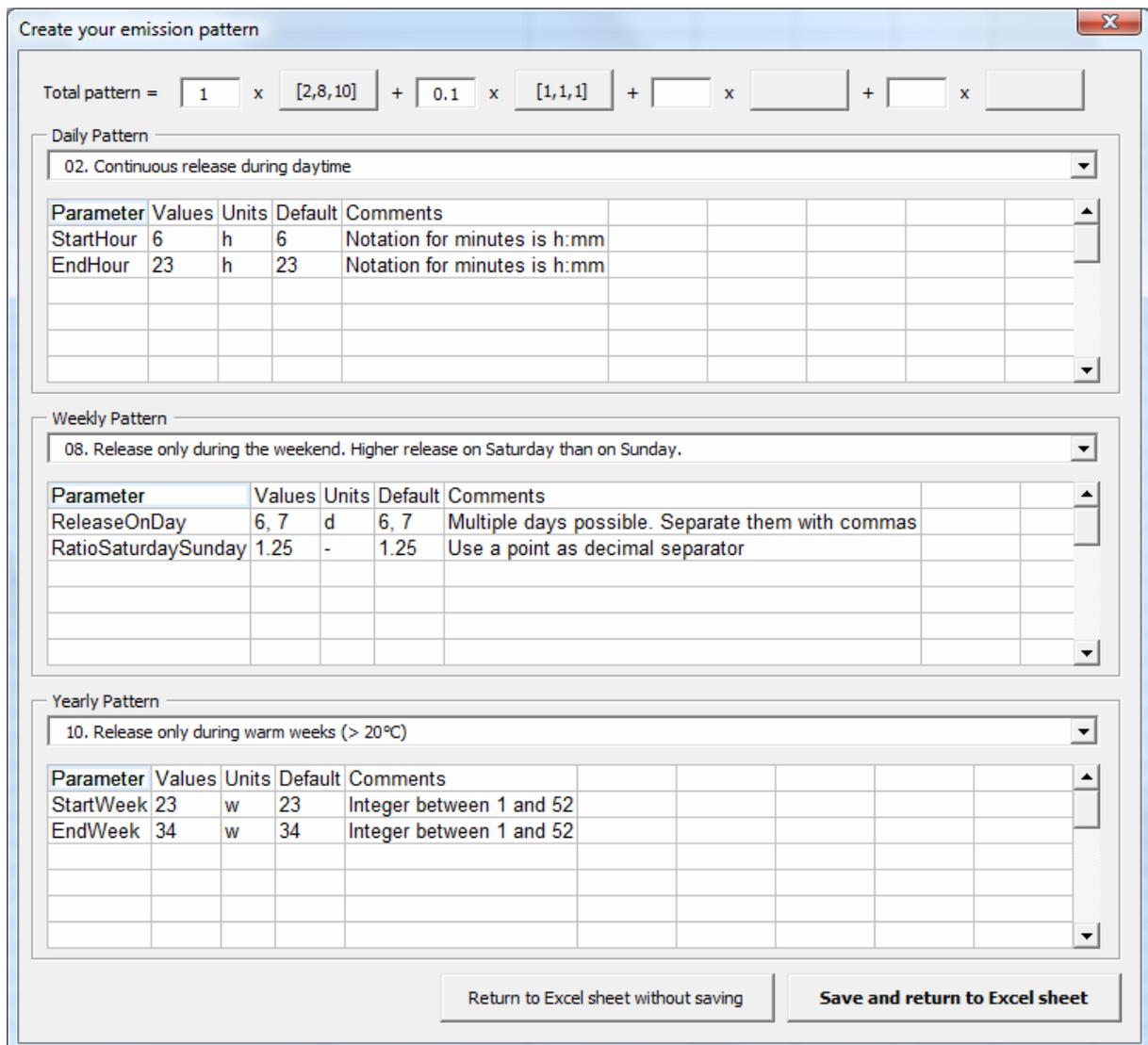
### 2.2 PP release patterns

Figure 2 illustrates the formulation of release patterns and how specific quantitative release time series are created out of generic pattern descriptions. In Task 3.3, the release dynamics of the 67 PPs were described by generic release patterns. These patterns consist of 3 entities: daily, weekly and yearly patterns. Diurnal effects are included in the daily pattern, which describes how the PP release fluctuates during a 24 hour period. The weekly pattern takes into account phenomena like the “weekend effect”. The yearly pattern describes the yearly fluctuations in PP release magnitude due to e.g. seasonal variations or “holiday effects”. Each daily, weekly and yearly release pattern has several parameters. (For a complete list of those parameters, see D3.3.)



**Figure 2.** An example of the combination of a daily, weekly and yearly pattern to obtain a one year time series. To keep the graph readable, only weeks 31 to 39 are shown instead of weeks 1 to 52.

To each of the PP sources identified in Task 3.2, a release pattern was assigned and stored in the database. These specific release patterns are a combination of a generic daily, weekly and yearly release pattern, with a certain set of applicable parameters. The release pattern is stored in the database as a vector with three elements:  $[DD, WW, YY]$ , in which  $DD$  is the daily pattern identifier,  $WW$  the weekly pattern identifier and  $YY$  the yearly pattern identifier. Each pattern also requires a set of pattern-specific parameters. These parameters are also stored in the database in a well defined vector format. To facilitate this database input, a VBA macro was developed as a tool in Excel to serve as a graphical user interface (see Figure 3).



**Figure 3.** A screenshot of the graphical user interface developed in Task 3.3 to create and edit release patterns.

In this task, the pattern vector was extended with a fourth “multiyear” pattern, to reproduce e.g. the decision to phase out the production or use of certain chemicals over a period of several years. This multiyear component was not needed in WP3 but is useful for scenario analysis in WP7. Table 1 gives an overview of the implemented multiyear patterns. If not specified, multiyear pattern number 1 (constant release) is applied by default.

**Table 1.** Possible multiyear patterns with their parameters and default values.

ID	Text	Parameters	Units	Default value
01	Constant release	-	-	-
02	Release fading linear from 1 to a certain value	FinalValue StartOfFadingPeriod EndOfFadingPeriod	- y y	0 0 1

With this information about patterns, a quantitative release time series must be created to feed the integrated model of Task 7.4. For this purpose, a stand-alone application was developed. The development and functioning of this application is explained in section 23 of this deliverable.

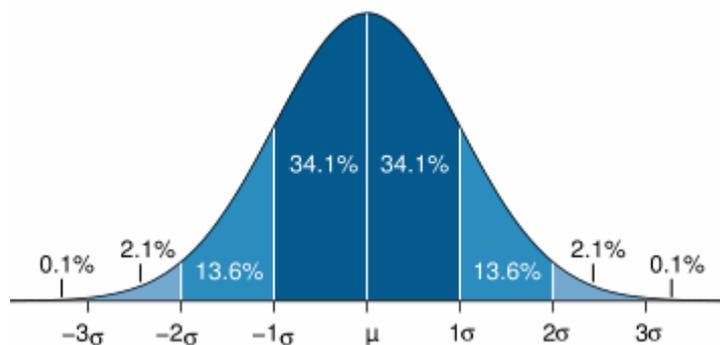
### 2.3 Deterministic and stochastic models

Most of the release patterns developed in Task 3.3 are deterministic. This means that the PP release is well defined both in time and magnitude. E.g. when a pattern with an alternation of release and no release is assigned to a certain Emission String (i.e. the combination of a PP and the process or product by which the PP is emitted or released), the magnitude of the PP release and the points in time on which the release starts and stops will be equal every time the pattern is repeated. Because this does not resemble natural conditions, randomness or stochasticity could be introduced in several ways.

A first way to include stochasticity in the deterministic patterns is to add random noise to the obtained time series. This noise can be generated on every time step as a random number sampled from a uniform distribution within given boundaries. The uniform distribution implies that the chance of a number to be sampled is equal for all values in the interval. With this kind of noise, the answer to the question about which boundaries should be set becomes important, as these boundaries will determine the hard limits of the release flux. In most patterns, it will be very difficult to determine these uncertainty boundaries. Therefore, it was decided to include the possibility of applying a normally distributed noise, also called Gaussian White Noise, on the obtained time series by setting a magnitude for the noise, called  $A_{\text{noise}}$  (e.g. if the values in the time series are subject to a maximum variation due to noise equal to  $\pm 10\%$ , the value of  $A_{\text{noise}}$  would be 0.1). At each time step, a value for this noise is sampled from the normal probability distribution with mean  $\mu = 0$  and standard deviation  $\sigma = A_{\text{noise}}/3$ , because in the normal distribution almost the whole range of possible values is comprised within the interval  $\mu \pm 3\sigma$  (see Figure 4). Such sampled value is then added to the value in the time series:

$$x_{\text{with noise}} = x_{\text{without noise}} \times (1 + p)$$

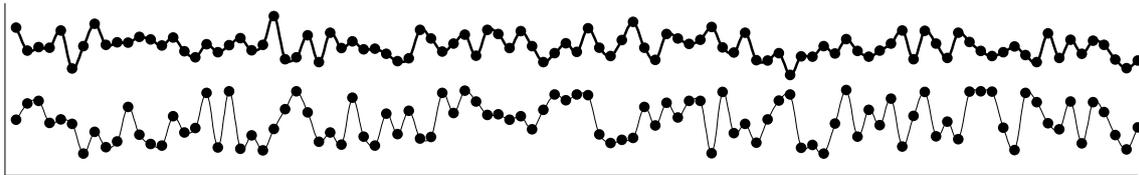
with  $p$  a random sampled value from the distribution  $N(0, \sigma^2)$ , so 99.8% sure to be in the interval  $[-A_{\text{noise}}, A_{\text{noise}}]$ .



**Figure 4.** Normal distribution (mean  $\mu$ , standard deviation  $\sigma$ ), which shows that 99.8% of the sampled values will be between  $\mu - 3\sigma$  and  $\mu + 3\sigma$ .

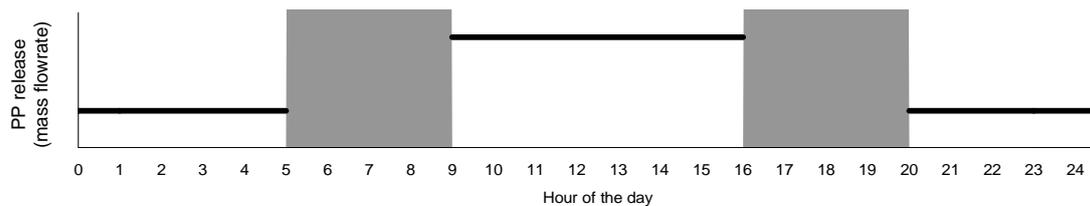
From Figure 5 it can be seen clearly that the normally distributed noise (upper line) shows less sudden variations compared to the uniformly distributed noise (lower line), although the limits (minimum and maximum value of the noise) are quasi the same. An extra step to minimise the occurrence of sudden oscillations could be the application of a moving window average function,

like a Kalman filter. An alternative is to generate noise by means of a random walk, in which the noise value on moment  $t$  is defined as the noise value on moment  $t-1$  plus a random number. In these cases however, the purpose of applying random noise could in certain cases be undermined. As an example, assume that a pattern defines that a certain PP releasing process takes place at a random time instance every Saturday of the year (the pattern definition could be  $[10,7,1,1]$  with parameters  $[[9,20,1,1.00,block,0:15],[6],[],[]]$ ). Without applying noise, the same quantity of PP would be released every Saturday. If the process concerns a human source for the PP release, this does not resemble reality. But neither would be setting the emitted quantity in one week function of the emitted quantity in the previous weeks, so applying a Kalman filter would not be appropriate in this case.



**Figure 5.** Two series of random numbers connected with curves. The lower series was sampled from the uniform distribution with interval  $[-3, 3]$ , while the upper series was sampled from a normal distribution with mean 0 and standard deviation 1.

A second (complementary) way to provide stochasticity is to create a dynamic release pattern which is not completely predefined but which allows certain parameters to be randomly sampled from a specified range. An example is the daily pattern number 23 (see D3.3), which is a diurnal pattern with switch times (parameters *StartOfDay* and *StartOfNight*) that are randomly sampled from the uniform distribution with mean *StartOfDay* and *StartOfNight* and with interval width equal to the value of the parameter named *Fuzziness* (indicated by the grey zone in Figure 6).



**Figure 6.** Example of fuzziness in the diurnal pattern number 23: the parameters *StartOfDay* and *StartOfNight* are sampled randomly from the grey intervals  $[7-2,7+2]$  and  $[18-2,18+2]$ .

### 3 Development of a model input generator

#### 3.1 Programming environment

The most efficient and least time consuming programming environment to develop the dynamic input generator would probably be Matlab®, because of its built-in features for numeric data handling, mathematical processing and graphical data visualisation. These advantages were balanced against the idea of having a stand alone application – for which no expensive license is needed – with a user friendly Graphical User Interface (GUI).

Finally, it was decided to create an independent application developed in Visual Basic .NET, which would serve as a GUI to create and edit PP release time series. Functions, patterns and default parameters were not hard coded but were stored in separate tab delimited text files, to preserve the tool's transparency and flexibility in case patterns would be edited or added to the pattern library.

#### 3.2 Final aim of the input generator

The time series of PP mass fluxes at discrete time steps serve as model inputs in Task 7.4. The modelling and simulation software used in Task 7.4 is WEST (MOSTforWATER, Kortrijk, Belgium; Vanhooren *et al.*, 2003). This modelling environment requires one or more tab delimited input text files in which one column contains the time values and another column contains the mass fluxes of the PP at those time steps. At runtime, the discrete input values can be kept constant until the next time step or they can be linearly interpolated at each time instance in the calculations. The input generator must thus be able to construct a time series of PP mass fluxes to each modelled compartment (air (A), groundwater (G), sewer (WI), urban impervious surface (UI), urban pervious surface (UP), surface water (WD)).

#### 3.3 Implemented functions

##### 3.3.1 Patterns

Each of the release patterns created in Task 3.3 (see §2.2) must be translated into a mathematical formulation. First, a vector of time instances for a certain simulation period (e.g. one year) is generated with the temporal resolution specified by the user (e.g. half an hour). To each time instance a PP mass flux has to be assigned. Most patterns can be implemented with simple if-then-else statements. E.g. for continuous patterns, each time instance in the time step vector is compared to the Start and End parameters. The resulting PP flux is set to 1 if the time step is contained within the interval [Start,End[ and to the value 0 in the other case.

A complete list of daily, weekly and yearly pattern functions can be found in the appendix.

##### 3.3.2 Linear combinations of patterns

In Task 3.3, the possibility was created to define a certain PP release pattern as a linear combination of two or more different release patterns. E.g. the total pattern “9[01,02,01,01]1[10,07,01,01]” divides the total PP load between pattern [01,02,01,01] (continuous release except on Friday afternoon) and pattern [10,07,01,01] (peak loads on Friday afternoon) with respective coefficients of 90% and 10%.

The input generator application reads the pattern string and splits it into two sub-patterns. It first generates the two individual time series, normalizes them (see §3.3.4), then multiplies them with the factors 9 and 1 respectively, sums both scaled time series and finally normalizes the total time series again.

### 3.3.3 Stochasticity

As explained in §2.3, two kinds of stochasticity can be distinguished. First, in several patterns not all parameters were fixed but some randomness was allowed. This is implemented as a function that samples a random value from the uniform distribution with a certain lower and upper bound for the parameter to be determined. The random number is generated by the built-in random number generator with a time-based seed value.

The second form of stochasticity is the addition of normal distributed random noise to the created time series. As explained in §2.3, the standard deviation for the zero-mean normal distribution is one third of the specified noise magnitude, which is entered as a percentage of the actual output value. Two modes to enter the noise magnitude were provided: as a global percentage applicable to all created patterns, or as a percentage specific to each PP source.

### 3.3.4 Normalization

After each (sub)pattern is created, it is normalized to ensure that the integral of the total yearly pattern equals one. The one year time series is then simply multiplied with the yearly PP load for that particular source. Normalization is obtained by dividing each element of the time series vector by the total sum of all values.

### 3.3.5 Multiple occurrences of a pattern

When modeling a catchment with a certain number of e.g. households, a dynamic release pattern can be assigned to each of the households individually or a pattern can be created for the whole group of households. In the latter case, the households are not considered to be separate emission points but e.g. a whole neighborhood which is connected to the same sewer network is considered to be one point source, just like the household point source is an accumulation of the toilet, dishwasher, wash basin, shower, etc.

As the application is designed to create patterns for each emission point source in a GIS, both possibilities exist, depending on the precision of the spatial data. On the other hand, the dynamic release patterns for the PPs are as much as possible collected from a database. The situation might occur that the GIS treats a certain number ( $X$ ) of households as a group, while the pattern stored in the database is applicable to a single household instead of a group. In this case the pattern generator should generate the pattern  $X$  times (each time with stochasticity, if present) and sum those  $X$  patterns to obtain the time series for the group of emission sources, this way reproducing a stochastic behavior similar to the one of the real system. For these cases, the possibility was built in to repeat the pattern creation a number of times, specified by the value of a field (e.g. in Table 2 Table 3 called "Occurrences") which is set to 1 by default.

## 3.4 Dynamics of generic pollutants

To run dynamic simulations on the fate of PPs in a catchment, not only dynamic release time series for priority pollutants are needed, but also dynamic input series of water and "generic" pollutants N, P, COD and BOD. These time series can also be created with the input generator. For this, the yearly pollutant load must be entered as an array of values instead of a single value, as is the case for PPs. The first element in the array represents the water flow rate in  $\text{m}^3/\text{year}$ , the next elements in the array represent the concentrations of Biochemical Oxygen Demand after 5 days ( $\text{BOD}_5$ ), total Chemical Oxygen Demand ( $\text{COD}_t$ ), soluble COD ( $\text{COD}_s$ ),  $\text{NH}_4\text{-N}$  and  $\text{PO}_4\text{-P}$ , all expressed in  $\text{g}/\text{m}^3$ . The input generator recognises the array format and automatically creates 6 time series for each source instead of one. The water flow rate time series is created first, according to the specified pattern. Next, the pollutants' time series are created by rescaling the water pattern and applying Gaussian white noise on it. So next to the loads array [ $\text{H}_2\text{O}(\text{m}^3/\text{year}), \text{BOD}_t(\text{g}/\text{m}^3), \text{COD}_t(\text{g}/\text{m}^3), \text{COD}_s(\text{g}/\text{m}^3), \text{NH}_4\text{-N}(\text{g}/\text{m}^3), \text{PO}_4\text{-P}(\text{g}/\text{m}^3)$ ], also a noise amplitude array is needed: [ $A_{\text{H}_2\text{O}}, A_{\text{BOD}_t}, A_{\text{COD}_t}, A_{\text{COD}_s}, A_{\text{NH}_4\text{-N}}, A_{\text{PO}_4\text{-P}}$ ]. A

clustering algorithm was implemented so that summation of the time series per catchment and per compartment is performed separately per array element. The final result is a file for each sub-catchment with 7 columns per compartment: Time (d), H<sub>2</sub>O (m<sup>3</sup>/d), BOD<sub>t</sub> (g/d), COD<sub>t</sub> (g/d), COD<sub>s</sub> (g/d), NH<sub>4</sub>-N (g/d), PO<sub>4</sub>-P (g/d). In this format, the file can directly be fed to the integrated dynamic model of T7.4.

#### 4 The creation of PP release time series: an illustrated example

In this section, the use of the application will be illustrated with a practical example of naphthalene releases in a small sub-catchment. A second illustrated example concerning Mercury releases can be found in De Keyser *et al.* (2008).

Assume that the project's database contains the Emission String Sentences on naphthalene listed in Table 2.

**Table 2. Emission String Sentences considering the release of naphthalene (CAS 91-20-3).**

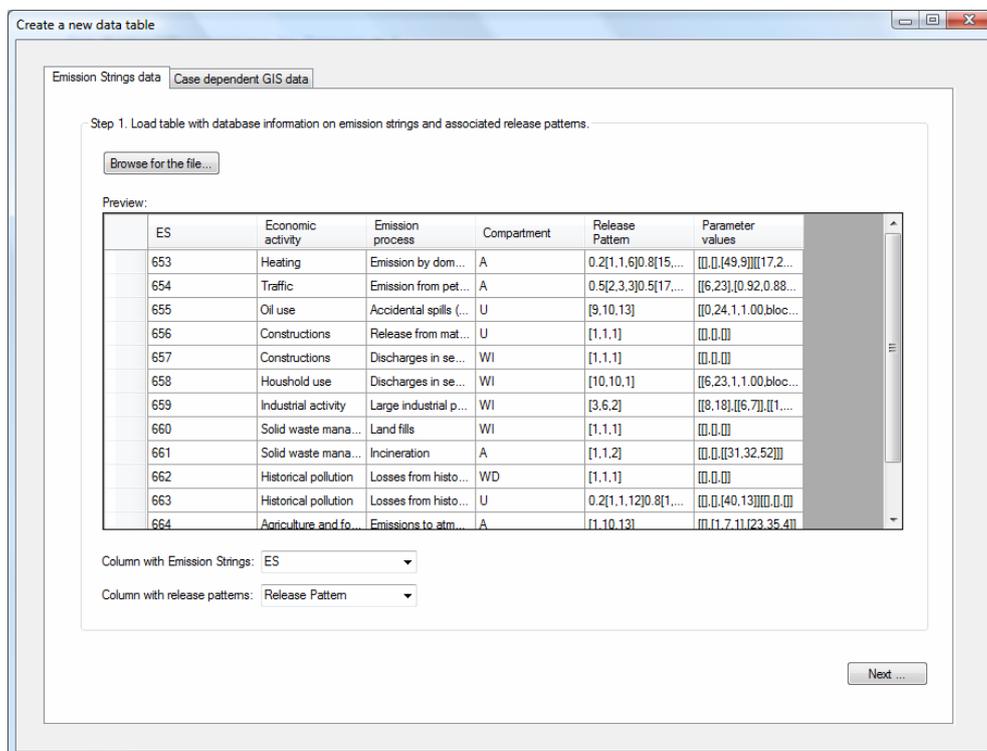
ES_ID	Economic activity	Emission process	Comp.	Release Pattern	Parameter values
653	Heating	Emission by domestic wood burning	A	0.2[1,1,6]0.8[15,1,6]	[[[],[],[49,9]][[17,21,23,block],[],[49,9]]
654	Traffic	Emission from petrol, diesel, ethanol mixtures and tyres	A	0.5[2,3,3]0.5[17,3,3]	[[6,23],[0.92,0.88],[[1,30,31,32,33,34,52],0.75]][[7,8,9,15,17,19,1.25,triangular],[0.5,0.5],[[1,30,31,32,33,34,52],0.6]]
655	Oil use	Accidental spills (via oil)	U	[9,10,13]	[[0,24,1,1.00,block,0:30],[1,5,1],[1,52,1]]
656	Constructions	Release from materials and constructions in non-sewered areas	U	[1,1,1]	[[[],[],[]]]
657	Constructions	Discharges in sewage effluents or storm water as a result of run-off from buildings and constructions in paved urban areas. Use of creosote and concrete plasticiser. Use in pyrotechnics/grinding wheels	WI	[1,1,1]	[[[],[],[]]]
658	Household use	Discharges in sewage effluents or storm water as a result of households, consumer use (e.g. in mothballs)	WI	[10,10,1]	[[6,23,1,1.00,block,0:15],[1,7,2],[[]]]
659	Industrial activity	Large industrial point sources, direct or via STP (IPPC installations for the manufacture of other organic basic chemicals)	WI	[3,6,2]	[[8,18],[[6,7]],[[1,30,31,32,33,34,52]]]
660	Solid waste management	Landfills	WI	[1,1,1]	[[[],[],[]]]
661	Solid waste management	Incineration	A	[1,1,2]	[[[],[],[[31,32,52]]]]
662	Historical pollution	Losses from historically contaminated soils. Historical pollution of sediments	UP	[1,1,1]	[[[],[],[]]]
663	Historical pollution	Losses from historically contaminated soils. Historical pollution from contaminated land. Use of creosote.	UP	0.2[1,1,12]0.8[1,1,1]	[[[],[],[40,13]][[[],[],[]]]]
664	Agriculture and forestry	Emissions to atmosphere. From agriculture and forestry (forest fires)	A	[1,10,13]	[[[],[1,7,1],[23,35,4]]]
665	Industrial activity	Emissions to atmosphere. From industry IPPC categories	A	[1,6,2]	[[[],[[6,7]],[[1,30,31,32,33,34,52]]]]

Further we assume that the GIS with spatial data on the studied sub-catchment (as provided by WP 6) provides the information shown in Table 3. The ES\_ID field couples the data of Table 3 with the emission strings in Table 2. The “Occurrences” field refers to the number of times the source occurs in the sub-catchment, and therefore the number of times the pattern generation has to be repeated for that source (see §3.3.5).

**Table 3. Information on the naphthalene sources in the studied sub-catchment.**

Source ID	ES_ID	Occurrences	Yearly load [kg/yr]
01. Young households	658	380	0.2
01. Young households	653	380	0.9
02. Elderly households	653	200	2.3
02. Elderly households	658	200	0.5
03. Forests	664	1	150
04. Phtalic anhydride producing facility	665	1	200
05. Phtalic anhydride producing facility	659	1	550
06. Dye producing facility	659	1	350
07. Highways	654	1	180
08. Local roads	654	25	2
09. Landfill	660	1	50

The input generator contains a wizard that is able to join the information of Table 2 and Table 3 and compile a working table, as illustrated in Figure 7 and Figure 8. As the release patterns only contain a daily, weekly and yearly component, the multiyear component is automatically set to the default multiyear pattern n° 01 (constant release). The result is a combined table with all information needed to generate the PP release time series (Figure 9). This table can be saved as a tab delimited text file and can be opened again in a later stage, so that the wizard has only to be used once for each case.



**Figure 7. The wizard to join Emission String data from a general database and case dependent emission source data from a GIS. Step 1: loading database information on emission strings and emission patterns.**

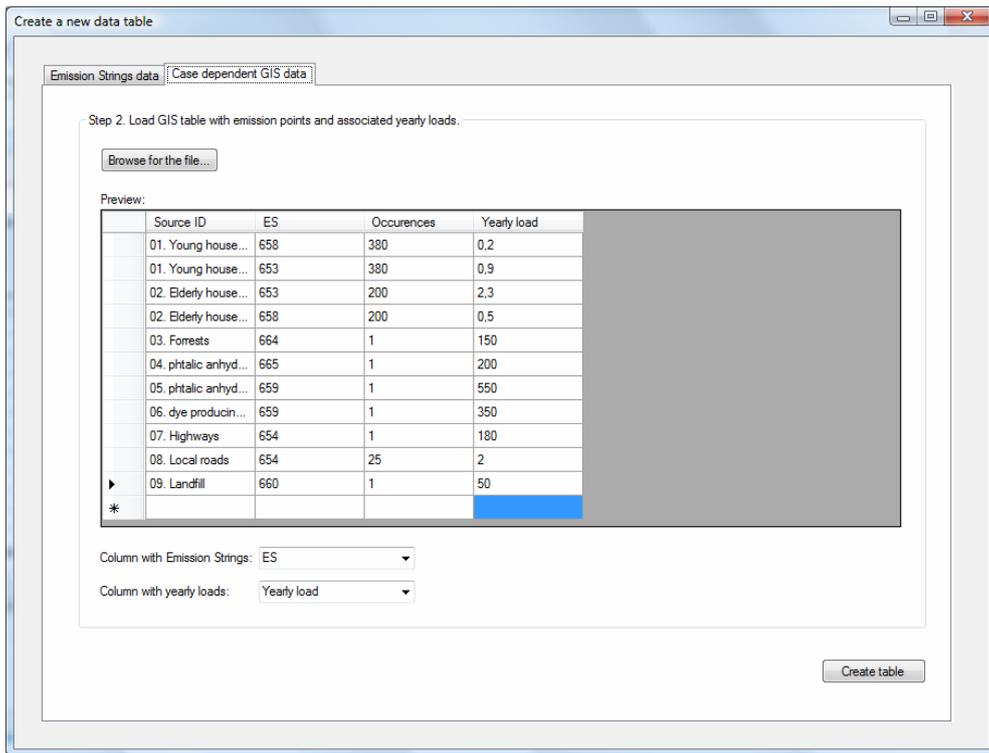


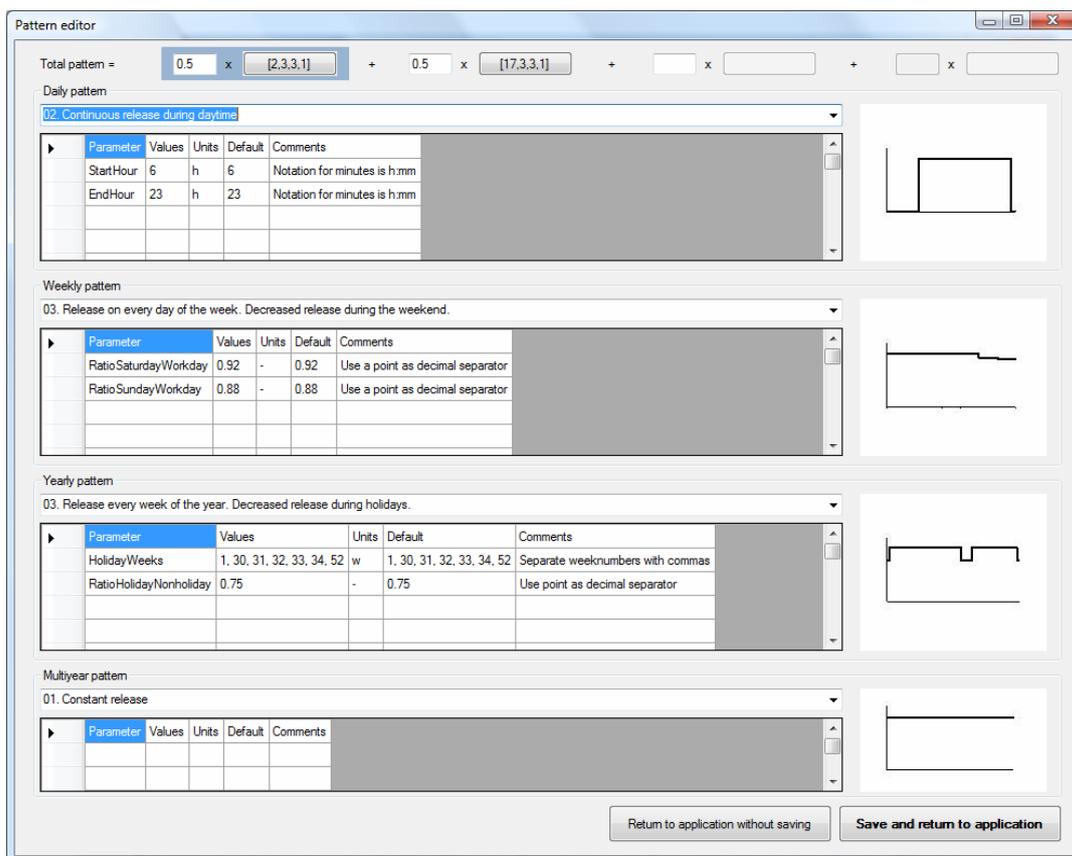
Figure 8. The wizard to join Emission String data from a general database and case dependent emission source data from a GIS. Step 2: loading case dependent information on emission sources, their Emission Strings and yearly loads.

Source ID	ES	Occurrences	Yearly load	Release Pattern	Parameter values	Compartment	Economic activity	Emission process
01. Young households	658	380	0.2	[10,10,1,1]	[16,23,1,1,00,block,0,15],[1,7,21,0,0]	WI	Household use	Discharges in sewage effluents or s
01. Young households	653	380	0.9	0,2[1,1,6,110,8[15,1,6,1]	[1,0],[49,91],[117,21,23,block,1,0],[49,91]	A	Heating	Emission by domestic wood burning
02. Elderly households	653	200	2.3	0,2[1,1,6,110,8[15,1,6,1]	[1,0],[49,91],[117,21,23,block,1,0],[49,91]	A	Heating	Emission by domestic wood burning
02. Elderly households	658	200	0.5	[10,10,1,1]	[16,23,1,1,00,block,0,15],[1,7,21,0,0]	WI	Household use	Discharges in sewage effluents or s
03. Forrests	664	1	150	[1,10,13,1]	[1,1,7,11],[23,35,41]	A	Agriculture and forestry	Emissions to atmosphere. From agr
04. phtalic anhydride producing facility	665	1	200	[1,6,2,1]	[1,16,71],[1,30,31,32,33,34,52],[1]	A	Industrial activity	Emissions to atmosphere. From ind
05. phtalic anhydride producing facility	659	1	550	[3,6,2,1]	[18,18],[16,71],[1,30,31,32,33,34,52],[1]	WI	Industrial activity	Large industrial point sources, direc
06. dye producing facility	659	1	350	[3,6,2,1]	[18,18],[16,71],[1,30,31,32,33,34,52],[1]	WI	Industrial activity	Large industrial point sources, direc
07. Highways	654	1	180	0,5[2,3,3,110,5[17,3,3,1]	[16,23],[0,92,0,88],[1,30,31,32,33,34,5...	A	Traffic	Emission from petrol, diesel, ethano
08. Local roads	654	25	2	0,5[2,3,3,110,5[17,3,3,1]	[16,23],[0,92,0,88],[1,30,31,32,33,34,5...	A	Traffic	Emission from petrol, diesel, ethano
09. Landfill	660	1	50	[1,1,1,1]	[1,0,0,0]	WI	Solid waste management	Land fills

Figure 9. The table with all combined information on all PP release sources considered in the case catchment.

It is important to note that the compiled data must have consequent units. For example, if the 25 occurrences of the source “08. Local roads” indicates that there are 25 km of roads in the catchment, the yearly load value of “2” must also be expressed in a unit per km. If it indicates that there are 25 roads, the yearly load must also be expressed as an average load per road, not per km of road. The same holds for the pattern.

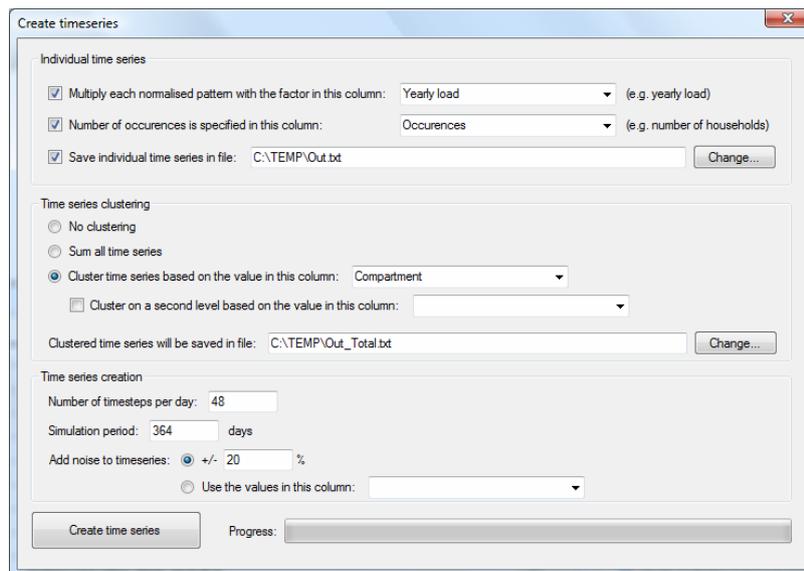
New rows can be added easily by filling in the bottom line. As can be seen from Figure 9, the release patterns and their parameter sets are recognised automatically and are shown as clickable links. By clicking on a pattern, this pattern and its parameter set can be changed easily in a pattern editor tool (Figure 10), resembling the one used in WP 3 to assign the patterns in Excel (see Figure 3).



**Figure 10.** The pattern editor to edit existing patterns and their parameter sets or to enter new patterns; note that here also the multiyear pattern can be adjusted, which was not the case in the pattern editor developed in WP 3.

The buttons at the top of the form allow to edit the different sub-patterns and their coefficients in a linear combination of patterns. For each pattern, a sketch of the pattern shape with the default parameter set is shown on the right. To preserve transparency and to keep the application extendable to other patterns, the images are loaded from a separate folder and have to be named in the format “AB.jpg”, in which A is the abbreviation of type of release pattern (daily (D), weekly (W), yearly (Y) or multiyear (M) release pattern) and B is the pattern number without leading zeroes, e.g. “W9.jpg” for weekly pattern number 09.

To create the needed time series for all sources listed in the combined table, a wizard (Figure 11) can be called from the menu File > Generate time series... (in Figure 9).



**Figure 11.** The wizard to generate a release time series for each line in the sources table.

First, for each PP releasing source a normalised individual time series is created according to the pattern and the parameter values in the table. If the first checkbox is checked, this normalised time series is multiplied with the load in the specified column. The second checkbox allows to repeat the time series generation for a certain number of times (see §3.3.5). In the example used here, the emission patterns were created for a single source (e.g. one household). Assuming that the catchment under study hosts 380 ‘young households’ and 200 ‘elderly households’ (see Figure 9), the PP release pattern generation will be repeated 380 times for each line which applies to young households (lines 1 and 2) and 200 times for each line which applies to elderly households (lines 3 and 4). The same holds for the 25 occurrences of ‘local roads’. E.g. if the yearly load is expressed in terms of kg PP released per km roads per year, this could mean that the catchment contains 25 km local roads and 1 km highway.

After the individual time series are created and saved into a tab delimited text file, they can be aggregated according to a value in a certain column, like the receiving compartment, the economic activity or the source ID. If all entrances apply to the same category (e.g. all sources are releasing the same PP to the same compartment), a total summed time series can also be generated and saved to a text file. As can be seen from Figure 11, aggregating can be performed on 2 levels. In this example, all entries in the working table apply to the same catchment, so the time series is only aggregated by the receiving compartment. The data in the resulting text files can easily be imported in e.g. Excel for manual (visual) control by copy/paste and in WEST for use in the integrated model.

Finally, three more options can be chosen: the temporal resolution of the time series (number of time steps per day, default set to 48), the simulation period (default set to one year of 364 days) and whether there should be noise added to the individual time series or not. The noise magnitude can either be a fixed value for all entries or a value specified for each source in the working table. It is specified as a percentage of the current value and is sampled randomly from a normal distribution (see §3.3.3 and §2.3).

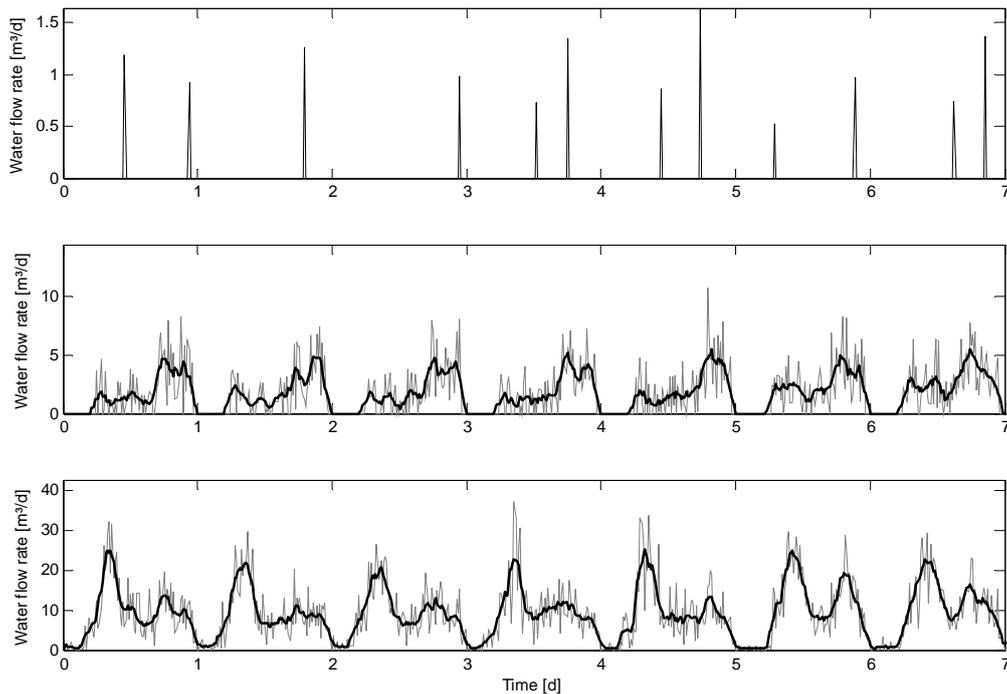
## 5 The creation of generic pollutants release time series

A time series is generated for the wastewater generating appliances shown in Table 4. Note that the pollutant loads are now expressed as vectors instead of single numbers, as was the case in the naphthalene example above. The first element in the vector represents the water flow rate in  $\text{m}^3/\text{year}$ , the next elements in the vector represent the concentrations of Biochemical Oxygen Demand after 5 days ( $\text{BOD}_5$ ), total Chemical Oxygen Demand ( $\text{COD}_t$ ), soluble COD ( $\text{COD}_s$ ),  $\text{NH}_4\text{-N}$  and  $\text{PO}_4\text{-P}$ , all expressed in  $\text{g}/\text{m}^3$ . The water flow rates and pollutant concentrations were compiled from experimental studies found in literature (Friedler, 2004; Kujawa-Roeleveld *et al.*, 2003; Gunther, 2000; Butler *et al.*, 1995). Noise estimates were adopted from Friedler & Butler (1996). In case no value was found in literature, noise was put to  $\pm 50\%$ . Also the noise magnitude to be applied on the time series can be specified for each individual component in the noise vector.

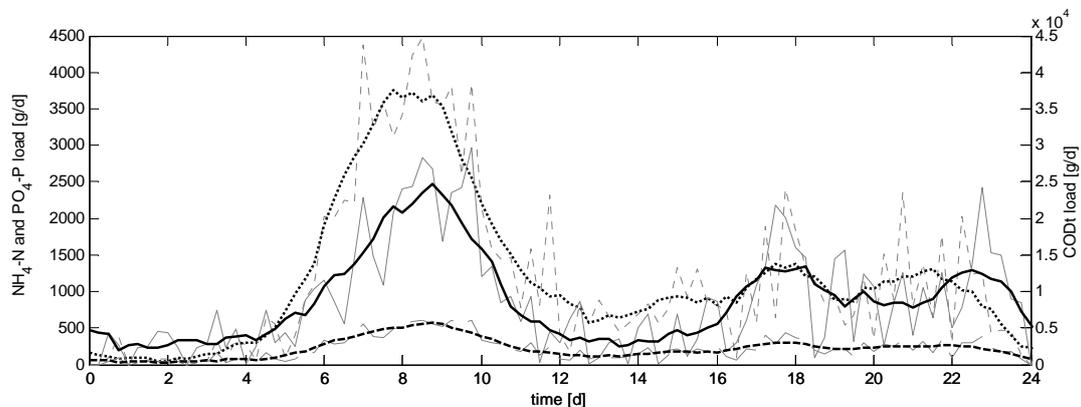
**Table 4. The wastewater generating appliances present in the virtual catchment; source categories: TU: toilet - urine; TF: toilet - faeces; BS: bath and shower; WB: wash basin; KS: kitchen sink; DW: dishwasher; WM: washing machine; the load vectors should be interpreted as  $[\text{H}_2\text{O}(\text{m}^3/\text{year}), \text{BOD}_t(\text{g}/\text{m}^3), \text{COD}_t(\text{g}/\text{m}^3), \text{COD}_s(\text{g}/\text{m}^3), \text{NH}_4\text{-N}(\text{g}/\text{m}^3), \text{PO}_4\text{-P}(\text{g}/\text{m}^3)]$ .**

Source	Category	PE	Loads	Noise	Pattern
TU	Households	100	[4.94,592,740,659,543,49]	[30,50,50,50,50,50]	2[24,8,1,1]5[24,6,1,1]7[11,1,1,1]5[10,10,1,1]
TF	Households	100	[4.68,3354,4192,671,117,39]	[20,50,50,50,50,50]	2[24,8,1,1]5[24,6,1,1]2[10,10,1,1]
BS	Households	100	[10.53,350,504,295,1.35,8.7]	[62,50,52,73,134,108]	65[24,6,8,1]156[24,10,2,1]26[24,8,8,1]39[24,10,2,1]
WB	Households	100	[2.912,215,406,284,0.41,15.8]	[118,50,63,60,77,113]	5[24,6,1,1]2[24,8,1,1]7[10,1,1,1]14[24,1,1,1]
KS	Households	100	[6.24,1068,1608,814.8,0.72,26.4]	[123,50,121,132,164,90]	7[11,1,1,1]2[10,9,1,1]3[10,10,1,1]
DW	Households	60	[1.144,712,1320,557,5.5,32]	[10,50,50,50,50,50]	[24,10,1,1]
WM	Households	100	[4.1,785,2276,1693,8.33,39]	[10,50,34,36,46,17]	52[24,10,1,1]30[24,10,13,1]
TU	Offices	20	[1.485,970,1213,1080,890,81]	[20,50,50,50,50,50]	47[10,6,2,1]47[11,6,2,1]3[10,6,8,1]2[10,6,6,1]
TF	Offices	20	[0.99,3354,4192,671,117,39]	[20,50,50,50,50,50]	94[10,10,2,1]3[10,10,8,1]2[10,10,6,1]

As an example, Figure 12 shows the generated time series for (1) the kitchen sink for one PE, (2) the kitchen sink for 100 PE and (3) all considered household appliances for 100 PE. This last time series is used as dry weather flow input for the dynamic catchment model, together with generic pollutant time series as shown for one day and 100 PE in Figure 13. A “tanks in series” approach is used to model the sewer system dynamics. The generated water and generic pollutants profiles are comparable to the experimental data found in literature (e.g. Friedler, 2004; Friedler & Butler, 1996; Butler *et al.*, 1995).



**Figure 12.** One week of the generated time series (with 15 minute temporal resolution) representing the kitchen sink waste water flow rate for one PE (top), the aggregated kitchen sink time series for 100 PE (middle) and the aggregated wastewater flow rate for all considered household appliances for 100 PE (bottom); the grey line is the instantaneous flow rate, the bold line is the 2 hour moving average flow rate.



**Figure 13.** One day of the aggregated time series for the generic pollutants CODt (dotted line, right axis), NH<sub>4</sub>-N (solid line, left axis) and PO<sub>4</sub>-P (dashed line, left axis) produced in a catchment with 100 PE household effluents; the grey line is the instantaneous flow rate (temporal resolution of 15 minutes), the bold line is the 2 hour moving average flow rate.

The larger the studied catchment, the more important the effect of residence time in the sewer system becomes (Gernaey *et al.*, 2005). As this aspect is not taken into account in the proposed input generation methodology, it must be considered in the sewer model. Large catchments should therefore be split up in sub-catchments. A second consequence of working with larger catchments (e.g. 1,000 PE instead of 100 PE), is that the effect of the Gaussian white noise on the individual time series will be hardly visible in the total time series, as the noises of the aggregated time series will compensate for each other.

## **6 Conclusions**

In the framework of WP 3 and WP 7, the need for an automatic tool to generate PP release time series was identified. For each PP release pattern type determined in Task 3.3, a mathematical function was defined and implemented in a stand-alone software tool realized in Visual Basic .NET. Additionally, functions for adding stochasticity to the deterministic patterns of Task 3.3 were included. The software tool was also extended to handle vectors of dependent components. Like this, the tool can be used to generate time series for generic pollutants (BOD, COD, N, P) next to priority pollutants. User interfaces and wizards were programmed in order to facilitate data management and the use of the functions.

## 7 References

- Butler, D., Friedler, E. & Gatt, K. (1995). Characterising the quantity and quality of domestic wastewater inflows. *Water Science and Technology*, 31(7), 13-24.
- De Keyser, W., Gevaert, V., Verdonck, F. & Benedetti, L. (2008). Modelling of Priority Pollutants releases from Urban Areas. Paper submitted to the 11th International Conference on Urban Drainage, Edinburgh, Scotland, UK, 10p.
- Friedler, E. (2004). Quality of individual domestic greywater streams and its implication for on-site treatment and reuse possibilities. *Environmental Technology*, 25, 997-1008.
- Friedler E. & Butler D. (1996). Quantifying the inherent uncertainty in the quantity and quality of domestic wastewater. *Water Science and Technology*, 33(2), 65-78.
- Gernaey, K.V., Rosen, C., Benedetti, L., and Jeppsson, U. (2005). Phenomenological modeling of wastewater treatment plant influent disturbance scenarios. Proceedings, 10th International Conference on Urban Drainage, Copenhagen, Denmark. 8p.
- Günther, F. (2000). Wastewater treatment by greywater separation: Outline for a biologically based greywater purification plant in Sweden. *Ecological Engineering*, 15, 139-146.
- Jeppsson, U., Rosen, C., Alex, J., Copp, J., Gernaey, K.V., Pons, M.-N., and Vanrolleghem, P.A. (2006). Towards a benchmark simulation model for plant-wide control strategy performance evaluation of WWTPs. *Water Science and Technology*, 53(1), 287-295.
- Kujawa-Roeleveld, K., Elmitwalli, T., Gaillard, A., van Leeuwen, M. & Zeeman, G. (2003). Co-digestion of concentrated black water and kitchen refuse in an accumulation system within the DESAR (decentralized sanitation and reuse) concept. *Water Science and Technology*, 48(4), 121-128.
- Ort, C. (2006). Short-term dynamics of micropollutants in sewer systems. PhD, Swiss Federal Institute of Technology Zurich, Gossau, Zurich. 260p.
- Vanhooren, H., Meirlaen, J., Amerlinck, Y., Claeys, F., Vangheluwe, H. and Vanrolleghem, P.A. (2003). WEST: Modelling biological wastewater treatment. *Journal of Hydroinformatics*, 5(1), 27-50.

## **Appendix: Implemented functions for the creation of daily, weekly and yearly patterns.**

In the tables on the next pages, the core of the pattern generating code is shown in pseudo code. Words in *italics* refer to parameter names. The ID numbers in the first column refer to the daily, weekly and yearly pattern ID assigned in Deliverable 3.3. When one function is representative for different patterns, multiple IDs are listed.

Of course also additional ancillary functions were implemented, e.g. to check whether the used parameter values are within feasible boundaries.

Each pattern starts with a vector called "f" filled with zeroes, so that only for the periods in which a PP release occurs the release magnitude has to be specified.

A more detailed qualitative discussion of all patterns can be found in Deliverable 3.3.

### A. Daily patterns

ID	Description	Function in pseudo code
01 02 03	Continuous release	If $t \geq StartHour$ AND $t < EndHour$ then $f(t) = 1$
04 05	Block pattern	$TimeOfOneCycle = ReleaseDuration + IntervalDuration$ If $t \geq StartHour$ AND $t < EndHour$ AND $t - StartHour - \left\lfloor \frac{t - StartHour}{TimeOfOneCycle} \right\rfloor \times TimeOfOneCycle < ReleaseDuration$ then $f(t) = 1$
06 07	Peak with decline to continuous level	If $t \geq StartOfPeak$ AND $t < EndOfPeak$ then Case $PeakShape = Triangular$ If $t < ClimaxOfPeak$ then $f(t) = \frac{t - StartOfPeak}{ClimaxOfPeak - StartOfPeak}$ else $f(t) = 1 - \left( \frac{t - ClimaxOfPeak}{EndOfPeak - ClimaxOfPeak} \times (1 - RatioContinuousClimax) \right)$ Case $Peakshape = Block$ $f(t) = 1$  If $t \geq EndOfPeak$ AND $t < EndOfContinuous$ then $f(t) = RatioContinuousClimax$
08	Two peaks with decline to continuous level	If $t \geq StartOfFirstPeak$ AND $t < EndOfFirstPeak$ then Case $PeakShape = Triangular$ If $t < ClimaxOfFirstPeak$ then $f(t) = RatioFirstSecondPeak \times \frac{t - StartOfFirstPeak}{ClimaxOfFirstPeak - StartOfFirstPeak}$ else $f(t) = RatioFirstSecondPeak \times \frac{EndOfFirstPeak - t}{EndOfFirstPeak - ClimaxOfFirstPeak}$  Case $Peakshape = Block$

		$f(t) = 1$ <p>If <math>t \geq \text{StartOfSecondPeak}</math> AND <math>t &lt; \text{EndOfSecondPeak}</math> then  Case <i>PeakShape</i> = Triangular</p> $\text{If } t < \text{ClimaxOfSecondPeak} \text{ then } f(t) = \frac{t - \text{StartOfSecondPeak}}{\text{ClimaxOfSecondPeak} - \text{StartOfSecondPeak}}$ $\text{else } f(t) = 1 - \left( \frac{t - \text{ClimaxOfSecondPeak}}{\text{EndOfSecondPeak} - \text{ClimaxOfSecondPeak}} \times (1 - \text{RatioContinuousSecondPeak}) \right)$ <p>Case <i>Peakshape</i> = Block  <math>f(t) = 1</math></p> <p>If <math>t \geq \text{EndOfPeak}</math> AND <math>t &lt; \text{EndOfContinuous}</math> then <math>f(t) = \text{RatioContinuousClimax}</math></p>
<p>09 10 11</p>	<p>Random peak(s)</p>	$\text{PeakDuration}_{\text{timesteps}} = \text{Round} \left( \frac{\text{PeakDuration}_{\text{time}}}{\text{TimestepDuration}} \right)$ $\text{LowerBound}_{\text{timesteps}} = \left\lfloor \frac{\text{LowerBound}_{\text{time}}}{\text{TimestepDuration}} \right\rfloor$ $\text{UpperBound}_{\text{timesteps}} = \left\lfloor \frac{\text{UpperBound}_{\text{time}}}{\text{TimestepDuration}} \right\rfloor$ <p>for <math>i = 1</math> to <i>NumberOfPeaks</i></p> $\text{PeakHeight}(i) = \frac{\text{RatioPeaks} \left( i - \left\lfloor \frac{i}{\text{CountElements}(\text{RatioPeaks})} \right\rfloor \times \text{CountElements}(\text{RatioPeaks}) \right)}{\text{RatioPeaks}(0)}$ <p>If <math>i = 0</math> then <math>\text{PeakOccurenceTimes}(i) = \text{GetRandomNumber}(\text{LowerBound}_{\text{timesteps}}, \text{UpperBound}_{\text{timesteps}})</math>  else  Do  <math>\text{PeakOccurenceTimes}(i) = \text{GetRandomNumber}(\text{LowerBound}_{\text{timesteps}}, \text{UpperBound}_{\text{timesteps}})</math>  For <math>j = 0</math> to <math>i - 1</math>  If <math> \text{PeakOccurenceTimes}(i) - \text{PeakOccurenceTimes}(j)  &lt; \text{PeakDuration}_{\text{timesteps}}</math> Then</p>

		<pre> distance = False Exit For Else distance = True End If Next j Loop Until distance = True End If next i SortAscending(PeakOccurenceTimes) for i = 1 to NumberOfPeaks for j = <math>\left( \text{PeakOccurenceTimes}(i) - \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right)</math> to <math>\left( \text{PeakOccurenceTimes}(i) + \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right)</math> Case PeakShape = Triangular If <math>j \leq \text{PeakOccurenceTimes}(i)</math> Then <math display="block">f(j) = f(j) + \frac{j - \left( \text{PeakOccurenceTimes}(i) - \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right)}{\left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor} \times \text{PeakHeight}(i)</math> Else <math display="block">f(j) = f(j) + \frac{\left( \text{PeakOccurenceTimes}(i) + \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right) - j}{\left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor} \times \text{PeakHeight}(i)</math> End If Case PeakShape = Block f(j) = f(j) + PeakHeight(i) next j next i </pre>
--	--	--

12 13 14 15 16	One peak	<p>If <math>t \geq StartOfPeak</math> AND <math>t &lt; EndOfPeak</math> then  Case <i>PeakShape</i> = Triangular</p> $\text{If } t < ClimaxOfPeak \text{ then } f(t) = \frac{t - StartOfPeak}{ClimaxOfPeak - StartOfPeak} \text{ else } f(t) = \frac{EndOfPeak - t}{EndOfPeak - ClimaxOfPeak}$ <p>Case <i>Peakshape</i> = Block  <math>f(t) = 1</math></p>
17 18 19 20	Two peaks	<p>If <math>t \geq StartOfFirstPeak</math> AND <math>t &lt; EndOfFirstPeak</math> then  Case <i>PeakShape</i> = Triangular</p> $\text{If } t < ClimaxOfFirstPeak \text{ then } f(t) = RatioFirstSecondPeak \times \frac{t - StartOfFirstPeak}{ClimaxOfFirstPeak - StartOfFirstPeak}$ $\text{else } f(t) = RatioFirstSecondPeak \times \frac{EndOfFirstPeak - t}{EndOfFirstPeak - ClimaxOfFirstPeak}$ <p>Case <i>Peakshape</i> = Block  <math>f(t) = 1</math></p> <p>If <math>t \geq StartOfSecondPeak</math> AND <math>t &lt; EndOfSecondPeak</math> then  Case <i>PeakShape</i> = Triangular</p> $\text{If } t < ClimaxOfSecondPeak \text{ then } f(t) = \frac{t - StartOfSecondPeak}{ClimaxOfSecondPeak - StartOfSecondPeak}$ $\text{else } f(t) = \frac{EndOfSecondPeak - t}{EndOfSecondPeak - ClimaxOfSecondPeak}$ <p>Case <i>Peakshape</i> = Block  <math>f(t) = 1</math></p>
21 22	Three peaks	<p>If <math>t \geq StartOfFirstPeak</math> AND <math>t &lt; EndOfFirstPeak</math> then  Case <i>PeakShape</i> = Triangular</p> $\text{If } t < ClimaxOfFirstPeak \text{ then } f(t) = RatioFirstSecondPeak \times \frac{t - StartOfFirstPeak}{ClimaxOfFirstPeak - StartOfFirstPeak}$ $\text{else } f(t) = RatioFirstSecondPeak \times \frac{EndOfFirstPeak - t}{EndOfFirstPeak - ClimaxOfFirstPeak}$

		<p>Case <i>Peakshape</i> = Block <math>f(t) = 1</math></p> <p>If <math>t \geq \text{StartOfSecondPeak}</math> AND <math>t &lt; \text{EndOfSecondPeak}</math> then Case <i>PeakShape</i> = Triangular</p> <p>If <math>t &lt; \text{ClimaxOfSecondPeak}</math> then <math>f(t) = \text{RatioSecondThirdPeak} \times \frac{t - \text{StartOfSecondPeak}}{\text{ClimaxOfSecondPeak} - \text{StartOfSecondPeak}}</math></p> <p>else <math>f(t) = \text{RatioSecondThirdPeak} \times \frac{\text{EndOfSecondPeak} - t}{\text{EndOfSecondPeak} - \text{ClimaxOfSecondPeak}}</math></p> <p>Case <i>Peakshape</i> = Block <math>f(t) = 1</math></p> <p>If <math>t \geq \text{StartOfThirdPeak}</math> AND <math>t &lt; \text{EndOfThirdPeak}</math> then Case <i>PeakShape</i> = Triangular</p> <p>If <math>t &lt; \text{ClimaxOfThirdPeak}</math> then <math>f(t) = \frac{t - \text{StartOfThirdPeak}}{\text{ClimaxOfThirdPeak} - \text{StartOfThirdPeak}}</math></p> <p>else <math>f(t) = \frac{\text{EndOfThirdPeak} - t}{\text{EndOfThirdPeak} - \text{ClimaxOfThirdPeak}}</math></p> <p>Case <i>Peakshape</i> = Block <math>f(t) = 1</math></p>
23	Day/night pattern	If $t < \text{StartOfDay}$ OR $t \geq \text{StartOfNight}$ then $f(t) = 1$ else $f(t) = \text{RatioDayNight}$
24	Random peaks sampled from user specified distribution	$\text{PeakDuration}_{\text{timesteps}} = \text{Round} \left( \frac{\text{PeakDuration}_{\text{time}}}{\text{TimestepDuration}} \right)$ <p>For each <math>i</math> in <i>DistributionShape</i>: <math>\text{DistributionTimes}(i) = i / \text{CountElements}(\text{DistributionShape})</math>  <math>\text{DistributionShape}(\text{CountElements}(\text{DistributionShape})+1) = \text{DistributionShape}(0)</math>  <math>\text{DistributionTimes}(\text{CountElements}(\text{DistributionTimes})+1) = 1</math></p> <p>For each <math>i</math> in <i>TemporalResolution</i>  <math>j = -1</math>  Do</p>

	<pre> j = j + 1 Loop Until ((i / TemporalResolution) ≥ DistributionTimes(j)) And ((i / TemporalResolution) &lt; DistributionTimes(j + 1))  Distribution(i) = Round <math>\left[ 100 \times \left( \frac{\text{DistributionShape}(j) + \frac{\text{DistributionShape}(j+1) - \text{DistributionShape}(j)}{\text{DistributionTimes}(j+1) - \text{DistributionTimes}(j)} \times \left( \frac{i}{\text{TemporalResolution}} - \text{DistributionTimes}(j) \right)}{\right)} \right]</math>  Next For each i in Distribution   If Distribution(i) ≠ 0 then     For j = 1 to Distribution(i)       SamplingPool(CountElements(Samplingpool)+1) = Distribution(i)     Next j   End If Next i For i = 1 to <i>NumberOfPeaks</i>   PeakHeight(i) = <math>\frac{\text{RatioPeaks} \left( i - \left\lfloor \frac{i}{\text{CountElements}(\text{RatioPeaks})} \right\rfloor \times \text{CountElements}(\text{RatioPeaks}) \right)}{\text{RatioPeaks}(0)}</math>   If i = 0 then PeakOccurrenceTimes(i) = SamplingPool(GetRandomNumber(0, CountElements(SamplingPool)-1))   else     Do       PeakOccurrenceTimes(i) = SamplingPool(GetRandomNumber(0, CountElements(SamplingPool)-1))       For j = 0 to i - 1         If  PeakOccurrenceTimes(i) - PeakOccurrenceTimes(j)  &lt; PeakDuration<sub>timesteps</sub> Then           distance = False           Exit For         Else           distance = True         End If       Next j     Loop Until distance = True   End If </pre>
--	---

	<pre> next i SortAscending(PeakOccurenceTimes) for i = 1 to NumberOfPeaks     for j = <math>\left( \text{PeakOccurenceTimes}(i) - \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right)</math> to <math>\left( \text{PeakOccurenceTimes}(i) + \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right)</math>         Case PeakShape = Triangular             If j ≤ PeakOccurenceTimes (i) Then                 <math display="block">f(j) = f(j) + \frac{j - \left( \text{PeakOccurenceTimes}(i) - \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right)}{\left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor} \times \text{PeakHeight}(i)</math>             Else                 <math display="block">f(j) = f(j) + \frac{\left( \text{PeakOccurenceTimes}(i) + \left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor \right) - j}{\left\lfloor \frac{\text{PeakDuration}_{\text{timesteps}}}{2} \right\rfloor} \times \text{PeakHeight}(i)</math>             End If         Case PeakShape = Block             f(j) = f(j) + PeakHeight(i)     next j next i </pre>
--	---

## B. Weekly patterns

ID	Description	Function in pseudo code
01	Continuous release	$f(t) = 1$
02 04 06	Workweek/weekend pattern, with no release on a certain day	If $t \in \text{NoReleaseOnDay}$ then $f(t) = 0$ else If $t \leq 5$ then $f(t) = 1$ If $t = 6$ then $f(t) = \text{RatioSaturdayWorkday}$ If $t = 7$ then $f(t) = \text{RatioSundayWorkday}$
03 05	Workweek/weekend pattern	If $t \leq 5$ then $f(t) = 1$ If $t = 6$ then $f(t) = \text{RatioSaturdayWorkday}$ If $t = 7$ then $f(t) = \text{RatioSundayWorkday}$
07	Release on a certain day	If $t \in \text{ReleaseOnDay}$ then $f(t) = 1$
08 09	Release on certain days, with different magnitude during weekends	If $t \in \text{ReleaseOnDay}$ then $f(t) = 1$ $f(\text{Saturday}) = f(\text{Sunday}) \times \text{RatioSaturdaySunday}$
10	Release on random day(s) (sampled from uniform distribution)	For $i = 1$ To $\text{NumberOfDays}$ Do $\text{SampledDay} = \text{GetRandomNumber}(\text{LowerBound}, \text{UpperBound})$ Loop Until $f(\text{SampledDay}) \neq 1$ $f(\text{SampledDay}) = 1$ Next $i$
11	Release on random day(s) (sampled from user specified distribution)	For each $i$ in $\text{DistributionShape}$ : $\text{DistributionTimes}(i) = i / \text{CountElements}(\text{DistributionShape})$ $\text{DistributionShape}(\text{CountElements}(\text{DistributionShape})+1) = \text{DistributionShape}(0)$ $\text{DistributionTimes}(\text{CountElements}(\text{DistributionTimes})+1) = 1$ For $i = 0$ to 6 $j = -1$ Do $j = j + 1$

	<pre>                 Loop Until ((i / 7) ≥ DistributionTimes(j)) And ((i / 7) &lt; DistributionTimes(j + 1)) Distribution (i) = Round <math>\left( 100 \times \left[ \text{DistributionShape}(j) + \frac{\text{DistributionShape}(j + 1) - \text{DistributionShape}(j)}{\text{DistributionTimes}(j + 1) - \text{DistributionTimes}(j)} \times \left( \frac{i}{7} - \text{DistributionTimes}(j) \right) \right] \right)</math> Next For each i in Distribution     If Distribution(i) ≠ 0 then         For j = 1 to Distribution(i)             SamplingPool(CountElements(Samplingpool)+1) = Distribution(i)         Next j     End If Next i For i = 1 To <i>NumberOfDays</i>     Do         SampledDay = SamplingPool(GetRandomNumber(0, CountElements(SamplingPool)-1))         Loop Until f(SampledDay) ≠ 1         f(SampledDay) = 1     Next i             </pre>
--	--

### C. Yearly patterns

<b>ID</b>	<b>Description</b>	<b>Function in pseudo code</b>
01	Continuous release	$f(t) = 1$
02	Continuous release pattern with no release during certain weeks	If $t \notin \text{HolidayWeeks}$ then $f(t) = 1$
03 05	Continuous release pattern with decreased release during certain weeks	If $t \in \text{HolidayWeeks}$ then $f(t) = \text{RatioHolidayNonHoliday}$ else $f(t) = 1$
04	Continuous release pattern with decreased release during 2 independent sets of certain weeks	If $t \in \text{HolidayWeeks}$ then $f(t) = \text{RatioHolidayNonHoliday}$ else If $(\text{StartOfDarkWeeks} > \text{EndOfDarkWeeks} \text{ AND } (t \geq \text{StartOfDarkWeeks} \text{ OR } t < \text{EndOfDarkWeeks})) \text{ OR}$ $(\text{StartOfDarkWeeks} < \text{EndOfDarkWeeks} \text{ AND } t \geq \text{StartOfDarkWeeks} \text{ AND } t < \text{EndOfDarkWeeks})$ then $f(t) = \text{RatioDarkdaysLightdays}$ else $f(t) = 1$
06 07 08 09 10 11 12	Continuous release from a certain startweek to a certain endweek	If $\text{StartWeek} > \text{EndWeek}$ AND $(t \geq \text{StartWeek} \text{ OR } t < \text{EndWeek})$ then $f(t) = 1$ If $\text{StartWeek} < \text{EndWeek}$ AND $t \geq \text{StartWeek}$ AND $t < \text{EndWeek}$ then $f(t) = 1$
13	Release in random week(s), sampled from uniform distribution	For $i = 1$ To $\text{NumberOfWeeks}$ Do SampledWeek = $\text{GetRandomNumber}(\text{LowerBound}, \text{UpperBound})$ Loop Until $f(\text{SampledWeek}) \notin 1$ $f(\text{SampledWeek}) = 1$

		Next i
14	Release in random week(s), sampled from user specified distribution	<pre> For each i in <i>DistributionShape</i>: <i>DistributionTimes</i>(i) = i / <i>CountElements</i>(<i>DistributionShape</i>) <i>DistributionShape</i>(<i>CountElements</i>(<i>DistributionShape</i>)+1) = <i>DistributionShape</i>(0) <i>DistributionTimes</i>(<i>CountElements</i>(<i>DistributionTimes</i>)+1) = 1 For i = 0 to 51     j = -1     Do         j = j + 1     Loop Until ((i / 7) ≥ <i>DistributionTimes</i>(j)) And ((i / 7) &lt; <i>DistributionTimes</i>(j + 1)) <i>Distribution</i>(i) = Round <math>\left( 100 \times \left[ \text{DistributionShape}(j) + \frac{\text{DistributionShape}(j + 1) - \text{DistributionShape}(j)}{\text{DistributionTimes}(j + 1) - \text{DistributionTimes}(j)} \times \left( \frac{i}{52} - \text{DistributionTimes}(j) \right) \right] \right)</math> Next For each i in <i>Distribution</i>     If <i>Distribution</i>(i) ≠ 0 then         For j = 1 to <i>Distribution</i>(i)             <i>SamplingPool</i>(<i>CountElements</i>(<i>SamplingPool</i>)+1) = <i>Distribution</i>(i)         Next j     End If Next i For i = 1 to <i>NumberOfPeaks</i>     If i = 0 then <i>PeakOccurrenceTimes</i>(i) = <i>SamplingPool</i>(<i>GetRandomNumber</i>(0, <i>CountElements</i>(<i>SamplingPool</i>)-1))     else         Do             <i>PeakOccurrenceTimes</i>(i) = <i>SamplingPool</i>(<i>GetRandomNumber</i>(0, <i>CountElements</i>(<i>SamplingPool</i>)-1))             For j = 0 to i - 1                 If <math> \text{PeakOccurrenceTimes}(i) - \text{PeakOccurrenceTimes}(j)  &lt; \text{PeakDuration}</math> Then                     distance = False                     Exit For                 Else                     distance = True                 End If             Next j         Loop Until distance = True </pre>

	<pre>                 End If             next i             for i = 1 to <i>NumberOfPeaks</i>                 for j = <math>\left( \text{PeakOccurrenceTimes}(i) - \left\lfloor \frac{\text{PeakDuration}}{2} \right\rfloor \right)</math> to <math>\left( \text{PeakOccurrenceTimes}(i) + \left\lfloor \frac{\text{PeakDuration}}{2} \right\rfloor \right)</math>                     f(j) = 1                 next j             next i         </pre>
--	--