Max-Planck-Institut für Plasmaphysik
EURATOM Association

VLEEM – Very Long Term Energy Environment Modelling

EC/DG Research Contract ENG2-CT-2000-00441

Final report

August 2002

ANNEX 9

Short description of the simulation tool TASES (Time And Space resolve Energy Simulation)

T. Hamacher
M. Biberacher

Basis of the presented simulation tool is a flexible data structure that contains all data necessary for the scenario. To understand the whole procedure, it is essential to explain how this data structure works.

1. Data structure

The complete simulation program is separated in different modules, which are more or less independent. Out of these modules, the class “graph" stands for data-management.

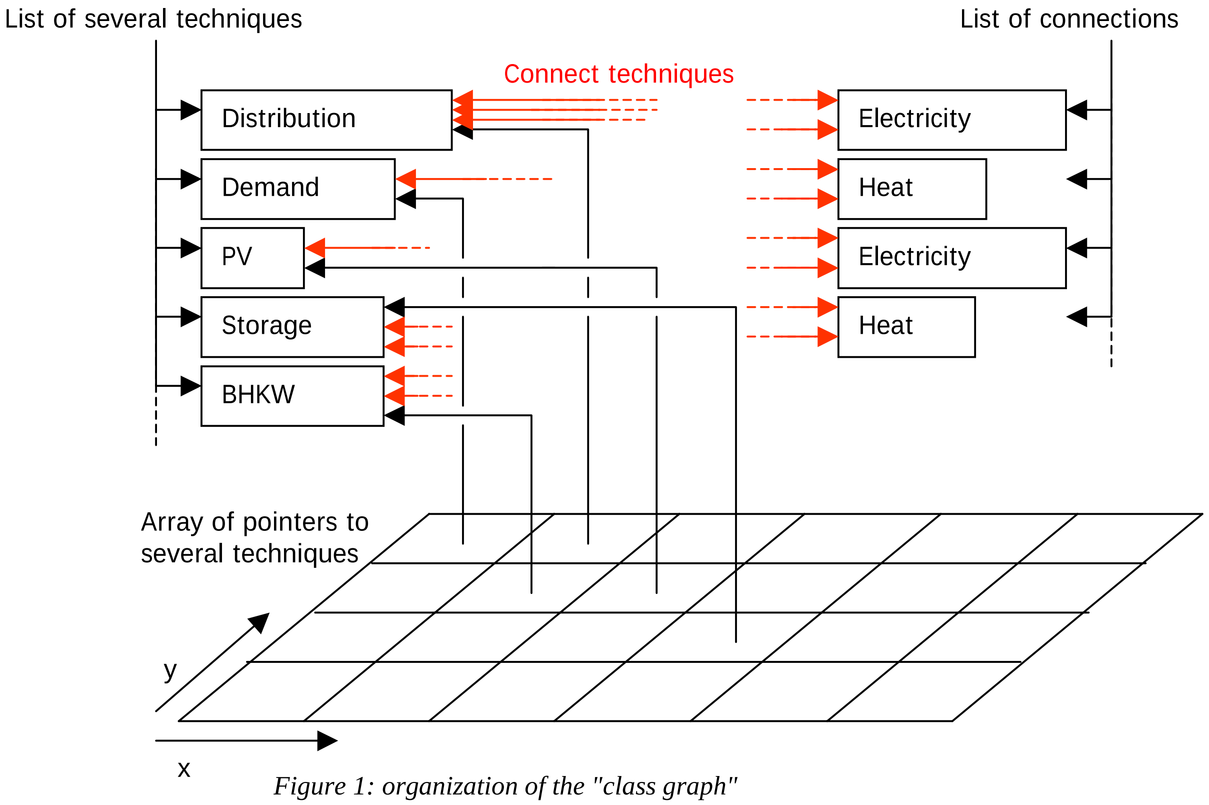


Figure 1: organization of the "class graph"

A fundamental array of pointers offers fast access regarding the spatial resolution in the given scenario. Each item in this array contains a pointer to a special member within the dataset. These connections realise descriptions of special supply techniques, demand, storage etc... and anything else involved in the scenario.
All different datasets are collected in a class "node". This class contains all data relevant to describe special proceedings such as "time dependence" (provided through an array).
Another similar class is called "link". This class contains variables to describe connections between two members of the class "node", for instance wires, pipes etc…
Members of "node" as well as members of "link" are collected in a list, in order to allow fast sequential access.

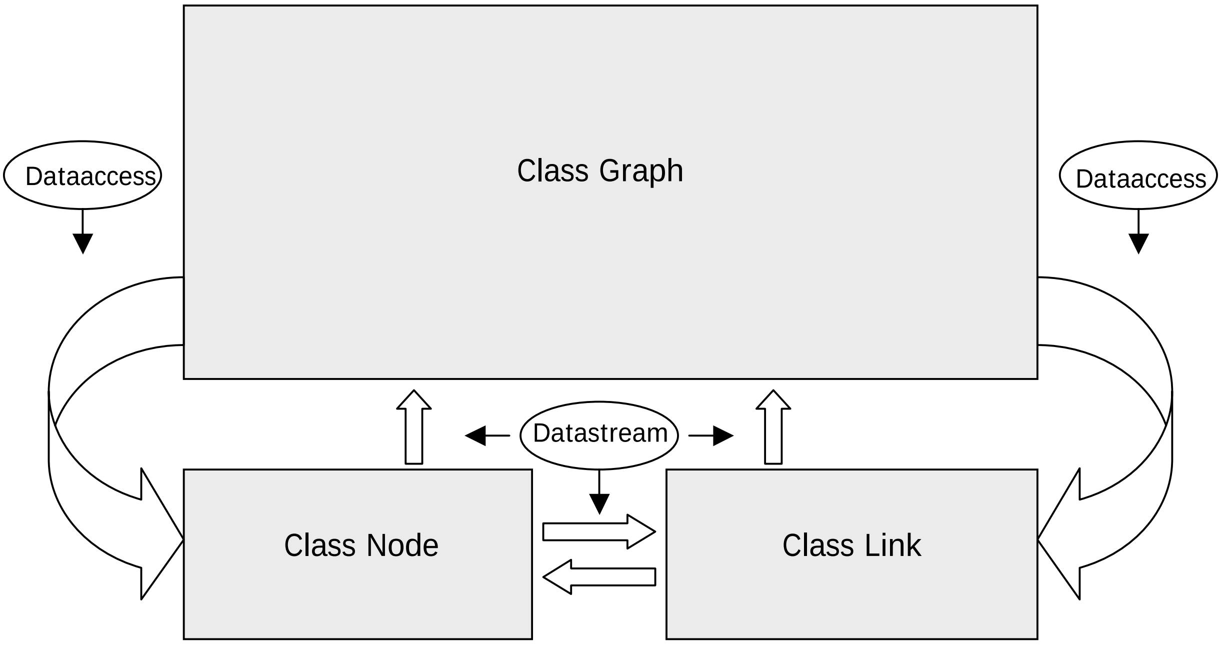
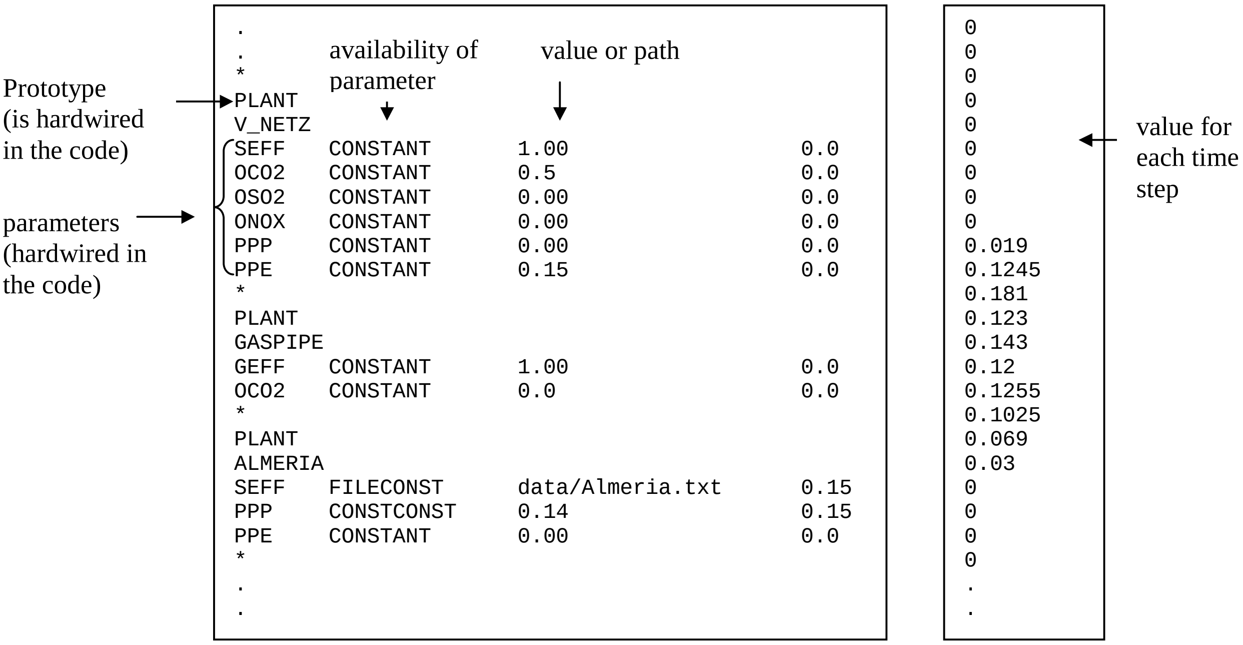


Figure 2: interaction of the classes

The three classes providing the complete data structure are implemented by means of independent modules. These classes are connected by defined interfaces.

There are several external ASCII-files providing parameters to feed the data structure with data regarding special supply, demand, storage or link techniques. All these external ASCII files are represented in a file called "typesdata" and guarantee time dependent descriptions of the mentioned functions. These files work according to the path declaration in the "typesdata"-file.

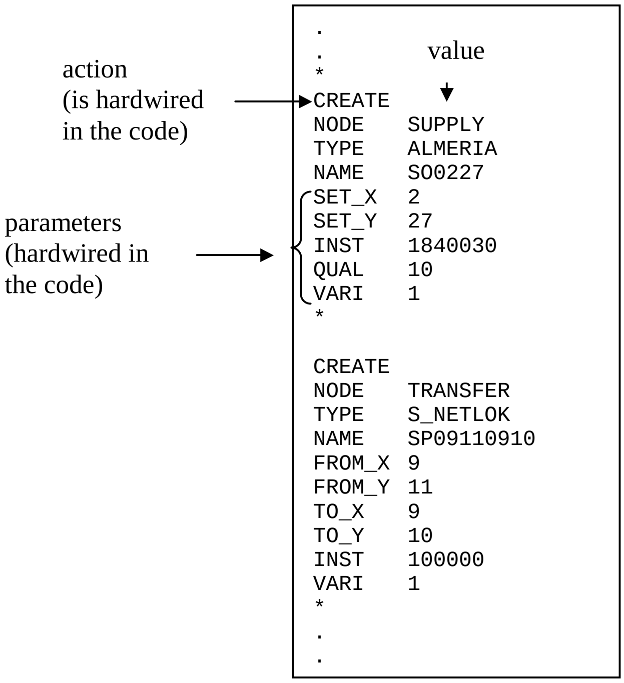
Additionally, the spatial resolution of the scenario to be simulated and ist assembling according to before mentioned functions are represented in a "scenario" - file.


availability of parameter
value or path
Prototype (is hardwired in the code)
parameters (hardwired in the code)
value for each time step

```
.
.
*
PLANT
V_NETZ
SEFF    CONSTANT      1.00                0.0
OCO2    CONSTANT      0.5                 0.0
OSO2    CONSTANT      0.00                0.0
ONOX    CONSTANT      0.00                0.0
PPP     CONSTANT      0.00                0.0
PPE     CONSTANT      0.15                0.0
*
PLANT
GASPIPE
GEFF    CONSTANT      1.00                0.0
OCO2    CONSTANT      0.0                 0.0
*
PLANT
ALMERIA
SEFF    FILECONST     data/Almeria.txt    0.15
PPP     CONSTCONST    0.14                0.15
PPE     CONSTANT      0.00                0.0
*
.
.
```

```
0
0
0
0
0
0
0
0
0
0.019
0.1245
0.181
0.123
0.143
0.12
0.1255
0.1025
0.069
0.03
0
0
0
0
 .
 .
```

Figure 3: part of „Typesdata“-file (left) and part of external file „Almeria.txt“ (right)


value
action
(is hardwired
in the code)
parameters
(hardwired in
the code)

```
.
.
*
CREATE
NODE   SUPPLY
TYPE   ALMERIA
NAME   S00227
SET_X  2
SET_Y  27
INST   1840030
QUAL   10
VARI   1
*

CREATE
NODE   TRANSFER
TYPE   S_NETLOK
NAME   SP09110910
FROM_X 9
FROM_Y 11
TO_X   9
TO_Y   10
INST   100000
VARI   1
*
.
.
```

Figure 4: part of „scenario“-file

Figure three and figure four show the syntax of the external files. These figures include all data necessary for the scenario and for the simulation.

It has to be noted, that the class “graph” does not only manage containing data, but does also offer search and select routines, e.g. search functions. This function provides basic features for higher classes.

2. Simulation

As soon as the data structure is feeded with all necessary information, the simulations module starts to create balanced energy estimations for each time step. The simulation module therefore uses a kind of hardwired heuristics.

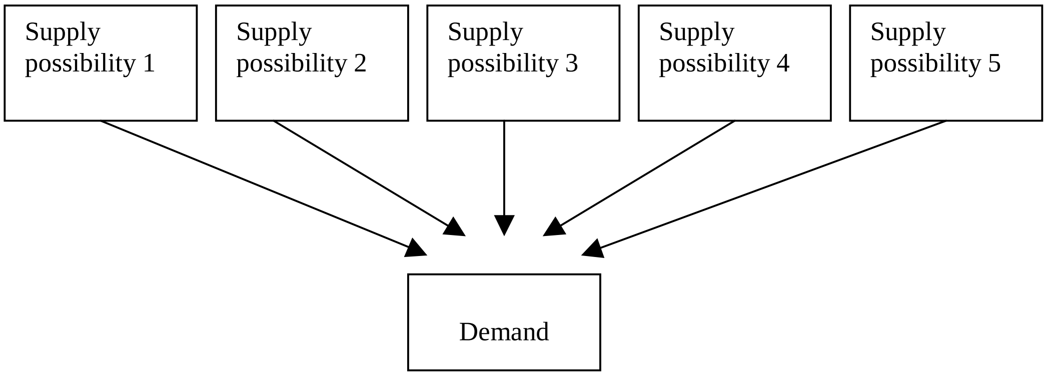


Figure 5: a given demand is satisfied following a quality sorted list of possibilities

As a result of the simulation, each supply technique participates in a balanced energy system. In order to satisfy a certain demand, a pretended quality factor is set in the scenario-file, as shown in figure five. The whole possible energy flows are collected in an extra list, classified by the quality factor and the efficiency of the connecting links.
Now the simulation starts. That means, for each time step all available demands have to be covered using the classified energy flows in the previously generated list.

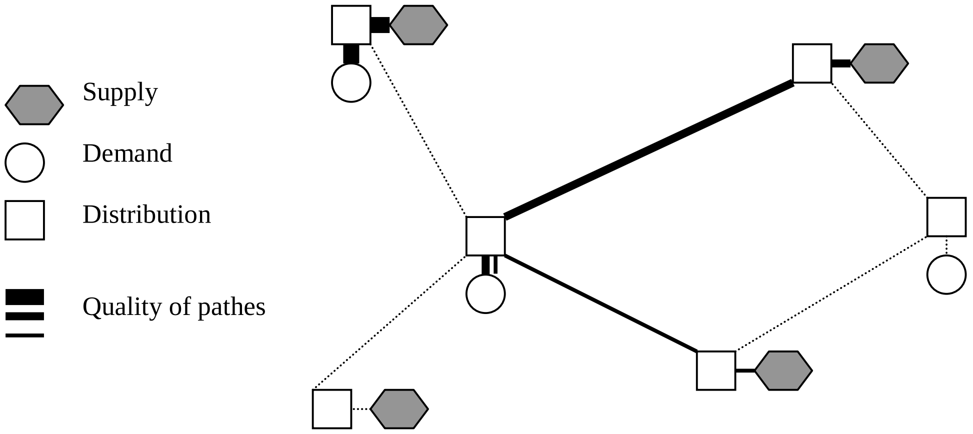
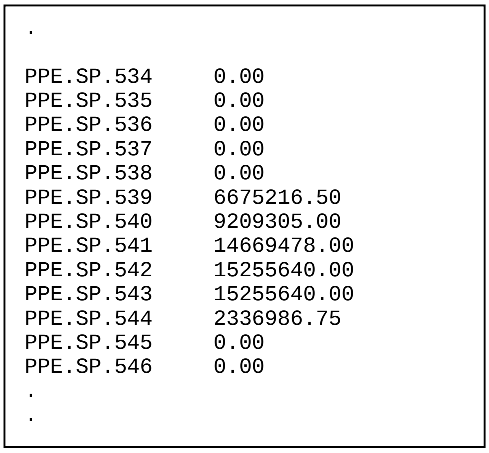


Figure 6: possible energy flows sorted by its ability

In a second step the storages which operated in the first step as supply techniques are operating as demands. However, one important difference is, that only energy flows fulfilling a certain hardwired quality level are accepted now.
The mentioned heuristic works as follows: Each storage has two quality levels - one for loading and one for unloading. This arrangement makes it possible to use different storages for loading and unloading.
The data strucuture manages the handling of three different types of energy flow and their conversions (gas, current and heat). Additionally, the heuristic manages CHP as well. If a supply on one end of the energy flow offers CHP, participation in demand is always driven by heat demand. That means, electric current can only be offered if a corresponding amount of heat is inquired.

As a result of the simulation, the load values are available for every link and node. These values are shown in figure seven.



```
.

PPE.SP.534     0.00
PPE.SP.535     0.00
PPE.SP.536     0.00
PPE.SP.537     0.00
PPE.SP.538     0.00
PPE.SP.539     6675216.50
PPE.SP.540     9209305.00
PPE.SP.541     14669478.00
PPE.SP.542     15255640.00
PPE.SP.543     15255640.00
PPE.SP.544     2336986.75
PPE.SP.545     0.00
PPE.SP.546     0.00
.
.
```

Figure 7: part of the „output“-file with the result of the simulation

3. Structure of the program

Figure eight shows the program's organisation. This is a very simplified flowchart of the input and output streams of the program.

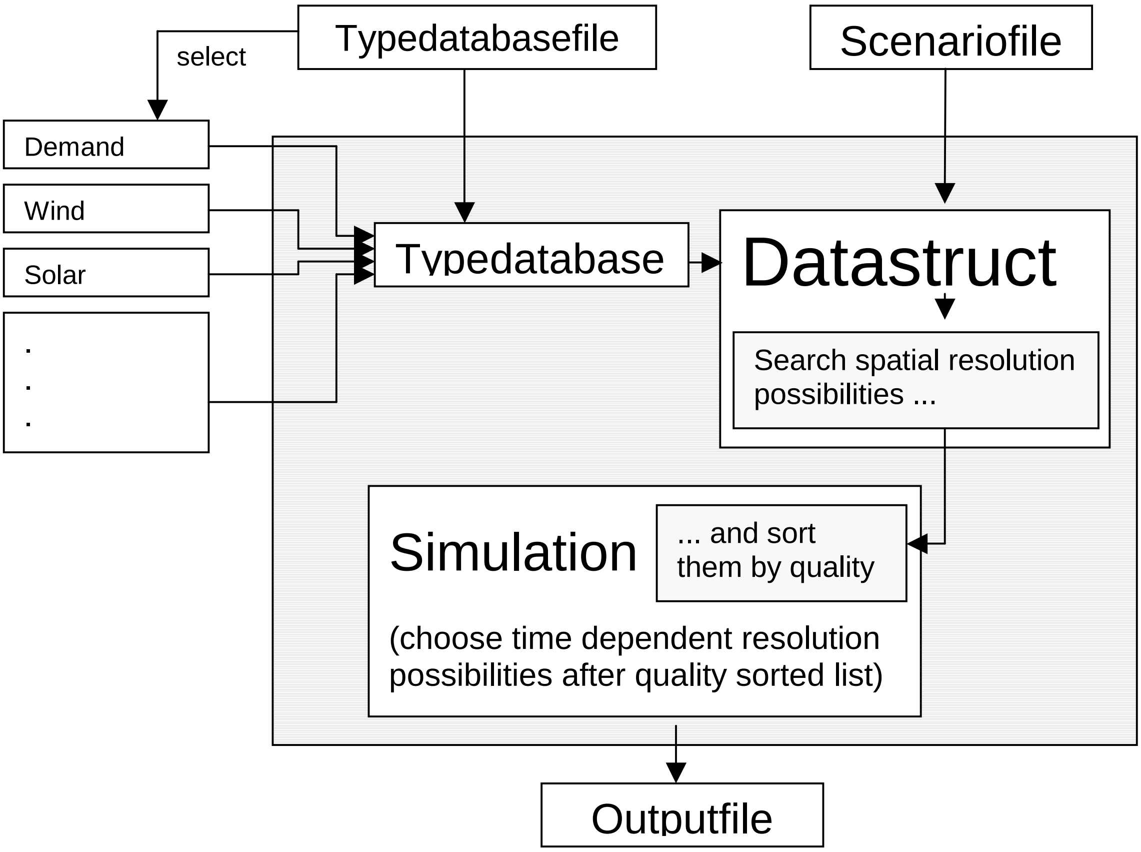
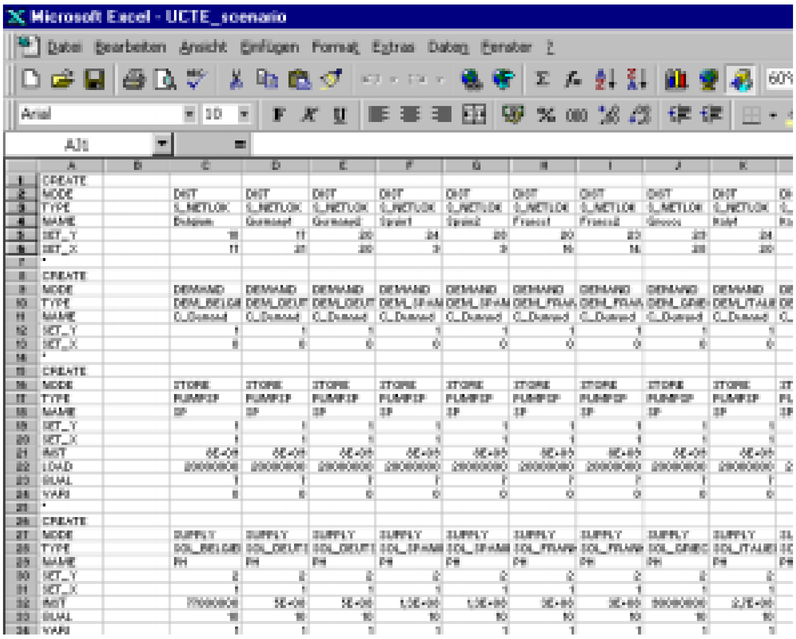

Typedatabasefile
Scenariofile
select
Demand
Wind
Solar
.
.
.
Typedatabase
Datastruct
Search spatial resolution possibilities ...
Simulation
... and sort them by quality
(choose time dependent resolution possibilities after quality sorted list)
Outputfile

Figure 8: flowchart of complete program and its in- and outputs

The main goal of the implementation of separated classes lies in aspired usage of the basic classes in different projects as well as in enlarged projects. Further steps might include optimisation of given scenarios with evolutionary algorithms. The class "datastructure" and the class "typedatabase" are both classes with only one duty - to handle all necessary data and to offer flexible interfaces for higher classes. The class "simulation" completes the main work. Consequently, most of the time for one run is necessary for this class scaling with
n (timesteps) • *k* (possible energy flows).

4. Handling

Due to the huge amount of input and output datasets, a well organised management of the mentioned data-sets is required. Therefore, all datasets are prepared in EXCEL. The scenario will be formulated tabularly in an EXCEL spreadsheet (figure nine). Additionally, a macro creates the already mentioned scenario-file.



Microsoft Excel - UCTE_scenario
Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster ?
Arial
10
60%
AJ1
=

	A	B	C	D	E	F	G	H	I	J	K	L
1	CREATE											
2	NODE		DIST	DIST	DIST	DIST	DIST	DIST	DIST	DIST	DIST	DI
3	TYPE		S_NETLOK	S_NETLOK	S_NETLOK	S_NETLOK	S_NETLOK	S_NETLOK	S_NETLOK	S_NETLOK	S_NETLOK	S_
4	NAME		Belgium	Germany1	Germany2	Spain1	Spain2	France1	France2	Greece	Italy	Ki
5	SET_Y		18	11	20	24	20	20	23	23	24	
6	SET_X		11	21	20	3	3	16	16	28	20	
7	*											
8	CREATE											
9	NODE		DEMAND	DEMAND	DEMAND	DEMAND	DEMAND	DEMAND	DEMAND	DEMAND	DEMAND	DE
10	TYPE		DEM_BELGIE	DEM_DEUT	DEM_DEUT	DEM_SPAN	DEM_SPAN	DEM_FRAN	DEM_FRAN	DEM_GRIE	DEM_ITALIE	DE
11	NAME		C_Demand	C_Demand	C_Demand	C_Demand	C_Demand	C_Demand	C_Demand	C_Demand	C_Demand	C_
12	SET_Y		1	1	1	1	1	1	1	1	1	
13	SET_X		0	0	0	0	0	0	0	0	0	
14	*											
15	CREATE											
16	NODE		STORE	STORE	STORE	STORE	STORE	STORE	STORE	STORE	STORE	ST
17	TYPE		PUMPSP	PUMPSP	PUMPSP	PUMPSP	PUMPSP	PUMPSP	PUMPSP	PUMPSP	PUMPSP	PL
18	NAME		SP	SP	SP	SP	SP	SP	SP	SP	SP	SP
19	SET_Y		1	1	1	1	1	1	1	1	1	
20	SET_X		1	1	1	1	1	1	1	1	1	
21	INIT		5E+05	5E+05	5E+05	5E+05	5E+05	5E+05	5E+05	5E+05	5E+05	
22	LOAD		20000000	20000000	20000000	20000000	20000000	20000000	20000000	20000000	20000000	2
23	DUAL		1	1	1	1	1	1	1	1	1	
24	VARI		0	0	0	0	0	0	0	0	0	
25	*											
26	CREATE											
27	NODE		SUPPLY	SUPPLY	SUPPLY	SUPPLY	SUPPLY	SUPPLY	SUPPLY	SUPPLY	SUPPLY	SU
28	TYPE		SOL_BELGIE	SOL_DEUT	SOL_DEUT	SOL_SPAN	SOL_SPAN	SOL_FRAN	SOL_FRAN	SOL_GRIEC	SOL_ITALIE	SC
29	NAME		PH	PH	PH	PH	PH	PH	PH	PH	PH	PH
30	SET_Y		2	2	2	2	2	2	2	2	2	
31	SET_X		1	1	1	1	1	1	1	1	1	
32	INIT		77000000	5E+06	5E+06	1,5E+06	1,5E+06	3E+06	3E+06	88000000	2,7E+06	
33	DUAL		10	10	10	10	10	10	10	10	10	
34	VARI		1	1	1	1	1	1	1	1	1	

Figure 9: managing of the input datasets in EXCEL

For the output file the same procedure is used in the opposite direction. A visual basic macro selects the needed data rows from the output file and creates tabulated sorted files which are readable in EXCEL. These files contain the hourly scattered (depending on the scattering of the input data rows) load curves for every node and every link as it is exemplary shown in figure ten.

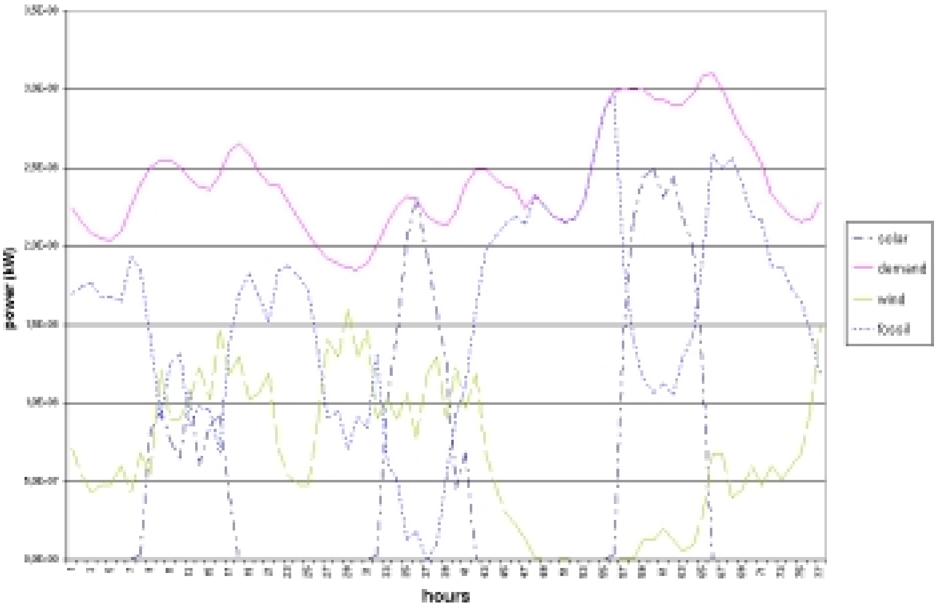


Figure 10: visualized load curves as result of the simulation

Finally it must be mentioned, that the restrictions for the scenario to be calculated are only given by the used computer system. Scenarios with about 100 different nodes (supplies, demands, storages etc.) and the corresponding links are no problem to be simulated on a normal PC if the time resolution is not higher than 8760 time steps (one year – hourly scattered).

All in all, TASES is a very powerful tool, to image virtual energy scenarios, especially then the option of outspread restrictions on more powerful machines is considered.