SIXTH FRAMEWORK PROGRAMME

Project contract no. 043363

MANMADE
Diagnosing vulnerability, emergent phenomena and volatility in man-made networks

SPECIFIC TARGETED PROJECT
NEST PATHFINDER
Sub-Priority Tackling Complexity in Science

Work Package 5 – D5.4
Methodology for studying market dynamics and power systems chains

Due date of delivery: Month 30
Actual submission date: 31st July

Start date of project: 1st of January 2007 Duration: 36 months

Lead authors for this deliverable: [C. Noé, T. Rossi, M. Serati, D. Sorrenti (Università Cattaneo – LIUC)]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

Introduction

This research work presents a methodology useful to evaluate the impact of the electric power supply on both supply chains and production plants performance.

Since this work originates from the previous study "Logical framework of the impact of the electric power supply on a logistic-production system", it uses the approximations of electric power supply and logistic-production systems performance defined into Deliverable 5.1. With reference to the electric power supply, they are electric faults and black-outs occurrence. The considered supply chain performance approximations are: (i) the stock-outs at the retailer stage, (ii) the backlogs at the supply chain nodes other than retailers, (iii) the average inventory level of the whole supply chain, (iv) the total transport distance covered. Finally, with reference to the production plants performance approximation, they are the percentages of defective parts produced as well as of on-time delivered orders.

Besides the electric power supply and the logistic-production system performance approximations, this research work derives from Deliverable 5.1. also the logical framework explaining the relations among the occurrence of different types of electric faults and black-outs and the performance of both supply chains and production plants (see the causal diagram depicted in the abovementioned deliverable).

At the origin, the occurrence of black-outs is supposed to be leaded by a growth of the electricity prices and their volatility which in turn are triggered from a mismatch between electricity demand and supply. A demand/supply gap, in other terms, reveals both a market disequilibrium and a grid congestion.

Here it is worth to notice that, when the supply chain is studied, a lower detail level is used. In particular, the different nodes the supply chain is composed of are treated as black-boxes and the dynamics of the nodes elements (e.g. the machines of a production plant) are neglected. As a consequence, on the supply chain performance only the impacts of black-outs are considered.

The methodology proposed in this work is based on an object oriented simulation meta-model based in turn on Arena™ software tool. In particular, in the first section the overall architecture of the simulation meta-model is presented, while the second and the third sections are devoted to outline the main elements of the meta-model itself when the meta-model is used for studying supply chains and production plants respectively. Finally, the fourth section depicts the multi-step procedure based on a sequence of three econometric models developed to represent into the simulation meta-model the black-outs occurrence as a function of the spot electricity price evolution (in this document the model developed for representing the electric faults occurrence is not presented since it has been illustrated in Deliverable 5.1).

Meta-model architecture

The meta-model allows to define the characteristics of the electric power supply and market as well as the configuration and the management policies of the logistic-production system (supply chain or production plant) to be studied. Then it automatically builds the corresponding Arena™ simulation model. Finally, through experimental campaigns the impact of electric faults and black-outs occurrences on the supply chain/production plant performance can be figured out.

The meta-model is made up from an Excel™ interface with database, a library of objects written in Siman™, i.e. the programming language which Arena™ refers to, and a Visual Basic™ application. Figure 1 shows the components of the meta-model and their interactions.
The Excel™ interface allows the user to specify both configuration and management policies of the logistic-production system. In the case of a supply chain, the configuration is described by: (i) the number of stages of the supply chain, (ii) the number of nodes at each stage; (iii) the node type (manufacturer, distributor or retailer) and the corresponding capacity; (iv) the suppliers of each node; (v) the node-to-node distance.
In the case of a production plant, the configuration is given by: (i) the production phases characterizing the system, (ii) the machines, which perform each phase; (iii) the production capacity of each machine; (iv) the electric faults each machine can suffer from and the corresponding inter-arrival probability distribution (see Deliverable 5.1); (v) the machine-to-machine distances and (vi) the characteristics of the transportation sub-system, which links the different machines.
Then, when a supply chain is studied, the management policy followed by each node should be specified because of the need to add some parameters. If the node adopts a push policy, forecasting, orders fulfilment and transport parameters (if applicable) are to be added. If the node adopts a pull policy, inventory management parameters are to be added.
When a production plant is studied, the dispatching rule followed by each machine (i.e. the rule defining the sequence according to which the items seize the machine) must be defined (at the moment the dispatching rules considered by the simulation meta-model are the following: first in first out (FIFO), earliest due date (EDD), shortest processing time (SPT) and user defined priority).
All the values entered via Excel™ interface are recorded into the Excel™ database.
The Siman™ objects library specifically conceived for supply chains contains all the six available combinations of node type (manufacturer, distributor and retailer) and management policies (push and pull).

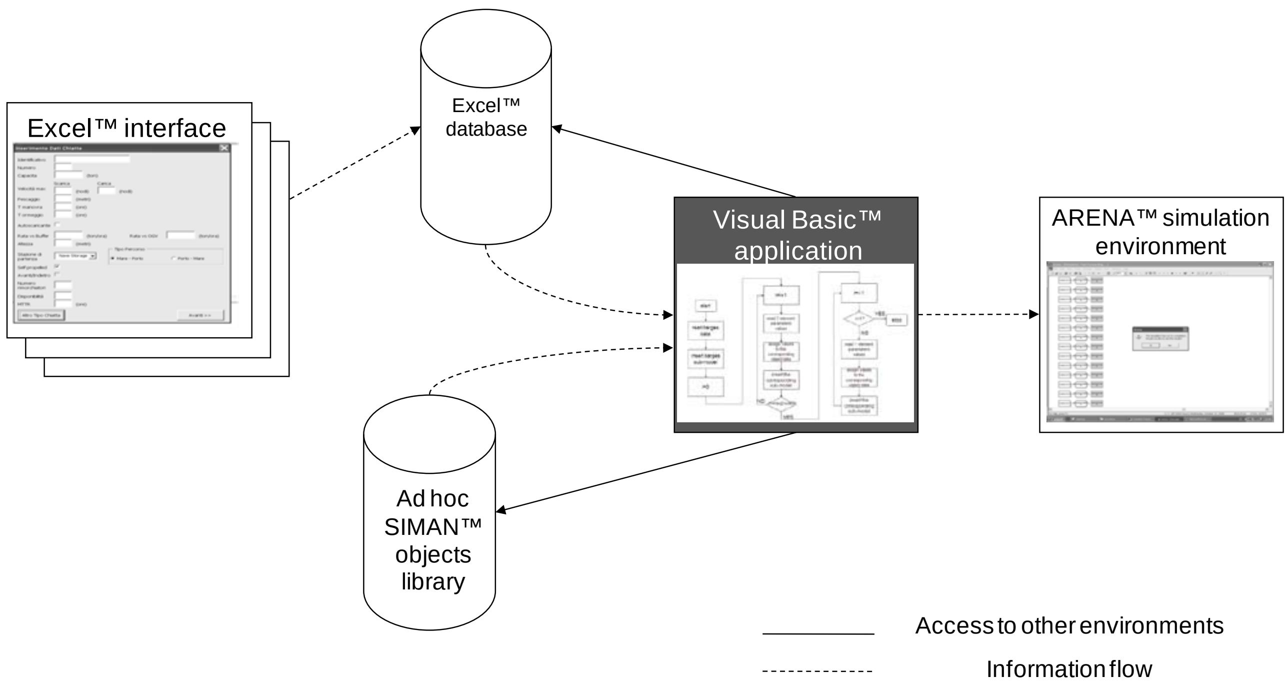

Figure 1. Simulation meta-model architecture

Each combination is an object. The *ad hoc* Siman™ production plant objects library, instead, contains the 'machine' object which a generic plant can be composed of.
Each object is described by behaviour and data. An object behaviour is a Siman™ simulation sub-model that operating on the object data represents how the corresponding node/machine behaves in the real word and how it interacts with the other nodes/machines of the supply chain/production plant. Object data are represented in a parametric form and consist, in the supply chain case, of the node typology, its position in the network, the sourcing strategy, etc.. In the production plant case, of the machine code, its position in the production flows, its production capacity, etc. These data assume the values entered via Excel™ interface and recorded into the Excel™ database through the Visual Basic™ application. Such an application allows the Arena™ simulation model of the supply chain or production plant under study to be automatically built. From the Excel™ database the Visual Basic™ application reads the nodes/machines and, for each of them it: (i) selects the Siman™ object from the *ad hoc* library; (ii) selects from the Excel™ database the values to be assigned to the object data; (iii) makes the assignments (in other words, it generates the instance); (iv) inserts into the Arena™ environment the instance, that is the Siman™ sub-model representing the object behaviour, which operates on the parameterized object data after the assignments. Once the Visual Basic™ application has completed the above mentioned steps for each node/machine, experimental campaigns can be performed on the Arena™ model.
In the following paragraphs more details are given with reference to the Siman™ objects and Visual Basic™ application in the supply chain case and in the production plant case respectively.

Supply chains simulation meta-model

Siman™ objects library

Within the *ad hoc* Siman™ objects library three classes of items have been defined, that correspond to the three types of nodes (manufacturer, distributor and retailer). The objects belonging to each class are given by the combination of the node type and management policy (push and pull). For each of them, data and behaviours are to be specified.
Referring to the data, which are synthesized in table 1, the manufacturer class objects, is characterized by: (i) the code, which univocally identifies the node within the supply chain; (ii) the topological parameters, i.e. the supply chain level which the node belongs to and the number of production resources characterizing the node as well their production rates; (iii) the common management parameters, i.e. the desired safety stock level and the number of transport resources as well their average speed; (iv) the initial values of the node inventory.
In addition, the object manufacturer-push is characterized by the data connected to the forecasts (the parameter, the time bucket of the forecasting exponential smoothing method and the initial expected demand). In turn, the object manufacturer-pull is characterized by the data connected to the inventory management policy (economic order quantity and reorder point).
With reference to the distributor class objects, the data they are characterized by are: (i) the code, the topological parameters, the common management parameters and the initial values of the node inventory, as above; (ii) the desired safety stock level and the number of transport resources as well their average speed, common to push and pull. In addition, the object 'distributor-push' is characterized by the data connected to the forecasts (the parameter, the time bucket of the forecasting exponential smoothing method and the initial expected demand). In turn, the object

‘distributor-pull’ is characterized by the data connected to the inventory management policy (economic order quantity and reorder point).

Node type (class)	Management policy (object type)	Data	Note
Manufacturer	Both push and pull	Code & level	
		Resources number	
		Production rate [units/hour]	
		Safety stock [units]	
		Average speed of transport means [km/hours]	
		Inventory initial value [units]	
	Push-specific only	Alpha (α)	(1)
		Expected demand – initial value [units]	(4)
		Time bucket duration (T) [hours]	(2)
	Pull-specific only	Economic Order Quantity (EOQ) [units]	
		Re-Order Point (ROP) [units]	
Distributor	Both push and pull	Code & level	
		Sources	(5)
		Distances [km]	(6)
		Safety stock [units]	
		Speed [km/hour]	
		Inventory initial value [units]	
	Push-specific only	Alpha (α)	(1)
		Expected demand – initial value [units]	(4)
		Time bucket duration (T) [hours]	(2)
	Pull-specific only	Economic Order Quantity (EOQ) [units]	
		Re-Order Point (ROP) [units]	
Retailer	Both push and pull	Code & level	
		Sources	
		Distances [km]	
		Safety stock [units]	
		Inventory initial value [units]	
		Average inter-arrival (INT) [hours]	(7)
		Mean (μ) [units]	(8)
		Standard deviation (σ) [units]	(9)
	Push-specific only	Alpha (α)	(1)
		Expected demand – initial value [units]	(4)
		Time bucket duration (T) [hours]	
	Pull-specific only	Economic Order Quantity (EOQ) [units]	
		Re-Order Point (ROP) [units]	

Notes:
(1) Parameter of the forecasting smoothing method
(2) At the end of each time bucket a forecast for the next period must be done
(4) It represents the demand expected by the manufacturer during the first time bucket
(5) The list of the codes corresponding to the nodes from which the distributor can be supplied
(6) Distances between the distributor and each source (for all the sources)
(7) Mean of the exponential distribution from which the final customers inter-arrival time is drawn
(8) Mean value of the normal distribution from which the quantity requested by the customer is drawn
(9) Std deviation of the normal distribution from which the quantity requested by the customer is drawn

Table 1. List of the objects data within the Siman™ library

With reference to the retailer class, the ‘retailer-push’ and the ‘retailer-pull’ objects are characterized by the same data of the ‘distributor-push’ and the ‘distributor-pull’ objects respectively. In addition, the objects belonging to the retailer class are characterized by the data connected to the final customers demand (the mean of the exponential distribution from which the customers inter-arrival time is drawn, and mean and standard deviation of the normal distribution from which the quantity requested by the single customer is drawn).
The behaviours of the different objects are Siman™ simulation sub-models which represent how each node acts in the real world and how it interacts with the other nodes of the supply chain. The entities characterizing the sub-models corresponding to the retailer class objects are: (i) the final customers and (ii) the products sold by the retailer. The entities which flow along the sub-models corresponding to the manufacturer and the distributor classes objects are: (i) orders and (ii) products. The black-outs, treated in the same way described for the logistic-production systems case, are the entities characterizing all the sub-models.
The Siman™ simulation sub-models representing the objects behaviours are depicted in tables 2, 3 and 4 by means of the attributed Petri nets formalism. The Petri-nets of the object behaviour for the retailer-pull, distributor-push and manufacturer-pull will be described in details in tables 2, 3, 4 as well. The decision to depict the object behaviour only for the abovementioned instances of nodes is that such a sample allows for describing the behaviours characterizing all the supply chain nodes classes (i.e. manufacturer, distributor and retailer) as well as the two types of management policies considered (i.e. pull and push).

Visual Basic™ application

The Visual Basic™ application is described in figure 2.

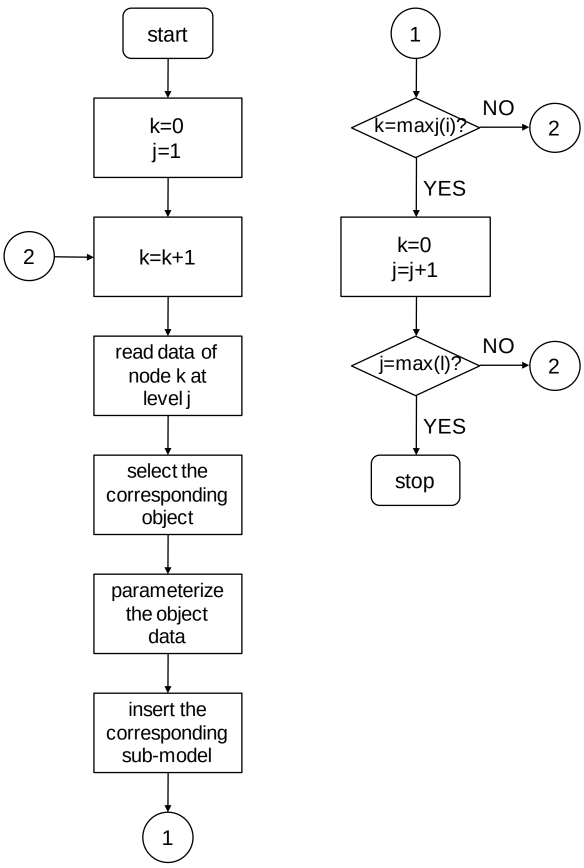

Figure 2. Visual Basic application flow diagram

It starts by setting k = 0 and j= 1. Counter j indicates the considered supply chain level, while counter k addresses the considered node of the level. Then, the Visual Basic™ application increments the counter k, accesses the Excel™ database and reads the values of the data characterizing the node k at level j.
After that, the application accesses the *ad hoc* Siman™ objects library, selects the object corresponding to the considered node, parameterizes its data according to the previously read values (from the Excel™ database) and it inserts the sub-model representing the object behaviour into the Arena™ simulation environment. At this point, the Visual Basic™ application checks if the value of k is equal to the maximum node code for the supply chain level j, that is it checks whether all the nodes belonging to the level j have been already processed.
If not, the above described sub-procedure is performed again, starting from the k counter increment. If yes, k is re-initialized to 0 and the counter j is incremented. Then, the Visual Basic™ application checks if the value of j is equal to the supply chain farthest level from the final customer (that is if all the levels of the considered supply chain have been already processed). If not, the Visual Basic™ application re-starts from the increment of counter k; if yes, the Visual Basic™ application stops and the Arena™ simulation model of the considered supply chain is ready to use.

Production plants simulation meta-model

Siman™ objects library

Within the *ad hoc* Siman™ objects library only one class of items has been defined: the machine. For such an object, data and behaviours are specified.
Referring to the data, they are: (i) the code, which univocally identifies the machine within the logistic-production system; (ii) the configuration parameters, i.e. the production phase performed by the machine, its production capacity, the distance between the machine and the other ones; (iii) the probability distribution of electric faults occurrence; (iv) the probability distribution of the electric faults effects; (v) the probability distribution of the restoring time (depending on the electric faults effects); (vi) the management parameters, i.e. the applied dispatching rule (for a complete overview of the objects data, see table 5).
The behaviour of the object ‘machine’ is a Siman™ simulation sub-model. As every Siman™ simulation model, also the considered sub-model is characterized by entities, which represent the elements of the real world that influence the real system functioning. The Siman™ simulation sub-model represents how the machine behaves in the real world, i.e. how it interacts with the other machines of the logistic-production system and with the entities. In particular, the entities characterizing the machine sub-model are: (i) the batch orders; (ii) the items each order is composed of; (iii) the electric faults and (iv) the black-outs. Each of these entities is characterized in turn by attributes according to which the low of the entity along the simulation sub-models is managed.
With reference to the entity ‘batch order’, its attributes are: (i) the type of the product the order refers to; (ii) the number of items the order is composed of (batch order size); (iii) the date when the order has been placed; (iv) the order delivery date.
The first two attributes of the entity ‘order’ are also attribute of the entity ‘item’. Besides them, the ‘item’ entity is also characterized by the array ‘production route’ (i.e. the sequence of machines the entity must visit), by the cycle times at the different machines and, finally, by the attribute ‘defect’, which records if the item is defective or not.

Manufacturer pull functioning. When there is a token in place PO_{l-1}, i.e. when an order has been placed by a node belonging to the downstream supply chain level, if the token attribute s is equal to i, i.e. if the order has been placed to the manufacturer under study, transition $T_{i,1}$ becomes active. It removes the token from PO_{l-1} and creates one token in $P_{i,2}$, which represents the order placed at the manufacturer. Moreover, transition $T_{i,1}$ initializes to 0 the pc attribute of such a token (this means that the order does not deal with the completion of a partial consignee) and assigns the value of the token attribute OQ to the variable Q. Once the token is in $P_{i,2}$, if the manufacturer inventory (given by the number of tokens held by place $P_{i,3}$) is sufficient for satisfying the order (i.e. if the number of tokens in $P_{i,3}$ is higher than Q) transition $T_{i,2}$ fires, otherwise (and if pc is equal to 0) transition $T_{i,3}$ is activated. $T_{i,2}$ removes Q tokens from $P_{i,3}$ as well the token from $P_{i,2}$ and creates one token in place $P_{i,4}$. Moreover, it assigns to the token attribute DQ the value of the token attribute OQ. $T_{i,3}$, instead, removes the token from $P_{i,2}$ and creates one token in place $P_{i,7}$ as well one token in $P_{i,2}$ again. It also assigns to the attributes DQ, pc and OQ of such tokens the values INV_i, 1 and Q-INV_i respectively. The token in place $P_{i,7}$ makes active transition $T_{i,8}$ that removes DQ tokens from place $P_{i,3}$ and creates one token in place $P_{i,4}$. When there is the token in $P_{i,4}$ and one token at least in $P_{i,5}$ (i.e. one of the distributor transport resources at least is available) transition $T_{i,5}$ starts to fire. It removes the token from $P_{i,4}$ and one token from $P_{i,5}$ and, after a duration given by the ratio between the distance of the manufacturer (node i) from the distributor who made the order (indicated by the token attribute c) and the average speed of the manufacturer transport resources, it creates one token in PS_{l-1} (i.e. in the place where the tokens representing the performed consignments directed towards the downstream supply chain level (l-1) are collected). Moreover, transition $T_{i,5}$ assigns to the variable rti the value of its duration d. Once the token is in PS_{l-1} and the value of its attribute s is equal to i, transition $T_{i,6}$ starts to fire and, after its duration (equal to the value of the rti variable), it creates one token in place $P_{i,5}$ (i.e. it makes the manufacturer transport resource again available). When the manufacturer inventory is lower than the re-order point (i.e. when the number of tokens in $P_{i,3}$ are less than ROP_i) and the value of variable order_i is equal to 0 (i.e. the manufacturer has not already placed a new production order) transition $T_{i,4}$ becomes active and creates one token, which represents a production order, in place $P_{i,6}$. Moreover, $T_{i,4}$ assigns the value 1 to the variable order_i and it parameterizes the token attributes OQ with the economic order quantity of the manufacturer under study (EOQ_i). When there is one token in $P_{i,6}$ and one token at least is in place $P_{i,7}$ (i.e. one of the manufacturer production resources at least is available), transition $T_{i,7}$ starts to fire. After its duration, given by the ratio between the quantity to be produced (OQ) and the production rate (pr_i), it assigns the value 0 to the variable order_i and creates OQ tokens in place $P_{i,3}$, i.e. it increases the manufacturer inventory level of the ordered quantity, and creates one token in place $P_{i,7}$ (i.e. it makes the manufacturer production resource again available).

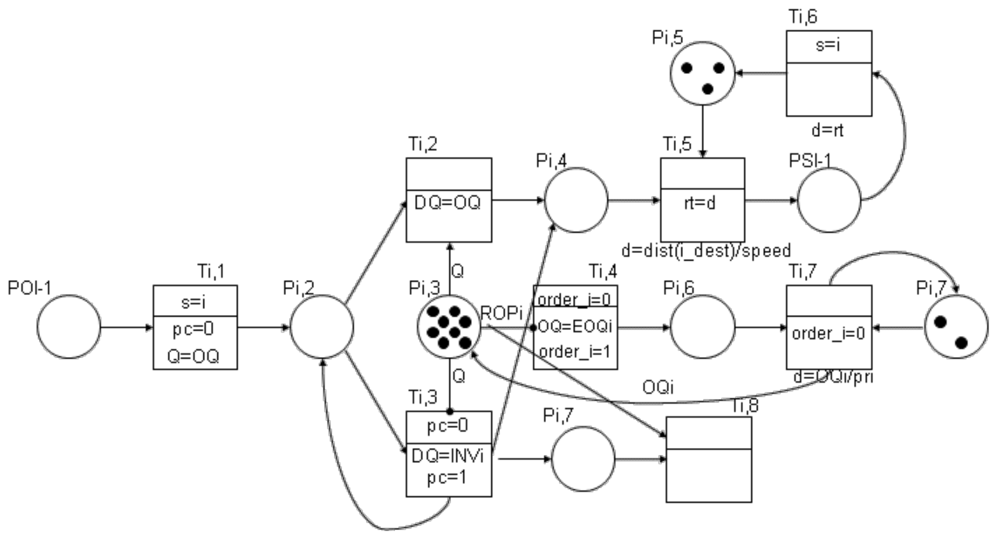

Table 2. Petri net describing the object manufacturing pull functioning behaviour

Distributor push functioning. When there is a token in place PO_{I-1}, i.e. when an order has been placed by a node belonging to the downstream supply chain level, if the token attribute s is equal to i, i.e. if the order has been placed to the distributor under study, transition $T_{i,1}$ becomes active. It removes the token from PO_{I-1} and creates one token in $P_{i,2}$, which represents the order placed at the distributor. Moreover, transition $T_{i,1}$ initializes to 0 the pc attribute of such a token (this means that the order does not deal with the completion of a partial consignee) and assigns the value of the token attribute OQ to the variable Q. Once the token is in $P_{i,2}$, if the distributor inventory (given by the number of tokens held by place $P_{i,3}$) is sufficient for satisfying the order, i.e. if the number of tokens in $P_{i,3}$ is higher than Q, transition $T_{i,2}$ fires, otherwise (and if pc is equal to 0) transition $T_{i,3}$ is activated. $T_{i,2}$ removes Q tokens from $P_{i,3}$ as well the token from $P_{i,2}$ and creates one token in place $P_{i,4}$. Moreover, it assigns to the token attribute DQ (which stands for delivered quantity) the value of the token attribute OQ and updates the value of the variable AD_i (such a variable represents the actual demand, i.e. the sum of the retailers demands collected during the whole period t by the distributor). $T_{i,3}$, instead, removes the token from $P_{i,2}$ and creates one token in place $P_{i,7}$ as well as one token in $P_{i,2}$ again. It also assigns to the attributes DQ, pc and OQ of these tokens the values INV_i, 1 and Q-INV_i respectively. The token in place $P_{i,7}$ makes active transition $T_{i,8}$ that removes DQ tokens from place $P_{i,3}$ and creates one token in place $P_{i,4}$. When there is a token in $P_{i,4}$ and one token at least in $P_{i,5}$, i.e. one of the distributor transport resources at least is available, transition $T_{i,5}$ starts to fire. It removes the token from $P_{i,4}$ and one token from $P_{i,5}$ and, after a duration given by the ratio between the distance of the distributor (node i) from the retailer who made the order (indicated by the token attribute c) and the average speed of the distributor transport resources, it creates one token in PS_{I-1}, i.e. in the place where the tokens representing the performed consignments directed towards the downstream supply chain level (I-1) are collected. Moreover, transition $T_{i,5}$ assigns to the variable rti the value of its duration d. Once the token is in PS_{I-1} and the value of its attribute s is equal to i, transition $T_{i,6}$ starts to fire and, after its duration (equal to the value of the rti variable), it creates one token in place $P_{i,5}$, i.e. it makes the distributor transport resource again available. Finally, place $P_{i,6}$ and transition $T_{i,7}$ allow to represent in a push context the placing of the orders by the distributor. As a matter of fact, when there is the token in $P_{i,6}$ transition $T_{i,7}$ is active. Such a transition removes the token from place $P_{i,6}$ and after its duration (given by the time bucket duration), it creates one token in place PO_I. Moreover, it parameterizes the variable ED_i (the expected demand for the next period) and the token attribute OQ according to equations y and x respectively. It also parameterizes attributes c and s, which represent the node's code and the source to which the node places the order, and re-initializes the variable AD_i.

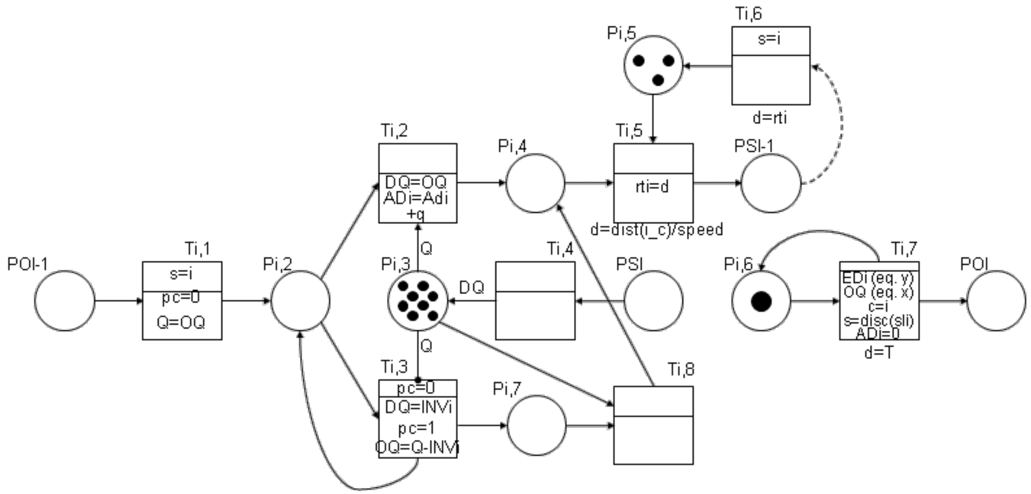

Table 3. Petri net describing the object distributor push functioning behaviour

Retailer pull functioning. Place $P_{i,1}$ and transition $T_{i,1}$ allow the final customers demand to be represented. When there is one token in $P_{i,1}$ transition $T_{i,1}$ becomes active. It cancels the token from $P_{i,1}$ and after its duration, whose value is drawn from an exponential distribution with mean INT_i, it creates one token in $P_{i,1}$ and one token in $P_{i,2}$. The latter token represents the final customer arrived. Moreover, transition $T_{i,1}$ assigns to the attribute q of such a token a value drawn from a normal distribution with mean μ_i and variance σ_i (the q attribute represents the number of items required by the customer) and to the variable Q the value of the attribute q. Once the token is in $P_{i,2}$, if the retailer inventory (given by the number of tokens held by place $P_{i,3}$) is sufficient for satisfying the customer demand, i.e. if the number of tokens in $P_{i,3}$ is higher than Q, transition $T_{i,2}$ fires, otherwise transition $T_{i,3}$ is activated. $T_{i,2}$ removes Q tokens from $P_{i,3}$ and the token from $P_{i,2}$ and creates one token in $P_{i,4}$, which records the number of satisfied customers. $T_{i,3}$, instead, only removes the token from $P_{i,2}$ and creates one token in place $P_{i,5}$, which records the number of stock-outs experienced by the retailer. When the retailer inventory is lower than the re-order point, i.e. when the number of tokens in $P_{i,3}$ are less than ROP_i, and the value of variable order_i is equal to 0, i.e. the retailer has not already placed a new order to the upstream supply chain stage. transition $T_{i,4}$ becomes active. It creates one token, which represents an order, in place PO_I, i.e. in the place that collects all the tokens representing orders of nodes belonging to the same level I. Moreover, $T_{i,4}$ assigns the value 1 to the variable order_i and the values of the economic order quantity, of the node code and of the selected source code to the token attributes OQ, c and s respectively. Finally, when there is a token in place PS_I, i.e. when a consignment directed towards the supply chain level I has been performed, if the token attribute c is equal to i, i.e. if the consignment is directed towards the node under study, transition $T_{i,5}$ becomes active. It cancels the token from PS_I, it assigns the value 0 to the variable order_i and it creates DQ tokens in place $P_{i,3}$, i.e. it increases the retailer inventory levels of the delivered quantity.

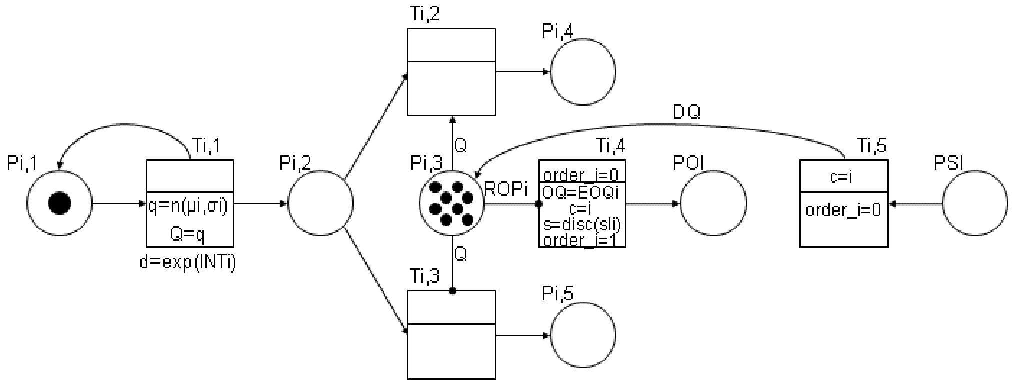

Table 4. Petri net describing the object retailer pull functioning behaviour

Object type	Data	Notes
Machine	Code (C)	
	Production phase	(1)
	Production capacity [hours]	(2)
	Distances [m]	(3)
	Electric faults inter-arrival times [hours]	(4)
	Probability of each inter-arrival time	(5)
	Probability of each electric fault	(6)
	Restoring times for each electric fault [hours]	
	Probability of each restoring times	
	Dispatching rule	(7)

Notes:
(1) Processing phase performed by the machine
(2) Number of daily working hours
(3) Distances between the machine and each machine of the logistic-production system (for all the machines)
(4) They represents the different values that can be assumed by the electric faults inter-arrival time
(5) Probability according to which the electric faults inter-arrival time can assume the abovementioned value (for all the values)
(6) Popularity coefficients of the different electric faults (once an electric fault is occurred, they allow for defining its typology)
(7) Such a data can assume the values: 1 (FIFO), 2 (EDD), 3 (SPT) and 4 (user defined priority)

Table 5. Object data

With reference to the entity ‘electric fault’, its attributes are: (i) the electric fault type (the values of this attribute is drawn from the model described into Deliverable 5.1) and (ii) the ‘idle’ attribute, which records if the machine is idle or not when the electric fault occurs.

Finally, no attribute characterizes the entity ‘black-out’. Actually, a black-out does not occur according to a certain inter-arrival probability distribution but its occurrence depends on the evolution of the spot electricity price (for a deep explanation of how the flow of the ‘black-out’ entity along the simulation sub-models is triggered, see the section devoted to the econometric model).

A synthetic overview of the entities attributes is depicted in table 6.

The Siman™ simulation sub-model representing the object ‘machine’ behaviour is depicted in figure 3 by means of the attributed Petri nets formalism. In the following, the Petri nets of the object behaviour is described in details.

When there is one token at least in place $P_{C,in}$ (i.e. when one order at least is waiting for being processed by the generic machine ‘C’) and there is the token in $P_{C,1}$ (i.e. the generic machine ‘C’ is not processing any order), transition $T_{C,1}$ becomes active. On the one side, it cancels the token from $P_{C,1}$ and one token from $P_{C,in}$ (when there is more than one token in $P_{C,in}$, the cancelled token depends on the machine dispatching rule). On the other side, the transition creates in $P_{C,2}$ a number of tokens equal to the value of the attribute ‘order size’ of the token removed from $P_{C,in}$. Finally, the transition $T_{C,1}$ records on the attribute ‘t1C’ of each token the actual time ‘tnow’ (i.e. the time at which the order starts to be processed by the generic machine ‘C’).

When there is one token at least in place $P_{C,2}$ (i.e. when one item at least is waiting for being processed by the generic machine ‘C’) and there is the token in $P_{C,3}$ (i.e. the generic machine ‘C’ is not processing any items), transition $T_{C,2}$ becomes active. One token is removed from both place

$P_{C,2}$ and $P_{C,3}$ (i.e. one item seizes the generic machine ‘C’) and one token is created in $P_{C,4}$ (the actual time ‘tnow’ is recorded on its attribute ‘t1’).

Entity	Data	Notes
Batch order	Type	(1)
	Order size	(2)
	Order date	(3)
	Order delivery date	
Item	Type	
	Order size	
	Production route [array]	(4)
	Cycle times [array]	(5)
	Defect	(6)
Electric fault	Type	(7)
	Idle	(8)
Black-out		(9)

Notes:
(1) Type of the product the order refers to
(2) Number of items the order is composed of
(3) Date at which the order has been placed
(4) Sequence of machine the item must visit
(5) Item cycle time at each machine (for all the machines)
(6) Binary attribute: its value is 1 if the item is defective, 0 otherwise
(7) Electric fault type
(8) Binary attribute: its value is 1 if the machine is idle when the electric fault occurs, 0 otherwise
(9) Entity without attributes

Table 6. Entities attributes

The token in $P_{C,4}$ makes active both transitions $T_{C,3}$ and $T_{C,4}$. The transition $T_{C,3}$ represents the item processing by the machine. It cancels the token in $P_{C,4}$ and, after a duration given by the item cycle time (i.e. by the value of the token attribute ‘ct’), creates one token in $P_{C,5}$. The transition $T_{C,4}$ allows for representing the electric faults effects on the generic machine ‘C’ by means of the token it creates in place $P_{C,6}$ (here it is worth to notice that transition $T_{C,4}$ assigns to the attribute ‘t1’ of this token the actual time ‘tnow’). As a matter of fact, only when there is one token in both places $P_{C,5}$ and $P_{C,6}$, transition $T_{C,5}$ is activated.
It removes the token from the two mentioned places and, after a duration given by the expression ‘ct-wt’ (for the meaning of the attribute ‘wt’ and the calculation of its value see in the following). Moreover, the transition $T_{C,5}$ creates one token in $P_{C,3}$ (i.e. it makes the generic machine ‘C’ available for being seized by another item) and another one in $P_{C,7}$ and records on the ‘w_C’ attribute of the latter token the value of the variable ‘$defect_C$’. The value of this variable, whose definition is explained in the following, allows for specifying if the item produced by the generic machine ‘C’ is defective or not.
Once a number of tokens equal to their ‘order size’ attribute value is in $P_{C,7}$, i.e. when all the items of the order have been processed by the ‘C’ machine, transition $T_{C,6}$ becomes active.

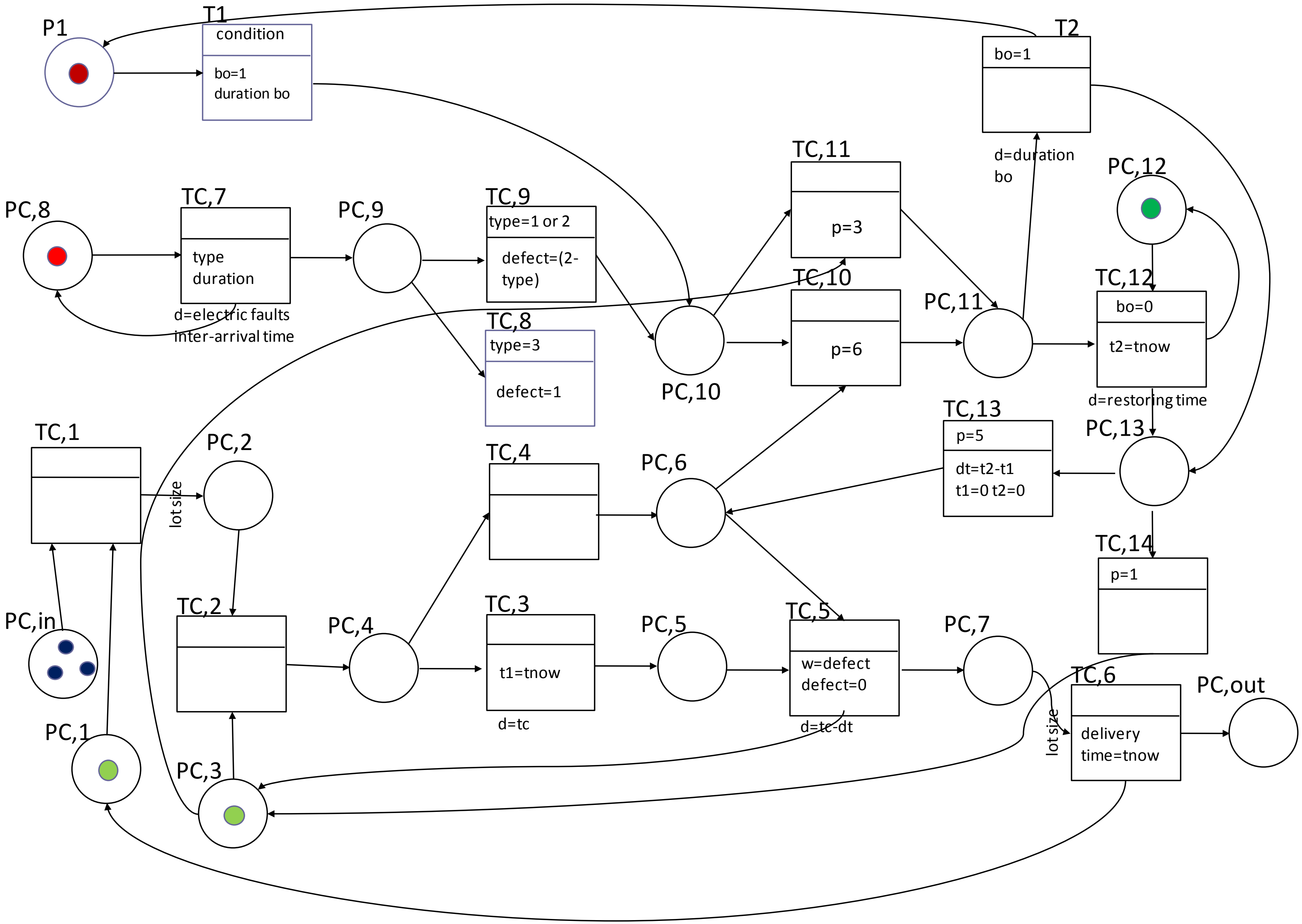

Figure 3. Petri net describing the object 'machine' behaviour

It removes the ‘order size’ tokens from $P_{C,7}$ and creates one token in $P_{C,1}$ (i.e. allows machine ‘C’ for processing a new order) and another one in place $P_{C,out}$. Finally, transition $T_{C,6}$ assigns to the attribute ‘delivery time’ of the last token the actual time ‘tnow’.
When there is a token in place $P_{C,8}$ transition $T_{C,7}$ starts to fire. It cancels the token from $P_{C,8}$ and, after a duration drawn from the electric faults inter-arrival time probability distribution (for details see Deliverable 5.1), it creates one token in $P_{C,8}$ (in this way the occurrence of the next electric fault on the generic machine ‘C’ can be simulated) and one token in $P_{C,9}$. Moreover, it assigns to the attribute ‘type’ of the latter token the code representing the typology of the electric fault occurred (also for the popularity of each electric faults typology see Deliverable 5.1).
Depending on the value of its attribute ‘type’, the token in place $P_{C,9}$ allows for activating transition $T_{C,8}$ or transition $T_{C,9}$. The first corresponds to the electric faults, which cause defective parts and do not stop the production process. As a consequence, transition $T_{C,8}$ cancels the token from $P_{C,9}$ and assigns to the variable ‘$defect_C$’ the value ‘1’. The transition $T_{C,9}$ corresponds to the electric faults which stop the production process (for details on the electric faults typologies see Deliverable 5.1). This transition cancels the token from $P_{C,9}$, creates on token in place $P_{C,10}$ and assigns to the ‘$defect_C$’ variable the value given by the expression ‘2-type’.
Depending on the status of the generic machine ‘C’, the token in $P_{C,10}$ activates transition $T_{C,10}$ or transition $T_{C,11}$. In particular, if the machine is idle, i.e. there is the token in place $P_{C,3}$, transition $T_{C,11}$ fires, whereas if the machine is processing an item, i.e. there is a token in place $P_{C,6}$, transition $T_{C,10}$ becomes active. Both transitions cancel the token from $P_{C,10}$, create one token in place $P_{C,11}$ and cancel the tokens from $P_{C,6}$ and $P_{C,3}$. Moreover, $T_{C,10}$ and $T_{C,11}$ assign the values ‘6’ and ‘3’ respectively to the attribute ‘p’ of the token created in $P_{C,11}$. The attribute ‘p’ substantially records if the electric fault is occurred when the machine was occupied by an item, i.e. when a token was in place $P_{C,6}$ (attribute value equal to 6), or if the electric fault is occurred when the machine was idle, i.e. when a token was in place $P_{C,3}$ (attribute value equal to 3). Finally, $T_{C,10}$ assigns to the attribute ‘t2’ of the token created in $P_{C,11}$ the actual time ‘tnow’.
The token in $P_{C,11}$, if there is one token at least in place $P_{C,12}$ (i.e. at least one maintenance operator is available), activates transition $T_{C,12}$. It cancels the tokens from $P_{C,11}$ and $P_{C,12}$ and, after a duration drawn from the restoring time probability distribution (such distribution is specified by the user via Excel™ interface), creates one token both in places $P_{C,13}$ and in $P_{C,12}$ (i.e. it makes again available the maintenance operator).
The token in $P_{C,13}$, depending on the value of its attribute ‘p’, activates transition $T_{C,13}$ or transition $T_{C,14}$. The transition $T_{C,13}$ cancels the token from $P_{C,13}$, creates one token in $P_{C,6}$ and assigns to the attribute ‘wt’ the value ‘t2-t1’. The attribute ‘wt’ stands for ‘worked time’ and records the time already spent for item processing. Obviously, for completing the production process onto machine ‘C’, the item must be worked for a number of time units given by the difference between its cycle time and the already spent processing time. The transition $T_{C,14}$, instead, cancels the token from $P_{C,13}$ and creates one token in $P_{C,3}$ (i.e. makes the machine in the idle status again available).

The black-outs occurrence is not represented by any of the places and transitions above described. However the Petri nets that model the machine behaviour and the black-outs occurrence are strictly linked. For this reason, hereinafter Petri net of the black-outs occurrence is described until it flows into the machine Petri net.
When a token is in place P_1 and a certain condition in the spot electricity price evolution is reached (for details see the section of this document devoted to the econometric model) transition T_2 fires. It cancels the token from P_1, creates one token in the place $P_{C,10}$ of the generic machine Petri net and assigns to the attribute ‘bo’ of this token the value ‘1’ (the attribute ‘bo’ is equal to ‘1’ if a black-out is occurred, ‘0’ otherwise). When one token with the attribute ‘bo’ equal to ‘1’ is in the place $P_{C,11}$

of the generic machine Petri net, the transition T_3 of the black-outs occurrence Petri net starts to fire. It cancels the token from $P_{C,11}$ and, after a duration given by the black-out duration (for details see the section of this document devoted to the econometric model), creates one token in P_1 (in this way the next black-out can be simulated) and one token in the place $P_{C,11}$ of the generic machine Petri net.

Visual Basic™ application

The flow diagram of the Visual Basic™ application is represented in figure 4.

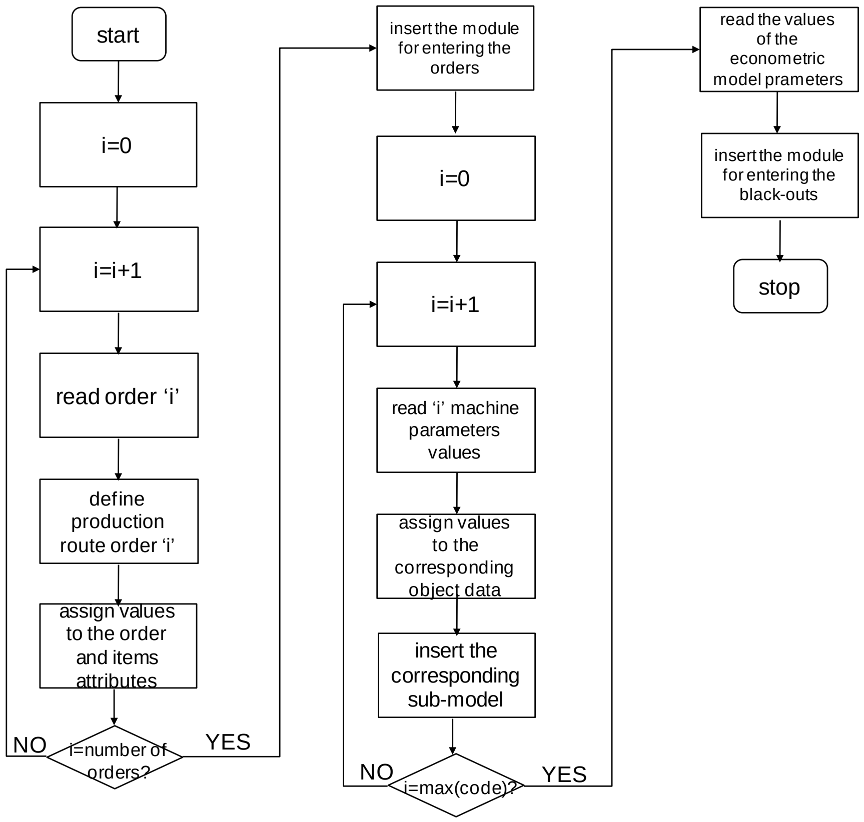

Figure 4. Flow diagram of the Visual Basic™ application

The program starts by initializing the counter i to the value 0. Such a counter indicates time by time the order considered by the Visual Basic™ application. Then, it increments the counter i, accesses the Excel™ database and reads the values of the data (i.e. the product the order refers to, the order size, the date when the order has been placed and the order delivery date) characterizing the i-th order. Due to the product the order refers to, the Visual Basic™ application reads also the sequence of machines such a product must visit and then assigns all the read values to the corresponding attributes of the entity, which represent the considered order.
At this point, the Visual Basic™ application checks if the value of i is equal to the number of orders recorded into the Excel™ database (i.e. if all the orders have been already processed). If not, the above described sub-procedure is again performed starting from the i counter increment; if yes, the simulation sub-model for creating orders is entered into the Arena™ environment and the counter i is re-initialized to 0.

Then, the Visual Basic™ application increments the value of counter i, accesses the Excel™ database and reads the values of the data characterizing the i-th machine (i.e. production capacity, distance from each of the other machines, restoring times as well their occurrence probabilities and applied dispatching rule).

After that, the application accesses the *ad hoc* Siman™ objects library, selects the object 'machine', parameterizes its data according to the previously read values (from the Excel™ database) and it inserts the sub-model representing the object behaviour into the Arena™ simulation environment. At this point, the Visual Basic™ application checks if the value of i is equal to the maximum machine code for the considered logistic-production system (i.e. if all the machines have already been considered).

If not, the above described sub-procedure is again performed starting from the i counter increment; if yes, the Visual Basic™ application accesses again the Excel™ database, reads the values of the parameters characterizing the econometric model, which is responsible for the black-outs generation (for details see the section of this document devoted to such econometric model), makes the necessary assignments and enters into the Arena™ environment the simulation sub-model for creating black-outs.

At this point, the Visual Basic™ application stops and the Arena™ simulation model of the considered logistic-production system is completed and ready to be used.

Econometric model

The adoption of an econometric model aimed at finding (in a dynamic framework) the main determinants of the electricity prices behaviour and produce joint forecasts for their evolution and the occurrence of grid black-outs and disturbances requires to take into account six fundamental points arising from the analysis of the theoretical and empirical econometric literature on electricity prices:

- The electricity market retains absolutely peculiar characteristics: it is an auction market that, although liberalised, is not strictly a spot one, but it requires both price and quantity of equilibrium to be defined one day in advance on the basis of expected supply and demand. This guarantees a good match among supply and demand, that, due to the non-storability of electricity, to unexpected peaks in demand and to congestions over the distribution network, could fail, causing jumps in prices and leading in extreme cases to the system blackout.
- The series of electricity prices have complex statistical properties that vary depending on spectral frequency to which data are measured and on sample size. Depending on the cases, it is possible to notice phenomena of seasonality at different frequencies, trends which are more or less linear at low frequencies, phenomena of auto-correlated volatility at high frequencies, and combinations of outliers apparently managed by non standard distributions.
- A wide range of models dedicated to the analysis of the properties of price series follow an approach that can be defined as being agnostic from the point of view of economic interpretation, meaning they do not foster the inference on (economic) factors that influence prices, but they limit the analysis to only their statistical properties.
- However, it seems evident that the evolution of prices over time is driven by the interaction between supply and demand of electricity, that is, from two phenomena not directly

measurable and in someway latent. Therefore, in order to effectively model demand and supply it would be suitable to include in the model those factors that determine their trend: for example climatic factors or the business cycle state that affect demand; productivity, size of the plant and costs of production concerning supply. It is an insidious approach, as these determinants play a role at different frequencies and usually statistical data on them are characterised by significant measurement errors, which makes more difficult the correct identification of the effects caused by each phenomenon on prices.

- Even for the hidden dangers previously mentioned, the econometric models dedicated to the analysis of electricity prices adopt very simplified specifications, often uniequational, taking into account only a few aspects of the issue at a time.
- Among the models proposed by the literature, none of them seems to be characterised by a uniformly better capability of fitting the data and by an outperforming forecasting behaviour; depending on the market taken as reference, on the sample of data being considered and on the measure of forecasting performance chosen, now prevail very simple autoregressive models, whereas other times Markow switching models with changing regimes.

The Multistep procedure

In the light of the previous stylized issues, we consider the necessity of adopting a completely new methodological framework in order to efficiently specify and forecast the behaviour of electricity prices; an eclectic approach is needed which enables the estimate and the effective identification of the unobservable dynamics of electricity demand and supply, the management of extremely wide datasets containing high frequency data, the coexistence of short term determinants of electricity prices with those of long term[1], the creation of forecasts on future trends as well as simulations of the impacts of structural shocks.

This innovative methodological tool is represented by a sequence of three different models (three steps procedure):

1. A dynamic factor model (henceforth DFM). These models were introduced in the late ‘70s and present characteristics which are definitely appropriate for the resolution of the six problems highlighted in the analysis of the literature on modelling and forecasting the electricity prices. Within the DFM framework it is possible to:
   - Reduce the problem size by extracting from a larger database a small set of synthetic measures: the Factors. This is a crucial point, given that SVAR models (step 2 of this procedure) were born to manage small-medium groups of variables.
   - Identify, estimate and analyse properties of widespread but unobservable variables; this is another basic point since within an economic framework usually we have no available data on supply and demand.
   - Clean the data, separating measurement errors and idiosyncratic behaviours from the economic structural signal.

   Our DFM allows to identify and estimate, although not observable variables, two orthogonal factors i.e. the market electricity demand and supply that seem to be the main determinants of prices.

1 Extracting the economic signal from the noise

2. A Structural VAR model (henceforth FASVAR) including the electricity price series, their volatility, the series of grid disturbances and the demand and supply factors obtained at the previous step. Within the FASVAR framework it is possible to:
    - Simulate the joint behaviour of the system taking into account all the dynamic and instantaneous cross-correlations among the included variables.
    - Generate the Dynamic Multipliers of the system and the so-called Impulse Response Functions (IRF), that provide a picture of the dynamic reaction of a target variable with respect to a shock occurring on a trigger variable.

   In this case we are able to evaluate the existence of a significant and positive dynamic correlation between price peaks and grid disturbances where prices lead disturbances; moreover model simulation reveals a strong correlation even between price volatility and grid disturbances. In other terms an unbalanced gap between demand and supply triggers both some market turbulence inducing a price unstability and also a grid congestion.
3. A Bayesian VAR model (henceforth BVAR) based on the same group of variables as FASVAR model, but estimated with Bayesian techniques. Such an approach, which is particularly useful for forecasting purposes, in this case has been specified on the basis of the output of steps 1 and 2. In particular SVAR simulation has been the starting point for the calibration of the BVAR hyper-parameters that tune the relative strength of priors and data. Within the FASVAR framework it is possible to:
    - Produce (both unconditional and conditional) forecasts for all the endogenous variables and a measure of uncertainty around them

   Within the BVAR framework it is possible to anticipate the future occurrence of a black-out conditionally on a forecasted growth of prices and their volatility.

In synthesis, the DFM combines the role of all the electricity price contributors in a small and manageable set of determinants; the SVAR model uses this synthetic information set to simulate existence, size and timing of the impact of a price (volatility) shock on the probability of a grid black-out and vice versa. Finally, the BVAR model provides joint forecasts of prices and grid black-outs using the first as leading (both in the logical sense and also in the timing sense) for the second.
In the figure 5 the Multistep procedure flow diagram is depicted.

The DFM model

Since the end of eighties it has clearly emerged that Dynamic (common) Factors Models could provide a "natural" way of summarizing in a formal framework the informational content of large macroeconomic datasets and provide a sounder statistical basis for the construction of composite measures of some target phenomena. Their great advantage is to efficiently reduce the large dimensional problem of handling tons of variables to identify and estimate a very small number of components. In a sequence of cornerstone papers, Stock and Watson (1989 - SW89, 1991, 1992) show how to obtain through the Kalman filter the maximum likelihood estimation of the parameters and the factors in a DFM cast into state space form and within this framework they rationalize and refine the U.S. Business cycle coincident composite index produced by the Conference Board.
Since SW89, a large body of literature has been developed on DFMs and focused on their forecasting ability (Stock and Watson 2002b), the adoption of different weighting schemes of variables contained in the original dataset (Stock and Watson 2002a) and different estimation techniques (FHLR) based on the use of the Principal Components.

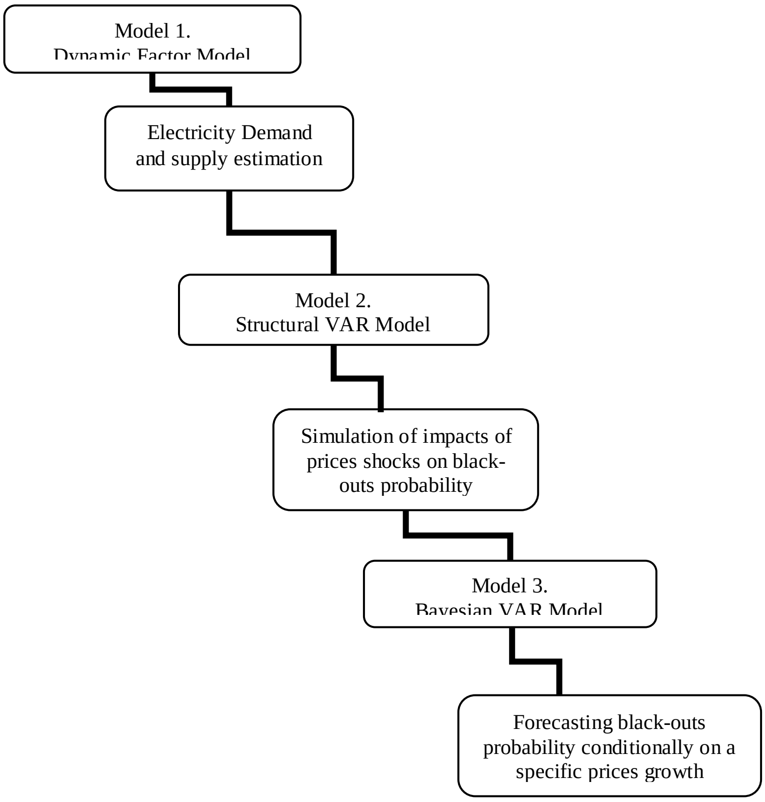

Figure 5. Flow diagram of Multistep procedure

In substance, Dynamic Factor Models (DFMs) have been developed as a powerful tool for exploiting the information contained in large datasets and summarizing the covariances among the variables contained therein. DFMs allow to describe the behaviour of each series as the sum of two components: the dynamics of a reduced number of common factors and an idiosyncratic shock. Let us collect the n variables of the dataset in the vector X_{t} and q common factors in vector f_t

The dynamic form of a DFM may be expressed as follows (SW, 2005):

$$\underset{(n\times 1)}{X_t} = \underset{(n\times q)}{\Lambda(L)}\underset{(q\times 1)}{f_t} + \underset{(n\times n)}{D(L)}X_{t-1} + \nu_t$$
$$f_t = \underset{(q\times q)}{\Gamma(L)}f_{t-1} + \eta_t \qquad [1]$$

where n (usually large) is the number of variables in the model, q the number of dynamic, primitive factors, D(L) is a diagonal matrix lag polynomial D(L)=diag(δ_1(L),...,δ_{n}(L)) and Λ(L) has degree p-1.

$$E(f_t) = 0, E(v_t) = 0, E(f_t v_\tau) = [0]$$

Common factors (f_{t}) and idiosyncratic shocks are uncorrelated at all leads and lags.

Chamberlain and Rothschild (1983) make a distinction between exact and approximate DFMs; in the former case E(v_{it}v_{jτ})=0,∀i≠j, in the latter there exists some contemporaneous correlation.

Let us define a vector containing the so-called static factors:

$$F_t = \begin{bmatrix} f_t' & f_{t-1}' & \cdots & f_{t-p+1}' \end{bmatrix}'.$$

The static form corresponding to the previously presented dynamic model is as follows:

$$\underset{(n\times 1)}{X_t} = \underset{(n\times r)}{\Lambda}\underset{(r\times 1)}{F_t} + \underset{(n\times n)}{D(L)}X_{t-1} + \nu_t$$

$$\underset{(r\times 1)}{F_t} = \underset{(r\times r)}{\Phi(L)}F_{t-1} + \underset{(r\times q)}{G}\eta_t$$

[2]

- $r=(p\times q)$ is the number of static factors (F_{t}).
- $\Phi(L)$ consists of the coefficients of $\Gamma(L)$ and zeros
- If order of $\Gamma(L)$ not higher than p,then $\Phi(L)=\Phi$
- If p=1, static factors coincide with dynamic factors.

The VAR form of a DFM (FAVAR model; Bernanke, Boivin and Eliasz, 2005) might be obtained by substituting equation 2 of system 2 into equation 1:

$$F_t = \underset{(r\times r)}{\Phi(L)}F_{t-1} + \varepsilon_{Ft}$$

$$\underset{(n\times 1)}{X_t} = \underset{(n\times r)}{\Lambda}\underset{(r\times r)}{\Phi(L)}\underset{(r\times 1)}{F_{t-1}} + \underset{(n\times n)}{D(L)}X_{t-1} + \varepsilon_{xt}$$

$$\varepsilon = \begin{bmatrix} \varepsilon_{Ft} \\ \varepsilon_{xt} \end{bmatrix} = \begin{bmatrix} G\eta_t \\ \Lambda G\eta_t + \nu_t \end{bmatrix}$$

$$E(\varepsilon\varepsilon') = \Sigma_\varepsilon = \begin{bmatrix} \underset{(r\times r)}{G\Sigma_\eta G'} & \underset{(r\times n)}{G\Sigma_\eta G'\Lambda'} \\ \underset{(n\times r)}{\Lambda G\Sigma_\eta G'} & \underset{(n\times n)}{\Lambda G\Sigma_\eta G'\Lambda' + \Sigma_\nu} \end{bmatrix}$$

[3]

The (1,1) block of Σ_{ε} contains the variance and covariance matrix of the static factors which is a function of its dynamic counterpart Σ_{η}; matrix G relates dynamic and static factor innovations. Notice that:

- the $\varepsilon_{x,t}$ have factor structure
- the $\varepsilon_{F,t}$ have factor structure without idiosyncratic noise
- $\mathrm{rank}(G)=\mathrm{rank}(G\Sigma_{\eta}G')=q$.
- $G\Sigma_{\eta}G'$ is positive semidefinite

Inverting the system 3 and focusing on X_{t} yields its MA representation in terms of current and lagged orthogonal innovations η_{t} to the dynamic factors:

$$\underset{(n\times 1)}{X_t} = \underset{(n\times q)}{B(L)}\eta_t + u_t$$

where:

- $B(L)=[I-D(L)L]^{-1}\Lambda[I-\Phi(L)L]^{-1}G$ and $u_{t}=[I-D(L)L]^{-1}\nu_{t}$
- impact multipliers: $B_0=\Lambda G$,
- long run multipliers: $B(1)=[I-D(1)]^{-1}\Lambda[I-\Phi(1)]^{-1}G$

Estimation may be obtained following a three step approach

For simplicity let us assume that we are in the condition in which the dynamics of $\Lambda(L)$ is no higher than p (the loadings have lags which do not exceed the dynamics of dynamic factors.

Step 1: Given the number of dynamic factors q, get F, Λ, D(L), by solving iteratively the following minimization problem:

$$min_{F_1,..F_T,\Lambda,D(L)} T^{-1} \sum \left[I - D(L)L)X_t - \Lambda F_t\right]' \left[I - D(L)L)X_t - \Lambda F_t\right]$$

Solution requires:

- step 1a: F_{t} can be computed by applying static PCA to $X_{t}=[I_{n}-D(L)L]X_{t}$
- step 1b: regress X_{it} on F_{t} and on $X_{it-1},...,X_{it-m}$ to get estimate of $\delta_{i}(L)$ and Λ

  Each step of this procedure reduces (does not increase) the sum of squares and the procedure can be iterated to convergence.
- step 1c: estimate the number of static factors r using Bai and NG (2002) IC criteria.

Step 2: get $\Phi(L)$, by auxiliary regressions

Step 3: Let us consider the simplest case when $\Phi(L)=\Phi$ and $D(L)=D$. The VMA representation of the FAVAR becomes:

$$X_t = (I - DL)^{-1}\Lambda(I - \Phi L)^{-1}G\eta_t + v_t$$
$$= A(L)G\eta_t + v_t = B(L)\eta_t + v_t$$
$$B_i = A_i G$$
$$A_0 = \Lambda, A_i = DA_{i-1} + \Phi^i$$

With G in hand we can obtain the IRFs and FEVDs for structural common shocks.

It is possible to exploit the factor structure of ε_{xt} in order to get estimate of G and the space spanned by the dynamic factor innovations η_{t}, and recover the dynamic factors.

Let us normalise η_{t} to have identity matrix; then we can write:

$$\Sigma_{\varepsilon x} = E(A(L)G\eta_t\eta_t' G' A(L)') + \Sigma_v$$

and taking trace

$$trace(\Sigma_{\varepsilon x}) = trace\left[G\prime\left(\sum_{i=0}^{\infty} A_i' A_i\right)G\right] + trace(\Sigma v)$$

therefore we are able to estimate G to max trace R^2, by computing G as the q eigenvectors associated to the highest q eigenvalues of $\sum_{i=0}^{\infty} A_i' A_i$. G is then normalised to generate orthonormal disturbances via the relation $\varepsilon_{Ft}=G\eta_{t}$

The number of dynamic factors q is estimated by applying the Bai-Ng (2002) procedure to the sample covariance matrix of ε_{xt}, yielding an estimator q. It is worth to note that this procedure finds the estimates of the innovations to the dynamic factors η_{t} on the basis of an arbitrary statistical normalization and not a theoretical structural economic model; in other words the impulse responses and variance decompositions delivered by the VMA representation of the DFM can be thought of as the factor version of impulse responses and variance decompositions with respect to Cholesky factorizations of conventional VAR innovations. The dynamic factor structural shocks ζ_{t}, that is the orthogonal shocks admitting an economic interpretation, are assumed to be linearly related to the reduced form dynamic factor innovations by:

$\zeta_{t}=H\eta_{t}$

where H is an invertible $q\times q$ matrix and $E(\zeta_{t}\zeta_{t}')=I$,so that $H\Sigma_{\eta}H'=I$.

In order to achieve the really structural dynamic factor shocks ζ_{t} Stock and Watson (2005) illustrate a set of different strategies, all based on zero restrictions on dynamic multipliers, as they have been proposed in the structural VAR literature. Christiano, Eichenbaum and Evans (1999) adopt a recursive identification scheme based on restrictions on the impact multipliers and inspire the Bernanke, Boivin and Eliasz (2005) FAVAR proposal, whereas Blanchard and Quah (1989) impose long run restrictions first used in FAVAR models by Giannone, Reichlin, and Sala (2002). Anyway, exclusion restrictions have been strongly criticized in the literature: Faust and Leeper (1997) show that small sample bias and measurement errors may induce substantial distortions in the estimations when using long run zero restrictions. On the other side, short run restrictions may be to much stringent and misleading: in many cases they are introduced not due to theoretical foundations but they are arbitrary imposed to respect order and rank conditions for identification. Moreover, Peersman (2004) shows that a large number of impulse responses based on zero restrictions are located in the tails of the distributions of all possible impulse responses.

In order to avoid technical problems of this sort in this paper we follow an identification strategy based on sign restrictions (Faust, 1998; Uhlig, 1999; Canova e De Nicolò, 2002): different dynamic factor shocks are identified according to the direction of their impact on the variables in the system.

In details, we specifyd a DFM related to the NORDPOOL grid including data on:

- **Electricity prices** (hourly sampled)
- Price volatility (daily based)
- **Electricity Production, Consumption and Net Imports** (monthly sampled)
- **Installed capacity** (yearly sampled)
- **Exchange of electricity between the countries** (yearly sampled)
- **Maximum system load** (effective) (yearly sampled)
- **Interconnections** (yearly sampled)
- **Black-out and disturbances** (yearly sampled)

Data have been collected for each one of the member countries and each kind of power, like Nuclear, Hydro and Thermal, for example.

We include into the model a quite large autoregressive structure (24 lags) and we estimate and identify on the basis of sign restrictions two orthogonal factors representing the electricity demand and supply. All the usual standard test controlling for the optimal number of factors and the quality of estimates confirm the reliability of our results.

The (Factor Augmented)SVAR Model

At the second step of our procedure, the demand and supply Factor measures generated through the DFM enter a SVAR model, jointly with the series of electricity prices and the series of grid black-outs.

A VAR model is a system of seemingly unrelated equations (SURE model; Zellner, (1962)) able to representing the whole set of dynamic correlations linking the interest variables; as a consequence in a VAR model all the phenomena are supposed to be jointly endogenous.

In formal terms the VAR representation for a $(n\times1)$ vector of series $\mathbf{y}_t$ is as follows:

$$\mathbf{y}_t = \mathbf{\Phi}\, \mathbf{d}_t + \mathbf{A}_1 \mathbf{y}_{t-1} + \mathbf{A}_2 \mathbf{y}_{t-2} + \ldots + \mathbf{A}_k \mathbf{y}_{t-k} + \boldsymbol{\varepsilon}_t \quad \boldsymbol{\varepsilon}_t \sim \mathrm{VWN}(\mathbf{0},\mathbf{\Sigma}), \qquad [4]$$

where $\mathbf{A}_i$ are square autoregressive matrices which size is n, whereas $\mathbf{d}_t$ is a deterministic components vector.

Equation [4] describes the evolution of each component collected in vector $\mathbf{y}_t$ as driven both by its own past behaviour and by the past behaviour of all the other endogenous in the sistem. For this reason, VAR models can be viewed as the conditional reduced forms of the following structural model:

$$\mathbf{B}\mathbf{y}_t = \boldsymbol{\Theta}\,\mathbf{d}_t + \boldsymbol{\Gamma}_1\mathbf{y}_{t-1} + \boldsymbol{\Gamma}_2\mathbf{y}_{t-2} + \ldots + \boldsymbol{\Gamma}_k\mathbf{y}_{t-k} + \boldsymbol{\upsilon}_t \qquad [5]$$

where $\mathbf{A}_i=\mathbf{B}^{-1}\times\boldsymbol{\Gamma}_i$ e $\boldsymbol{\Phi}=\mathbf{B}^{-1}\times\boldsymbol{\Theta}_i$ e $\boldsymbol{\varepsilon}_t=\mathbf{B}^{-1}\times\boldsymbol{\upsilon}_t$

It is worth noting that <u>simultaneous</u> linkages among the variables $\mathbf{y}_i$ are hidden in the variance and covariance matrix of the system error terms.

After having estimated a VAR model through ML estimation, one could look for some measure of the impact of a shock affecting one of the endogenous variables (the trigger one) onto another variable, the target one. This simulation step needs to move from the estimated coefficients of equation [4] to those of equation [5]: in other terms we have to solve an identification problem, switching from an unconstrained VAR to a Structural VAR approach (SVAR)[2].

The ratio is to make explicit the usually hidden instantaneous correlations among the endogenous by imposing them a direction of causality; this is the same as to identify a set of original orthogonal shocks and analyse the dynamic reaction of all the system variables with respect of these shock. The way3 is to pre-multiply equation [4] by the inverted Cholesky factor (P^{-1}) of Σ:

$$\mathbf{A}_0^{*}\mathbf{y}t = \mathbf{A}_1^{*}\mathbf{y}_{t-1} + \mathbf{A}_2^{*}\mathbf{y}_{t-2} + \ldots + \mathbf{A}_k^{*}\mathbf{y}_{t-k} + \mathbf{e}_t,\ \mathbf{e}_t \sim \mathbf{VWN}(0,\mathbf{I}_n), \qquad [6]$$

where $A_0^*=P^{-1}$, $A_i^*= P^{-1}A_i$ e $PP'=\Sigma$

A_0^* is a lower triangular matrix which main diagonal elements are equal to 1 which implies a recursive identification scheme: orthogonal shocks on the top variables istantaneously affect the bottom variables and not *vice versa*. Identification makes it possible to simulate over the relevant time horizon the Impulse Response Functions4 (IRFs) that describe the shape of the dynamic reaction of variable j with respect to a shock occurring on variable i.

The Factor Augmented SVAR model we specify for the NORDPOOL electricity market provides a quite interesting empirical evidence: there exists a significant and positive dynamic correlation between a price peak and grid disturbances where the first lead the second (see Figure 1). Moreover, model simulation reveals a strong correlation even between price volatility and black-outs. In other terms it seems that an unbalanced gap between demand and supply generates both a kind of market triggers both some market turbulence inducing a price unstability and also a grid congestion.

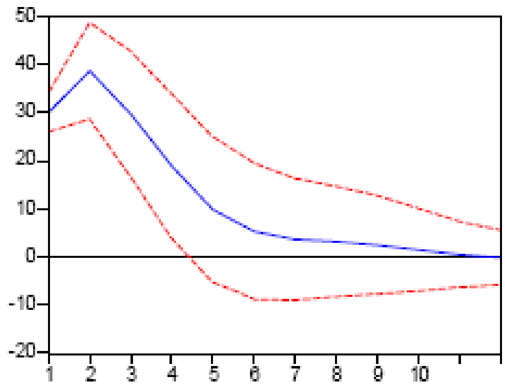

Figure 6. Response of black-outs to a prices shock

2 For a survey: Amisano and Giannini (1997).

3 We are referring to the exact identification case.

4 And their confidence bounds

Figure 6 shows that after a positive price shock there is a significant growth of the probability of grid black-outs over a period of four days. Similar results are found when a volatility shock does occur.

The forecasting Bayesian VAR model

The last step of our procedure is developed within a Bayesian VAR framework based on the same group of variables as the previous FASVAR model

One of the main drawbacks of (S)VAR models is their profligate parameterization which is particularly relevant whenever large systems have to be managed: too many parameters to estimate and an information set which is not large enough. This feature is typically reflected in a low efficiency level of estimates and an unsatisfactory degree of quality of the forecasts.

Doan, Litterman e Sims (1986) overcome this problem moving to a bayesian framework: all the model parameters are considered as random variables and their estimation combines (in an optimal way) the informations coming from data (synthetized by the likelihood function of the model) with theoretically inspired *a-priori* which role is both to enlarge the available information set (higher efficiency) and strengthen the model fit.

Let us consider the i^{-th} VAR equation:

$$y_{it} = \mathbf{x}_t'\boldsymbol{\beta}_i + \boldsymbol{\varepsilon}_{it},\ \boldsymbol{\varepsilon}_{it} \sim N(0, \sigma^2_i) \qquad [7]$$

Prior informations on the parameters are collected in a system of stochastic linear constraints:

$$\mathbf{R}\,\boldsymbol{\beta}_i = \mathbf{d} + \mathbf{e}_0,\ E(\mathbf{e}_0) = \mathbf{0},\ var(\mathbf{e}_0\,\mathbf{e}_0') = \mathbf{Q}_0. \qquad [8]$$

Priors represent a kind of extra-sample information and could be treated as *p* additional observations in the sample; on this basis it may be derived the mixed Bayesian GLS estimator proposed by Theil-Goldberger:

$$\tilde{\boldsymbol{\beta}}_i = [\sigma^{-2}\,\mathbf{X}'\mathbf{X} + \mathbf{R}'\,\mathbf{Q}_0^{-1}\mathbf{R}]^{-1}[\sigma^{-2}\mathbf{X}'\mathbf{y} + \mathbf{R}'\,\mathbf{Q}_0^{-1}\mathbf{d}],$$
$$var(\tilde{\boldsymbol{\beta}}) = [\sigma^{-2}\,\mathbf{X}'\mathbf{X} + \mathbf{R}'\,\mathbf{Q}_0^{-1}\mathbf{R}]^{-1} \qquad [9]$$

As for the specification of the prior distribution we follow the so called *Minnesota prior* which features are indexed to a small set of hyperparameters calibrated in order to optimize the forecasting performances of the model. In particular we follow the suggestions coming from step two (SVAR model): the hyperparameter tuning the intensity of the link between prices and black-outs has received a higher weight. The forecasting performance of this BVAR model has been measured over a five days horizon by means of the Theil's U indexes. The evidence on prices and black-outs series is reported in table 7 and suggests a quite encouraging model performance: in fact all the Theil's indexes are largely smaller than one.

Forecasting Horizon	Prices equation	Black-outs equation
1-step ahead	0.645	0.714
2-step ahead	0.681	0.706
3-step ahead	0.704	0.755
4-step ahead	0.735	0.802
5-step ahead	0.779	0.813

Table 7. Theil's indexes

On the basis of the previous results the BVAR model (step 3 of our procedure) seems to be a valuable tool for anticipating the future occurrence of a black-out (we forecast both the number of future black-outs and also their probability to occur) conditionally on a forecasted growth of prices and their volatility.
The output of the econometric model represents the condition that triggers the black-outs occurrence into the objects behaviours characterizing the simulation meta-model.

Concluding remarks

This deliverable is focused on the definition of a simulation meta-model, which allows for automatically building logistic-production systems/supply chains simulation models for evaluating the impact of electric faults and black-outs on the real systems. Logistic-production systems and supply chain managers can benefit from this work since the presented tool provides an effective support for assessing the vulnerability of the plant or of the supply chain to the power supply quality.
The reason for coping with this issue is twofold: first, simulation is one of the most suitable decision support tool for analyzing plants and supply chains; second, notwithstanding the above mentioned statement and the advantages, which can be easily demonstrated, in testing for instance, countermeasures to black-outs on a simulation model rather than in the real-life, simulation is not widely applied in industry. This is basically due to the fact that building a simulation model can be a very complex and time consuming task, which companies cannot often cope with since their human resources do not have the necessary competencies and/or enough time.
The simulation meta-model developed by the research work is made up from: (i) an Excel™ interface, which allows the user to define the characteristics of the logistic-production system or of the supply chain; (ii) an ad hoc SIMAN™ objects library, which contains the objects representing the machines or the nodes a plant or a supply chain can be composed of; (iii) a Visual Basic™ application, which starting from the data entered via Excel™ interface and from the ad hoc SIMAN™ objects automatically builds the ARENA™ simulation model corresponding to the considered logistic-production system or supply chain.
Among the information entering, as inputs, the simulation meta-model, a particular attention has been devoted to the specification of the probability distribution of black-outs occurrence. Its properties has been defined within a multi-step econometric procedure based on three models arranged in sequence. A Dynamic Factor Model allows to estimate the otherwise not measurable determinants of the electricity prices behaviour and in particular the market demand and supply. Demand and supply factors enter a Structural VAR model that identifies and estimates all the instantaneous and lagged correlations between prices (and their volatility) and black-outs. Finally, the set of simulated impacts of prices (volatility) on black-outs probability is the main reference in order to specify the prior distribution of a Bayesian forecasting VAR model: within this framework we forecast the probability of a black out occurrence, conditionally to a price growth.
Then, by experimenting on such a model and measuring the output of the experimental campaign (basically the percentages of defective parts produced as well of on-time delivered orders (for the logistic-production system case) and the number of stock-outs at the retailer stage, the number of backlogs at the nodes belonging to the other supply chain stages, the average inventory level of the whole supply chain and the total distance covered by the transport resources along the supply chain (for the supply chain case)), the user is able to verify in advance the effects of faults and black-outs on the system performance.

A company has two main advantages, strictly connected one to the other, in using the proposed simulation meta-model. First, it can finally exploit simulation techniques in coping with the power supply issue. As a matter of fact, since through the simulation meta-model the simulation model of the specified plant or supply chain is automatically built, neither the human resources competencies nor their impossibility to spend a lot of time in building the simulation model are no more hurdles. Second, the use of the proposed simulation meta-model allows to dramatically reduce the time required to assess the system vulnerability to electric faults and black-.outs and to test potential countermeasures. In few minutes the user can specify the logistic-production system/supply chain characteristics through the Excel™ interface; immediately the Visual Basic™ application builds the corresponding simulation model, which anyhow can be run in few time, even if depending on the simulation length and on the hardware.