FORAC

DE LA FORÊT AU CLIENT

LogiLab – User guide

Alexandre Morneau, Rémy Huot, Sébastien Lemieux, Line Simoneau

FORAC Research Consortium

Table of contents

1 LogiLab description

LogiLab enables linear programing based optimization. It's a mathematical programing method used to find a way to obtain the best possible result (like profit maximization) in a given mathematical model. It can be used to model many linear relations. Minimal or maximal capacities can be modeled on factory transformations, stocking or transport. Constrains like demand mandatory to fulfill can also be modeled. Linear programing can be applied on many domains likes business, economy and engineering. It has been used extensively in the past to solve distribution and production planification problems. It also has been used a lot to solve resources allocations and affectations problems.

Logilab can be used to design and plan supply chains, using graphic edition of mathematical models. Those models are solved by LogiLab using specialized optimization engines like Gurobi, CPLEX, etc. LogiLab allows users with no mathematical programing background to easily develop and solve mathematical models. In LogiLab environment, mathematical models are built by associating different real life objects like products (bought, sold, transformed or transported), suppliers (harvested forests, sawmills, chemical factories, etc), transformation centers (with different capacities and capabilities), clients (with different needs or orders). Each of these modeled object has a position, and different transport options can be defined with given capacities, speed and costs to move products around. Then, LogiLab automatically arrange all these objects in a coherent mathematical model and solve it. Once optimized, the optimal solution can be visualized on a schematic map (Figure 1) or on a geographic map (Figure 2). This visual representation of the results gives insight to facilitate analysis and presentation to stakeholders.

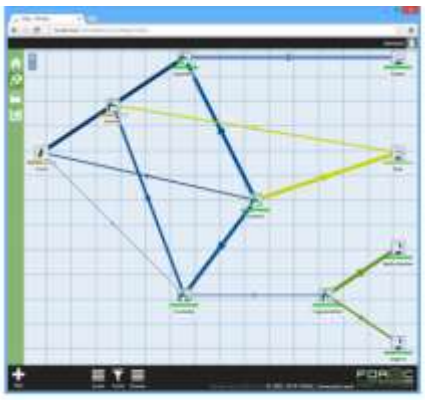

Figure 1: Schematic map

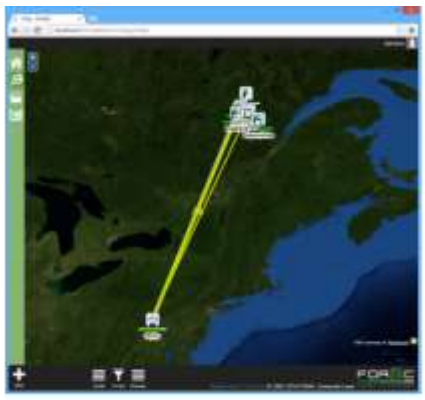

Figure 2: Geographic map

1.1 Breaking down the network in business units

A value creation network can be built of many companies, factories, harvest zones, etc.

1.1.1 Business units

A Business unit (BU) can be associated to a physical location (ex: factory) or it can be a virtual location (ex: classification center). A BU is a network node where it's possible to buy, stock, transforms or sell products.

1.1.2 Processes

Processes allows to define all product transformations in LogiLab. As shown in Figure 3, a process contains one or many input products, a transformation length and cost, and one or many output products.

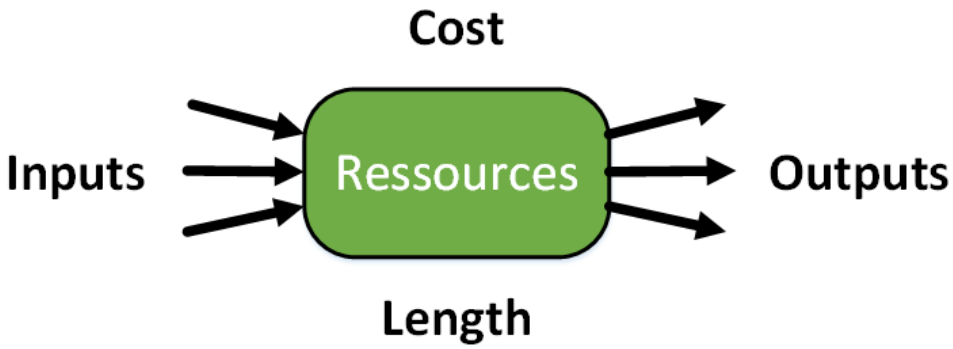

Figure 3: Process graphical representation

1.1.3 Example

Before building the model, it's important to know what specific problematic will be the subject of the analysis. Let's imagine a basic case built of one forest supplier, one sawmill, one paper mill and one client for each of the two factories. Here is how they should be modeled in function of the subject of the analysis:

- Base case:
  - Forest: Supplies an estimated volume of one transformable product
  - Sawmill: A very simple model: logs in, average product basket out
  - Paper mill: A very simple model: shavings in, paper out
  - Clients: Demands should be adjusted according to sales forecasts

- Measuring the impact of adding a new sawing line:
  - Forest, Paper mill and Client : Same as base case
  - Sawmill: Separated into two BU: (1) Sawing and (2) drying and planning together

- Measuring the impact of a modification in the paper making recipe:
  - Forest, Sawmill and Client: Same as base case
  - Paper mill: If the impact of the paper making recipe is generalized to the whole paper mill, it can modeled as one BU. If not, it will have to be decomposed into many BU.

- Measuring the impact of buying forest products at auctions:
  - Sawmill, Paper mill and Client : Same as base case
  - Forest: Modeled as many BU. One for the base case and one more for each different possible supply source.

The strategy is to keep the model as simple as possible and to only add details at the parts where we want the measure the impact of changes. Because, the simplest is the model, the easier it is to validate and to solve effectively.

1.2 Model temporal granularity

LogiLab needs to separate the temporal space into periods. A period is a time unit. It can represent an hour, a shift, a month, a year, etc. It is possible to use a finer time granularity, but it might not be recommended in most of the cases. The more there are periods, the harder the problem will be to solve. Very often, the difficulty of the problem grows faster than the number of periods.

The choice of the number of periods is important. For tactic level problems, periods usually last between a week and a month. This allows to consider the seasonality in sales, supplies or transformation processes while keeping the size of the model manageable.

1.3 Product aggregations

It's often a good idea to reduce the number of products to simplify the model. However, this must not impact the quality of the solution. Sometimes, it's interesting to keep a very detailed product hierarchy (ex. : 2x4 16', 2x4 14', etc.). However, most of the time keeping only generic product family is enough (2x4, 2x6, etc). It's also possible to use different product aggregations for each product families.

2 Using LogiLab

This section is about navigating the user interface of LogiLab and detailing the input data format.

2.1 Navigating LogiLab user's interface

The LogiLab user interface is separated in four main parts (see Figure 4).

- The menu bar (in black at the top of the screen)
- The context bar (in black at the bottom of the screen)
- The navigation bar (in green at the left of the screen, highlighted in a red rectangle in the screenshot)
- The main window (at the center of the screen)

The navigation bar allows to navigate to the 4 main sections of LogiLab :
(from top to bottom)

- Home section
- Map section
- Optimization section
- Analysis section

The home section is shown at Figure 4. At the rightmost of the menu bar, there is a button named "Sign in" that must be used to log into LogiLab.

Figure 4: LogiLab home page

2.1.1 Optimization section

LogLab optimization section (Figure 5) shows the list of every problems that you can access. This list can be filtered using the search bar at the top of it. Clicking on a problem will show its details. Creating a new problem is possible at any time by using the “new” button at the bottom left of the screen, on the context bar.

When creating a new problem, inputs files must be provided. Also, you must specified a group to which the problem will belong. By default, the problem will private, meaning only you and the system administrators will be able to see it. To be able to share a problem with other users, you must assign it to a group. The group “Public” will make the problem visible to all LogiLab users. If you work in a team, you can contact the LogiLab administrators and they will create a new group for your team.

When visualizing a problem, three pages are available. The first page is the problem summary (Figure 6). It shows the problem name, group, description, input files and list of its business units and flows. A button allows you to edit the information of the project and its group. Another button allows to protect the problem. Protecting a problem disables the option to delete it. A reason must be provided to protect the problem. These two buttons are highlighted in red in Figure 6.

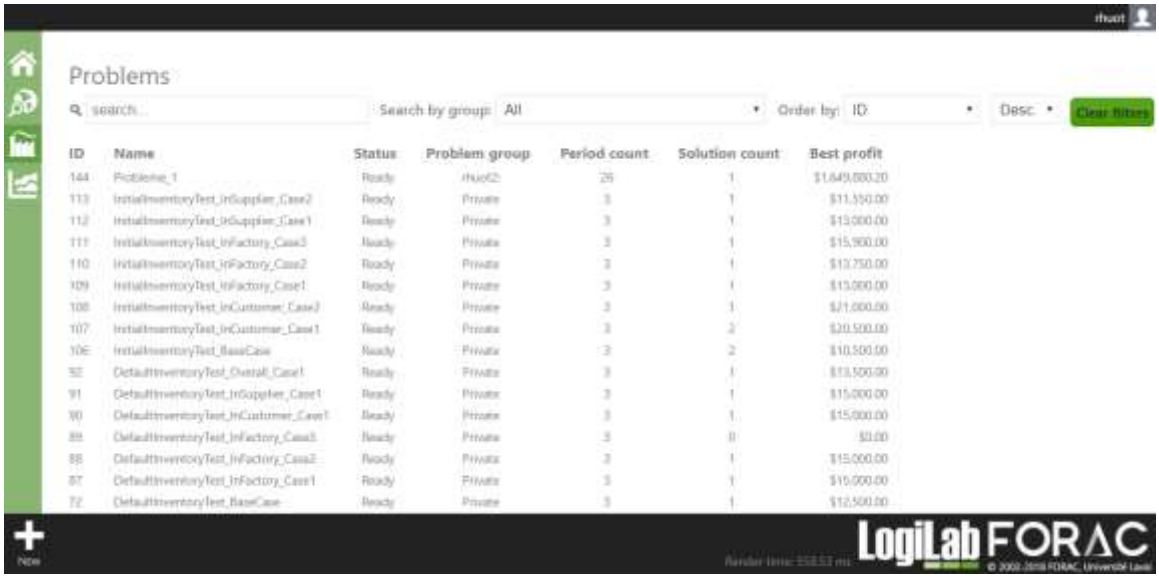

Figure 5 –LogiLab optimization section

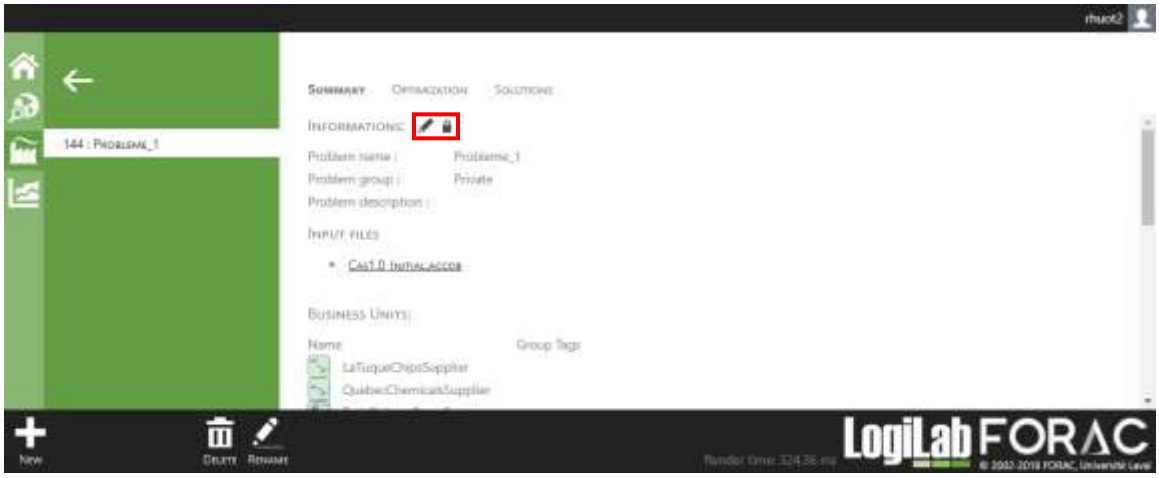

Figure 6 – Problem summary

The second page is the problem optimization page (Figure 7). It shows, and allows to modify, many parameters before launching the solver to optimize the problem:

- *SolverName*: Solver to use. Supported solvers are GLPK, CBC, Gurobi and CPLEX.
- *IsMip*: If true, the problem will be solved as an integer model. If false, the problem will be solved as a linear model.
- *TimeLimit*: Time (in seconds) allocated to the solver to find the optimal solution. When expired, the best solution found will be returned.
- *MipGap*: Minimal Gap (not percentage) to reach to return a solution. For example, a 0.01 Gap tells the solver to stop when it reach a solution not less than 1% inferior to the best possible solution.
- *RandomizeOrder*: If true, when building the model, write variables in a random order. This can be used to add variation to the time to solve a Mip model.
- *GetLpFile*: If true, a ".lp" will be generated. These files are a written version of the generated model. They can be very usefull when debugging models.
- *GetSolFile*: If true, a ".sol" file will be generated after the optimization. These files contain a written version of the found solution.
- *ExportExcelSolution*: If true, an Excel solution file will be generated after the optimization.
- *ExportAccessSolution*: If true, an Access solution file will be generated after the optimization.
- *FlowDiscretizationDelayPercentage*: Discretization rate used for transport delays. For example, a rate of 30% implies that transports delays lasting more than 30% of a period will be considered as lasting a full period.

- *ProcessDiscretizationDelayPercentage* : Discretization rate used for process delays. For example, a rate of 30% implies that process delays lasting more than 30% of a period will be considered as lasting a full period.
- *EnableInteractiveSolution* : Enable the interactive optimization feature.
- *Optimization file* : Optionally add a complementary Python script file to customize the LogiLab model.

The « *Start new optimization* » button launches a new optimization. The LogiLab server will enqueue the optimization job to be processed. The list of all previously enqueued optimization jobs for this particular problem is shown at the bottom of the page. The optimization log for a particular job can be shown by clicking on the job's status. If a job has generated a solution, it will be possible to visualize it by clicking the "View Solution" button at the right of its line.

Figure 7 –Problem optimization page

The third page is used to visualize the problem's solutions (Figure 8). The controls in the red rectangle allow to respectively select a solution, rename a solution, delete a solution and visualize a solution on the map. Just below, the first three tabs allow to visualize the solution from different points of views. The last tab allows to download the solution data using every format supported.

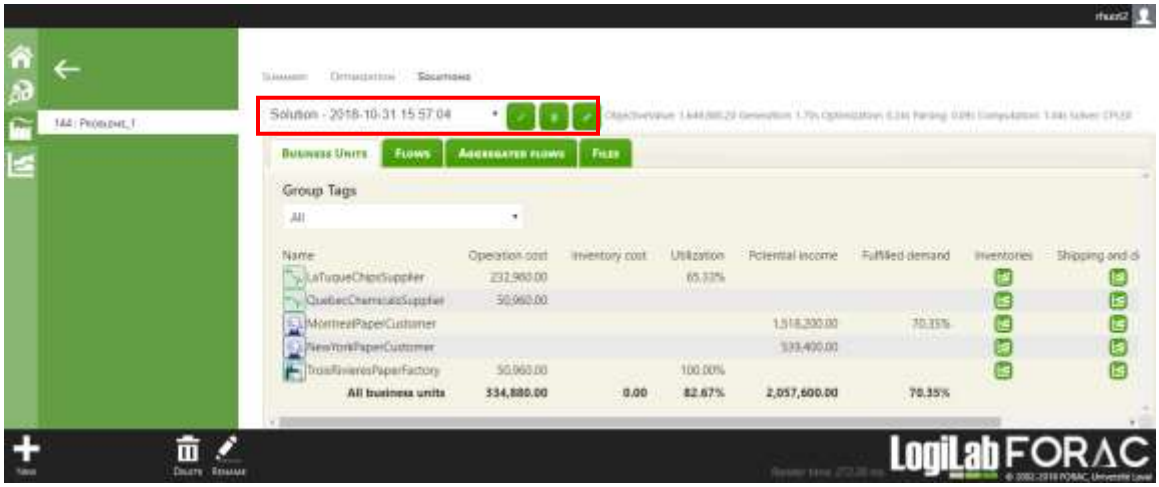

Figure 8 – Solution visualisation page

2.1.2 The mapping module

From the mapping module, solutions can be visualized on a map. When first opened, an empty map is shown (Figure 9). The “Dynamic” button at the lower left of the screen, in the contextual bar, allow to select a solution to visualize and to set the preferred display options. The display options include : showing or hiding business unit icons, business unit names, capacities usage. Business unit clustering based on zoom level can also be activated or deactivated.

On the map, product flows are illustrated as colored arrows. The width of the arrows is proportional to the quantity of product moving in the flow. Each product type will have a different arrow color.

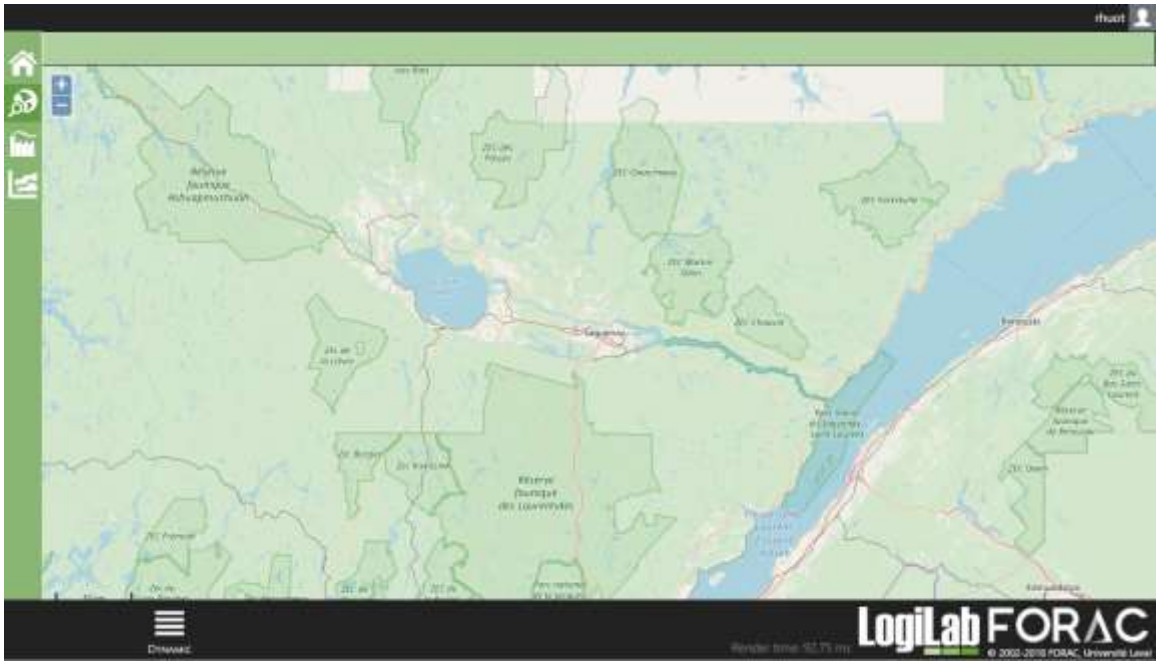

Figure 9 – Mapping module

The selected solution will be shown on the map and a “Toolbar” button will appear at the lower left of the screen, in the contextual bar (Figure 10). This button shows a toolbar used to change the display of the current solution. With this toolbar, the solution can be shown as a conceptual graph. It can also be used to show only one period of the solution.

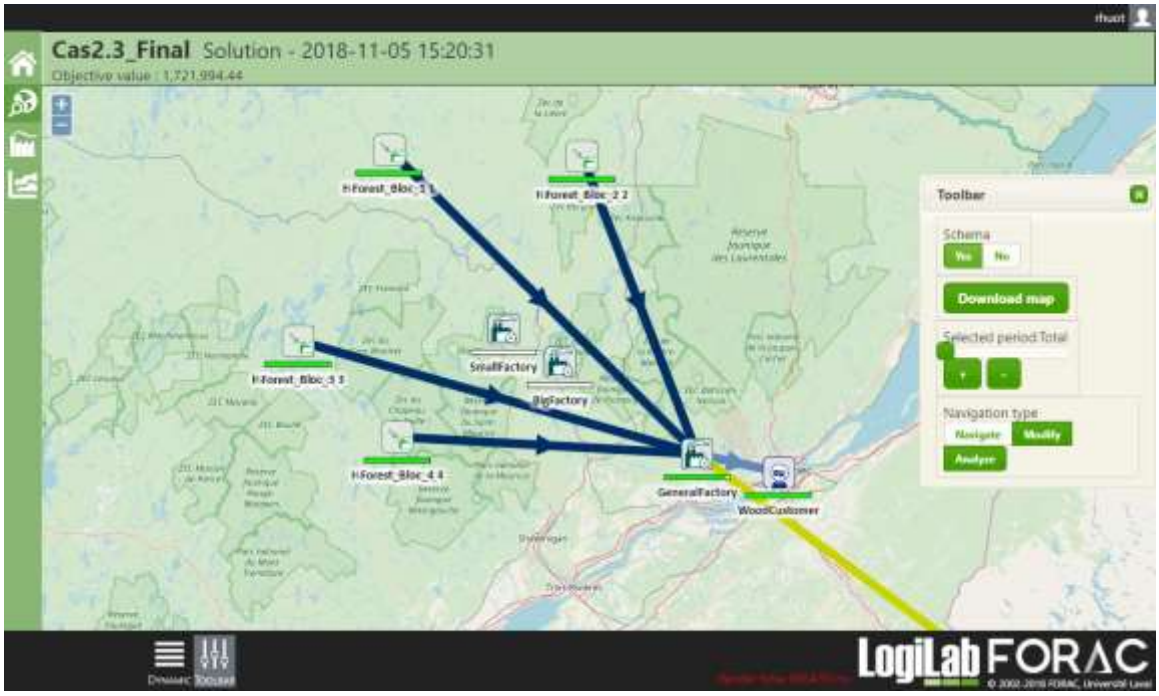

Figure 10 – Displaying a solution

2.1.3 The analysis module

The analysis module allows to visualize graphics showing different aspects of a solution (Figure 11). The “Data source » button, in the lower left of the screen, in the contextual bar, allow to select a solution to visualize. The green navigation bar at the left of the screen allows to select an aspect of the solution to visualize. The controls in the red frame allows to respectively: select a business unit, show data as a table, export the current graph as an image.

Figure 11 – Analysis module

3 Design a logistic network model for LogiLab

There are three supported way of creating a logistic network model for LogiLab. The first is to fill the preconfigured Access form. The second is to fill an Excel template, then to use the "AccesFileGenerator" to generate a valid Access input file. The third way was to use a legacy Excel template and is not supported anymore because it doesn't support every LogiLab features. All necessary forms, templates and examples are available for downloading in the subsection "Forms and Examples" on LogiLab home page. Also, every model designed can be customized dynamically with advanced features using a Python script file.

3.1 Access form

When opening a new Access form (EmptyInputFile.accdb), some information will first have to be recorded in a form (Figure 12). Mandatory fields are: problem name, number of periods, period length (in days).

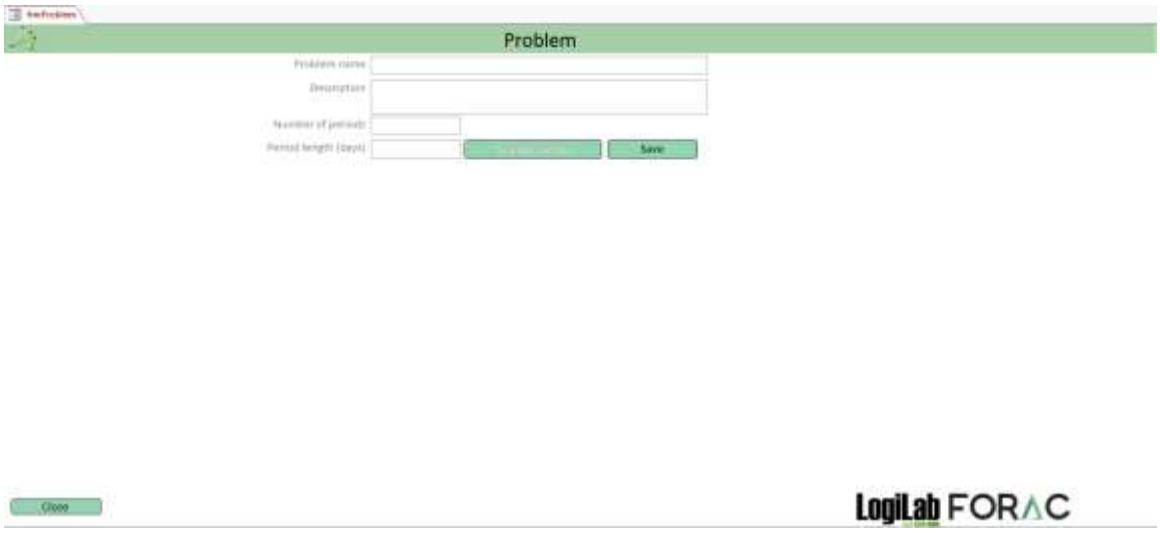

Figure 12 – New Access form

Once the mandatory fields filled and the "save" button pushed, the form main screen will be shown (Figure 13). Use the three buttons on the left side to navigate between the three main sections of the form: business units, flows and products.

Figure 13 – Access form main screen

3.1.1 Business units

The business unit form allows to manage the business units of the model. The button "Add Business Unit" open a form to create a new business unit. Double click on an existing business unit to edit it.

In the form to add a business unit (Figure 14), only the business unit name and the business unit type are mandatory. After pressing the "Confirm" button, new elements will be shown in the form. At the right of the screen, the "Group Tags" section will be shown and will allow to add a tag to the business unit to group it with other business units. At the bottom of the screen, the inventory form allow to manage the business unit storage capacities.

If the type is "Supplier" (Figure 15), the "See supplier's supply" button will appear to manage the supplier supplies. If the type is "Factory", the "See factory's processes" button will appear to manage the factory processes. If the type is "Customer", the "See customer's demand" button will appear to manage the customer demand.

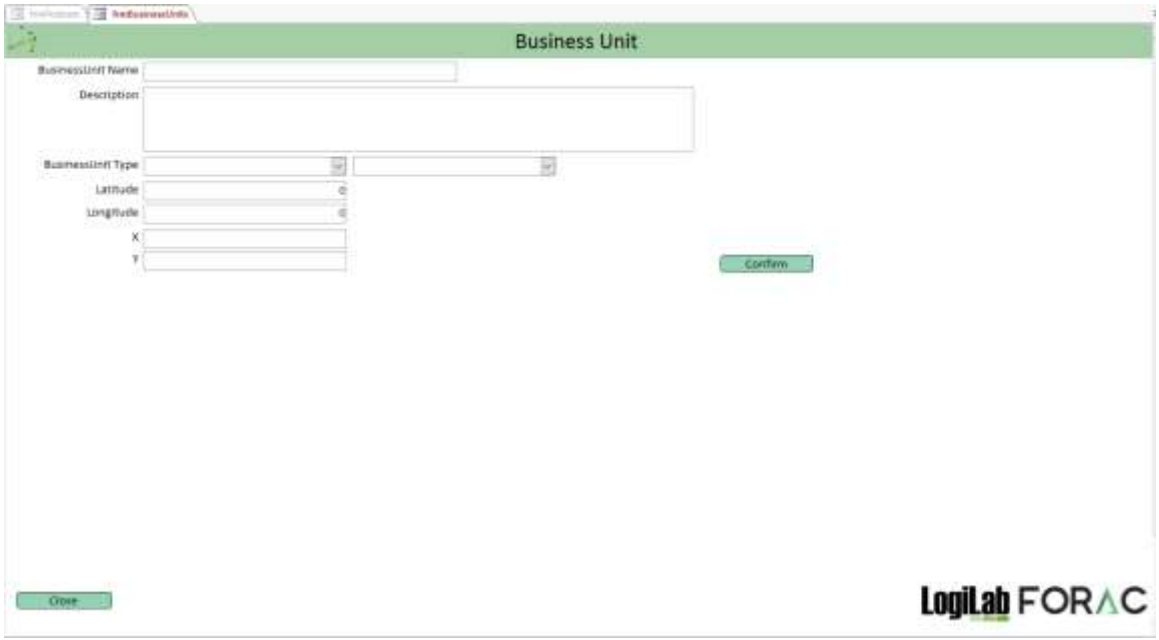

Figure 14 – Form to add a business unit

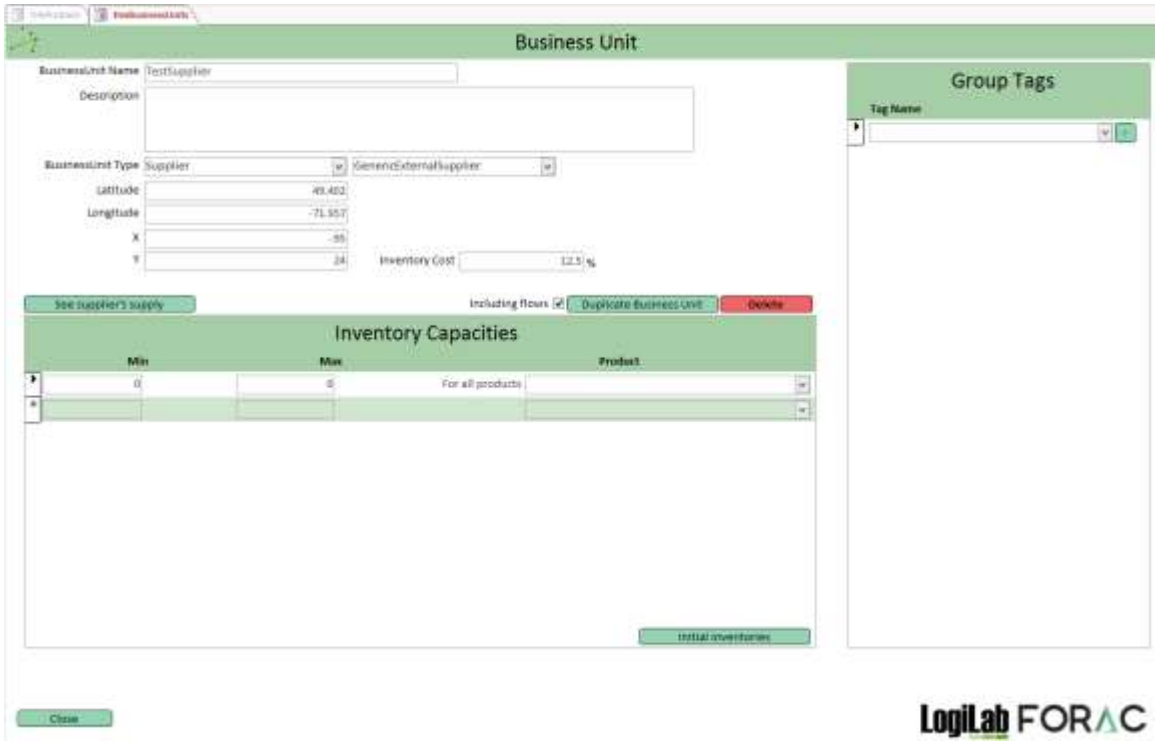

Figure 15 – Supplier form

3.1.1.1 Suppliers / Customers

While these two types of business units behave very differently in a logistic network, both there forms are very similar. Clicking the “See supplier’s supply” button will open the product supplied form. Clicking the “See customer’s demand” button will open the product demand form. In both of these forms, lines can be deleted by selecting them and pressing “Delete” on your keyboard. Also, some fields can be edited directly in the form. For the others, you must double-click on a line to open a more detailed form.

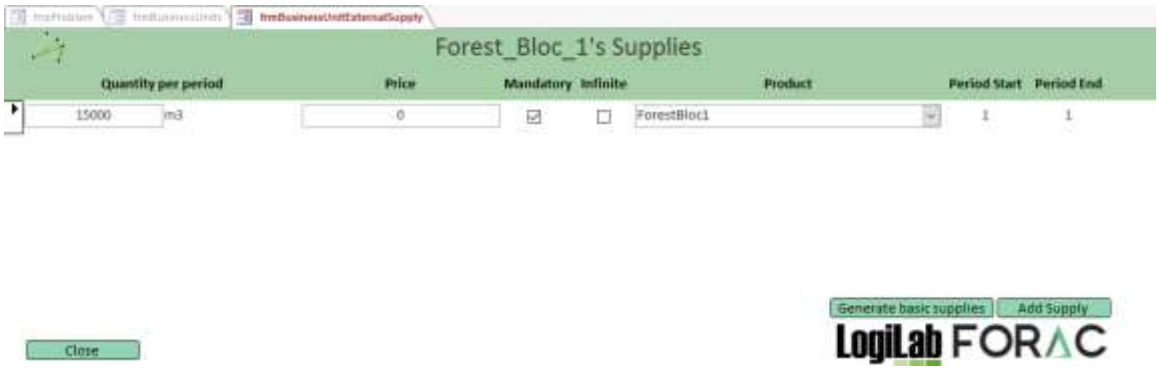

Figure 16 – Product supplied form

There are two way to add products to these two forms, one at the time with the “Add Supply / Add Demand” buttons or for many periods with “Generate basic supplies / Generate basic demands” buttons.

Pressing the “Add Supply / Add Demand” buttons will open the “supplier product form (Figure 17) / customer product form”. These forms mandatory fields are “Product”, “Quantity per period” and “Price” (per period). The option “Mandatory” allows to force the product creation (for supplier) or consummation (for customer). Be warned than using the “Mandatory” option can cause infeasible models if this constraint clashes with another one. The option “Infinite” allows the product to be created (for supplier) or consumed (for customer) without maximum boundaries, ignoring the “Quantity per period” field. Both options cannot be selected at the same time because it would be impossible to force an infinity of products to be created or consumed. Finally, the form allows to select the period interval where this supply or demand is valid.

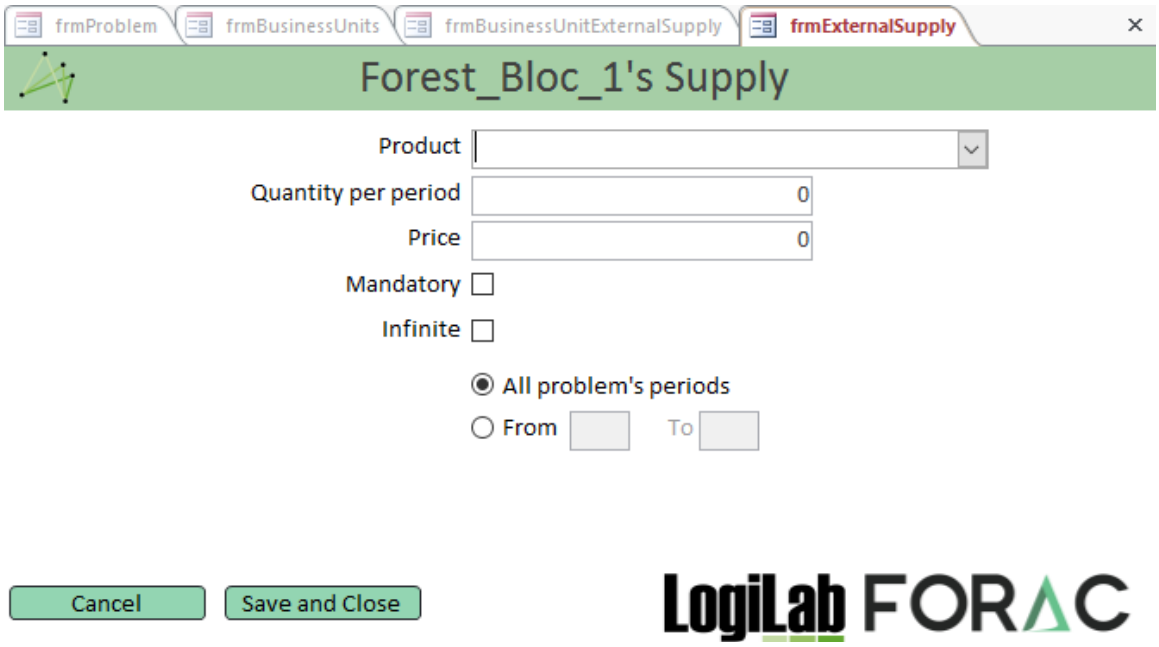

Figure 17 – Supplier product form

Pressing the "Generate basic supplies / Generate basic demands" buttons will open the "generate supplier products form (Figure 18) / generate customer products form". These forms facilitates the creation of many elements for many periods in a given interval. Instead of creating only on supply at the time, this form fills the interval with supplies. The field "NbPeriodPerDemands" allows to set the length in period(s) of each of the created supply. For an example, if a user sets 2 as "NbPeriodPerDemands" for a 6 period interval, 3 supply elements will be created, respectively for the periods 1-2, 3-4 and 5-6. Once created, these new elements can be edited the same way than the other ones.

Figure 18 – Supplier : generate many products form

3.1.1.2 Factories

For factories, the displayed button is “See factory’s processes”. This button opens the processes form (Figure 19). In the upper part of the form, the global production capacity of the factory can be edited and saved with the buttons “Change” and “Save” respectively. De default global production capacity is from 0 to the length of a period in hours. New processes can be added with the “Add Process” button. Existing processes can be edited by double clicking them.

Figure 19 – Factory process list

Basic informations like name, process usage, cost must be provided to create a new process (Figure 20). The process usage represents the portion of the factory global capacity that is spent each time the process is fully executed. The discrete checkbox can be used if the process must be discrete instead of continuous. To enable discrete processes, the option “IsMip” must be used when the optimisation is started.

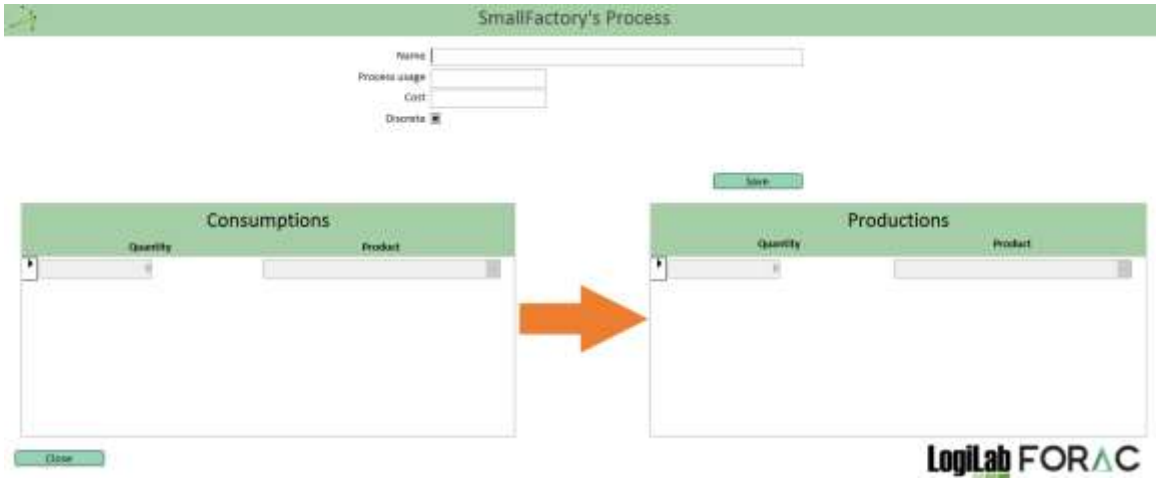

Figure 20 – New process form

Once the “Save” button is pushed, the sub-forms “Consumptions”, “Productions” and “Group Tags” will be displayed (Figure 21). “Group Tags” are going to be explained at section 3.1.4. A process models a transformation in which the quantities of products defined in the “Consumptions” form are consumed to produce the quantities of products defined in the “Productions” form. For the example shown in Figure 21, this process uses 1 m^3 of the factory global capacity of the factory (7000 as shown in figure 19). When completed, this process will use 1 m^3 of BigLogsPile and 1 SortTicket to produce 267 pmp of RegularQualityALumber, 0.0328 tmv of sawdust and 0.1107 tmv of wood chips.

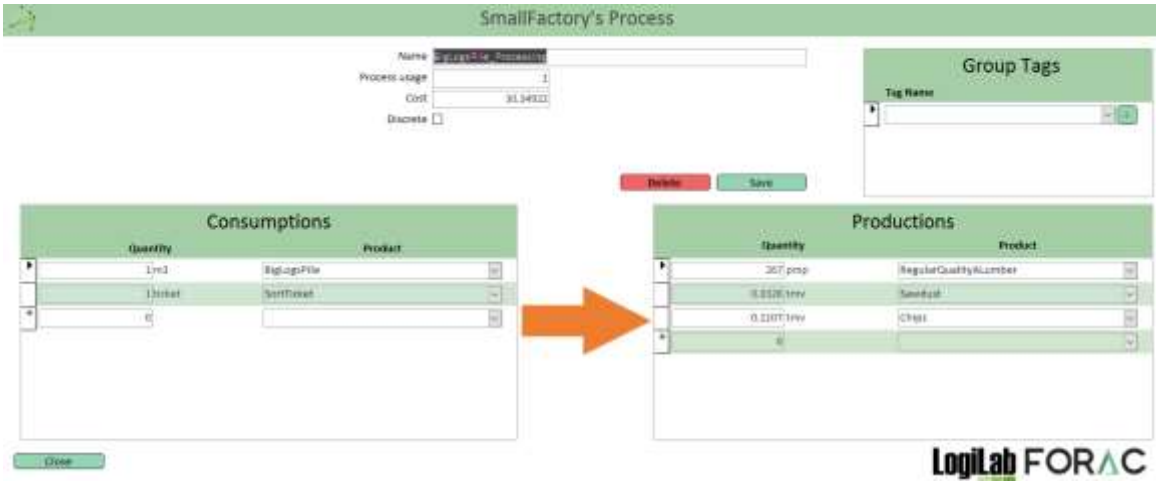

Figure 21 – Process form

3.1.2 Flows

The flow form allows to visualize, add and modify the flows of the logistic network (Figure 22). Flow can be edited by double-clicking them. The “Delete all flows” allows to quickly remove all flows of the network. The “Generate basic flows” button is used to automatically generate all possible flows based on the current network configuration.

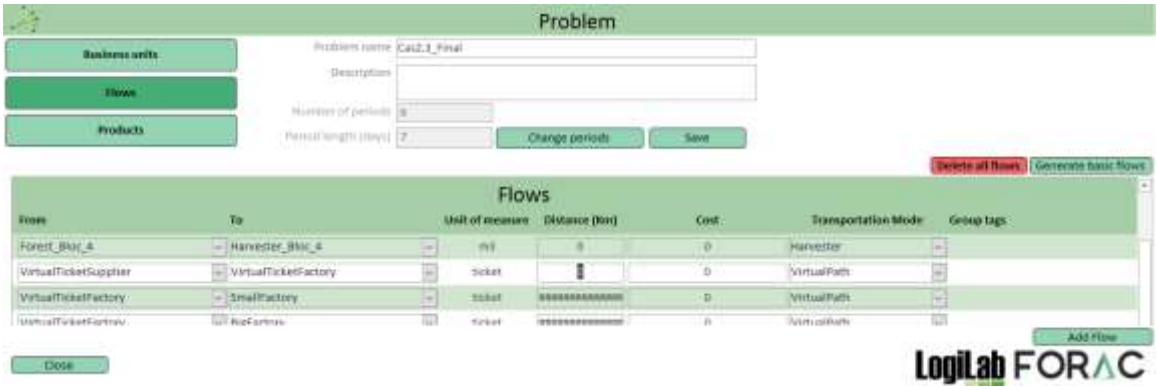

Figure 22 – Flow form

The “Add Flow” button displays the new flow form (Figure 23). Basic flows properties, like its origin and destination business unit, its transportation mode, its cost, its unit of measure, its distance in Km and its delay in hours, must be defined. If the wanted transport mode is not in the options, it can be added with the “+” button. The flow cost is define per unit of moved product per kilometer. For example, if a flux is transporting 10 m^3 of wood on 300 km à a cost of 0.05$, le total transport cost will be: 0.05*10*300 = 150$. A flow can only have one unit of measure defined. If another unit of measure is needed, another flow must be defined. If distance is not defined, LogiLab will automatically estimate the bird’s eye distance between the origin and the destination. If the delay is bigger than the percentage of the period defined in the optimization parameter “FlowDiscretizationDelayPercentage”, the product moving through the flow will arrive at the next period.

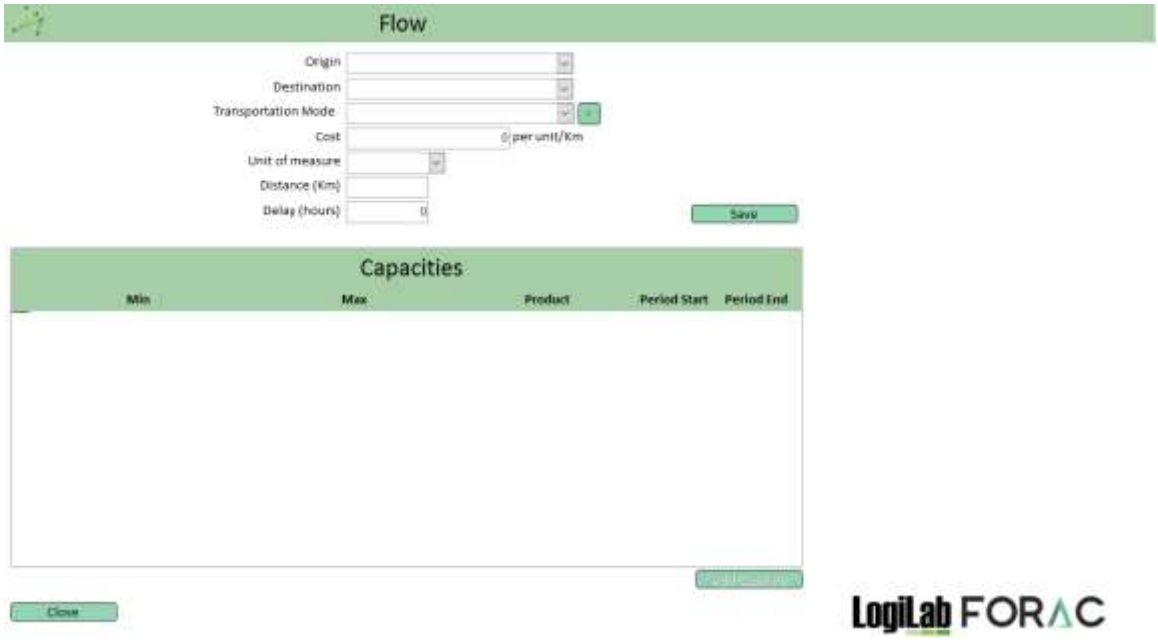

Figure 23 – New flow form

Once the “Save” button is pushed, the sub-forms “Capacities” and “Group Tags” will be displayed (Figure 23). “Group Tags” are going to be explained at section 3.1.4. The capacity sub-form allows to define product types and quantities that can be moved by this flow for given periods. New capacities can be added with the “Add Capacity” button. When selected, existing capacities can be deleted with the keyboard button “Delete”.

3.1.3 Products

The products form allows to visualize, add or modify the products used in the modeled logistic network (Figure 24). Products can be added and modified directly in the list.

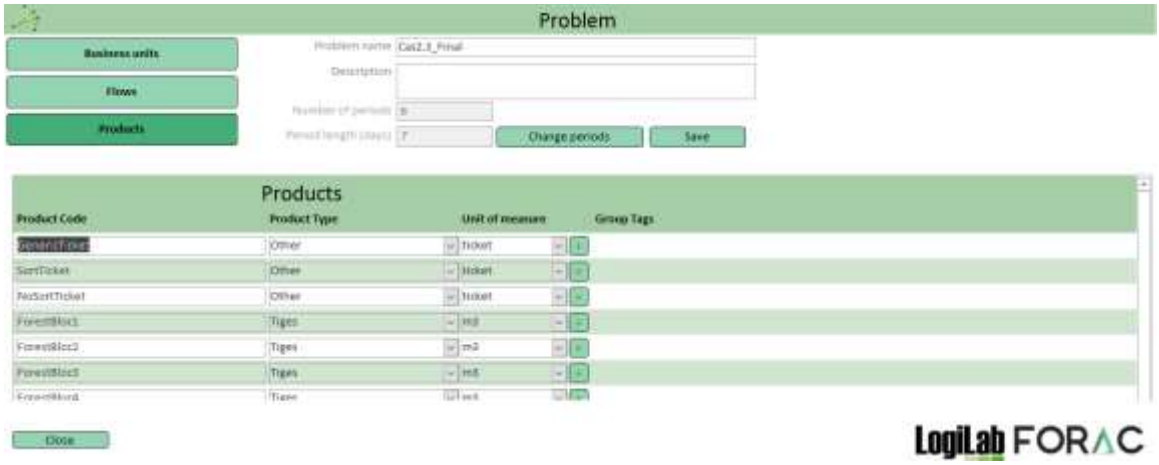

Figure 24 – Products form

The product edition form (Figure 25) can be opened by double-clicking on an existing product. This forms allows to visualize every business units that produce or consume a given product. This information is very useful when troubleshooting a model. It also contain a “Group Tags” sub-from. “Group Tags” are going to be explained at section 3.1.4.

Figure 25 – Product edition form

3.1.4 Group tags

Group tags allows to aggregate specific data with a given tag. This is very useful to customize the way the solution will be visualized. They can be defined for business units, processes, flows and products. For example, if a user wants to compare inventory costs for every business units located in an arbitrary region, he only has to apply a “RegionX” group tag on every business units to consider. When visualizing the solution, the “RegionX” group tag will be available to aggregate, filter and display the data associated with this specific region.

3.2 AccessFileGenerator: Generating an Ms Access file from an Ms Excel file.

While the Access Form is the best way to browse, visualize and modify a LogiLab case, it is not a particularly efficient way to import large quantities of data.

The “AccessFileGeneror” is specifically designed to ease the importation of massive data into LogiLab. First, data must be copy pasted into the Ms Excel importation template specific structure. Then, the application will automatically validate the data and import it into a functional Acces Form file, ready to be imported into LogiLab.

The “AccessFileGenerator.zip” can be downloaded from the LogiLab homepage, in the « Formulaires et exemples » or « Forms and Examples » sections.

Figure 26 – Téléchargement du fichier AccessFileGenerator.zip

This compressed archive contains the last version of the AccessFileGenerator “Forac.LogiLab.AccessFileGenerator.exe“ and its specific user guide: “ GuideUsager_AccessFileGenerator.pdf“.

3.3 Legacy Excel form

The legacy excel form was the first way used to define LogiLab problems. It was oversimplified and did not allow to harness the full power of LogiLab. Its use is not recommended anymore.

The “CasSimple.xlsx” example excel form can be downloaded from the LogiLab homepage, in the « Formulaires et exemples » or « Forms and Examples » sections. However, it won’t be covered in this document. The legacy French documentation is still available in the document “Guide_Utilisateur.pdf” available on the LogiLab homepage, in the « Documentation » section.

3.4 Model modification with a Python script

Once the LogiLab case is transformed into a mathematical problem, it can be customized by adding specific variables and constraints to it before it gets solved. To allow this quick customization, LogiLab supports Python scripts files to be imported to problems. This section describes how to add, use and remove a Python script file to LogiLab.

As shown at Figure 27, the Python script file can be added at the “PythonScriptFile” field of the “Optimization” page. The combo box allows to select between scripts currently associated with this problem, which one to use for the current solver execution. The “+” button allows to select a new Python script file to associate with this problem. The “-“ button allows to remove the association with the currently selected Python script file and this problem.

Figure 27 – Using a Python script file

To validate that the Python script working as expected, the optimization log must be validated. As shown at Figure 28, the line “Adding custom variables and constraints from file …” is shown when the script is launched and the line “Custom Computed Indicators generation competed” is shown after the script is ran.

Figure 28 – Python script file *Optimization log*

3.5 Using interactive optimization

Section to be completed…

Annex-A : Profit maximization model (in French)

T	l'ensemble des périodes de temps
U	l'ensemble des unités d'affaires
K	l'ensemble des types de capacité (capacité machine, limite des stocks)
W	l'ensemble des procédés (machines, inventaires)
$W_{tu} \subset W$	l'ensemble des procédés pouvant être effectué à l'unité u à la période t
P	l'ensemble des produits
E	l'ensemble des liens existant entre les unités
$\delta_u^+ \subset E$	l'ensemble des liens arrivant à u
$\delta_u^- \subset E$	l'ensemble des liens partant de u

Table 12 – Ensembles

q_{ktu}	capacité de type $k \in K$ de l'unité u au temps t disponible
f_{etp}^l	flux minimal du produit p passant sur l'arc e à la période t
f_{etp}^u	flux maximal du produit p passant sur l'arc e à la période t
f_{et}^l	flux minimal de tous les produits passant sur l'arc e à la période t
f_{et}^u	flux maximal de tous les produits passant sur l'arc e à la période t
c_w	coût du procédé w
c_{etp}^f	coût de transport du produit p sur l'arc e si le transport commence à la période t
l_{etp}	délai de transport du produit p sur l'arc e si le transport commence à la période t
s_w	délai de production du produit p avec le procédé w
α_{pw}	quantité de produit p requis par le procédé w
γ_{pw}	quantité de produit p produit par le procédé w
λ_{kuw}	quantité d'unité de capacité de type $k \in K$ de l'usine u consommer par le procédé w
d_{tup}	demande de produit p à l'unité u à la période t
ρ_{tup}	valeur de vente du produit p à l'unité u à la période t

Table 13 – Constantes

Y_{tuw}	La quantité du procédé w effectué à l'unité u et se terminant à la période t
D_{tup}	La quantité de produit p vendue à l'unité u à la période t
F_{ept}	Le flux de produit p sur l'arc e partant de i à la période t et allant à j

Table 14 – Variables

$$\max \sum_{t \in T} \left(\sum_{u \in U} \left(\sum_{p \in P | d_{tup} > 0} \rho_{tup} D_{tup} - \sum_{w \in W_{tu}} c_w Y_{tuw} \right) - \sum_{e \in E} \left(\sum_{p \in P} c^f_{etp} F_{etp} \right) \right)$$

Sujet à :

$$\sum_{t_1 \in T} \left(\sum_{w \in W_{t_1 u} | t_1 + s_w = t} \gamma_{pw} Y_{t_1 uw} \right) + \sum_{e \in \delta^+_u} \left(\sum_{t_2 \in T | t_2 + l_{et_2 p} = t} F_{et_2 p} \right) - \sum_{w \in W_{tu}} \alpha_{pw} Y_{tuw} - \sum_{e \in \delta^-_u} F_{etp} - D_{tup} = 0 \qquad \forall t \in T, u \in U, p \in P \quad (1)$$

$$D_{tup} \leq d_{tup} \qquad \forall t \in T, u \in U, p \in P \quad (2)$$

$$\sum_{w \in W_{tu}} \lambda_{kuw} Y_{tuw} \leq q_{ktu} \qquad t \in T, u \in U, k \in K \quad (3)$$

$$f^l_{et} \leq \sum_{p \in P} F_{etp} \leq f^u_{et} \qquad t \in T, e \in E \quad (4)$$

$$f^l_{etp} \leq F_{etp} \leq f^u_{etp} \qquad e \in E, t \in T, p \in P \quad (5)$$

$$Y, D \geq 0 \quad (6)$$