

Open Specification for Equipment Coupling Modeling Notation (ECMN)

developed by iMinds

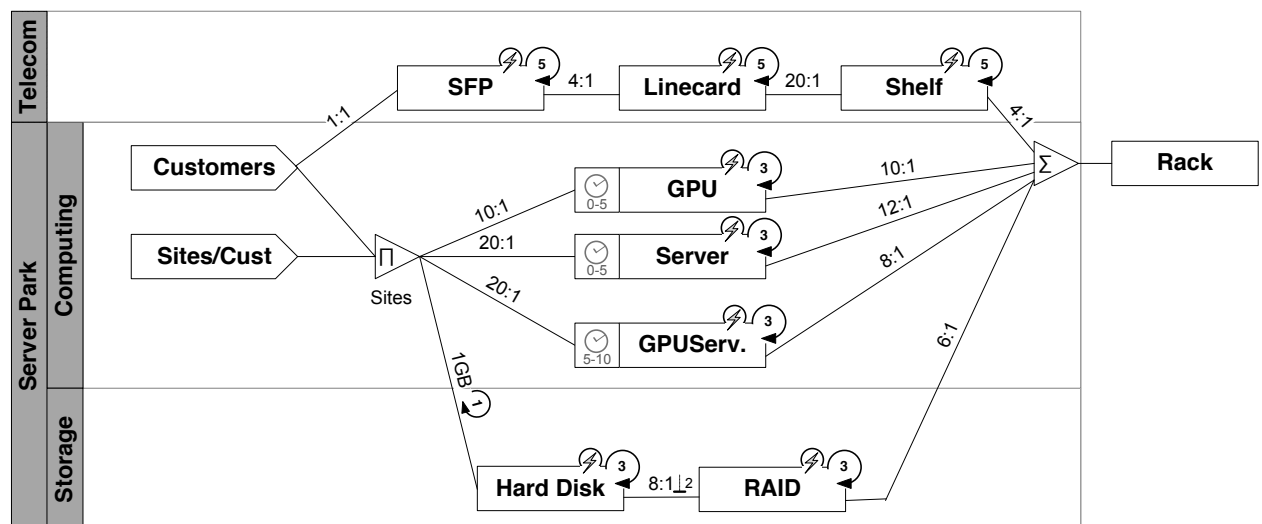
part of FI-ware open specification of the BusinessCalculator
<https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Apps.BusinessCalculator>

When tackling the problem of how to dimension a central office, a street cabinet, a server room, etc., one is repetitively having the same discussions and making very comparable calculations. For instance a street cabinet of a telecom operator contains one small rack and one or two power blocks in there. It contains at max 3 (sometimes 4) shelves and each of those shelves can be equipped with a predefined amount of line cards. Finally this amount of line cards will be directly correlated to the amount of customers to serve via this street cabinet. Calculations of the dimensioning are as such clearly hierarchical in nature. For instance the amount of shelves will depend on the amount of line cards which is in turn dependent on the amount of customers to connect.

Definitely the modeling and calculation are two separate steps in the process of dimensioning the equipment. The first subsection will detail the visual equipment coupling modeling notation. In the section following, this visual language is coupled to an XML format for which an XSD description will be added to make this more rigid and standardized. The third subsection is detailing how the calculation of the dimensioning (amount of needed equipment for each type) can be performed based on this visual ECMN. Finally, the last subsection will add examples to make things more concrete and at the same time provide cases that can be used both in testing as in the construction of templates.

VISUAL NOTATION

As mentioned, the equipment model consists of a hierarchical structure documenting how equipment is linked to each other and what the constraints on later calculations will be. The hierarchical model will be represented in a graph structure for which an example, on which we will extend, is shown below:



The basic building blocks of the equipment model are:

Driver

A driver with the name indicated in the arrow. The direction of the arrow is inside the equipment model and can be placed at any side of the model. Typically the drivers are placed at the left hand side or at the bottom (at the 'leaves' of the tree).

Additional information is to be coupled to this driver and some modeling specific information, e.g. recurrence, can be added directly attached to the driver block in the model

The following information can be coupled to the driver:

Name	Unique (model) String value
ID	String – for identification in a broader context

The driver is an input block and as such no other block can connect before this block. The driver can connect to multiple other equipment, aggregator and/or separator blocks.

Int. Driver

An intermediate driver is the same as a driver with the difference that it is used for making the link between two points in the model. It can connect at the one side to one incoming child and on the other side it can connect to multiple equipment, aggregator and/or separator blocks.

It is to be used in the following two cases:

1. Clarity of the model
In this case the intermediate driver will make the link from one point in the model to another point to allow having less connectors crossing other connectors or blocks.
2. Sub model
In the case of the sub model it is used in the expanded view to indicate clearly the end of the sub model and where it connects to the higher level model.

The following information can be coupled to the intermediate driver:

Name	String value which can be used twice in the same model
ID	String – for identification in a broader context

In the editor, clicking on the one intermediate driver should also indicate the other intermediate driver (when existing) it is linking to.

Equipment

Identifies a type of equipment based on its name. Additional information is to be coupled to this equipment and some modeling specific characteristics, e.g. reinstallation period, power usage, etc., can be added directly attached to the equipment block in the model.

The following information can be coupled to the equipment:

Name	Unique (model) String value
ID	For identification
Cost	Time function ➔ Should be extensible in the future
Capacity	Double
Size	Double + possibility to give its unit (e.g. gigabyte)
Maintenance	Fixed percentage or Time function
Energy Consumption	Double (possibly later Time function)
Installation	Fixed percentage Or fixed cost per unit

	Or Time function
Floor space	Double
Reinstallation time	Integer value
Time dependent character	Free to choose

Equipment can be coupled directly after one other equipment, one driver, one aggregator or one separator by means of a connector (shown below). An equipment block itself can couple to multiple other equipment, aggregator and/or separators.

Submodel 

Equipment models are inherently hierarchical and can be stacked in a hierarchical manner with more intuitive visualization and modeling by means of a sub model.

The editor should intuitively allow seeing the expanded view of this sub model (e.g. double click). The content of the sub model is similar to a regular ECMN using all blocks mentioned here. To start with we fix the constraints so that a sub model should connect to one input (child) indicated in the expanded view as a driver and one output (parent) indicated by an intermediate driver in the expanded view. In the future we might relax this constraint and open up the possibility to have multiple input drivers and multiple intermediate drivers. At that point we will have to make the link between the higher level model and its sub models clear (e.g. by ID or name).



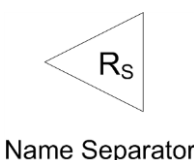
An aggregator will combine constraints into one new constraint to link to blocks further in the calculations (called down flow parent), thereby linking multiple children to one parent. Additionally aggregators can define how the combination of constraints should be performed by means of a flexible rule set (R_A) indicated directly into the symbol. By default the summing aggregator is assumed and the aggregator can be removed from the figure in this case.

The following information can be coupled to the aggregator:

Name	String value
ID	String – for identification in a broader context

An aggregator can be linked from multiple drivers, equipment, aggregators and/or separators (called children) up to several other equipment, aggregators and/or separators.

The example below should be read as the amount of servers is depending on the total number of sites which is the multiplication of the number of customers and average sites/customer.



A separator will separate constraints into multiple - not necessarily different - constraints to link to equipment further in the calculations, thereby linking one child to multiple parents. Additionally separators can define how the split of constraints should be performed by means of a flexible rule set (R_S) indicated directly into the symbol. By default the parallel separator is assumed and the separator can be removed from the figure in this case. If wished or required, a separator can have an identification or name.

The separator can be linked from one driver, equipment, aggregator or separator up to several other equipment, aggregators and/or separator.

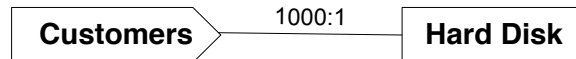
The following information can be coupled to the separator:

Name	String value
ID	String – for identification in a broader context

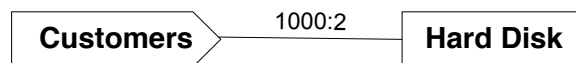
N : M

The connector is represented by a simple line and connects drivers, equipment, aggregators or separators to each other. The granularity factor links the different levels of equipment types, by indicating the maximum number of equipment of the lower type (children - N) that is needed before a new equipment of the higher type (down flow parent - M) is to be installed. A connector without any granularity indicated refers to a 1:1 connector.

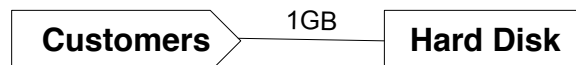
The following example should read as follows: For every 1000 customers a new hard disk should be installed.



It is not required that the parent side of the connector be equal to one or be smaller than the child side. The following example requires hard disks to be installed per 2 for every 1000 customers, for instance for the sake of redundancy.

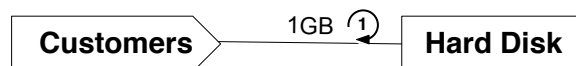


When a size is set for the parent of this connector link, a simple granularity relation to this size can be used as well. For instance the following example indicates that 1GB per user is reserved on the disk. When the size of the disk is set to 1000GB or 1TB, this means that a new hard disk has to be installed per 1000 new customers.



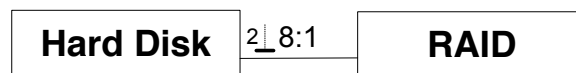
N : M

In case equipment needs to be installed on a yearly base (or with a predefined recurrence) in correspondence to its input (e.g. driver), this is indicated with the repetitive symbol on top of the connector. An example can include the yearly increase of the available disk space for each user. Extending the previous example, the example shown below reads as every year an extra 1GB hard disk space must be installed per existing customer. As mentioned before the 1 in the recurrence symbol can be omitted.

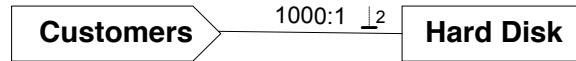


N : M

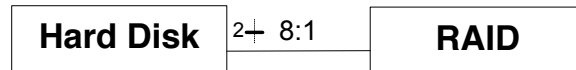
Certain types of equipment can only be installed in batches. The basis for the batch can be either the parent or the child. In case the child side has been selected, this refers to a minimal installation batch per installation of the parent (taking into account the granularity). This has been used in the example below and means: A RAID controller can contain 8 hard disks and for every new RAID controller at least 2 new hard disks should be installed.



In case the parent side has been selected, this refers to a minimal installation batch first time – so only for the first installation of the parent. This has been used in the example below and means: for every 1000 customers, a new hard disk should be installed, but the first time installation requires two hard disks to be installed.



A batch can also be the default way of installation, or a requirement for at every new installation. This is made clear by means of an averaging indicator. The example below uses this and this means: every addition of hard disks to the RAID should always happen in batches of 2.



As mentioned before, the links between drivers and equipment and from equipment to other equipment higher in the hierarchy can be tuned with calculation rules. The following calculation rules are defined in the equipment modeling language:



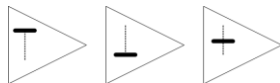
Summation *aggregator*
Calculates the amount of all down flow parent types based on the sum of the amount of all children.



Multiplication *aggregator*
Calculates the amount of all down flow parent types based on the multiplication of the amount of all children.



Difference *aggregator*
Calculates the amount of all down flow parent types based on the difference between two children (limited to two children).



Max, min and average aggregator
Calculates the amount of parent type equipment based on the highest, lowest or average amount of children type equipment

As mentioned before, additional annotations can be added to both the drivers and equipment to indicate additional aspects to take into account in discussions and in the calculations. The following annotations are defined in the equipment modeling language:



Recurrent *installation*
Any equipment annotated with this symbol will have to be reinstalled with a reinstallation period as indicated (t) in the loop.



Time *Dependent* *Equipment*
This equipment has to be included for a certain period in time. Applying this to different types of equipment allows substituting equipment for newer equipment at a later point in time.



Recurring maintenance
The maintenance of this type of equipment entails an extra cost, which is recurring periodically.



Maintenance upfront
The maintenance of this type of equipment entails an extra cost, which needs to be paid to the supplier upfront (fixed maintenance contract).



Floor space
The equipment consumes a reasonable amount of floor space inside the central office, cabinet, server room. A leasing cost for this floor space should be taken into account.



Active equipment type

This symbol indicates that the equipment under study contains active components, and will therefore consume electricity.

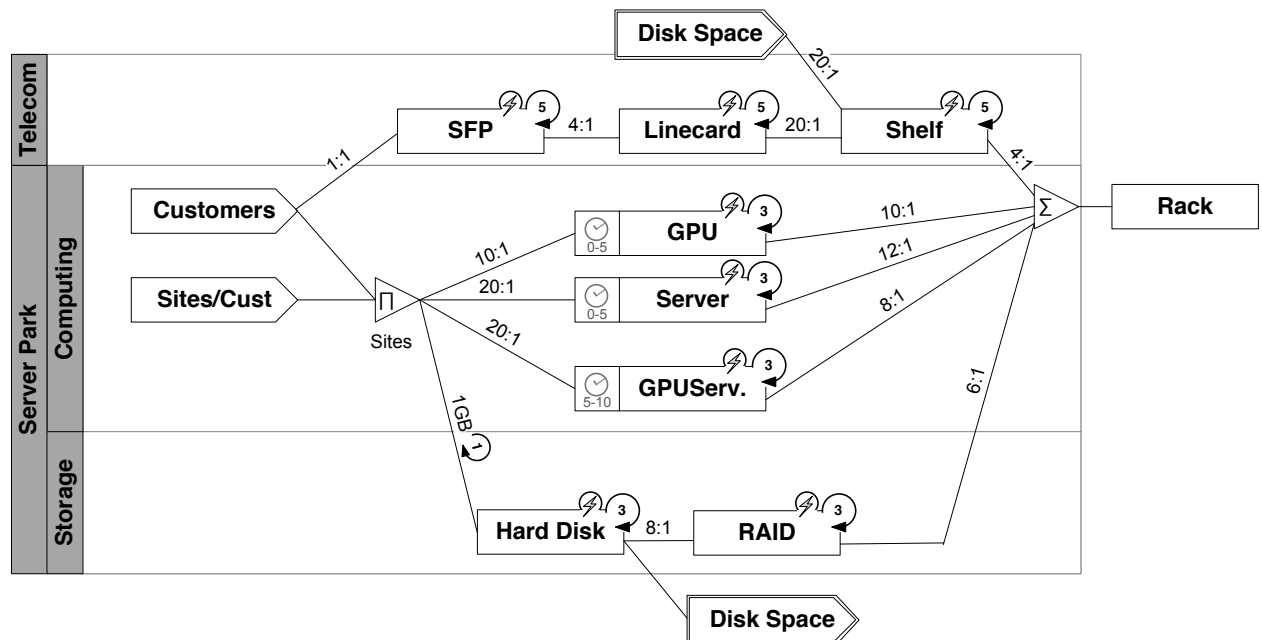
Finally the different building blocks of an ECMN can be grouped in swim lanes and pools as an indication of the responsibilities for the different equipment.

Swim lanes are indicated by large bars indicating the grouping in which the different equipment blocks are assigned. This can be based on different types of groupings, such as which business unit is going to pay or maintain them, etc. More in general all blocks in the model should be assignable to different groups and it should be possible to make this grouping visible by means of swim lanes or coloring.

EXAMPLE: USAGE OF AN INTERMEDIATE DRIVER:

An intermediate driver is an element added to the model in order to improve the models readability. The following figure gives an example of this usage. Connecting the Hard Disk to usage on the Shelf would require a major graphical restructuring of the model in order to keep its readability to the maximum. By means of the intermediate driver called Disk Space, the amount of Hard Disks can be linked to the Shelf as if connected directly with an aggregation factor of 20:1. As such an intermediate driver can be used to link two different locations in the model to each other as if connected by one connector.

Intermediate drivers are connected to each other by means of their name. In theory there can be more than two intermediate drivers with the same ID all connecting to the same point. We prefer not to encourage this usage as we believe this can be confusing for people reading complex models.



An intermediate driver is also used in case of a sub-model to indicate the equipment type for which the amount will (or can) be used further in the calculations. For example when using the following (fictive) Sub-model somewhere in the calculations, two inputs and two outputs are defined. The sub model is later used collapsed as shown in the figure at the bottom. As mentioned in the original description, we will start adding sub-models with only one input and one output.

