

How to represent the logical layers from the different categories with Hashtag #Notation with the CPW Process method and with extended possibilities?

Bernd J. Schneider

IC Informatica Consulting GmbH

Zurich, Switzerland

Email: Bernd.Schneider@ICInformaticaConsulting.com

URL: <http://www.cpw-method.com>

04 June 2016

In this chapter is described how the simplified logical layer of the CPW Process method with the Hashtag #Notation can be extended with more possibilities, to cover and to describe either other categories, areas or topics or then to cover the logical operations, which can be occur in the analysis, where then in some cases the complexity of the logical operations can be much more higher.

As said before, a catalog or a list of logical operations, which can be assigned to a category with areas and with different topics, is a logical layer.

In the previous chapter has been used the logical layer only for the selection. This simplified logical layer for the selection is as follows:

#Question? Yes/No

Here is a question, which can be answered with Yes or No. It could be also represented like this:

#Question? #Yes/No

In this case the answer of the question is separated through the hashtag #. You can characterize the answering possibilities with Yes or No also as a possible value set of the question.

Furthermore has been used the question also as a loop. This means, that the loop is repeated as long as the question is answered with Yes.

Therefore you could also represent the logical layer like this. Either as a selection or as an iteration:

#Selection #Question? #Yes/No

or

#Iteration #Question? #Yes/No

Or the loop or iteration could be also represented like this:

#Iteration #Repeat #Question? #Yes/No

#Iteration #Repetition #Question? #Yes/No

If there is the case, that a loop or iteration is represented with from/to with the respective number of the loop repetitions, then the logical layer can be represented like this:

#Iteration #Repetition #From = X To = Y

The increment of the loop or iteration would be equal 1 in this case.

As said a selection can be assigned a logical operation. Then the logical layer can also be represented in such a way:

#Selection #Logical operation #XOR #A XOR B

In this case the logical operation is a Boolean operation. The Boolean operation is an Exclusive Or and is represented in the Boolean Algebra as XOR. The value set can be A XOR B. In such a way can be represented the whole spectrum of the Boolean Algebra.

The logical operations can be also used for a #Sequence and is then represented like this:

#Sequence #Logical operation #XOR #A XOR B

But at a selection can be also a function. A question can be also seen as a function, which can be answered with Yes or No. In this case is then Yes or No the part of the value set.

But furthermore a function can also be a generic function, but also a mathematical function or formula, where can be the possible X values part of the definition set and the possible Y values part of the value set.

Then the logical layer would look like this:

#Selection #Function #Definition set $D = \{ 1,2,3,4 \}$ #Value set $V = \{ 1,2,3,4 \}$

If you want to have a specific value for the X value and Y value, then you can represent it like this:

#Selection #Function #Definition set $D = \{ 1,2,3,4 \}$ #Value set $V = \{ 1,2,3,4 \}$ #x = 1; y = 3

A function can be used not only for a #Selection of course, but also for a #Sequence. Then you can represent the logical layer like this:

#Sequence #Function #Definition set $D = \{ 1,2,3,4 \}$ #Value set $V = \{ 1,2,3,4 \}$ #x = 1; y = 3

The process step would have then with the function an Input and would generate or create then through the process of the function and of the respective process step an output.

In such a way it would be possible to represent any functions and also mathematical functions and formulas with definition sets and value sets.

So it would be possible to represent with the #Notation and with the possible #Hierarchies for a logical layer not only logical operations, but also functions and also mathematical functions and formulas. Furthermore then it would be also possible to represent other areas of mathematics (for example: other functions, integral, linear algebra, numerical mathematics, cybernetics,...) with the #Notation but also functions from the different categories, areas or topics.

In the last chapter has been represented at a Sequence with the #Notation among others also this.

With the #Notation and the possible #Hierarchies can be subdivided any things of the different categories, areas and topics.

In the most cases makes the subdivision according to hierarchies sense, but it could be based the subdivision in some cases not according to hierarchies but also according to other criteria. At the end you have a subdivision of the things either hierarchical or not hierarchical as follows:

#Thing A #Thing B #Thing C

If you have a subdivision according to hierarchies, then you can make the subdivision according to area or sub area or according to function or sub function or also according to organizational function and sub function with people.

Then the subdivision would look with the #Notation and with the #Hierarchies as follows:

#Area #Sub Area

#Function #Sub Function

#Org Unit Function #Org Unit Sub Function

The higher-level area, function and Org Unit Function could be assigned then to the header of the respective Swimlane and it creates then in the particular process steps a further subdivision or differentiation according to sub area, sub function and Org Unit Sub Function.

Is the representation of the subdivision with #Notation and with the possible #Hierarchies only to the respective categories, areas and topics, then you don't speak of a logical layer, but it is only a simple subdivision with the #Notation and with the possible #Hierarchies of the respective categories, areas and topics.

But a catalog or a list of logical operations, or of functions and but also mathematical functions and formulas, which can be assigned to a category with areas and with different topics, is then a logical layer.

Furthermore there is also the possibility for the logical operations or for the functions, but also for the mathematical functions and formulas, to define a definition set or value set or also for specific cases the concrete x values or y values.

If a logical operation and also a function doesn't exist, but there is only the case that attributes have to be forwarded from one process step to the other process step, then you speak of attribution. In some cases the attributes are forwarded only from one process step to the other process step from A to B and in other cases the attributes are changed and processed through logical operations or functions through a process step.

If there is only the case, that in a current process step the attributes are only forwarded from one process step to the other, it can be possible, that they can be used as an input for a logical operation or function in a following process step.

Therefore the attribution belongs also to the logical layer.

This means if you use the subdivision of the #Notation and the possible #Hierarchies in combination with attribution, then this belongs also to a logical layer, because it could be, that in the following process steps the attributes can be used for a logical operation or a function.

The representation with the #Notation and the possible #Hierarchies and subdivision could be also represented like this for example:

#Area #Sub Area
#Function #Sub Function
#Org Unit Function #Org Unit Sub Function

This representation with the #Notation and the possible #Hierarchies shows not only the subdivision, but also examples for characteristics and values:

Example with Area and Sub Area:

#Area #Sub Area
#Area = Europe #Sub Area = Hamburg
#Area: Europe #Sub Area: Hamburg
#Europe #Hamburg

Example with Function and Sub Functions:

#Function #Sub Functions
#Coffee Shop #Coffee #Coffee with milk
#Coffee Shop #Coffee #Coffee without milk

Example with Org Unit Function and Org Unit Sub Function:

#Org Unit Function #Org Unit Sub Function
#Coffee Shop Employee #Coffee Shop Assistant

This representation with the #Notation and the possible #Hierarchies shows not only the subdivision but also the definition set and the value set and in addition also the examples of characteristics and values:

Example with Function and Sub Functions:

#Function #Sub Functions #Definition set $D = \{ 1 \}$ #Value set $V = \{ 1,2 \}$ # $x = x$; $y = y$

The following example shows from the example #Coffee Shop the logical layer, if a #Coffee from a Client has been taken with #Coffee with milk or #Coffee without milk. Furthermore is here also the #Definition set $D = \{ 1 \}$. The #Definition set $D = \{ 1 \}$ means with $x = 1$, that the Client will take one #Coffee. At the #Value set $V = \{ 1,2 \}$ the values can be $y = 1$ and $y = 2$. The value $y = 1$ means, that the Client take the #Coffee without milk, and the value $y = 2$ means, that the Client take the #Coffee with milk:

#Coffee Shop #Coffee #Coffee with milk #Definition set $D = \{ 1 \}$ #Value set $V = \{ 1,2 \}$
$x = 1$; $y = 2$

#Coffee Shop #Coffee #Coffee without milk #Definition set $D = \{ 1 \}$ #Value set $V = \{ 1,2 \}$
$x = 1$; $y = 1$

The following figure shows the mentioned example #Coffee Shop with the extended logical layer and with the subdivision with the hashtag #Notation and hashtag #Hierarchies with the CPW Process method:

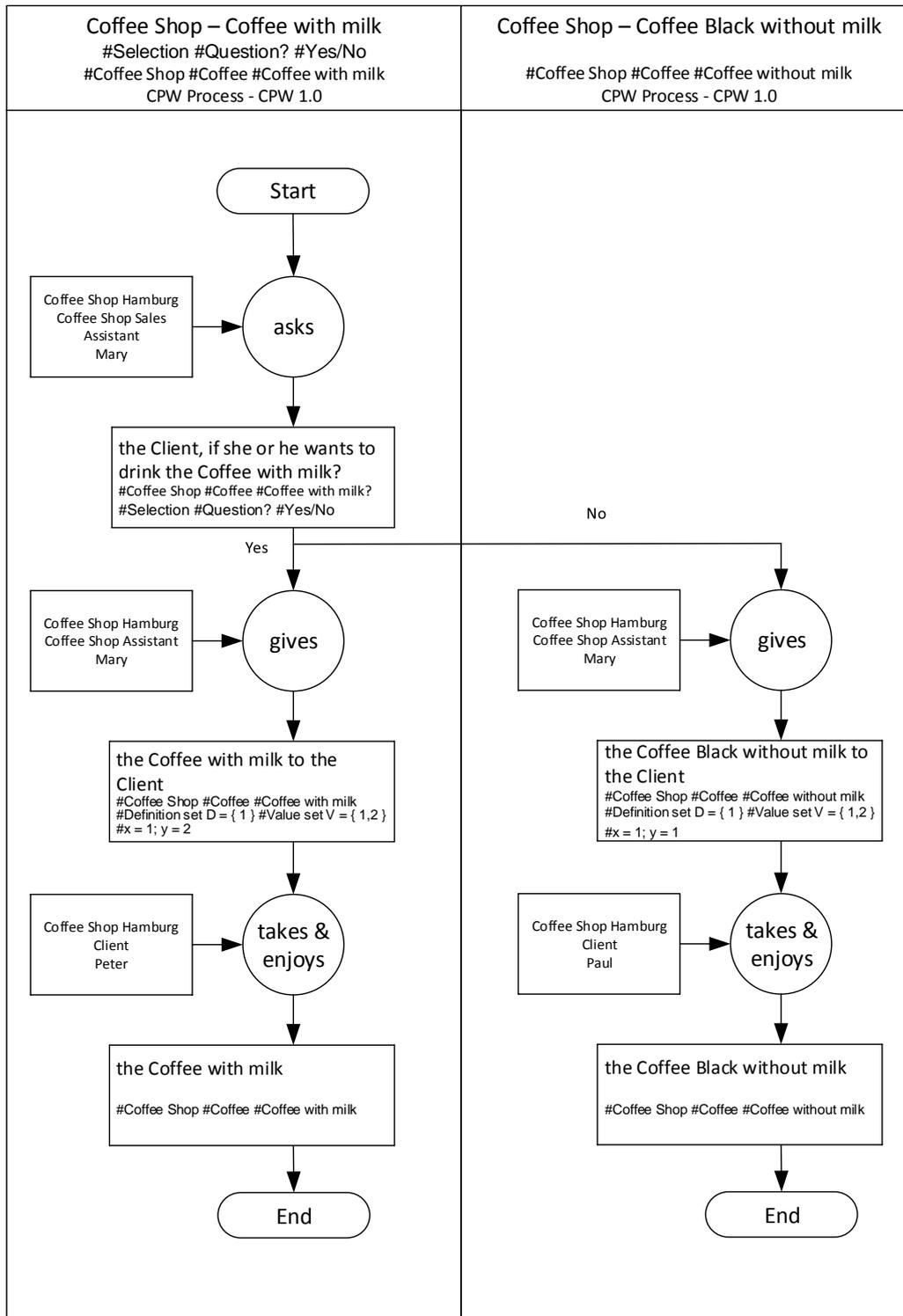


Fig. 1: Example for an extended logical layer and subdivision with Hashtag #Notation and with the CPW Process method

In this chapter has been described how the simplified logical layer of the CPW Process method can be extended with further possibilities of the #Notation, to cover either other categories, areas or topics or then to cover the logical operations, but also functions or mathematical functions and formulas, which can be occur in the analysis of business process reengineering.

References

CPW Method

- **CPW Method**
- **CPW Process method**

<http://www.cpw-method.com>