

Contents

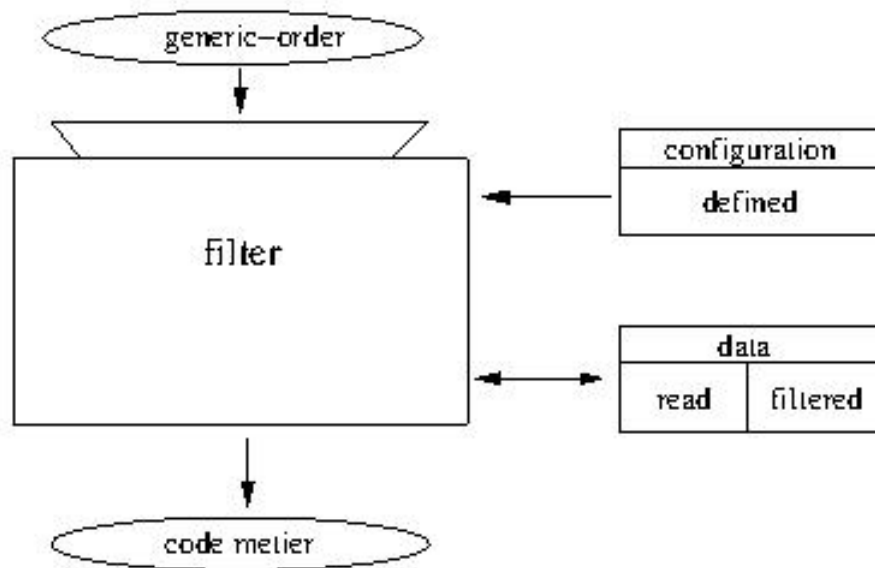
1	Introduction	1
2	Modèle conceptuel des traitements	1
2.1	In	1
2.2	Out	1
2.3	Data	1
3	Implementation	2

1 Introduction

Cette page décrit le traitement '**filter**' faisant partie de l'activité '**actors**'.

Le but de ce traitement est de traiter les données de l'expérience puis de les difuser aux autres acteurs.

2 Modèle conceptuel des traitements



2.1 In

Nous simplifions ici le modèle des traitements en ométant les ordres génériques décrits au niveau de l'activité.

2.2 Out

Ce traitement génère l'évènement suivant :

- *code-métier* à destination du traitement ???

2.3 Data

Un scénario spécifie la configuration le code métier à charger par l'acteurs.
Les données sont traitées à la volée. Chaque n-uplet étant indépendant des autres.

- `in` : Objet **scénarios**
- `in/out` : Objet **data**

3 Implementation

Cet acteur est implémenté par le fichier `/cvs/narval/actors_LC/filters/LC_filter.c`.

```
void process_block (struct my_struct *algo_data,
                   void *input_buffer,
                   unsigned int size_of_input_buffer,
                   void *output_buffer,
                   unsigned int size_of_output_buffer,
                   unsigned int *used_size_of_output_buffer,
                   unsigned int *error_code)
{
    int i;
    *error_code = 0;
    *used_size_of_output_buffer = 0;

    if (size_of_input_buffer != 8 || size_of_output_buffer < 8)
    {
        printf("filter:\terror\n");
        *error_code = 1;
        goto error;
    }

    printf("filter: \t%s\n", (char*) input_buffer);

    for (i=0; i<8; ++i)
    {
        ((char*)output_buffer)[i] = ((char*)input_buffer)[i];
    }

    *used_size_of_output_buffer = 8;
error:
    return;
}
```