

# Contrôle des observations pour la gestion des systèmes de flux de données.

Christophe Dousson\*, Pierre Le Maigat\*

\*Orange Labs – 2, avenue Pierre Marzin – 22300 Lannion  
{christophe.dousson, pierre.lemaigat}@orange-ftgroup.com  
<http://perso.rd.francetelecom.fr/dousson>

**Résumé.** Les systèmes d'analyse de flux de données prennent de plus en plus d'importance dans un contexte où les données circulant sur les réseaux sont de plus en plus volumineuses et où la volonté de réagir au plus vite, en temps réel, devient un besoin nécessaire. Afin de permettre des analyses aussi rapides et efficaces que possible, il convient de pouvoir contrôler les flots de données et de focaliser les traitements sur les données pertinentes. Le protocole présenté dans ce papier donne au module de traitement des capacités d'action et de contrôle sur les observations remontantes en fonction de l'état de l'analyse. La diminution des flux résultant de telles focalisations permet des traitements beaucoup plus efficaces, plus pertinents et moins consommateurs de ressources. Les premiers résultats montrent un réel gain de performances sur nos applications (facteur 100).

Nous proposons donc ici un protocole permettant de propager des informations de contrôle du plus haut-niveau de l'analyse jusqu'aux sources d'événements. L'architecture mise en œuvre, baptisée TESS (pour *Timestamped Event Stream System*) est de type « workflow » où les événements transitent de module en module par des « liens ». Ces liens sont orientés d'une interface dite « Producteur » vers une interface dite « Consommateur » (voir figure 1) sur lesquels vont circuler les données du flot. Cette architecture s'appuie sur les hypothèses suivantes :

- les événements sont tous instantanés (les informations avec durée pourront être modélisées avec un événement de début et un autre de fin),
- les événements sont tous datés (avec une date ponctuelle) et seront donc notés  $(e, t)$ ,
- les connexions entre producteur et consommateur sont de type FIFO (en revanche, il n'y a pas de contrainte sur le fonctionnement interne d'un module),
- les envois de messages et d'événements sont tous *asynchrones*.

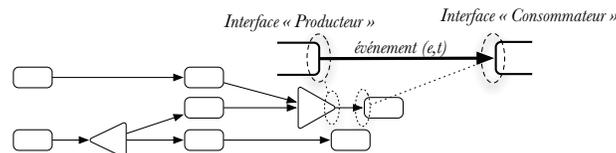


FIG. 1 – Architecture générale : producteurs et consommateurs