

DomVision : Intergiciel de gestion de données pour l'environnement domestique

Loïc Petit^{1,2}

Claudia Lucia Roncancio²

Cyril Labbé²

François-Gaël Ottogalli¹

¹ - Orange Labs, Grenoble, France `name.surname@orange-ftgroup.com`

² - University of Grenoble, LIG Laboratory, France `name.surname@imag.fr`

Résumé

La complexité toujours grandissante de l'environnement domestique rend la mise œuvre de services de plus en plus difficile. La maîtrise de ces équipements et de leurs interactions est central du point de vue de l'utilisateur final ou des fournisseurs de services. Nous nous intéressons à l'approche déclarative afin de permettre une gestion des données plus flexible dans cet environnement. En effet, le réseau local domestique pouvant être considéré comme un producteur de flux de données hétérogènes, nous souhaitons pouvoir l'interroger à la fois ces flux mais aussi leurs historiques. Nous présentons ici *DomVision* qui couple un système de gestion de flux de données et un SGBD relationnel pour profiter à la fois de la gestion de la dynamique et des capacités de persistances des différents moteurs. Nous décrivons ici l'architecture globale ainsi que son utilisation dans notre cadre applicatif.

1 Motivation

Le réseau domestique est le lieu principal d'utilisation des appareils de haute technologie. Que ce soient des décodeurs TV, des passerelles internet, des tablettes, les ordinateurs ou des disques durs réseau ; un grand ensemble d'équipements collaborent pour fournir des services à l'utilisateur de plus en plus évolués. Cependant, cette collaboration est connue [5] pour être extrêmement complexe à gérer principalement à cause de l'hétérogénéité des dispositifs et la dynamique de l'environnement.

L'ensemble du réseau local domestique peut être considéré comme un producteur de flux de données utiles à des diverses fins. Observer et analyser ces données localement, mais aussi d'un point de vue global devient un enjeu important.

De nombreux travaux de recherches [7, 4] et industriels développent de plus en plus de méthodes et de protocoles pour permettre la gestion des équipements domestique et notamment, l'accès à leurs données de contrôle et de configuration. Ces travaux sont supportés par des initiatives telles que le *BroadBand Forum*, la *HomeGateway Initiative* et le forum *Universal Plug'n'Play* (avec notamment le profil *Device Management*¹), rassemblant les plus grand acteurs industriels.

Comprendre cet ensemble de données est primordial pour les trois grands acteurs autour des systèmes d'informations domestiques : l'utilisateur, les constructeurs-développeurs et les opérateurs télécoms. Ce dernier doit notamment pouvoir gérer un parc à très grande échelle² pour ses besoins propre, mais aussi pour aider ses utilisateurs à maîtriser le fonctionnement des divers dispositifs et utiliser au mieux les données disponibles. Pour cela, les flux de données issus de l'environnement

1. UPnP Device Management :1 <http://upnp.org/specs/dm/dm1/>

2. De l'ordre de dizaines de millions de clients, ainsi qu'un parc de millions de boxes en croissance

local (la maison) doivent pouvoir être analysés, comparés à des historiques, mais aussi couplés aux données du système d'information de l'opérateur et autres prestataires de service.

Nos travaux contribuent à la construction des systèmes d'information domestiques en proposant une solution qui permet l'interrogation déclarative des flux et des historiques. L'évaluation des requêtes est faite au plus près des producteurs des flux de données. Cela permet, à un niveau local, plus de flexibilité et une meilleure adaptation à la complexité de l'environnement. Notre proposition couvre un large spectre de requêtes en s'appuyant sur l'algèbre Astral [10, 9]. Cette algèbre formalise les opérateurs sur flux et relations temporelles et a permis de mettre en évidence des propriétés d'associativité et commutativité sur ces opérateurs qui sont utilisables dans un but d'optimisation. L'utilisation d'une solution de gestion de données structurée et optimisable pour l'environnement local constitue un atout supplémentaire pour faciliter le passage à l'échelle des systèmes d'informations reliant plusieurs maisons, villes, etc.

La démonstration proposée dans cet article présente DomVision, l'intergiciel que nous proposons pour les systèmes d'informations domestiques. Il met en œuvre une architecture effectuant le couplage entre notre système de gestion de flux (Astral) à un SGBD relationnel dans un contexte de dispositifs domestiques basés sur le protocole *UPnP*. Cette solution permet de fonctionner dans un contexte dynamique en offrant aussi bien un traitement temps réel des données qu'un service d'historisation. Elle permet une gestion fine de la stratégie de persistance des données observées ainsi qu'une rétroalimentation pour influencer la politique de *monitoring*.

Nous détaillerons dans la section 2, comment un couplage peut s'établir entre les deux systèmes. Ensuite, nous aborderons en section 3 l'architecture interne de notre intergiciel. Nous poursuivrons par la description de l'environnement domestique en section 4. Et enfin, nous présenterons le périmètre de la démonstration en section 5.

2 Couplage d'un gestionnaire de flux et d'un gestionnaire de base de donnée

Les *Data Stream Management Systems* [6] résultent de l'application des principes de la gestion des bases de données aux flux de données. Il est ainsi possible d'utiliser une stratégie déclarative pour effectuer des interrogations au dessus de flux de données. Nous utilisons ici l'algèbre Astral [10, 11, 9], une algèbre de flux inspirée du modèle relationnel et de plusieurs travaux existants dans le domaine [3, 1]. Ce type de modèle permet notamment de manipuler les flux et les relations temporelles (relations évoluant au cours du temps) dans la même algèbre. Ainsi, la liaison entre un gestionnaire de flux et un gestionnaire de base de données relationnel est possible.

Il existe plusieurs applications qui expérimentent le couplage entre des données stockées et les systèmes de flux. L'application la plus évidente étant l'écriture des résultats de requêtes sur les flux en base de données [2]. De façon inverse, il est aussi possible de mêler des données stockées en entrée du système de gestion de flux en représentant les données sous forme de relation temporelles [12].

Ainsi le couplage entre un gestionnaire de flux et un gestionnaire de base de données est possible de façon conceptuelle. Nous allons décrire l'architecture logicielle mise en place pour réaliser ce couplage.

3 Architecture de DomVision

L'architecture de l'intergiciel s'établit en trois grands thèmes. Tout d'abord, la gestion des flux de données, l'intégration des historiques dans un système de gestion de base de données et l'écriture des composants dédiés au couplage.

3.1 Structure de la gestion flux

Les travaux sur l’algèbre Astral ont servis de base au développement d’un prototype³. Sa structure d’utilisation est classique dans le cadre des applications d’observation que l’on retrouve dans plusieurs prototypes de gestionnaire de flux [8]. Une exécution de requête est décomposable en trois parties :

- **Les sources** sont des composants *adaptateurs* qui représentent les équipements. En effet, chaque source se place aux dessus d’un protocole réseau (tel que *UPnP*, très présent dans l’environnement domestique) et forme un flux de n-uplet (ou une relation temporelle) à partir des données qu’il est possible de collecter sur le ou les équipements.
- **L’arbre d’opérateurs** est un enchaînement de composants. Les feuilles sont les sources de données et les composants effectuent les différents traitements (modèle Astral) sur les flux.
- **Les *handlers*** utilisent directement le résultat de l’arbre d’opérateur pour effectuer une action. Par exemple, notifier l’utilisateur d’un nouveau n-uplet dans son flux d’alerte.

La structure interne d’un gestionnaire de flux étant détaillé, la manière de créer des historique à partir des flux va être décrite.

3.2 Définir un historique à partir des observations

Supposons qu’il soit possible de faire évoluer la base de données pour représenter l’ensemble des concepts observables du système. Il est nécessaire de pouvoir lier l’historique d’une donnée (i.e. un flux en sortie d’Astral) à son instance dans le modèle. Une structure relationnelle est utilisé pour relier les concepts aux flux.

Nous définissons la notion de *paramètre* comme une variable d’état observable sur un des concepts. Par exemple, la *charge processeur* est un paramètre d’un équipement ou d’une application, *bande passante utilisée* est un paramètre d’un lien réseau. Dans notre cas, un paramètre sera un attribut particulier remonté dans les flux d’Astral. Nous pouvons attacher dans la base de données un *monitorable* (concept abstrait) et un paramètre à un historique. Il est donc possible de savoir, pour toute entité observable, les différents paramètres que nous avons pu obtenir et d’en récupérer l’historique. Il est important de noter que la sémantique de l’attachement d’un paramètre est très forte. En effet, si nous regardons le paramètre de charge processeur, l’attachement à une application indiquera la charge que consomme cette application. Alors que si l’attachement avait été fait sur l’équipement, elle indiquerait la charge totale. Lors de l’analyse des données, cette séparation sera extrêmement structurante.

Nous avons désormais décrit comment nous pouvons établir des historiques et décrire les composants permettant le couplage.

3.3 Couplage avec le gestionnaire de flux

Afin d’illustrer plus efficacement le couplage, la figure 1 résume l’architecture fonctionnelle de DomVision. Nous avons créé plusieurs composants Astral pour permettre de correctement coupler la gestion de flux et le gestionnaire de base de données précédemment présentés.

L’**alimentation** du modèle principal se fait de manière très ad hoc en créant différents *handlers* (composant H_3 sur la figure 1) étant capables de garantir les contraintes du modèle avec des informations fournies par différents flux. Cependant, nous avons écrit plusieurs *handlers* suffisamment génériques pour effectuer ce type de tâche.

La **persistance** d’un flux dans le but de stocker l’historique d’un (ou plusieurs) paramètre(s) est gérée par un autre *handler* (composant H_2 sur la figure 1). Celui-ci suppose qu’un attribut *monitorableId* existe. Si tel est le cas, il entretiendra la table *HasParameter* pour permettre d’associer

3. <http://astral.ligforge.imag.fr>

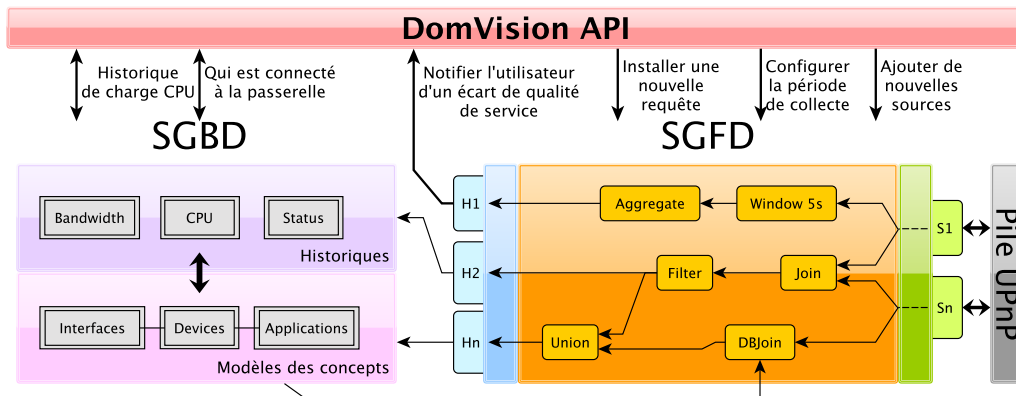


FIGURE 1 – Architecture interne de DomVision

les concepts observés à l'historique. Au besoin, il devra créer les entrées dans les tables *Parameter* et *ColumnPointer* à l'initialisation évidemment.

Afin de pouvoir **associer** un identifiant de base de données dans un flux, nous avons développé un opérateur léger⁴ (opérateur *DBJoin* sur la figure 1) qui permet pour tout tuple d'un flux d'aller chercher l'unique n-uplet de la relation *Table* qui vérifie que l'attribut *Column* est égal à un certain attribut du n-uplet du flux. Un exemple concret serait l'identification d'un équipement à partir d'un numéro de série.

Enfin pour toute **analyse** en temps réel plus poussée, nous avons créé une source qui représente le résultat d'une requête *SQL* qui peut être réactualisée périodiquement ou avec un *trigger* par le gestionnaire de base de données.

Nous avons vu précédemment comment il est possible de faire un système de gestion de données déclarative couplant flux et bases de données permettant d'exploiter les avantages de chacune des approches. Nous allons maintenant décrire comment cet ensemble s'instancie dans notre cadre applicatif.

4 L'environnement domestique

Nous devons décrire les concepts que nous pouvons trouver à l'intérieur de la maison. Nous avons représenté l'ensemble des classes du réseau local domestique ainsi :

- **Device** : Équipement matériel qui est raccordé au réseau local domestique.
- **Application** : Logiciel (ou service) embarqué sur un seul et unique *device*
- **Interface** : Interface réseau qui comporte une adresse logique (*ip*) et physique (*mac*) attaché à un seul et unique *device*.
- **Link** : Lien réseau qui relie deux interfaces réseau.

À chacun de ces entités, nous pouvons attacher des notions de *types* (*interface wifi*, *device* de type passerelle), que nous n'aborderons pas dans ce document.

Nous sommes conscients que ce modèle est simplifié et qu'un raffinement du modèle pourrait être de mise. Toutefois, ce modèle est suffisant pour la mise en œuvre de notre solution, des travaux futurs sont en cours pour améliorer la précision de ce modèle.

4. En réalité, une fonction tierce qui est exécutée dans un opérateur d'évaluation d'expression

5 Démonstration

Afin de démontrer l'efficacité et la pertinence de notre système d'information domestique, nous mettons en place un cadre assez typique d'une utilisation multimédia dans cet environnement. Tout d'abord, un routeur nous servira de nœud central du réseau. Ensuite, deux ordinateurs représenteront l'utilisation classique du multimédia avec un fournisseur de contenu (*Media Server*) et un lecteur (*Media Renderer*). Tous deux supportent le profil *UPnP-DM* qui nous permet de récupérer un grand ensemble de données.

DomVision a été développé en Java-OSGi. Il utilise directement les API d'Astral ainsi que la pile protocolaire *UPnP* pour *OSGi* de *Cling*. Le gestionnaire de base de données choisit est *Apache Derby* pour sa portabilité dans l'environnement Java-OSGi. Ainsi grâce à sa grande portabilité, DomVision sera déployé sur une petite plateforme embarquée⁵ qui sera elle-même connectée à la passerelle. Afin de pouvoir montrer les différentes capacités de notre solution, différents types de liens réseau seront établis (WiFi ou Ethernet).

Le scénario d'exécution montrera l'utilisation habituelle d'un réseau local domestique. Nous exécuterons ainsi plusieurs requêtes de différents types : requêtes sur flux simples ou compliquées ; requêtes instantanées et historiques sur la topologie ou sur des paramètres ; requêtes utilisant la base comme source.

Références

- [1] D. Abadi, D. Carney, U. gur Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora : a new model and architecture for data stream management. *VLDB '03 : Proceedings of the 29th international conference on Very large data bases*, 12(2), 2003.
- [2] K. Aberer, M. Hauswirth, and A. Salehi. Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks. In *MDM '07 : International Conference on Mobile Data Management 2007*, pages 198–205, 2007.
- [3] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, and M. Datar. STREAM : The Stanford Data Stream Management System. *Data Stream Management : Processing High-Speed Data Streams*, Jan. 2004.
- [4] S. Ben Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, and Y. Berbers. EASY : Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. *Journal of Systems and Software*, 81 :785–808, Aug. 2007.
- [5] K. Geihs. Middleware challenges ahead. *Computer*, 34(6) :24–31, June 2001.
- [6] L. Golab and M. T. Özsu. Issues in data stream management. *SIGMOD '03 : Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 32(2), 2003.
- [7] T. Gu, H. Pung, and D. Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1) :1–18, 2005.
- [8] S. Inc. SQLstream. <http://www.sqlstream.com>, 2010.
- [9] L. Petit. Astral : Advanced Stream Algebra, Wiki. <http://sigma.imag.fr/astral>, 2010.
- [10] L. Petit, C. Labbé, and C. L. Roncancio. An Algebraic Window Model for Data Stream Management. In *MobiDE '10 : Proceedings of the 9th International ACM Workshop on Data Engineering for Wireless and Mobile Access*, pages 17–24. ACM, 2010.
- [11] L. Petit, C. Labbé, and C. L. Roncancio. Revisiting Formal Streams and Temporal Relation in Data Stream Querying. In *Article soumis à BDA '2011*, pages 0–20, 2010.
- [12] F. Reiss, K. Stockinger, K. Wu, A. Shoshani, and J. M. Hellerstein. Enabling Real-Time Querying of Live and Historical Stream Data. In *SSDM' 07 : Proceedings of the 19th International Conference on Scientific and Statistical Database Management*, page 28, 2007.

5. GlobalScale GuruPlug Server Plus