Array-OL : proposition d'un formalisme Tableau pour le Traitement de Signal Multi-Dimensionnel

Alain Demeure* Anne Lafage** Emmanuel Boutillon** Didier Rozzonelli*
Jean-Claude Dufourd** Jean-Louis Marro*

* *Thomson Sintra ASM, 525, route des Dolines, BP 157, 06903 Sophia-Antipolis Cedex*
** *ENST Paris, 46, rue Barrault, 75634 Paris Cedex 13*

RÉSUMÉ

Array-OL est un formalisme permettant de spécifier graphiquement sous forme de manipulation de tableaux le traitement de signal multi-dimensionnel caractéristique des systèmes de détection. La saisie d'une application se fait à deux niveaux : à un niveau global, l'application complète est d'abord donnée par un graphe dont les nœuds échangent des tableaux; à un niveau local, on décrit ensuite les opérations sur tableaux effectuées dans chaque nœud. Les résultats de cette étude seront bientôt accessibles dans le *framework* Ptolemy.

ABSTRACT

Array-OL is a graphical formalism based on array transformations which allows the full specification of the multidimensional signal processing implemented in the detection systems. Array-OL proposes a two level approach: first, the global level describes the processing through a graph where the nodes exchange multidimensional arrays; second, the local level details the calculations performed on the arrays by each node. The results of this study will be soon avalailable in the Ptolemey framework.

1 Introduction

Parler de dimension pour un signal signifie qu'il existe un critère simple pour en ordonner les échantillons. Pour un signal vocal numérisé, il est ainsi tout naturel d'ordonner les échantillons suivant leur ordre d'arrivée. On constitue ainsi un vecteur d'échantillons qu'on peut voir aussi comme un tableau à une dimension. Cette opération équivaut à associer à chaque échantillon un indice qui représente l'âge de l'échantillon.

Si on considère ce signal vocal non plus avant mais pendant son traitement, l'organisation mono-dimensionnelle ci-dessus ne tient généralement plus : si on applique à ce signal vocal une FFT par exemple, le signal acquiert une deuxième dimension Fréquence, alors que la dimension temporelle reste toujours présente, mais indicée désormais par l'âge de la récurrence FFT.

Pour les systèmes de détection (ex. : Radar, Sonar, Goniomètre en Guerre Electronique), le signal a un caractère multi-dimensionnel dès son acquisition. Ces systèmes prennent en compte des échantillons qui s'identifient par leur âge mais aussi par le numéro du capteur dont ils sont issus. Ces échantillons constituent en fait les éléments d'un tableau au moins à deux dimensions : Capteurs et Temps. Le signal conserve ensuite pendant tout son traitement ce caractère multi-dimensionnel justifiable d'une organisation en tableau.

Ce papier propose un formalisme où une chaîne de Traitement de Signal (TS) apparaît comme une suite d'étapes opérant sur des tableaux.

Dans ce formalisme, on a recours successivement à deux modèles pour spécifier une application :

1. un premier modèle, dit **Modèle Global**, permet de décrire l'ensemble de la chaîne comme un graphe d'étapes échangeant des tableaux,
2. un deuxième modèle, dit **Modèle Local** permet de définir complètement pour chaque étape calculs et accès aux tableaux.

Ce formalisme que nous avons nommé Array-OL (*Array Oriented Language*) a été développé tout particulièrement pour décrire le TS des systèmes de détection. Sa spécification a été faite dans le cadre d'une coopération Thomson-Telecom Paris. Tant pour le Modèle Global que Local, choix a été fait d'une saisie graphique pour définir une application. Les éditeurs graphiques et le simulateur seront bientôt accessibles dans une première version à la communauté TS dans le *framework* Ptolemy [Pto94] où un domaine Array-OL a été créé[1].

Ce papier commence par un rappel sur la structure Tableau. Il précise ensuite la nature et les particularités des dimensions en TS. On introduit ensuite le Modèle Global et on détaille plus particulièrement le Modèle Local.

2 Rappels sur la structure Tableau

Tableau et Repère Cartésien (ou Système de Coordonnées) désignent en fait le même type d'objet. Ce premier rappel est destiné à préparer l'approche graphique choisie pour Array-OL, et suffira à ceux qui sont familiers avec la structure Tableau.

Un tableau peut être vu comme un ensemble de cases que l'on juxtapose en ligne, rectangle, parallélépipède, etc. Cet ensemble de cases constitue alors un repère cartésien,

[1] Ptolemy peut être obtenu gratuitement auprès de l'université de Berkeley : `http://ptolemy.eecs.berkeley.edu`.

généralement borné, dont on n'aurait gardé que les points de coordonnées entières. Placer des valeurs dans ces cases revient à ordonner ces valeurs suivant la structure du tableau. Dans la communauté TS, assimiler Tableau et Repère Cartésien se fait intuitivement puisqu'on utilise indifféremment les termes Dimensions ou Axes, Indices ou Coordonnées, Composantes ou Points, etc.

Tout l'intérêt de l'organisation Tableau réside dans la facilité de désigner une case pour y ranger ou y prélever une valeur, cette désignation se faisant quand on spécifie ou quand on exécute effectivement un traitement : la case est désignée succinctement par ses indices. Des groupements de cases peuvent être aussi désignés commodément s'ils correspondent à des agencements particuliers : lignes, colonnes, et plus généralement des éléments contenus dans des fenêtres elles-même structurées en fait en sous-tableaux. La désignation synthétique de ces groupements peut se faire en donnant les équations et inéquations affines vérifiées par les indices repérant les cases correspondantes.

3 Les dimensions usuelles en TS et leurs particularités

Les dimensions permettant d'ordonner les échantillons en TS correspondent à des grandeurs physiques. Trois classes peuvent être distinguées :

- les dimensions temporelles qui ordonnent suivant l'âge des échantillons,
- les dimensions spatiales qui utilisent un critère géométrique : numéro de capteur, direction, ...
- les dimensions fréquentielles usuellement absentes en entrée du système de traitement, mais créées lors de calculs le long des dimensions précédentes.

Pour les dimensions temporelles, une particularité fréquente est leur multiplicité qui correspond à des quantifications croissantes sur l'âge des échantillons (un peu comme le temps est donné en secondes, minutes, heures, etc.). En radar par exemple (figure 1), on ordonne en Cases Distance, puis en Récurrences, puis en Rafales, et davantage. Parallèlement à ces trois dimensions, on accède à des composantes groupées pour des traitements particuliers (accès parallèles à la dimension Case Distance pour la compression d'impulsion, parallèles à la dimension Récurrence pour le calcul Doppler, ...). On peut accéder parallèlement à une certaine dimension en plusieurs points de la chaîne (exemple : la compression et la TFAC sur la dimension Case Distance).

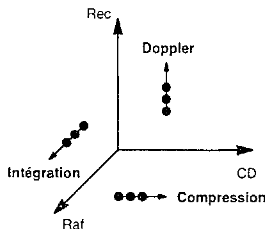

Figure 1: Exemple radar

Pour les dimensions temporelles, une seconde particularité est que la dimension correspondant à la plus grosse quantification peut avoir une taille infinie. Ceci est presque toujours la règle pour les systèmes de détection passifs, où faute de pouvoir borner le temps d'observation par la durée d'action d'un signal émis (contrairement aux systèmes actifs), on est amené à traiter un flot ininterrompu d'échantillons en provenance des codeurs.

Les dimensions spatiales peuvent présenter plus rarement une autre particularité : le rebouclage. Autrement dit, le dernier indice a un successeur, l'indice zéro. Pour une antenne Sonar dont les capteurs (hydrophones) sont disposés par exemple en cercle, la dimension spatiale n'a ni début ni fin. Le tableau Hydrophone et Temps acquis par le système doit être vu en fait comme cylindrique; la formation des voies nécessitera de mener le calcul sur les indices des hydrohones dans une arithmétique modulo le nombre total d'hydrophones.

Toutes ces particularités sont prises en compte dans Array-OL qui admet des dimensions de taille infinie et des groupements d'éléments sortant par le bord d'un tableau et rentrant par le bord opposé.

4 Variabilité des dimensions et des accès

La figure 2 symbolise dans une optique Tableau la tête d'un traitement de signal pour une chaîne Sonar. Elle fait apparaître trois étapes : FFT, Formation des Voies (FV), Regroupement Large Bande (RLB : sommation de tous les canaux Fréquence). Les tableaux transmis de proche en proche sont figurés avec leur nombre exact de dimensions et leur taille. On voit que des dimensions apparaissent (exemple : la dimension fréquence F en sortie de la FFT) ou disparaissent (exemple : cette même dimension F supprimée par le RLB). D'autres dimensions perdurent à travers leur caractère physique, mais subissent des modifications sur la signification de leur quantification. Par exemple, la dimension T, de taille infinie, est quantifiée avant la FFT en période d'échantillonnage d'entrée, et se retrouve après FFT, toujours avec sa caractéristique infinie, mais indicée désormais avec l'âge des récurrences FFT (ceci amène à renommer R cette dimension).

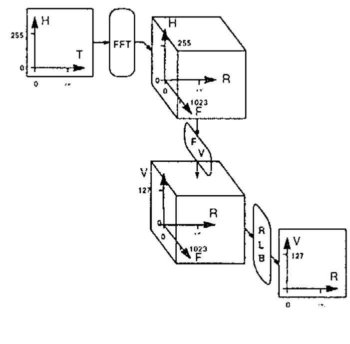

Figure 2: Tête de chaîne Sonar

Sur la figure, les flèches reliant une étape à ses tableaux sont censées indiquer la direction suivant laquelle les accès groupés se font dans un tableau pour alimenter le calcul. Par exemple, la flèche reliant le tableau en tête de chaîne (dimensions H, T) à l'étape FFT est dessinée parallèle à la dimension

T pour indiquer que la FFT accède à des échantillons alignés suivant la dimension temporelle. Sur l'ensemble de la figure, on remarque alors que les flèches en lecture s'alignent suivant des dimensions toujours différentes. De façon générale, on constate pour le TS des systèmes de détection, une réorientation constante des accès ("changement d'axe" ou encore "*corner turn*" en Anglais).

Dans la conception d'Array-OL, on a pris note que rendre compte des spécificités d'une chaîne de traitement de signal passait plus par la description de l'évolution des dimensions et la définition rigoureuse des accès, que par le détail des calculs (majoritairement: somme de produits, FFT, la panoplie complète ne composant qu'une petite bibliothèque). Array-OL est avant tout un formalisme pour décrire des tableaux et les accès dans ces tableaux.

5 Le Modèle Global

Le Modèle Global permet de donner graphiquement une vue synoptique de la chaîne de traitement.

L'application est décrite par un graphe dont les nœuds constituent les étapes et les arcs portent explicitement les tableaux (figure 3).

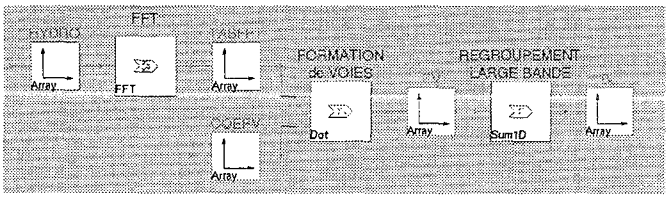

Figure 3: Modèle Global

L'information apportée par ce graphe reste limitée:

- dépendances entre étapes,
- consignes pour le Modèle Local portant sur le nom et la taille des tableaux.

Nous avons introduit le Modèle Global plutôt dans un souci d'ergonomie (les informations ci-dessus auraient pu être fournies au niveau du Modèle Local mais en alourdissant la saisie). Le Modèle Global garantit plus facilement la cohérence de toute l'application et améliore la lisibilité de la documentation.

La représentation sous forme de graphe pourrait laisser croire que notre Modèle Global est très proche du modèle Flot de Données très en usage dans la communauté TS. L'interprétation est cependant très différente :

- dans le modèle Flots de Données, les arcs acheminent un flux continu de jetons et toutes les étapes opèrent en parallèle,
- dans notre Modèle Global, un arc achemine un jeton unique: un tableau, dont la taille peut être infinie; de proche en proche les étapes ne sont mises à feu qu'une fois, et opèrent sans se recouvrir au moins sur les portions du graphe ne comportant pas de rebouclage.

6 Le Modèle Local

Le Modèle Local est utilisé pour donner, étape par étape, le détail des calculs et des accès dans les tableaux.

6.1 Principe du Modèle

Tout d'abord en Array-OL, on considère que le volume de calcul attaché à une étape est exactement défini par la taille de son tableau de sortie: l'étape doit en calculer toutes les composantes.

On considère ensuite que ce volume de calcul est décomposable en calculs plus fins, tous identiques, qui n'intéressent chacun qu'un groupe restreint de composantes, voire une seule. Pour désigner ces notions de 1. calcul plus fin, 2. groupe de composantes, nous avons choisi dans Array-OL les termes **Transformation Elémentaire** (TE) et **Motif.**

L'accomplissement d'une étape peut être vu alors comme une construction : l'étape doit édifier complètement le tableau Résultat en y juxtaposant des Motifs Résultats produits à partir de Motifs Opérandes via la TE.

La figure 4 schématise la construction du tableau H, F par itération de la TE FFT. A chaque itération, celle-ci produit un vecteur de fréquences (Motif Résultat) à partir d'un vecteur d'échantillons temporels (Motif Opérande).

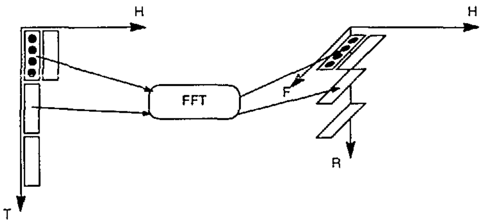

Figure 4: Construction du tableau FFT

6.2 Les simplifications offertes par le domaine TS

Array-OL se limitant au TS, il est possible de faire des restrictions sur les possibilités du formalisme, qui ne seraient bien sûr pas acceptables pour un langage à parallélisme de Données visant le calcul scientifique (ex. : HPF). Ces restrictions correspondent à des constatations simplificatrices qui n'ont jamais été mises en défaut dans les nombreuses applications utilisées pour tester Array-OL.

Première simplification: le Motif est un tableau. On constate que les Motifs sont eux-même structurés en tableau, les cas les plus courants étant le point unique ou le vecteur. Array-OL admet cependant les Motifs à dimensions multiples.

Deuxième simplification: des composantes alignées et équidistantes sur un Motif s'ajustent sur des composantes alignées et équidistantes sur un tableau. Cette simplification signifie qu'en particulier chaque dimension du Motif s'ajuste sur le tableau suivant une orientation et un pas d'échantillonnage restant constants pour toutes les itérations de la TE.

Troisième simplification : des Motifs Résultats alignés et équidistants sont produits par des Motifs Opérandes alignés et équidistants. En particulier, des Motifs Résultats juxtaposés suivant une dimension Résultat (ou encore "pavant" cette dimension) ont été produits par des Motifs Opérandes eux-même alignés et équidistants.

6.3 Spécification du calcul

Lors de la saisie d'une étape, on commence par sélectionner dans la bibliothèque des TE, celle mise en œuvre par cette étape. Un paramétrage de cette TE peut être nécessaire pour dimensionner ses Motifs (ex. : définir des Motifs 1D de 1024 points pour une FFT).

Sont donc complètement dissociées la spécification des calculs et la spécification des accès, ce qui constitue une particularité d'Array-OL. Le bénéfice est alors double :

- vis à vis des TE, les dimensions sont complètement banalisées; ex. : il n'y a pas de distinction à faire entre un FIR suivant une dimension temporelle et une Formation de Voies suivant une dimension spatiale, toutes deux ne faisant pour calcul qu'une somme de produits,
- la panoplie des TE nécessaires pour décrire un très grand nombre d'applications est réduite; on a une véritable notion de bibliothèque.

6.4 Spécification des accès

6.4.1 L'Ajustage

La deuxième simplification énoncée en 6.2 est concrètement mise à profit dans Array-OL : plutôt que de spécifier dans une forme littérale, mathématique ou autre, les relations affines entre indices des composantes du Motif et indices des composantes du tableau, l'utilisateur définit graphiquement comment s'ajustent l'origine du Motif puis le premier point de chacune des dimensions, ce qui est suffisant pour retrouver leur orientation et leur pas (figure 5). Dans la terminologie Array-OL, cette première partie de la spécification d'accès est appeleé **Ajustage**; la mise en correspondance graphique : un point du Motif, un point du tableau, est appelée **Exemple**.

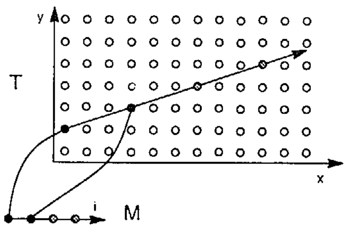

Figure 5: Ajustage

6.4.2 Le Pavage

Il reste encore à préciser comment, pour chaque itération de la Transformation Elémentaire, se disposent l'ensemble de tous les Motifs. Nous avons appelé Pavage cette dernière partie de la spécification des accès.

Une fois ajusté, un Motif voit la position de toutes ses composantes définie par la seule position de son origine. Seule cette dernière reste donc à considérer.

Tableaux Opérandes et tableaux Résultats n'ont pas un rôle symétrique dans l'accomplissement d'une étape :

- un élément résultat n'est calculé qu'une fois alors que les Motifs Opérandes se recouvrent fréquemment (ex. : FIR "glissant"), ce qui amène à utiliser plusieurs fois un même élément opérande,
- par principe dans Array-OL, tous les éléments résultats doivent être calculés, alors que les éléments opérandes ne sont pas nécessairement tous utilisés.

Ces remarques amènent à donner une fonction **Pilote** au tableau résultat (s'il y en a plusieurs, on en choisit un) : on va définir comment la position d'un Motif Résultat contraint en fait la position de tous les autres. Là encore, la spécification va se faire par des exemples graphiques, en utilisant la troisième simplification (cf. section 6.2). Un premier Motif Résultat (en fait son origine) est désigné arbitrairement dans le tableau résultat Pilote (au plus simple, on peut prendre celui qui comprend l'origine du tableau); pour ce premier Motif, on donne aussi la position de l'origine de tous les autres Motifs. Ceci constitue les premiers exemples de Pavage. Pour que la spécification du Pavage soit complète, il faudra ensuite donner des exemples autant de fois qu'il y a de dimensions dans le tableau résultat Pilote (figure 6).

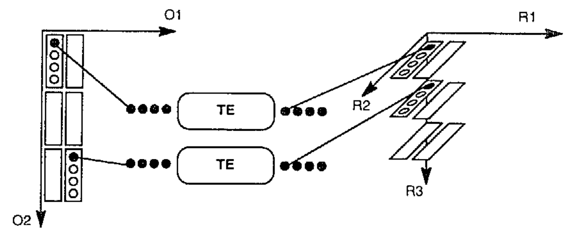

Figure 6: Pavage

7 Conclusion

Dans l'immédiat, nous voyons Array-OL comme un moyen très rapide de spécifier et de valider par simulation une chaîne TS. Travaillant au niveau du tableau, le concepteur d'application n'a plus en particulier à chercher comme dans l'approche Flot de Données la dimension du flot et la taille du jeton qui rendront compte des changements d'axe permanents (cf. section 4) dans le TS des systèmes de détection.

Contrairement à d'autres formalismes, une application écrite en Array-OL ne comporte pas de directives sur ses modalités d'exécution. Le placement des opérations, dans le temps sur machine séquentielle, dans le temps et l'espace sur machine parallèle, peut être envisagé d'innombrables façons, pourvu qu'il respecte les dépendances. Array-OL constitue alors un bon point de départ pour une problématique plus lointaine, mais que nous avons déja attaquée : le placement automatique d'applications TS sur machine dédiée.

Références

[Pto94] Ptolemy. An overview of the Ptolemy project. Technical report, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, March 1994.