Rapport final
TraLaLA

Site web : `http://www.cduce.org/tralala`
Descriptif du projet : `http://www.cduce.org/papers/tralala.pdf`

1 Liste des équipes impliquées

- Projet *Gemo*, INRIA Futurs, Saclay, en partenariat avec l'équipe *Vérification*, LIAFA, Paris.
- Équipe *Langages*, Laboratoire d'Informatique de l'École Normale Supérieure, Paris (jusqu'à septembre 2006); laboratoire Preuves Programmes et Systèmes, Université Denis Diderot, Paris (depuis octobre 2006).
- Équipe *Move* LIF, Marseille.
- Équipe *Bases de Données*, LRI, Université Paris 11, Orsay.
- Projet *MOSTRARE*, INRIA Futurs, Lille en partenariat avec l'équipe Spécification, Tests et Contraintes, LIFL, Université des Sciences et technologies de Lille.

2 Liste des participants

Nous énumérons toutes les personnes ayant participé au projet avec les dates de leur participation lorsqu'elles n'ont pas participé à toute la durée du projet. Le pourcentage indique le niveau de participation de chaque membre. Nous indiquons aussi les chercheurs invités par Tralala et qui ont travaillé sur le projet au moins un mois.

- Gemo + LIAFA
  - Andrei Arion, Gemo, doctorant, bourse MENSR, début de la thèse en 2004, 100%.
  - Claire David, LIAFA, doctorante, 100% (bourse BDI + monitorat), début de la thèse en 2005.
  - Ioana Manolescu, Gemo, Chargée de Recherche, 50%.
  - Anca Muscholl, LIAFA, Professeur des universités, 50%.
  - Mathias Samuelides, LIAFA, doctorant (bourse MENSR + monitorat), 01/09/2004-/01/09/2007, 100%.
  - Luc Segoufin, Gemo, Chargé de Recherche, 50%.
  - Cristina Sirangelo, Gemo, Post-doc, 100% (financement INRIA), 01/12/2005-31/12/2006.
- LIENS → PPS
  - Giuseppe Castagna, Directeur de Recherche, 30%,
  - Daniele Varacca, Maître de Conférences, 15%, depuis octobre 2006.

- Alain Frisch, Ingénieur Télécom Paris, 50% collaborateur exterieur dépuis 2005.
- Pietro Abate, Ingénieur de Recherche, 100% (CDD financement Tralala) 05/2007-07/2008,
- Nils Gesbert, Ingénieur de Recherche, 100% (CDD financement Tralala), 04/2006-09/2006,
- Karoline Malmkjær, Ingénieur de Recherche, 100% (CDD financement Tralala) 11/2007-01/2008,
- Luca Padovani, chercheur invité, 100% 06/2006 (financement Tralala), 12/2007 (financement Tralala) 05/2007 (invitation Paris 7).
- Mariangiola Dezani, chercheur invitée, 100% 01/2005 (financement Tralala), 01/2007 (financement Tralala).
- Haruo Hosoya, chercheur invité, 100% 10/2004-11/2004 (financement Tralala) et 10/2006 (invitation ENS).

• LIF

- Denis Lugiez, Professeur, 30%
- Silvano Dal Zilio, Chargé de Recherche, 50% jusqu'à juillet 2007.
- Lucia Acciai, Doctorant, bourse MENSR, début de la thèse en 2004, 100%

• LRI

- Benzaken Véronique, Professeur, 30%
- Bidoit Nicole, Professeur, 30%
- Dario Colazzo, Maître de Conférences, 30%.
- Burelle Marwan, Doctorant, $\frac{1}{2}$ ATER début de la thèse en 2002, 30%
- Miachon Cédric, Doctorant, $\frac{1}{2}$ ATER début de la thèse en 2003, 75%
- Objois Matthieu, Doctorant, bourse MENSR + monitorat, début de la thèse en 2003, 40%
- Kim Nguyen, Doctorant, bourse MENSR + monitorat, début de la thèse en 2004, 100%

• MOSTRARE

- Joachim Niehren, Directeur de Recherche INRIA, 50%
- Sophie Tison, Professeur, 50%
- Anne-Cécile Caron, Maître de conférences, 50%
- Jean-Marc Talbot, Maître de conférences jusque septembre 2006 (nommé professeur au LIF à partir de septembre 2006), 50%
- Iovka Boneva, Doctorante, Docteur depuis le 22 juin 2006 (début de la thèse sept. 2002), 75% jusque septembre 2006.
- Denis Debarbieux, Docteur depuis le 9 décembre 2005 (début de la thèse sept. 2002), 75% jusque septembre 2006.
- Emmanuel Filiot, Docteur depuis le 13 octobre 2008 (début de la thèse sept 2005), 75%
- Olivier Gauwin, Doctorant depuis décembre 2006, 50%
- Mathias Samuelides, Docteur depuis décembre 2007 (LIAFA), ATER à Lille en 2007-2008. 50%

3 Changements significatifs intervenus dans le projet

Partenaires Au niveau des partenaires il n'y a eu qu'un seul changement car l'ENS a été remplacée par l'Université Paris 7 suite à la mutation du responsable du projet, Giuseppe Castagna. Si d'un point de vue de la continuité de la recherche ce changement n'a eu que peu de répercussions, d'un point de vue administratif les conséquences ont été lourdes, tel l'indisponibilité des fonds pour le partenaire pendant 5 mois, une césure et d'importants délais dans l'embauche de l'ingénieur recherche en CDD et l'impossibilité d'obtenir un état précis du budget disponible.

Participants. Concernant les participants, le changement le plus significatif a été le départ vers l'INRIA Roquencourt d'Alain Frisch, le développeur principal de ℂDuce. Alain a continué activement sa collaboration dans le cadre de Tralala, en contribuant à la recherche, en coencadrant Kim Nguyen (doctorant participant à 100% à Tralala), et en participant aux réunions du projet. Toutefois, on ne peut formellement inclure dans Tralala que le travail effectué par Alain en collaboration avec des membres du projet. Ceci pénalise, quoique seulement en apparence, les sous-thématiques "streaming" et "polymorphisme" car même si les objectifs fixés dans ces sous-thèmes ont été atteints, certains l'ont été "extérieurement" au projet (par Alain Frisch).

Un autre changement majeur a été le départ de Silvano Dal Zilio (CR CNRS) du LIF vers le LAAS (Toulouse) ce qui a fortement limité le développement de la thématique sur le site de Marseille. Cela a limité le travail sur les questions liées à l'optimisation de requêtes sur documents XML à l'aide de technique d'automates.

Les autres changements ont été la mutation de Anca Musholl (en septembre 2006 du LIAFA au Labri) qui a néanmoins continué à assurer le travail de recherche pour le projet, l'arrivée de Dario Colazzo (recruté sur un poste MdC LRI), le départ de Virginie Thion (doctorante LRI), le départ de Charles Meysonnier (doctorant LIF), le départ de Denis Debarbieux (doctorant Mostrare), l'arrivée de Lucia Acciai (doctorante LIF) et l'arrivée d'Emmanuel Filiot et Olivier Gauwin (doctorants Mostrare). À cela s'ajoutent les recrutements effectués dans le cadre de Tralala, c'est-à-dire, Cristina Sirangelo (post-doctorante Gemo), Nils Gesbert (ingénieur de recherche CDD, LIENS), Pietro Abate et Karoline Malmkjær (ingénieurs de recherche CDD, PPS), ainsi que la participation active de Daniele Varacca, Maître de Conférences à Paris 7 suite à l'inclusion de Paris 7 dans le consortium.

Thématiques Il n'y a pas eu de changement significatif au niveau des thématiques, mis à part le renforcement de certains axes de recherche, en particulier les aspects liés à la distribution. Ceci concerne les services Web, le développement d'applications Web et l'étude de la concurrence (ayant comme finalité la définition d'une extension concurrente de ℂDuce). Les autres modifications sont essentiellement des recentrages et des modifications de calendrier qui prennent en compte les aspects contingents de la recherche, tels que la non-obtention des mois d'ingénieurs de recherche demandés (ce qui a affecté l'implantation et l'expérimentation de certaines solutions), des recrutements/invitations tardifs (avec des répercussions notamment sur le streaming) ou des départs de personnels.

4 Résumé des principales avancées

4.1 Avancées par thématique

Dans notre projet la recherche s'articule autour de quatre thématiques comme cela a été spécifié en section B2 du descriptif du projet. Nous allons décrire les avancées pour chaque thématique.

4.1.1 Étendre le langage CDuce

Responsable : Giuseppe Castagna — **Publications liées :** [66, 26, 24, 4, 77, 63, 84, 20, 65, 55, 21, 56, 22, 52, 17, 58, 75, 76, 85, 90, 86]

Comme nous l'anticipions dans notre proposition de projet, le langage ℂDuce joue deux rôles distincts dans Tralala, constituant à la fois une base d'expérimentation sur laquelle greffer les résultats obtenus dans le cadre du projet, et un sujet de recherche à part entière. Pour ce qui concerne les expérimentations, celles-ci n'ont pas pour vocation d'être intégrées —au moins dans un premier temps— dans ℂDuce. Il s'agit donc en général, de branches séparées dans le CVS de ℂDuce. C'est ainsi que nous avons implanté deux extensions de ℂDuce par des *crawlers* et par des filtres, deux expériences dans le cadre de la recherche sur la définition de filtrages et d'itérateurs "en profondeur", ainsi que d'un éventuel traitement à la volée des flux XML. Pour ce qui concerne la recherche ciblée à ℂDuce, elle se partage en deux volets l'un plus fondamental, l'autre plus technologique et visant les standards existants. Dans les extensions fondamentales l'une des deux avancées les plus importantes a été l'extension de ℂDuce par le langage de requêtes ℂQL [52, 63, 84], l'implantation de ses optimisations logiques, ainsi que la très récente expérimentation d'interfaces graphiques pour la définition des requêtes [24], ce qui a constitué le travail de thèse de Cédric Miachon. L'autre avancée fondamentale a été la définition d'un langage pour définir des itérateurs qui peuvent être typés de manière hautement polymorphe [66, 26, 77] qui a constitué les travaux de thèse de Kim Nguyen. Le dernier apport est d'importance cruciale car il permet de définir des transformations de documents XML avec un typage extrêmement précis. En effet ces itérateurs enrichissent ℂDuce par une nouvelle forme de polymorphisme très riche et puissante, dont la définition était nécessaire afin de manipuler des documents XML complexes; la définition de cette forme de polymorphisme constituait l'un des enjeux majeurs de cet axe de recherche. Une implantation de ces itérateurs pour ℂDuce est disponible et sera bientôt incluse dans la distribution officielle de ℂDuce. Nous pouvons donc affirmer que l'objectif du polymorphisme des transformations XML a été largement atteint et même dépassé car nous avons aussi exploré des formes plus classiques du polymorphisme pour XML. Notamment Castagna, Frisch et Hosoya ont étudié le polymorphisme paramétrique [58] et au moment de l'écriture de ce rapport, ce travail est poursuivi par Abate, Castagna et Nguyen afin de pouvoir l'étendre aux fonctions d'ordre supérieur et donc à ℂDuce. Nous avons aussi étudié l'"error mining" pour ℂDuce [55] mais cet aspect n'a pas été implanté en ℂDuce par manque de ressources humaines. Concernant les aspects plus technologiques nous voulons signaler le développement d'outils pour l'importation et l'exportation de web-services dans ℂDuce [85], d'un framework pour la gestion de documents XML intégrant du code ℂDuce (à la PHP) pour le développement d'applications web "server-side" [90, 86], l'interfaçage complet avec OCaml, la réécriture de la gestion de la validation XML Schema, ainsi que la préparation et le maintien de la documentation correspondante [75, 76]. L'apport des IRs financés par Tralala a été fondamental pour la maintenance du système et de la documentation de ℂDuce, ainsi que pour le developpement du langage, en particulier pour ce qui concerne la préparation d'un "portage" Windows.

Grâce à cet effort, ℂDuce est désormais disponible pour Mac OSX, Windows Vista/XP, et il est inclus dans des distributions majeures de Linux, notamment, Red-Hat Fedora, Ubuntu, Debian et Mandriva.

4.1.2 Langage de requêtes et optimisation

Responsables : Véronique Benzaken et Ioana Manolescu — **Publications liées :** [79, 2, 52, 18, 15, 43, 73, 99, 38, 50, 40, 65, 51, 32, 48, 6, 59, 36, 82, 31, 61, 70, 33, 34, 29, 28, 41, 27, 87]

Expressivité, complexité Dans [18] nous avons étudié la complexité de deux problèmes centraux pour l'évaluation de requêtes XML: l'évaluation de requêtes XPath et la conformité d'un document XML par rapport à un schéma XML donné. Nous donnons des algorithmes pour ces deux opérations et nous montrons que, asymptotiquement, ils sont optimaux.

La variété de représentation des documents et des requêtes font qu'il est indispensable d'étudier leur expressivité et leur complexité d'évaluation. Dans sa thèse [87], Denis Debarbieux définit des requêtes graphes, généralisation des Tree Pattern Queries. Ces requêtes permettent d'interroger des graphes, et en particulier des documents XML colorés, ensembles d'arbres XML (avec références) qui se partagent des noeuds. Il étudie la complexité d'évaluation et l'expressivité de ce langage, en comparaison avec Core-XPath et un fragment de XQuery adaptés aux documents XML-colorés. Cette étude est menée à la fois selon la structure du document (arbre ou graphe) et la structure de la requête.

Les documents XML étant pour l'essentiel des arbres étiquetés, l'évaluation de requêtes sur un document XML peut être faite de manière efficace à l'aide d'automates d'arbre. Ces automates ont soit un processus parallèle, par exemple la variante "bottom-up", ou *classique*, soit un processus plus séquentiel, on parle alors de *tree walking automata*. Ces derniers sont particulièrement pertinents dans le contexte de XML car ils permettent d'implémenter facilement les requêtes XPath. Ils sont aussi essentiels pour l'évaluation de requêtes à la volée car de telles évaluations correspondent à un parcours *main gauche* du document, un exemple typique de ce que peut facilement faire un automate séquentiel.

Afin de mieux comprendre comment de tels automates peuvent être utilisés dans le cadre de XML il est primordial de mieux comprendre ces automates. C'est ce que nous avons faits dans [15], [43], [36] et [31]. Dans cette série d'articles nous avons étudié les limites du pouvoir d'expression des automates d'arbre cheminant (tree walking automata) et de leurs extensions avec des jetons ou par imbrication. Nous avons aussi mis en évidence le lien entre ces automates et diverses variantes du langage de requête XPath. Nous avons aussi obtenu des résultats de complexité du test du vide de ces automates et donc des problèmes d'inclusion et d'équivalence des langages de requête associés, préliminaires essentiels à la compréhension de l'optimisation de l'évaluation de ces langages de requêtes. Une grosse partie de ces travaux a fait l'objet de la thèse de Mathias Samuelides [82].

Le langage XPath permet de sélectionner des noeuds dans un document XML. De façon plus large, on peut s'intéresser à la sélection de n-uplets de noeuds. Les requêtes n-aires dans les arbres sont définissables par des formules MSO avec n variables libres du premier ordre. Niehren et al. ont étudié dans [61] la représentation de requêtes n-aires et des automates d'arbres pour ces requêtes. Filiot et al. ont également proposé dans [70] une nouvelle classe de requêtes n-aires qui s'expriment sous la forme de composition de requêtes monadiques. Filiot et Tison ont étudié dans [29] le problème de l'indépendance des variables - introduit à l'origine pour les bases de données - dans le contexte des arbres.

Ils montrent comment décider qu'une requête régulière est équivalente à une union de produits cartésiens, ce qui permet de l'évaluer plus efficacement.

La logique monadique du second ordre (MSO) est plus expressive que la logique du premier ordre (FO). Le langage Core-XPath 2.0 navigationnel est connu pour capturer la logique du premier ordre, et le modèle checking y est PSPACE complet. Filiot et al ([70, 33]) ont distingué un fragment de Core XPath 2.0 appelé Polynomial-time Path Language (PPL). Ils montrent que PPL reste FO-complet, bien que le problème d'évaluation de requête soit polynomial (et donc le model checking également).

En général, les fragments de XPath étudiés ne prennent pas en compte les valeurs des données (contenu textuel, valeur des attributs, ...). Filiot, Talbot et Tison ont adapté la logique spatiale TQL (proposée par Cardelli et Ghelli) au contexte des arbres sur un alphabet infini. Le fait de traiter un alphabet infini permet de prendre en compte les valeurs des données, qui sont habituellement ignorées. Ils prouvent dans [34] la décidabilité du problème de la satisfiabilité pour plusieurs fragments expressifs de TQL. Ceci est réalisé en utilisant une nouvelle classe d'automates d'arbres avec tests d'égalités et de différences. Quelques résultats concernant ces automates avec contraintes sont publiés dans [28].

La logique FO^2 étendue avec des comparaisons de données s'est avérée décidable sur les arbres à données de rang non-borné [41], mais la complexité de l'algorithme demande des formalismes de spécification plus simples, ayant des algorithmes plus efficaces. Dans sa thèse, Claire David a étudié un formalisme alternatif à XPath, qui est basé sur des combinaisons booléennes de motifs d'arbres avec données (data tree patterns) [27]. Dans le cas général considéré par [27], la satisfaisabilité s'avère indécidable, et le model-checking est DP-complet. Pour des fragments syntaxiques de data tree patterns, la complexité de la satisfaisabilité se situe entre NExpTime et 2ExpTime.

Optimisation Nous avons étudié le problème de l'optimisation des requêtes XQuery dans le contexte des bases de données natives, sous deux angles dont la dualité est devenue évidente: *l'optimisation de l'accès aux données*, et *les algèbres logiques* pour XQuery. En parallèle, nous avons commencé une étude systématique des performances de systèmes de traitement de requêtes XML.

En l'absence d'un modèle de stockage de référence pour les bases de données XML, le problème de choisir, parmi un ensemble de structures de stockage de données (telles que tables, index, ...) la meilleure manière d'accéder aux données nécessaires pour une requête, a une importance particulière due au coût élevé d'accès aux données. Nous avons proposé *XML Access Modules (XAMs)* [73], un modèle générique décrivant les structures de données persistantes XML à l'aide d'un formalisme algébrique (décrit dans [59]). Afin de faciliter la réécriture des requêtes à l'aide des XAMs, nous avons proposé un algorithme d'extraction de motifs XAM à partir d'un sous-ensemble de XQuery [38], ainsi qu'une approche d'optimisation algébrique de XQuery intégrant les XAMs avec les plans logiques [99]. L'algorithme de réécriture des requêtes XQuery à l'aide des vues matérialisées (XAM) sous des contraintes structurelles de type Dataguide est décrit dans [32]. La totalité de notre approche d'optimisation algébrique d'accès aux données est implantée dans le prototype ULoad [51].

En parallèle, nous nous sommes intéressés à la problématique de la méthodologie de comparaison de performances des systèmes de requêtes XML. Nous avons proposé les *micro-benchmarks* comme outil le plus approprié pour mesurer les performances d'un processeur XQuery [50]. Un micro-benchmark permet de réaliser une mesure pointue sur un point précis de performance. Suivant cette approche, nous avons mesuré les performances de six systèmes de traitement de requêtes XML, sur treize micro-benchmarks XPath et XQuery [48]; nous avons ainsi mis en évidence les multiples aspects qui influent sur les performance d'un

processeur, et validé l'intérêt de micro-benchmarks.

Nous nous sommes également interessés à l'étude de l'optimisation de requêtes XQuery en présence de typage. Ce type de techniques consiste à exploiter tant le type du document que celui de la requête afin de ne charger en mémoire principale que la partie du document strictement nécessaire au calcul du résultat. Nous avons ainsi défini une analyse statique de la requête qui permet d'inférer le "Type Projector" associé. Ce type projector consiste en un élagage du document. Enfin, afin d'établir la pertinence d'une telle optimisation nous avons implanté un prototype et opéré des mesures de performance qui permettent de valider l'approche suivie. Ces résultats ont été publiés dans [40, 65].

4.1.3 Efficacité : traitement à la volée et compression

Responsables : Anca Muscholl et Ioana Manolescu — **Publications liées :** [19, 8, 39, 37, 67, 5, 98, 30]

Streaming Cette sous-thématique a pâti du départ d'Alain Frisch (cf. §3). Dans sa nouvelle affectation Alain a implanté deux différents langages spécifiquement définis pour le traitement à la volée de documents XML, le premier basé sur les travaux de Akihiko Tozawa, le deuxième original. Ce dernier donne d'excellents résultats pour l'utilisation de mémoire au prix d'un overhead d'exécution plus qu'acceptable.

Le traitement à la volée est aussi au centre de la définition et de l'implantation (pour l'instant à niveau de prototypes séparés) des "crawlers" pour ℂDuce (LRI et LIF). Donc, au niveau pratique, cette thématique a dépassé les objectifs préfixés.

Parmi les études théoriques, Cristina Sirangelo et Luc Ségoufin ont travaillé sur la validation et l'évaluation de requêtes en streaming, afin d'améliorer la compréhension de ce qu'il est possible (ou impossible) de faire par un streaming. En particulier, ils ont étudié la quantité de mémoire minimale nécessaire afin de faire de la validation de documents [37] en streaming par rapport à une DTD. Dans le même ordre d'idée, il est intéressant d'identifier les requêtes adaptées au streaming. C'est un des objectifs du travail d'Olivier Gauwin. Il y a (au moins) deux critères possibles. Tout d'abord, le critère de la concurrence bornée : est-ce que la requête a besoin d'une mémoire bornée pour son évaluation en streaming, la borne ne dépendant pas du document interrogé. Le second critère est le délai borné: est-ce que l'on peut décider après un temps borné (le temps est ici compté en nombre d'évènements liés à la lecture du document) qu'un noeud est solution de la requête. Là encore la borne ne doit pas dépendre du document interrogé. Une publication [98] est en cours de soumission sur ce travail.

Enfin, Gauwin et al. se sont intéressés à la définition formelle de la réponse "au plus tôt" d'une requête en streaming. Le document est lu dans l'ordre du document, sans retour en arrière possible, et répondre au plus tôt signifie sortir chaque noeud solution de la requête dès que l'on a suffisamment d'information pour être certain de sa sélection. Dans ce contexte, ils utilisent dans [67] des automates particuliers, appelés Streaming Tree Automata [5], afin de décider au plus tôt si un noeud est solution d'une requête.

Dans un cadre plus général, Kuhlmann et Niehren [30] proposent des logiques et automates pour des arbres dont les noeuds sont totalement ordonnés. La motivation initiale de l'ajout d'un ordre provient de la linguistique, où il représente l'ordre des mots dans une phrase, mais définir un ordre total sur les noeuds d'un arbre a bien sûr du sens lorsque l'on étudie le streaming.

Documents compressés Le travail sur le traitement de requêtes sur des données XML compressées a consisté principalement à finaliser les travaux sur le système préexistant

XQueC. Ainsi, nous avons introduit un modèle détaillé de coût de la compression ainsi qu'un nouveau algorithme de choix de la meilleure méthode de compression [8]. Le modèle de coût prend en compte tous les aspects influencés par le choix d'un certain algorithme de compression à appliquer sur un certain ensemble de valeurs, notamment: le coût de calcul associé à la compression; le facteur de compression; et l'ensemble des opérations qui peuvent être appliquées directement sur les valeurs compressées (donc, directement sur les données stockées). Ce dernier aspect est surtout intéressant par rapport à un jeu de requête données, qui peut comporter différentes comparaisons sur les valeurs.

A partir de ce modèle de coût, nous avons montré qu'un choix optimal des algorithmes de compression ne peut pas être fait en pratique, car il faudrait connaître d'avance les facteurs de compression atteints par tous les algorithmes sur tout sous-ensemble des données. Nous avons en échange proposé un ensemble d'algorithmes heuristiques qui, partant d'un jeu de données connu, d'un jeu de requêtes (optionnel), et d'un ensemble d'algorithmes de compression disponibles, recommandent des algorithmes à utiliser. Nous avons montré que dans plusieurs cas concrets, ces algorithmes heuristiques font des choix assez proches de l'optimum, tout en gardant des performances acceptables.

Des développements intéressants issus de XQueC portent sur l'analyse statique des requêtes [19, 39, 2]. En effet, le système de stockage de XQueC s'appuie sur un résumé de données, ou *dataguide*, qui facilite l'identification des structures de stockage pertinentes pour une requête donnée. Nous avons identifié et résolu des nouveaux problèmes d'optimisation statique intéressants, s'appuyant sur le dataguide comme un ensemble de contraintes d'intégrité [2]. Il s'agit notamment du problème de minimisation de requêtes en présence des contraintes de dataguides.

4.1.4 Contraintes et Typage de Documents

Responsable : Anne-Cécile Caron — **Publications liées :** [77, 66, 56, 45, 44, 68, 25, 96, 40, 41, 65, 42, 58, 72, 13, 47, 97, 54, 62, 53, 83, 7, 69, 64, 9, 95]

Typage et optimisation Les données semi-structurées sont modélisées parfois par des arbres, parfois par des graphes orientés dont les arcs sont étiquetés. Dans les deux cas, la définition d'un schéma sous la forme de contraintes et de types peut être très utile lors de l'analyse et l'optimisation de requêtes ou de transformations.

La spécification de XML par le W3C offre la possibilité de typer les documents XML au moyen d'un formalisme appelé *XML schema* qui peut être utilisé pour faire de l'analyse statique de transformations de documents. Supposons qu'on possède un document spécifié par un schéma XML x et qu'on veuille le transformer afin qu'il vérifie le schéma XML y. Est-ce que le programme q faisant cette transformation est correct? Par exemple est-ce que pour tout t vérifiant x, $q(t)$ vérifie y? Ce problème est réputé difficile en présence de contraintes d'intégrité.

Nous avons étudié un cas très particulier de ce problème: étant donné un schéma XML x, est-ce qu'il existe un arbre t vérifiant la spécification x? Il se trouve que ce problème est indécidable en général. Dans [41] et [42] nous avons exhibé un formalisme basé sur des logiques et des automates manipulant des alphabets infinis (qui nous servent à coder les valeurs de données) afin de donner des conditions suffisantes et robustes sur le langage de spécification garantissant la décidabilité du vide.

XML schema n'est pas la seule manière de typer un document XML. Le langage ℂDuce possède l'algèbre de types la plus riche parmi les langages de transformation pour XML. Pour ce qui concerne l'ajout du polymorphisme à ℂDuce, nous avons commencé l'étude

du polymorphisme paramétrique [58] et nous sommes en train d'étudier son extension aux fonctions d'ordre supérieur (et donc à la version complète de $\mathbb{C}$Duce). Mais nous avons préféré recentrer notre recherche sur le polymorphisme des itérateurs, car il nous semble que ce dernier puisse avoir un impact bien plus important sur les transformations XML. Ceci parce que les transformations typiques de documents XML demandent en général deux ou trois itérations sur le document et rarement doivent faire appel à du backtraquage. En se concentrant sur un formalisme plus simple, les itérateurs, car non-Turing complets, plutôt que sur les fonctions, nous avons pu capturer une nouvelle forme de polymorphisme dont la précison n'était pas à la portée du polymorphisme existant, qu'il soit paramétrique, de sous-typage ou ad hoc [66]. Cette recherche constitue le travail de thèse de Kim Nguyen [77]. Alain Frisch a aussi défini et implanté l'utilisation des types et patterns $\mathbb{C}$Duce sur un langage hôte polymorphe (OCaml), même si formellement cette contribution n'a pas été développée dans une équipe de Tralala (cf. §3).

Indépendamment du modèle utilisé pour représenter les données semi-structurées, il existe un lien entre la notion de type et celle de *guide*. Lorsqu'on modélise les données semi-structurées par des graphes, le guide d'une donnée D est un graphe dont les noeuds sont les types de D, et les transitions modélisent les chemins de D. Un guide permet de décrire le contenu d'une donnée dont on ne connaît pas le schéma, permettant ainsi à l'utilisateur de formuler des requêtes. Il permet aussi d'améliorer le stockage et d'optimiser l'évaluation des requêtes. En effet, indexer une donnée consiste à construire un guide à partir duquel on peut évaluer une requête sans consulter la donnée. Dans [72], nous avons étudié les index qui préservent les contraintes d'inclusion. Ainsi, les optimisations utilisant les contraintes d'inclusion sont valables, que la requête soit exécutée sur la donnée ou directement sur l'index.

En sus d'étudier le typage des transformations de documents XML, nous avons ouvert un nouvel axe de recherche dans Tralala qui vise à formaliser et étudier le typage des interactions et échanges de documents XML. Daniele Varacca et Giuseppe Castagna en collaboration avec Rocco De Nicola ont commencé à étudier les aspects de la théorie de la concurrence liés au typage de $\mathbb{C}$Duce, ce qui a conduit à la définition de $\mathbb{C}\pi$, une extension du π-calcul par le sous-typage sémantique [56]. Cette étude a été poursuivie en collaboration avec Mariangiola Dezani par la définition d'une correspondance formelle entre $\mathbb{C}$Duce et $\mathbb{C}\pi$ [45]. Ceci est un pas préliminaire pour une extension concurrente de $\mathbb{C}$Duce et son intégration avec des mécanismes d'orchestration de Web-services, de façon que ce dernier puisse anticiper les évolutions futures de la programmation Web. En effet, une tendance forte dans l'industrie pour la conception de systèmes logiciels complexes est basée sur la notion de "services web", qui suggère des scenarii futurs d'utilisation dans lesquels un "web-architecte" créera de nouveaux services par assemblage des services pré-existants disponibles sur le Web et qu'il aura découvert par une recherche interactive. Pour qu'une telle recherche et un tel assemblage soient possibles, il faut que le comportement des services soit spécifié de manière formelle par des "contrats". C'est pourquoi nous avons développé une théorie pour la spécification de services web, qui vise à définir de manière formelle la compatibilité entre clients et services et la substituabilité et évolution de services. Ceci a été commencé en collaboration avec Samuele Carpineti, Cosimo Laneve et Luca Padovani [44] ensuite avec la participation de Nils Gesbert [68, 25] et actuellement se poursuit en collaboration avec l'Universié de Turin et celle d'Urbino [96]. Ceci est un premier pas vers la définition des langages d'assemblage de composants qui permettront non seulement de décrire la coordination des différents services mais aussi de prouver formellement et statiquement des propriétés satisfaites par ces assemblages.

Séquences ordonnées et non-ordonnées Un des objectifs de la thématique "contraintes et typage de documents" concerne la vérification (efficace) de contraintes d'intégrité plus fines que celles des langages traditionnels. Nous nous sommes en particulier intéressés aux contraintes d'intégrité satisfaites indépendamment de l'ordre d'apparition des champs dans un document. Le choix de ce type de contraintes est naturel lorsqu'on travaille avec des données obtenues à partir de la fusion de bases de données relationnelles (un cas important de documents de grande taille), puisque l'ordre des champs est alors sans intérêt. Ces contraintes apparaissent également dans l'étude de XML Schema, qui possède un opérateur de composition d'éléments XML indépendant de l'ordre.

Nous avons montré dans des travaux précédents (voir par exemple [13]) que l'étude de séquences "non-ordonnées" d'éléments XML pouvait se réduire à l'utilisation de contraintes arithmétiques sur le nombre d'occurrences des éléments. Ce type de contraintes se retrouve couramment dans l'utilisation du format XML pour l'agrégation de données. Malheureusement, les classes d'automates d'arbres utilisés dans le cadre de la manipulation de documents XML ne sont pas totalement satisfaisantes en présence de ce type de contraintes. En effet, il faut pouvoir raisonner sur des arbres non ordonnés et de degrés non bornés. Dans ce projet, parmi d'autres modèles d'automates, nous avons travaillé sur une nouvelle classe d'automates d'arbres développée par l'équipe du LIF qui permet de traiter ce cas précis.

Les derniers développements de ces travaux nous ont amenés à étudier la logique modale PML [47, 97] (pour Presburger Modal Logic), qui étend une série d'approches logiques dans lesquelles les modalités peuvent être contraintes par des expressions arithmétiques (e.g. les logiques modales graduées ou les logiques de descriptions.) Nous avons montré que le problème de la satisfaisabilité dans PML est PSPACE-complet, ce qui donne une logique avec une meilleure complexité que la logique SSMH et améliore le résultat sur la logique SL. Des extensions avec contraintes arithmétiques de la logique PML sont montrées indécidables, ce qui permet d'avoir une meilleure appréciation des rapports entre ces différentes logiques.

Dans le cadre de la thèse de Iovka Boneva [83], nous avons étudié la logique spatiale présente dans le langage TQL, proposé par Cardelli et Ghelli pour les arbres non-ordonnés d'arité non-bornée, nous intéressant aux problèmes de décision (model-checking, satisfiabilité) ainsi qu'à l'expressivité de cette logique. Nous avons établi des résultats précis de complexité pour le problème de model-checking. De plus, nous avons montré que le problème de satisfiabilité était indécidable et ce, pour divers fragments de la logique [54]. Ceci nous a amenés à définir des fragments de cette logique spatiale pour lesquels la satisfiabilité serait décidable et à comparer celle-ci avec d'autres logiques permettant d'exprimer des propriétés sur les arbres non-ordonnés d'arité non-bornée. Nous avons identifié des fragments de la logique pour laquelle le problème de satisfiabilité est décidable et avons relié dans [54] ces fragments respectivement à la logique monadique du second-ordre (MSO) et à la logique monadique du second-ordre de Presburger (PMSO) de Seidl *et al.* Nous avons aussi comparé l'expressivité de MSO, de PMSO et CMSO et proposé un cadre uniforme pour ces logiques en termes d'automates d'arbres [53]. Dans ce dernier travail, nous avons aussi relié l'expressivité de la logique PMSO et les automates d'arbres modulo AC d'Ohsaki, automates dont nous avons étudié une version généralisée dans [62].

Contraintes de référence Les langages permettant de définir le schéma d'un document XML (par exemple les DTDs) n'offrent pas de mécanismes directes pour spécifier les références entre éléments et tout particulièrement pour préciser le type des sous-documents référencés. A l'évidence, ceci contraste avec une fréquente utilisation de références dans les documents XML ou semi-structurés. Ce volet de notre travail a consisté à proposer une

extension de la notion de DTD permettant de donner aux références un statut de "citoyens de première classe". Ainsi un ref-schema intègre dans la spécification d'un schéma la déclaration des références comme tout autre lien entre éléments. Un ref-schema est défini par une grammaire d'expressions régulières et un document valide pour un ref-schema donné est un graphe. Dans une étape suivante, l'objectif a été de disposer d'un formalisme de représentation et de raisonnement (plus précisément une logique) unique permettant de décrire schémas, contraintes et requêtes. Les logiques modales sont des formalismes simples dont l'adéquation à exprimer des propriétés de graphes est bien connue. Nous avons choisi d'étudier les problèmes d'expression de schémas et de contraintes ainsi que de satisfiabilité de contraintes en présence de schéma en utilisant la logique modale hybride (HML). Intuitivement, HML étend la logique modale par l'ajout de mécanismes permettant d'identifier en les nommant les noeuds d'un graphe. Ces mécanismes se révèlent nécessaires par exemple pour énoncer des propriétés de réflexivité, de symétrie, et ainsi de suite. Les résultats obtenus sont les suivants. (1) Nous avons montré de manière constructive et modulaire que tout ref-schema normalisé est "équivalent" à une formule HML. (2) Ce résultat a été étendu dans le cas général aux ref-schemas en s'appuyant sur la traduction précédente. (3) Nous avons aussi proposé un système de preuve par tableau préfixé pour la satisfiabilité de contraintes en présence de ref-schéma. [69, 64, 9, 95]

4.2 Résultats attendus vs. résultats obtenus

Dans les tableaux qui suivent nous indiquons l'état d'avancement des résultats attendus tels que nous les avions spécifiés dans la section B3 du programme détaillé

Première année			
Résultat attendu	Durée déb fin	Avancement	Rapports
développement CDuce, version distribuable, documentation	0–8	**Complété**	[92, 75, 76]
étude comparative des langages XDuce, XQuery et CDuce à partir du banc de test proposé par le W3C pour XML-Query	6–12	**Complété**	[52, 50, 48]
modules d'accès en mémoire pour documents XML, définition d'algorithmes d'optimisation physique	0–12	**Objectifs complétés** [mais la recherche continue]	[73, 59, 38, 32, 51]
étude du polymorphisme et de l'algèbre de filtrage pour CDuce	0–...	**Objectifs complétés** [mais la recherche continue]	[58, 89, 77, 26, 66]
noyaux de langages de requêtes à greffer à CDuce	0–12	**Complété implanté**	[52, 76, 75]
mise au point d'un système de types pour le/les langages de requête envisagés	0–12	**Complété**	[52, 40]
état de l'art sur l'évaluation à la volée de documents XML, avec ou sans compression, et définition de nouvelles stratégies d'évaluation dans ce cadre	0–12	**Complété**	[37]
étude de web-services pour CDuce	...–...	**Complété prototypé**	[85]

Deuxième année				
Résultat attendu	Durée déb	Durée fin	Avancement	Rapports
implantation du, ou des, langages de requêtes et des outils du typage	12	24	**Complété**	[52]
définition et implantation d'un optimiseur générique	12	36	**Complété**	[79, 51, 73, 38, 32]
expérimentation avec des exemples de grands volumes de données	18	24	**Pas encore complété** [en retard sur le programme prevu]	[93]
étude des aspects liés à la préservation statique de contraintes d'intégrité: il s'agit de proposer des analyses correctes, garantissant la validité d'une transformation respectivement à une contrainte d'intégrité donnée.	12	24	**Complété**	[27, 41, 42, 69, 64, 95]
mise au point d'algorithmes pour l'évaluation à la volée de documents XML, avec ou sans compression, et caractérisation (logique) de classes de requêtes évaluable avec ces méthodes	12	30	**Pas encore complété** [en retard sur le programme prevu]	[15, 43, 31, 36, 82].

Troisième année				
Résultat attendu	Durée déb	Durée fin	Avancement	Rapports
fin de l'implantation du langage de requête avec ajout d'optimisations qui dépendent de contraintes (statiques) d'intégrité	24	36	Objectifs partiellement atteints	[63, 84, 52, 24]
implantation des méthodes d'interrogation à la volée de données compressées.	30	36	Objectifs non atteints [l'étude de l'interrogation de données compressées s'est arretée à la modalité *batch*]	
optimisation à base de résumé structurel pour les requêtes XML (avec ou sans compression)	24	36	**Complété**	[39, 2, 8, 94]
Evaluation de requêtes à la volée, délai et mémoire bornés	24	36	Objectifs partiellement atteints	[5, 67, 98]

Le projet a enfin obtenu des résultats non prévus au moment de la rédaction du projet.

Nouveaux résultats non attendus		
Résultat	Avancement	Rapports
possibilité d'intégration de code CDuce dans des pages XHTML et interfaçage avec le serveur web (à la PHP)	version beta implanté et distribuée	[90, 86]
interfaces graphiques pour la programmation de requêtes	premier prototype en phase de test, mais pas encore distribué	[24]
itérateurs polymorphes pour XML	prototype distribué et en cours d'intégration dans la distribution de ℂDuce	[66, 26]
interfaces graphiques interactives pour l'extraction d'information à l'aide d'un processus d'apprentissage par raffinement.	prototype distribué	[91]
optimisation de requêtes XML par l'utilisation d'élagages basés sur le typage	prototype distribué	[40, 46, 12]
micro-benchmark pour XQuery et mesures de performance	prototype distribué	[50, 48, 6]
théorie des contrats pour les services web	bases théoriques	[96, 68, 44, 3, 45, 25]

Formation Enfin nous tenons à souligner la contribution du projet Tralala à la formation de jeunes chercheurs car, en sus des nombreux stagiaires, **neuf thèses de doctorat** sont issues de ce projet: Acciai, Arion, Boneva, David (soutenance prevue début 2009), Debarbieux, Nguyen, Miachon, Objois, Samuelides. A ces thèses il faut ajouter la HDR de Jean-Marc Talbot.

5 Réalisations obtenues dans le cadre du projet

5.1 Logiciels.

Dans le cadre du projet nous avons développé un nombre relativement important de logiciels, dont la qualité varie du simple prototype à l'outil de démonstration technologique, jusqu'au logiciel fini utilisé en production par des utilisateurs industriels. Voici une liste des réalisations.

CDuce CDuce a été étendu pour permettre un interfaçage complet et flexible avec OCaml. L'utilisation de XML Schema et la validation a été réécrite et améliorée. Il a été enrichi par un sous-langage de requêtes et la prochaine release majeure (0.6.0) permettra la définition d'itérateurs polymorphes grâce à l'introduction des filtres. CDuce est désormais distribué en format source et binaire pour plusieurs plates-formes (OSX, Windows et Linux) et inclus dans des distributions majeurs de Linux. La documentation à été enrichie et elle est désormais disponible aussi en version cartacée. Nombreuses améliorations et corrections ont été apportées au typage et au moteur d'exécution.

ECDuce Nous avons commencé à étudier l'utilisation de CDuce pour le développement d'applications web côté serveur. La solution retenue a été celle de permettre l'utilisation de commandes CDuce à l'intérieur de XHTML (à la PHP). Pour cela nous avons développé deux applications : la première transforme une page XML contenant du code CDuce en un programme CDuce qui produit la même page XML où le code a été remplacé par le résultat de son exécution; le deuxième est un script CGI qui appelle la première fonction et exécute le code résultant. Ces applications, actuellement en version alpha, ont été utilisées pour développer un Wiki, lui même utilisé pour décrire la documentation de ECDuce. Le tout est disponible sur le Web [90].

CDuce-Crawlers Cette extension de CDuce est une implantation des *crawlers* ou patterns avec accumulateurs. Le prototype permet à l'utilisateur de définir de tels patterns et de s'en servir pour extraire des sous-parties d'un document XML. On peut en particulier simuler la partie descendante du standard XPath grâce à ces crawlers. Le typage de tels termes est donné par un algorithme d'approximation. Cette extension demande cependant encore un travail d'intégration avec le langage pour être pleinement utilisable et donc distribuée.

CQL Nous avons défini CQL, un langage de requêtes pour CDuce et étudié son optimisation. CQL est inclus dans CDuce depuis la version 0.3.

Patterns by examples Nous avons étudié comment définir des interfaces pour permettre une programmation complètement graphique en CQL. Un premier prototype de test pour une évaluation interne a été développé en GTK, mais pas distribué.

Web-services Nous avons étudié la possibilité d'écrire des services web en CDuce. Le résultat de cette étude est la réalisation de deux programmes pour l'importation et l'exportation de services web écrits en CDuce. Pour l'importation, le programme prend en entrée une description de service web sous le format standard WSDL et génère le code CDuce nécessaire au programmeur qui souhaite appeler le service distant depuis CDuce. Pour l'exportation, le programme prend le code CDuce à exporter ainsi qu'une description WSDL fournie par le programmeur et génère le code nécessaire à la publication sur un serveur web du programme original en tant que service web.

Type-projectors Nous avons implanté un prototype pour tester les types projectors. Il consiste en un programme indépendant, prenant en entrée un document, une requête XPath et une DTD et fournissant en sortie le document projeté, c'est à dire une sous-partie du document nécessaire au traitement de la requête par un moteur XQuery externe tel que Galax.

Filters Nous avons défini et implanté un langage de combinateurs, nommés *filtres*. Les filtres sont suffisamment expressifs pour écrire des transformations complexes de documents XML tout en gardant une discipline de typage simple. Les filtres sont intégrés à ℂDuce, et seront inclus dans sa distribution dès la prochaine release. L'intérêt des filtres est que le programmeur n'est pas limité à un langage restreint pour toutes les tâches "non-XML" et peut par exemple bénéficier des fonctions système et autres facilités fournies par ℂDuce et plus généralement par tout vrai langage de programmation. Le programmeur peut donc définir ses propres itérateurs polymorphes et bénéficier d'un typage précis, tout en restant dans le cadre d'un langage (hôte) complet.

ULoad est un prototype permettant aux utilisateurs de déclarer des vues matérialisées (XAMs) sur des documents XML, de charger le contenu de ces vues dans un stockage XML (qui peut être aussi bien basé sur une base de données relationnelle qu'un stockage natif), et d'exploiter ces vues afin de répondre à des requêtes exprimées dans un sous-langage de XQuery. ULoad réécrit des requêtes sous un ensemble de contraintes structurelles sur les documents XML ciblés par la requête, et utilise ces contraintes pour signaler soit l'insatisfiabilité d'une requête qui viole les contraintes, soit l'absence de vues matérialisées convenables pour réécrire la requête.

XSum est un module d'extraction et d'exploitation de contraintes structurelles sur des documents XML (gemo.futurs.inria.fr/software/SUMMARY); XSum est issu du travail sur la compression de données, en particulier du modèle d'organisation de données dans le système de gestion de données XML compressées XQueC. La fonctionnalité la plus intéressante de XSum est la capacité d'associer de l'information structurelle à des motifs d'arbres représentant soit des requêtes, soit des vues matérialisées (XAMs).

Squirrel Squirrel est un outil interactif d'extraction d'information sur le Web, qui implémente un algorithme d'apprentissage interactif de requêtes monadiques représentées par des automates d'arbres [11]. Un des problèmes fondamentaux résolu pour Squirrel a été de trouver une bonne notion d'automates pour les arbres d'arité non-bornée, tel que le théorème de Myhill-Nerode pour les automates d'arbres standard se généralise, et tel que les automates déterministes minimaux soient uniques pour tout langage régulier d'arbres, et les plus petits possible [14, 60]. Squirrel possède une interface Web, qui peut-être téléchargée et intégrée au navigateur Mozilla-Firefox [91]. Partant d'un document HTML exemple, l'utilisateur doit annoter quelques éléments qu'il veut voir extraire. Par exemple, sur une page de vente aux enchères, l'utilisateur sélectionne quelques noms de produits et leur prix. Le système génère alors un *Wrapper* (outil d'extraction d'information) et colorie sur la page Web les noeuds que ce wrapper a extraits. L'utilisateur peut corriger la proposition du système en donnant d'autres exemples d'éléments à extraire ou en désélectionnant des éléments extraits par le système. Le système propose alors un nouveau wrapper, et ce processus continue jusqu'à ce que l'utilisateur soit satisfait. En pratique, le nombre d'exemples à annoter est très petit et l'apprentissage du wrapper nécessite peu d'interactions. Ceci a été confirmé par des expériences.

6 Réunions et Conférences organisées dans le cadre du projet

Nous avons effectué huit réunions plénières dont quatre étalées sur deux jours. A ces réunions nous avons invité des chercheurs extérieurs au projet (notamment des étrangers) et, lorsque c'était possible, des représentants d'ACI de thèmes voisins de notre projet.

1. 3 novembre 2004 Paris.
2. 10-11 mars 2005 Marseille.
3. 7-8 juillet 2005 Lille.
4. 26-27 janvier 2006 Paris.
5. 23-24 mai 2006 Marseille.

6. 21 janvier 2007 Nice.

7. 22 février 2008 Paris.

Le programme détaillé de chaque journée ainsi que les transparents de nombreuses présentations sont disponibles en ligne sur le site de Tralala: `http://www.cduce.org/tralala_reunions.html`.

En plus de ces réunions plénières il y a eu des réunions bi/tri-latérales et des échanges de chercheurs. Outre les échanges et visites entre équipes parisiennes, on peut citer:

- 19-20 avril 2005 à Paris. Réunion restreinte LRI, ENS, Marseille. Thèmes abordés: le langage Reduce, les itérateurs pour XML et leur typage.
- 9 décembre 2005. Visite des membres du LRI et de l'en à Lille. Thèmes abordés: algèbres de processus, données sémi-structurées, automates d'arbres.
- 16 février 2006 à Lille. Réunion de travail LIFL, Gemo. Thèmes abordés: automates d'arbres, transducers et requêtes n-aires.
- 24 mars 2006. Visite d'Alain Frisch à Lille. Thèmes abordés: automates d'arbres, transducers et typage.
- 17-21 mai 2006 à Marseille. Groupe de travail LRI Marseille. Thèmes abordés: Les Crawlers dans CDuce.
- 22 Juin à Lille 2006. Réunion de travail LIFL, Gemo. Thèmes abordés: automates d'arbres, transducers et requêtes n-aires.

7 Soutiens obtenus en liaison avec ce projet

7.1 Postes chercheurs

Nous avons obtenu de la part de l'INRIA le financement d'un post-doc pour une durée de 1 an. Nous avons recruté Cristina Sirangelo, qui a soutenu sa thèse à l'université de Calabre en Italie en 2005, sur ce financement depuis le 1er Décembre 2005.

7.2 Postes ingénieurs

Le financement Tralala pour recruter 18 mois de poste d'ingénieur a été utilisé pour embaucher par un CDD Pietro Abate (05/2007-07/2008) qui s'est occupé de la maintenance du langages ℂDuce et a collaboré à l'étude du polymorphisme paramétrique, Nils Gesbert (04/2006-09/2006) qui a travaillé sur les services Web et Karoline Malmkjær (11/2007-01/2008), qui a effectué le portage de ℂDuce sur Windows.

7.3 Contrats nationaux

- ACI "Sécurité", Projet Casc (LIENS, LRI)
- ANR Blanc ENUM (2008-2010). Collaboration entre Lille, Paris VII, Caen, Marseille.

7.4 Contrats européens

EGIDE PAI Procope et Polonium (2006-2008) "Vérification et requêtes en présence de données" (GEMO)

7.5 Contrats internationaux hors CEE

néant

7.6 Contrats industriels

- ANR RNTL ATASH (2006-2009), collaboration Mostrare, LIP6, et XRCE (Xerox Research Center Europe).
- ANR RNTL WebContent (2006-2009)

7.7 Contacts internationaux financés dans le cadre de ce projet

- Mikolaj Bojanczyk, University of Warsaw, Poland.
- Michele Boreale, Dipartimento di Sistemi e Informatica, Università di Firenze
- Mariangiola Dezani, Dipartimento d'Informatica, Università degli Studi di Torino, Italie.
- Haruo Hosoya, Computer Science Department, University of Tokyo, Japon
- Cosimo Laneve, Dipartimento d'Informatica, Università degli Studi di Bologna, Italie.
- Sebastian Maneth, Kensington Research Lab, Sydney, Australie
- Wim Martens, Hasselt University, Belgique
- Hitoshi Ohsaki, NAIST, Amagasaki, Japon
- Luca Padovani, ISTI, Universitá di Urbino, Italie.
- Thomas Schwentick, Universität Dortmund, Allemagne.
- Helmut Seidl, TU Munich, Allemagne.

8 Publications obtenues dans le cadre du projet

Nous ne reportons ici que les articles issus du projet. Parmi ceux-ci, les articles [66, 26, 24, 63, 84, 4, 1, 40, 65, 52, 55, 40, 51, 32, 48, 6, 26, 66, 38] ont été réalisés par des auteurs qui proviennent d'au moins deux équipes différentes du projet, tandis que [62, 2, 19, 39, 96, 25, 68, 1, 45, 44, 58, 14, 10, 41, 42, 43, 71] ont été réalisés avec des chercheurs extérieurs invités dans le cadre du projet.

Oeuvres de vulgarisation scientifique

[1] V. Benzaken, G. Castagna, H. Hosoya, B.C. Pierce, and S. Vansummeren. *The Encyclopedia of Database Systems*, chapter "XML Typechecking". Springer, 2008. To appear.

Articles dans des revues internationales avec comité de lecture

[2] Andrei Arion, Angela Bonifati, Ioana Manolescu, and Andrea Pugliese. Path summaries and path partitioning in modern XML databases. *WWW Journal*, 11(1):117–151, 2008.

[3] G. Castagna, R. De Nicola, and D. Varacca. Semantic subtyping for the π-calculus. *Theoretical Computer Science*, 398(1-3):217–242, 2008. Essays in honour of Mario Coppo, Mariangiola Dezani-Ciancaglini and Simona Ronchi della Rocca.

[4] A. Frisch, G. Castagna, and V. Benzaken. Semantic subtyping: dealing set-theoretically with function, union, intersection, and negation types. *Journal of the ACM*, 55(4):1–64, 2008.

[5] Olivier Gauwin, Joachim Niehren, and Yves Roos. Streaming tree automata. *Information Processing Letters*, 2008. To appear.

[6] P. Michiels, I. Manolescu, and C. Miachon. Toward microbenchmarking XQuery. *Elsevier Journal on Information Systems*, 33(3), 2008.

[7] Yves André, Anne-Cécile Caron, Denis Debarbieux, Yves Roos, and Sophie Tison. Path constraints in semi-structured data. *Theoretical Computer Science*, 385(1-3):11–33, 2007.

[8] Andrei Arion, Angela Bonifati, Ioana Manolescu, and Andrea Pugliese. XQueC: A query-conscious compressed XML database. *ACM Transactions on Internet Technologies (TOIT)*, 7, 2007.

[9] Nicole Bidoit and Dario Colazzo. Testing XML constraint satisfiability. *Electronic Notes in Theoretical Computer Science*, 174(6):45–61, 2007. Version complète de [69].

[10] Wim Martens and Joachim Niehren. On the minimization of XML schemas and tree automata for unranked trees. *Journal of Computer and System Sciences*, 73(4):550–583, 2007.

[11] Julien Carme, Rémi Gilleron, Aurélien Lemay, and Joachim Niehren. Interactive learning of node selecting tree transducer. *Machine Learning*, 2006.

[12] Dario Colazzo, Giorgio Ghelli, Paolo Manghi, and Carlo Sartiani. Static analysis for path correctness of XML queries. In *Journal of Functional Programming*, volume 16, pages 621–661, 2006.

[13] Denis Lugiez and Silvano Dal Zilio. XML schema, tree logic and sheaves automata. *Applicable Algebra in Engineering, Communication and Computing (AAECC)*, 17(5):337–377, 2006.

[14] Wim Martens and Joachim Niehren. On the minimization of XML schemas and tree automata for unranked trees. *Journal of Computer and System Science*, 2006.

[15] Anca Muscholl, Mathias Samuelides, and Luc Segoufin. Complementing deterministic tree-walking automata. *Information Processing Letters*, 2006.

[16] Joachim Niehren, Jan Schwinghammer, and Gert Smolka. A concurrent lambda calculus with futures. *Theoretical Computer Science*, 364(3):338–356, November 2006.

[17] G. Castagna, J. Vitek, and F. Zappa Nardelli. The Seal Calculus. *Information and Computation*, 201(1):1–54, 2005.

[18] Georg Gottlob, Christoph Koch, Reinhard Pichler, and Luc Segoufin. The parallel complexity of XML typing and XPath query evaluation. *Journal of the ACM*, 52(2):284–335, 2005.

Articles dans des revues nationales avec comité de lecture

[19] Andrei Arion, Angela Bonifati, Ioana Manolescu, and Andrea Pugliese. Un modèle de stockage XML basé sur les séquences. *Ingénierie des Systèmes d'Information*, 10(2), 2005.

Articles invités

[20] G. Castagna. CDuce, an XML processing programming language: From theory to practice. In *Proc. of SBLP 2007, XI Brazilian Symposium on Programming Languages*, pages 3–4. SBC - Brazilian Computer Society, 2007.

[21] G. Castagna. Patterns and types for querying XML. In Proceedings of *DBPL 2005, 10th International Symposyum on Database Programming Languages* Lecture Notes in Computer Science, n.3774 Springer (full version) *and XSym 2005, 3rd International XML Database Symposium* Lecture Notes in Computer Science n.3671:1-3, Springer (summary), 2005. Joint invited talk.

[22] G. Castagna. Semantic subtyping: challenges, perspectives, and open problems. In *ICTCS 2005, Italian Conference on Theoretical Computer Science*, number 3701 in Lecture Notes in Computer Science, pages 1–20. Springer, 2005.

[23] G. Castagna and A. Frisch. A gentle introduction to semantic subtyping. In Proceedings of *PPDP '05, the 7th ACM SIGPLAN International Symposium on Principles and Practice of Declarative Programming,* pages 198-208, ACM Press (full version) and *ICALP '05, 32nd International Colloquium on Automata, Languages and Programming,* Lecture Notes in Computer Science n. 3580, Springer (summary), Lisboa, Portugal, 2005. Joint ICALP-PPDP keynote talk.

Communications dans les actes de conférences internationales avec comité de lecture

[24] V. Benzaken, G. Castagna, D. Colazzo, and C. Miachon. Pattern by Example: type-driven visual programming of XML queries. In *PPDP '08: 10th international ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming*, pages 131–142. ACM, 2008.

[25] G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. In *POPL '08, 35th ACM Symposium on Principles of Programming Languages*, pages 261–272, January 2008.

[26] G. Castagna and K. Nguyen. Typed iterators for XML. In *ICFP '08: 13th ACM-SIGPLAN International Conference on Functional Programming*, April 2008.

[27] Claire David. Complexity of data tree patterns over XML documents. In *MFCS*, pages 278–289, 2008.

[28] Emmanuel Filiot, Jean-Marc Talbot, and Sophie Tison. Tree automata with global constraints. In *Developments in Language Theory (DLT 2008)*, Lecture Notes in Computer Science. Springer Verlag, 2008.

[29] Emmanuel Filiot and Sophie Tison. Regular n-ary queries in trees and variable independence. In *5th IFIP International Conference on Theoretical Computer Science (IFIP TCS 2008)*. Springer Verlag, 2008. To appear.

[30] Marco Kuhlmann and Joachim Niehren. Logics and automata for totally ordered trees. In *19th International Conference on Rewriting Techniques and Applications (RTA 2008)*, Lecture Notes in Computer Science. Springer Verlag, July 2008.

[31] Balder ten Cate and Luc Segoufin. Xpath, transitive closure logic, and nested tree walking automata. In *PODS*, 2008.

[32] Andrei Arion, Véronique Benzaken, Ioana Manolescu, and Yannis Papakonstantinou. Structured materialized views for XML queries. In *33rd International Conference on Very Large Databases VLDB*, pages 87–98, Vienna, Austria, 2007.

[33] Emmanuel Filiot, Joachim Niehren, Jean-Marc Talbot, and Sophie Tison. Polynomial time fragments of XPath with variables. In *26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2007)*, pages 205–214. ACM-Press, 2007.

[34] Emmanuel Filiot, Jean-Marc Talbot, and Sophie Tison. Satisfiability of a spatial logic with tree variables. In *16th EACSL Annual Conference on Computer Science and Logic (CSL 2007)*, volume 4646 of *Lecture Notes in Computer Science*, pages 130–145. Springer Verlag, 2007.

[35] Joachim Niehren, David Sabel, Manfred Schmidt-SchauSS, and Jan Schwinghammer. Observational semantics for a concurrent lambda calculus with reference cells and futures. In *23rd Conference on Mathematical Foundations of Programming Semantics (MFPS 2007)*, volume 173 of *Electronical notes in theoretical computer science*, pages 313–337. Elsevier, April 2007.

[36] Mathias Samuelides and Luc Segoufin. Complexity of pebble tree-walking automata. In *FCT*, 2007.

[37] Luc Segoufin and Cristina Sirangelo. Constant-memory validation of streaming xml documents against dtds. In *ICDT*, 2007.

[38] Andrei Arion, Véronique Benzaken, Ioana Manolescu, Yannis Papakonstantinou, and Ravi Vijay. Algebra-based tree pattern identification in XQuery. In *Flexible Query Answering Systems (FQAS)*, Milano (Italy), June 2006.

[39] Andrei Arion, Angela Bonifati, Ioana Manolescu, and Andrea Pugliese. Path summaries and path partitioning in modern XML databases (poster). In *WWW Conference*, Edinburgh (Great Britain), May 2006.

[40] V. Benzaken, G. Castagna, D. Colazzo, and K. Nguyen. Type-based XML projection. In *VLDB 2006, 32nd International Conference on Very Large Data Bases*, pages 271–282, 2006.

[41] Mikolaj Bojanczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data trees and applications to XML reasoning. In *PODS*, 2006.

[42] Mikolaj Bojanczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin, and Claire David. Two-variable logic on words with data. In *LICS*, 2006.

[43] Mikolaj Bojanczyk, Mathias Samuelides, Thomas Schwentick, and Luc Segoufin. Expressive power of pebble automata. In *ICALP*, 2006.

[44] S. Carpineti, G. Castagna, C. Laneve, and L. Padovani. A formal account of contracts for Web Services. In *WS-FM, 3rd Int. Workshop on Web Services and Formal Methods*, number 4184 in LNCS, pages 148–162. Springer, 2006.

[45] G. Castagna, M. Dezani-Ciancaglini, and D. Varacca. Encoding $\mathbb{C}$Duce into the $\mathbb{C}\pi$-calculus. In *CONCUR 2006, 17th. International Conference on Concurrency Theory*, number 4137 in Lecture Notes in Computer Science, pages 310–326. Springer, 2006.

[46] Dario Colazzo and Carlo Sartiani. An efficient algorithm for XML type projection. In *ACM-SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP)*, 2006.

[47] S. Demri and D. Lugiez. Presburger modal logic is PSPACE-complete. In *IJCAR 06, 3rd International Joint Conference on Automated Reasoning*, LNCS. Springer, August 2006.

[48] Ioana Manolescu, Cédric Miachon, and Philippe Michiels. Towards micro-benchmarking XQuery. In *EXPDB*, juin 2006.

[49] L. Acciai and M. Boreale. XPi: a typed process calculus for XML messaging. In *FMOODS 05, 7th IFIP International Conference on Formal Methods for Object-Based Distributed Systems*, number 3335 in LNCS, pages 47–66. Springer, June 2005.

[50] Loredana Afanasiev, Ioana Manolescu, and Philippe Michiels. MemBeR: A micro-benchmark repository for XQuery. In *XML Symposium*, Trondheim (Denmark), septembre 2005.

[51] Andrei Arion, Véronique Benzaken, Ioana Manolescu, and Ravi Vijay. ULoad: Choosing the right storage for your XML application. In *VLDB*, pages 1330–1333, 2005.

[52] V. Benzaken, G. Castagna, and C. Miachon. A full pattern-based paradigm for XML query processing. In *PADL 05, 7th International Symposium on Practical Aspects of Declarative Languages*, number 3350 in LNCS, pages 235–252. Springer, January 2005.

[53] Iovka Boneva and Jean-Marc Talbot. Automata and logics for unranked and unordered trees. In *International Conference on Rewriting Techniques and Applications (RTA 2005)*, volume 3467 of *Lecture Notes in Computer Science*, pages 500–515. Springer, 2005.

[54] Iovka Boneva, Jean-Marc Talbot, and Sophie Tison. Expressiveness of Spatial Logic for Trees. In *Twentieth Annual IEEE Symposium on Logic in Computer Sience (LICS 2005)*, pages 280–289. IEEE Press, 2005.

[55] G. Castagna, D. Colazzo, and A. Frisch. Error mining for regular expression patterns. In *ICTCS 2005, Italian Conference on Theoretical Computer Science*, number 3701 in Lecture Notes in Computer Science. Springer, 2005.

[56] G. Castagna, R. De Nicola, and D. Varacca. Semantic subtyping for the π-calculus. In *LICS '05, 20th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 2005.

[57] Dario Colazzo and Carlo Sartiani. Mapping maintenance in XML p2p databases. In *International Symposium on Database Programming Languages (DBPL)*, pages 74–89, 2005.

[58] H. Hosoya, A. Frisch, and G. Castagna. Parametric polymorphism for XML. In *POPL '05, 32nd ACM Symposium on Principles of Programming Languages*. ACM Press, 2005.

[59] Ioana Manolescu and Yannis Papakonstantinou. XQuery midflight: Emerging database-oriented paradigms and a classification of research advances. In *International Data Engineering Conference (ICDE)*, Tokyo (Japan), April 2005.

[60] Wim Martens and Joachim Niehren. Minimizing tree automata for unranked trees. In *10th International Symposium on Database Programming Languages*, volume 3774 of *Lecture Notes in Compute Science*. Springer Verlag, August 2005.

[61] Joachim Niehren, Laurent Planque, Jean-Marc Talbot, and Sophie Tison. N-ary queries by tree automata. In *10th International Symposium on Database Programming Languages*, volume 3774 of *Lecture Notes in Computer Science*, pages 217–231. Springer Verlag, September 2005.

[62] Hitoshi Ohsaki, Jean-Marc Talbot, Sophie Tison, and Yves Roos. Monotone AC-tree automata. In *International Conference on Logic for Programming and Automated Reasoning (LPAR 2005)*, volume 3835 of *Lecture Notes in Computer Science*, pages 337–351. Springer Verlag, 2005.

Communications dans les actes de conférences nationales avec comité de lecture

[63] V. Benzaken, G. Castagna, D. Colazzo, and C. Miachon. Pattern by Example: type-driven visual programming of XML queries. In *Bases de Données Avancées*, 2007.

[64] Nicole Bidoit and Dario Colazzo. Capturing well typed references in DTDs. In *Bases de données advancées*, 2006.

[65] V. Benzaken G. Castagna, D. Colazzo, and K. Nguyen. Type-based XML projection. In *Bases de données advancées*, 2006. Version préliminaire de [40].

Communications dans des ateliers internationaux avec actes électroniques sur invitation ou sélection par comité de lecture

[66] G. Castagna and K. Nguyen. Typed iterators for XML. In *PLAN-X '08, 6th ACM-SIGPLAN Workshop on Programming Language Technologies for XML*, January 2008. Version préliminaire de [26]. Sélection par CdL.

[67] Olivier Gauwin, Anne-Cécile Caron, Joachim Niehren, and Sophie Tison. Complexity of earliest query answering with streaming tree automata. In *ACM SIGPLAN Workshop on Programming Language Techniques for XML (PLAN-X 2008)*, January 2008. PLAN-X Workshop of ACM POPL.

[68] G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. In *PLAN-X '07, 5th ACM-SIGPLAN Workshop on Programming Language Technologies for XML*, January 2007. Sélection par CdL. Version préliminaire de [25].

[69] Nicole Bidoit and Dario Colazzo. Testing XML constraint satisfiability. In *International Workshop on Hybrid Logic 2006 (HyLo), colocated with LICS 2006*, 2006.

[70] Emmanuel Filiot, Joachim Niehren, Jean-Marc Talbot, and Sophie Tison. Composing monadic queries in trees. In Giuseppe Castagna and Mukund Raghavachari, editors, *PLAN-X 2006 Informal Proceedings*. Basic Research in Computer Science, 2006.

[71] Lucia Acciai, Michele Boreale, and Silvano Dal Zilio. A Typed Calculus for Querying Distributed XML Documents. In *NWPT 2005 – 17th Nordic Workshop on Programming Theory*, October 2005.

[72] Yves Andre, Anne-Cecile Caron, Denis Debarbieux, and Yves Roos. Indexes and path constraints in semistructured data. In *DEXA Workshop on Logical Aspects and Applications of Integrity Constraints*, pages 837 – 841. IEEE Comp. Soc. Press, aug 2005.

[73] Andrei Arion, Véronique Benzaken, and Ioana Manolescu. XML Access Modules: Towards physical data independence in XML databases. In *Second International Workshop on XQuery Implementation, Experience and Perspectives (XIME-P)*, 2005.

[74] Dario Colazzo and Carlo Sartiani. Typechecking queries for maintaining schema mappings in XML P2P databases. In *ACM Workshop on Programming Language Technologies for XML (PLAN-X)*, 2005.

Notices descriptives, manuels d'initiation ou de référence de logiciels ou de langages

[75] V. Benzaken, G. Castagna, and A. Frisch. ℂDuce *Tutorial*, 2006. Available on line at `http://www.cduce.org/tutorial.html`.

[76] G. Castagna, J. Demouth, A. Frisch, and S. Zacchiroli. ℂDuce *User Manual*, 2006. Available on line at `http://www.cduce.org/manual.html`.

Thèses et rapports de stage

[77] Kim Nguyen. Langage de combinateurs pour XML: Conception, typage, implantation. PhD thesis, Université Paris Sud, May 2008.

[78] Lucia Acciai. Algèbres de processus pour les architectures orientées service,. PhD thesis, Université de Provence, Aix-Marseille I, 2007. PhD thesis. Université de Provence.

[79] Andrei Arion. Modules d'accès XML: vers l'indépendance physique dans les bases de données XML. PhD thesis, Université de Paris XI, Décembre 2007.

[80] Yann Barsamian. *Contrôle d'accès et composition de Services Web*. Master's thesis, Master Parisien de Recherche en Informatique, 2007.

[81] Matthieu Objois. Langages de requêtes temporels, extraction de connaissances temporelles et application aux flux de données. PhD thesis, Université de Paris XI, 2007.

[82] Mathias Samuelides. Automates d'arbres à jetons. PhD thesis, Université de Paris 7, Décembre 2007.

[83] Iovka Boneva. Logics for unranked and unordered trees and their use for querying semistructured data. PhD thesis, Université des Sciences et Technologies de Lille, Lille I, 2006. PhD thesis. Université des Sciences et Technologies de Lille - Lille 1.

[84] Cédric Miachon. Langages de requêtes pour XML à base de patterns : conceptions, optimisation et implantation. PhD thesis, Université Paris Sud, 2006.

[85] Philippe Tagoum. *Importation et exportation de services web en* ℂDuce. Rapport de ter-m1, Université Paris 11, mai 2006.

[86] Till Varoquaux. *E-*ℂDuce*: une intégration de* ℂDuce *avec XHTML.* Rapport de ter-m1, Université Paris 11, mai 2006.

[87] Denis Debarbieux. Modélisation et requêtes des documents semi-structurés: exploitation de la structure de graphe. PhD thesis, Université des Sciences et Technologies de Lille, Lille I, dec 2005.

[88] Jean-Marc Talbot. Habilitation. model-checking pour les ambients : des algèbres de processus aux données semi-structurées, December 2005. Habilitation à diriger les recherches. Université des Sciences et Technologies de Lille - Lille 1.

[89] Kim Nguyễn. *Une algèbre de filtrage pour le langage* ℂDuce. Mémoire de DEA, Université Paris 7, 2004.

Miscellanea: articles soumis ou en préparation, rapports non publiés, URI's, cités dans le rapport

[90] ECDuce. `http://www.reglisse.ens.fr/ecduce`.

[91] Squirrel: Structural query induction relying on regular languages. `http://mostrare.futurs.inria.fr/software/squirrel`.

[92] The CDuce Programming Language. `http://www.cduce.org`.

[93] ULoad experiments. `http://gemo.futurs.inria.fr/projects/XAM/experiments.html`.

[94] XSum: the XML Summary tool. `http://gemo.futurs.inria.fr/software/SUMMARY/`.

[95] Nicole Bidoit and Dario Colazzo. Hybrid logic for expressing XML schemas with typed references. En cours de soumission, 2008. Version complète de [64].

[96] G. Castagna, M. Dezani-Ciancaglini, E. Giachino, and L. Padovani. General session types. Submitted, July 2008.

[97] Stéphane Demri and Denis Lugiez. The complexity of modal logics with presburger constraints. Submitted, 2008.

[98] Olivier Gauwin, Joachim Niehren, and Sophie Tison. Bounded delay and concurrency for earliest query answering. 2008. submitted.

[99] Andrei Arion, Véronique Benzaken, Ioana Manolescu, and Yannis Papakonstantinou. Trading with plans and patterns in XQuery optimization. Gemo technical report, 2006.