

# Avant-propos

---

Une rencontre fortuite entre ses deux auteurs participe de la genèse de cet ouvrage. Le hasard a fait que nous soyons mis en présence un soir, l'un parlant de UML, l'autre de XML. Notre échange a mis au jour le vide qui existait entre la réalité et les discours de salon : contrairement aux idées reçues, utiliser UML pour concevoir des modèles XML n'est pas direct, et confrontées à la spécificité de XML, les anciennes méthodes de modélisation semblent échouer. Nous avons eu envie de clarifier la situation au moyen d'un ouvrage. L'évidence du problème s'est révélée encore plus dès que nous avons commencé à rédiger l'ouvrage ; nous n'imaginions pas à quel point notre intuition initiale était en deçà de la réalité.

Chacun dans sa spécialité a vu ses idées converger avec celles de l'autre ; cela s'est fait dans l'harmonie. C'est important. Comme Marcel Dassault, qui disait souvent qu'« un bon avion est un bel avion », nous pensons qu'un bon modèle est un beau modèle.

Faute de pouvoir tout connaître des applications de XML, nous étions certains d'au moins une chose : XML est bien plus qu'un nouveau format d'échange de données entre ordinateurs, bien plus qu'un alignement de balises et d'attributs. C'est un nouveau type de modèle de données qui, à ce titre, doit recevoir de nouvelles techniques de modélisation.

XML est pour les ordinateurs une nouvelle langue universelle qui va peut-être enfin leur permettre de s'ouvrir à la communication. Une langue qui serait toutefois aujourd'hui comme folle, parlée par mille personnes qui ne chercheraient ni à s'écouter, ni même à apprendre la langue de son interlocuteur. L'image est caricaturale mais l'impression générale laissée par l'avalanche de normes, standards et protocoles XML y confine parfois.

Les possibilités de la matière XML sont encore sous-estimées. Le *basus XMLius* ne fait que découvrir la puissance réelle de XML, qui va bien au-delà d'un simple format d'enregistrement. Voyez un peu : on exprime en XML des structures de données, des documents pourtant usuellement assimilés à des *données non structurées*, des descriptions de processus, des ressources informatiques physiques, mais également des concepts totalement abstraits, des langages de programmation, et la liste est encore longue... XML pourrait même envahir sous peu les couches basses des systèmes d'exploitation.

Une telle omniprésence ne peut qu'inciter les concepteurs d'applications à comprendre, bon gré mal gré, comment toutes ces architectures de données et d'information s'intègrent et cohabitent. XML permet de manipuler des espaces d'information à  $n$  dimensions. Au cœur d'applications de plus en plus interconnectées, les données XML, de par les enjeux économiques qu'elles représentent, devront être modélisées avec le plus grand soin.

## Objectifs de ce livre

Comme tout langage, XML a ses règles et, comme toute création humaine, il n'expose sa puissance qu'à ceux qui le lui demandent.

Nous espérons répondre dans cet ouvrage à de nombreuses questions, à commencer par une explication du vocabulaire qui s'y trouve utilisé. Diverses influences font que des termes abscons sont trop souvent utilisés par les spécialistes : nous voulons les rendre aussi simples à comprendre que possible. Ces mots doivent, comme *frigorifère*, devenir courants. C'est ce souci de clarté, de simplicité, qui nous a guidés tout au long de cet ouvrage.

Au-delà de la théorie, nous mettons notre expérience à votre disposition : d'un côté, dix années de modélisation, et de l'autre, vingt ans de SGML et XML. Cette expérience nous autorise à formuler quelques conseils dans le choix des modèles et de la méthode. Ceux que nous présentons ont tous été confrontés à des projets réels.

Ce livre apporte des réponses aux questions suivantes : Quel est le bon modèle ? Comment concevoir un modèle ? Existe-t-il une méthode ? Comment mettre au point les schémas XML ? Comment mesurer la qualité d'un modèle ?

Un schéma XML est la traduction physique d'un modèle. Nous expliquons ici comment conduire la conception d'un tel modèle. Nous décrivons les différentes étapes qui permettent de passer du modèle théorique au schéma concret.

Peu à peu, la prise de conscience se fait que XML a un rôle à jouer bien plus grand que celui de simple format de fichier. Cet ouvrage est destiné tant à ceux qui en sont convaincus qu'à ceux qui découvrent la modélisation XML.

Ayant aboli les frontières traditionnelles de la donnée structurée, du document et de la transaction, XML s'est imposé *de facto* comme modèle universel dans la définition de schémas de stockage des données (XML Schema et Relax NG), d'échange (SOAP, UDDI, WSDL), d'application de règles métier (XSLT, XQuery, Schematron) et d'orchestration des processus (BPEL).

Au regard des couches logicielles qui composent les systèmes d'information apparaît naturellement le concept d'architectures de données. L'empilage des couches de données va confronter les concepteurs à un problème du type tour de Babel. On le voit déjà avec des applications telles que SOAP et JSP qui permettent de faire cohabiter dans un même fichier informatique plusieurs modèles XML, voire plusieurs langages de programmation.

C'est à ce stade que XML devient un outil de modélisation, passant du statut de simple descripteur de données (le schéma des données) à celui de chef d'orchestre (le schéma des processus qui le font vivre).

La question de la représentation des nouveaux systèmes n'a pas de réponse officielle. Il n'existe pas à ce jour de méthode de modélisation des systèmes qui aurait été spécialement étudiée pour XML. Cela viendra probablement un jour, mais le travail est difficile : XML est multiforme, la « chose » est plus complexe qu'il n'y paraît. Si l'information est la plus humaine des caractéristiques animales, sa gestion avec XML est la plus humaine des créations de l'informatique.

Nous cherchons désormais à représenter des flux d'informations qui, par nature, ont pour vocation à circuler, s'échanger, se transformer. XML est aujourd'hui la seule réponse que nous ayons face au besoin grandissant de gestion d'informations changeant d'état continuellement. Nous en tenons compte ici dès le chapitre 1.

Aussi, produire des schémas XML ne se limite pas à écrire des schémas statiques de données mais à représenter les différents états de l'information, de sa création à son obsolescence.

Concrètement, vous allez découvrir :

- une démarche de modélisation XML en utilisant UML ;
- les impacts de XML sur la conception des systèmes d'information ;
- les trucs et astuces de certains schémas XML.

## À qui s'adresse cet ouvrage ?

« On comprend mieux nos ignorances d'avant » (témoignage d'un chef de projet ayant participé à un cours sur la modélisation XML).

Le livre s'adresse à ceux qui, pour concevoir des systèmes d'information, doivent recevoir, gérer et traiter des volumes importants de données XML. Aujourd'hui, ils se heurtent à des techniques de conception inadaptées au XML. Ce livre leur offre les explications dont ils ont besoin pour adapter leurs méthodes de conception aux cas auxquels ils sont confrontés.

Tous les informaticiens sont concernés par cet ouvrage.

Nous pensons tout d'abord aux étudiants en informatique qui ne manqueront pas d'être confrontés, dès leur entrée dans la vie professionnelle, à des projets faisant largement appel à XML.

Ensuite, nous pensons à ceux pour qui le XML était initialement dédié : les développeurs de sites de commerce électronique et, plus largement, ceux qui conçoivent et développent des applications de gestion de contenu.

Bien sûr, sont également concernés les développeurs de systèmes de gestion et production de documents techniques, juridiques ou autres.

Enfin, viennent les informaticiens du monde de la gestion qui évolue rapidement vers une informatique d'entreprise et, de ce fait, ouvre ses applications à la gestion des documents d'entreprise.

## Sujets couverts par cet ouvrage

Ce livre fournit théorie et pratique :

- Les sept premiers chapitres sont autant d'étapes de la démarche de modélisation que nous expliquons.
- Les six suivants sont des exemples concrets de cas particuliers.

Dans notre **introduction**, nous revenons sur la dualité XML/UML avec des explications sur les concepts qui se cachent derrière les termes de modèle et de schéma.

Dans le **chapitre 1**, nous présentons une méthode de préconception d'application. Elle permet d'identifier, organiser et analyser les fonctions du futur système d'information. Cette méthode d'ébauchage se veut simple mais efficace. Elle offre aux chefs de projet qui la pratiquent la possibilité de rationaliser leur projet.

Dans le **chapitre 2**, nous analysons les modèles conceptuels des deux mondes XML et UML.

Dans le **chapitre 3**, nous expliquons comment passer du modèle conceptuel UML à un schéma XML. Nous y détaillons l'utilisation de la notation graphique du diagramme des classes de UML en vue de la production d'un schéma XML.

La réalisation d'un modèle physique constitue le **chapitre 4** où nous exposons les spécificités du stockage de documents XML. C'est ici qu'intervient la possibilité de prendre en compte les mécanismes de liaison de XML.

Dans le **chapitre 5**, qui traite de la séparation des données et des traitements, nous expliquons pourquoi l'approche dite par objets métier n'est pas bonne ; nous lui opposons en effet celle orientée service, plus conforme aux fondamentaux de XML.

Dans le **chapitre 6**, nous nous attachons à la problématique des variantes. À partir d'un modèle de base, il est fréquent de devoir développer des variantes. Ce chapitre expose tout ce qu'il faut savoir sur l'écriture de telles applications.

Le **chapitre 7** est fondamental. Comme nous l'avons vu en introduction de cet avant-propos, un fichier-programme peut contenir aujourd'hui jusqu'à quatre ou cinq langages de programmation différents ! Il faut savoir représenter ces programmes dans différents plans logiques pour s'assurer de leur cohérence.

Le **chapitre 8** est le premier d'une série traitant de cas réels. Les modèles choisis ont tous une originalité. Ils peuvent servir d'exemple. Dans ce chapitre, nous présentons les modèles modulaires qui sont articulés et peuvent être décomposés en petits morceaux.

Dans le **chapitre 9**, nous expliquons le rôle des éléments purement structuraux, qui sont aux schémas XML ce que les articulations sont au corps humain.

Le cas des modules d'information est traité au **chapitre 10**. Nous fournissons en particulier les règles de base qui gouvernent la conception de systèmes basés sur le principe des modules d'information.

Les métadonnées, qui font le pont entre la gestion du contenu et celle du contenant, sont présentées au **chapitre 11**. Les modèles RDF, Dublin Core, xHTML et S1000D sont présentés accompagnés de nombreux exemples concrets.

Les possibilités de liaison sont un point fort de XML. Sans les liens, le Web ne serait pas, à coup sûr, ce qu'il est devenu aujourd'hui. C'est un sujet important mais complexe. Nous revenons dans le **chapitre 12** sur les modèles ID/IDREF, XLink, XPointer, XTM, et expliquons en détail le principe des liens logiques.

Nous terminons notre voyage dans le monde des cas réels avec le **chapitre 13** qui aborde la question de la gestion des révisions. Garder la trace et l'historique des informations modifiées est aujourd'hui un impératif pour de nombreuses applications ; mieux vaut connaître les possibilités offertes par XML dans ce domaine.

## Quelques règles de lecture de l'ouvrage

Les chapitres de l'ouvrage sont relativement indépendants les uns des autres. Vous pouvez donc les aborder librement. Chacun d'entre eux cerne un des aspects du sujet général de l'ouvrage, de la façon la plus didactique possible, étayée par moult exemples et illustrations.

L'ouvrage contient des extraits de schémas XML, des exemples de documents XML et des figures. Les tableaux sont également utilisés pour présenter certains cas de figure. La syntaxe des schémas XML fait que les codes sources sont longs et pénibles à lire. Parfois, nous avons préféré simplifier la lecture des schémas sources en les écrivant sous la forme DTD ou de représentations graphiques.

En ce qui concerne le vocabulaire, nous utilisons la terminologie suivante :

- Après avoir très longtemps combattu le mot « parseur » au profit de la seule expression consacrée d'« analyseur lexico-syntaxique », il apparaît que le mot « parseur », largement utilisé, s'impose. Il faut lui reconnaître l'avantage de la concision. Attention toutefois, ce mot est souvent galvaudé. Nous vous encourageons à vous référer au glossaire pour connaître sa définition exacte.
- Le terme de schéma pour XML est utilisé pour désigner un schéma conforme à la recommandation XML Schema du W3C. Pour la norme ISO Relax NG, nous parlerons de grammaire. Le terme DTD est utilisé pour parler d'un schéma conforme à la norme ISO 8879, à savoir SGML, ou la recommandation XML 1.0 du W3C. Pour éviter de devoir préciser le type de schéma dans les cas où cela n'est pas utile, nous parlerons simplement de schéma.
- Nous utilisons le terme d'« instance » pour désigner un ensemble d'information SGML ou XML valide par rapport à une DTD, et l'expression de « document XML » pour un ensemble d'information XML valide par rapport à un schéma XML.

Les premiers chapitres se lisent l'un à la suite de l'autre, mais ceux de la deuxième partie de l'ouvrage peuvent être lus dans n'importe quel ordre.

## À propos des exemples

Nous nous interdisons de développer des théories sur des exemples imaginaires. Il existe un gouffre entre la réalité et les sempiternels exemples de bons de commandes, bibliothèques virtuelles et autres recettes de pizza *al dente* dont la littérature technique nous abreuve. Proposer un ouvrage sur la modélisation XML qui ne serait pas l'image la plus précise de la réalité serait une démarche *de facto* pervertie. Toutefois,

nous sommes parfois obligés de faire l'impasse sur ce principe et d'avoir recours à des exemples figurés. Ils sont alors la plupart du temps simplement détournés de cas réels.

Notre politique sur cette question est nette. En accord avec Kant qui s'attache à la manière dont les chercheurs font l'acquisition de leurs connaissances scientifiques, nous pensons que le véritable savoir ne s'acquiert que par la confrontation de postulats à la réalité.

« Avec XML, je sais travailler pour les générations futures,  
c'est du développement durable ! »

Jean-Jacques Thomasson

« XML ouvre de nouveaux horizons encore inexplorés  
à l'application des techniques de modélisation »

Antoine Lonjon

## Les auteurs

Jean-Jacques Thomasson a suivi depuis 1984 toute l'évolution des langages de balisage, de SGML à XML. Il dirige depuis vingt ans des équipes spécialisées dans le développement de systèmes de gestion de l'information. Riche de cette longue expérience de la mise en œuvre de XML au sein des systèmes documentaires, il milite chaque fois qu'il le peut en faveur d'une nouvelle approche de la modélisation des données, seule capable selon lui, de faire évoluer de manière décisive la relation homme-machine ainsi que la puissance des applications informatiques. Précédemment à cet ouvrage, il a traduit plusieurs recommandations du W3C (XSLT, DOM, Xpath, XML Schema), les normes WebDAV de l'IETF et XTM de XML Topic Map, le livre d'Eric van der Vlist sur XML Schema paru aux éditions O'Reilly. Il est également l'auteur de *Schémas XML* paru aux éditions Eyrolles en 2002, ainsi que de cinq articles édités depuis 2004 par la revue *Informatique Professionnelle* du Gartner Group. Contact : [jjthomasson@free.fr](mailto:jjthomasson@free.fr).

Antoine Lonjon a plus de 15 ans d'expérience dans le domaine de la modélisation appliquée à l'analyse des systèmes d'information et des métiers de l'entreprise. Il a participé à de nombreux projets internationaux de standardisation des techniques de modélisation, dont le projet des *Core Component* de ebXML et le projet *UML 2.0* à l'OMG (Object Management Group). Antoine Lonjon a aussi contribué à la conception des outils de modélisation de la suite logicielle de MEGA International dont il est aujourd'hui directeur de l'offre.

## Remerciements

Nous tenons à remercier tous ceux qui, de près ou de loin, ont croisé la route de ce livre et accepté de nous accompagner par la discussion, l'écriture ou la relecture :

- Libero Maesano, le bénédictin de notre ouvrage et grand maître des services Web depuis la parution aux éditions Eyrolles de son ouvrage de référence sur le sujet. Sa vision sur notre travail était fondamentale : les architectures orientées service (SOA) et les modèles de données XML sont comme des prises mâles et femelles.
- Jean Delahousse et son équipe de Mondeca prônent l'utilisation du modèle XTM (XML Topic Map) pour la gestion des connaissances. Michel Biezunski a contribué de manière importante à la rédaction de la norme ISO 13250 spécifiant les Topic Maps dont est hérité le modèle XTM. Tous deux sont des amis qui ont accepté de relire les parties de l'ouvrage concernant ces sujets.
- Michel Doméon, de la société Dassault-Aviation, qui m'a soutenu depuis des années dans mes travaux et a apporté son savoir sur la norme S1000D. Il fait référence sur la question de la modularisation de l'information et intervient au plus haut niveau des instances décisionnaires concernant les normes et modèles XML applicables au monde de l'aéronautique et de la défense.
- Bruno Estrade, qui a travaillé depuis plusieurs années sur les systèmes de bases de données mêlant relationnel et XML, a joué un rôle important dans la rédaction du chapitre dédié au stockage (chapitre 4) en multipliant les tests et en confrontant la théorie à la réalité. Ce chapitre, commencé par Julien Viet, n'aurait pu être terminé sans son précieux concours.
- Guillaume Lebleu, de la société Brixlogic, a apporté son regard et validé nos propos sur les modèles orientés données utilisés dans les milieux de la banque et de la finance. Brixlogic développe une offre de produits qui comprennent la logique métier de ces modèles XML et permettent donc aux utilisateurs de se concentrer sur leur seule problématique métier.
- Stéphane Mariel nous a autorisé à reproduire à l'identique l'application PiloteWeb qu'il avait conçue et développée pour son propre ouvrage paru aux éditions Eyrolles : *PostgreSQL* dans la collection des *Cahiers du programmeur*. Qu'il en soit ici remercié, ce cas d'école est intéressant à plus d'un titre.
- Nicolas Rabaté, qui a toujours cru en nos travaux (depuis près de vingt ans !), nous a soutenu, et a relu le chapitre et la théorie sur les éléments purement structurels à une époque où le texte était loin d'être abouti...
- Arthur Ramiandrisoa, architecte de la société SQLI, qui, il y a plusieurs années déjà, fut l'un des premiers à comprendre le parti qu'il pourrait tirer des infrastructures XML, notamment grâce aux bases de données natives XML. Nous regret-

---

tons encore aujourd'hui que son nom ne vienne pas compléter les deux nôtres au générique de cet ouvrage.

- Julien Viet, jeune chercheur du NIST à l'époque où nous l'avons connu. Julien a fait un travail important sur la problématique de stockage de données XML dans des bases relationnelles avant de quitter le projet pour devenir chef développeur de JBoss.