

XML – Un instrument stratégique pour les Archives?

Basé sur l'article *Softening the Borderlines of Archives through XML — a Case Study, Proceedings of the ERPANET workshop 'XML as a Preservation Strategy Urbino/Italy', Octobre 2002.*

Stephan Heuscher

Archives fédérales suisses
Responsable du domaine Architecture des données
dans le projet commun eGouvernement ARELDA
Stephan.Heuscher@bar.admin.ch

Peter Keller-Marxer

Archives fédérales Berne
Chef du Centre ARELDA et
Chef du projet commun eGouvernement ARELDA
Peter.Keller@bar.admin.ch

(Traduction française de Jean-Daniel Zeller (rév. Myriam Erwin), de l'article original allemand paru dans ARBIDO, no 3, 2003, pp. 16-18)

Il n'y a aujourd'hui presque aucun champ d'application de l'informatique dans lequel l'Extensible Markup Language (XML)¹ n'entre pas en jeu, d'une manière ou d'une autre, avant tout lors de l'échange d'informations structurées entre divers systèmes. Dans le domaine de l'édition, des bibliothèques et des archives, XML n'est en rien fondamentalement nouveau, puisqu'il s'agit d'un pur sous-ensemble du Standardized General Markup Language (SGML, standard ISO 8879:1986) qui est connu dans ces domaines depuis presque vingt ans. Comme un "SGML à usage domestique", XML a contribué en peu d'années à faire basculer ce langage de balisage complexe et difficile à manipuler, conçu pour des usages hautement spécialisés, dans des cercles de domaines d'application bien plus vastes.

Ainsi, ce n'est pas le concept, depuis longtemps connu par le SGML, d'un langage balisage sémantique pour les données structurées qui rend XML intéressant pour les archives et les bibliothèques aujourd'hui, mais le fait qu'XML est devenu une norme acceptée et répandue universellement dans l'industrie, où (contrairement au SGML) une large palette de logiciels économiquement avantageux, voire gratuits, sont à disposition, avec lesquels les documents XML peuvent être produits et traités de manière simple. En outre, une multiplicité de logiciels d'application courants (par exemple dans le domaine des bases de données) supportent aujourd'hui au moins l'importation et l'exportation de leurs données dans les formats XML. Ainsi apparaît une situation nouvelle permettant avec une relative simplicité et efficacement la création de "flux de données ouverts" entre les diverses applications générales des producteurs des données et les applications spécifiques aux archives, de telle façon que les données à transférer ne rencontrent plus aucune barrière technique fondamentale, grâce à l'utilisation d'XML de part et d'autre.

XML: des données structurées préservées d'une façon compréhensible?

Dans la discussion actuelle autour de l'utilisation XML pour l'archivage à long terme, on oublie souvent qu'XML ne représente ni une stratégie d'archivage ni un commencement d'archivage, mais seulement une technologie, donc un instrument pour l'élaboration des solutions d'archivage possibles.

Un malentendu également répandu est qu'XML représente la fin du désordre des formats de données et que les documents XML sont fondamentalement "compréhensibles" ou très "auto-explicatifs". Ces deux propositions sont inexactes. XML n'est pas un format, mais une définition du

¹ <http://www.w3.org/TR/REG-xml>

mode de définition des formats sémantiques. Il constitue la base d'un nombre toujours croissant de formats XML.²

Comme technologie, XML est soumis à la même obsolescence que les autres formats de données. Aujourd'hui, personne ne peut prédire comment XML se développera ni si XML existera encore dans 20 ans. Les documents XML sont directement lisibles, certes, de purs documents texte donc "human-readable", au contraire des données binaires qui sont illisibles pour les humains; mais cela ne veut donc pas dire que la sémantique, qui est la marque directe de la compréhension d'un document XML, est absolument lisible également. Deux documents XML au contenu trivial permettent de l'expliquer (cf. encadré). Ils contiennent tout deux la même information, en fait, deux inscriptions d'une liste des personnes avec le nom, l'abréviation, le numéro de téléphone, le login informatique et le bureau desdites personnes.

La première version est totalement incompréhensible pour une personne sans connaissance exacte de la signification des structures sémantiques (balises XML). La deuxième version est compréhensible pour une telle personne, mais intuitivement seulement, puisque les noms des paramètres "Angestellte" ("employé"), "Person" ("personne"), "Telefon" ("téléphone") etc., ont une signification compréhensible en général. Mais : leur signification ici et le contexte exact restent inconnus.

Document XML (Exemples, voir le corps du texte)

Document XML « incompréhensible » :

```
<?xml version="1.0"?>
<a a="BAR">
  <pl "U1977" k="Hs" t="031 3241095" r="E43">Heuscher</p>
  <pl "U1976" k="Zt" t="031 3250017">Zürcher</p>
</a>
```

Document XML « compréhensible » :

```
<?xml version="1.0" encoding="UTF-16"?>
<a:Angestellte amt="BAR" xmlns:a="http://namespaces.arelda.ch/mitarbeiterliste"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://namespaces.arelda.ch/mitarbeiterliste
    http://schemata.arelda.ch/mitarbeiterliste_2003.xsd">
  <a:Person login="U 1977" kuerzel="Hs" telefon="+41 313241095" raum="E43">
    Heuscher</a:Person>
  <a:Person login="U1977" kuerzel="Zt" telefon="+41 313250017"
  >Zürcher</a:Person>
</a:Angestellte>
```

Et : qui saura encore dans 30 ans ce qu'était en 2003 un "login" ou quel format avait un numéro de téléphone? Il est aussi improbable qu'en 2103 on comprenne la notion d'"employé" de la même façon qu'aujourd'hui. Et est-ce une erreur de l'enregistrement que chez la première personne un numéro du bureau est donné, mais pas chez la deuxième personne ? Une solution à ces questions est indiquée dans les lignes avec les qualifications "xmlns" et "xsd", qui réfèrent quelles spécifications extérieures d'un « espace de nom » XML (namespace)³, appelé "a:" et qui définissent la structure des schémas XML⁴. valable pour ces documents XML.

(Exkursus: Cet espace de nom définit un contexte de signification historique des balises sémantiques par lequel les significations exactes de la sémantique utilisée dans le document XML

² On trouve plus de 500 formats XML sur : <http://www.oasis-open.org/cover/xml.html#applications>.

³ <http://www.w3.org/TR/REC-xml-names/>

⁴ <http://www.w3.org/TR/xmlschema-0/>

sont déterminées. Celles-ci apparaissent également sous forme XML. Tous les documents XML peuvent se rapporter alors universellement à cet espace de nom que cette sémantique utilise. La question se pose alors de savoir comment peuvent se réaliser des recherches sur de grandes quantités de telles données significatives (par exemple les métadonnées, les données d'enregistrement, les instruments de recherche, etc.), utilisant diverses sémantiques dont certaines d'une part respectent les espaces de nom strictement, et d'autre sont plus tolérantes et admettent des relations de ressemblance (par exemple les thesauri) entre plusieurs sémantiques.)

Ces questions fondamentales - les contextes de signification historique, les mises en valeur multiples par diverses sémantiques et d'après diverses ontologies, la recherche au moyen de thesauri, etc. - sont connues naturellement depuis longtemps dans le domaine des archives et des bibliothèques et sont mises en application de diverses manières par les logiciels d'enregistrement, d'indexation et de catalogage. Ce sont cependant presque toujours des systèmes dans lesquels les données, les modèles de données, la consistance des données et leur fonctionnalités sont intrinsèquement liées aux produits propriétaires spécifiques à un vendeur (par exemple les bases de données et les interfaces bureautiques) – ce qui devient un problème pour l'archivage à long terme.

Nous en sommes actuellement au point où les développements autour de XML (et même fondés sur XML) des "standards X" comme Namespaces, XPath⁵, XPointer⁶, XMLSchema, XSLT⁷, XQuery⁸ etc. permettent avec une relative simplicité et efficacement la résolution de tous ces fonctionnements archivistiques connus depuis longtemps, et en principe presque complètement indépendamment des logiciels propres à un fournisseur, et de mettre à disposition une forme interchangeable, apte à l'archivage intermédiaire comme à l'archivage à long terme.

Ici se révèle si une importance stratégique ressort réellement de l'implémentation de XML comme base technologique d'une solution d'archivage ou de bibliothèque, ou si XML représente seulement un phénomène de mode et ne fait que déplacer les mêmes problèmes dans une nouvelle technologie. Il s'agit d'identifier avec soin quelles fonctions d'archive seront simplifiées par la mise en oeuvre d'instruments - et de quels instruments - du "brave new world XML", et peuvent ou doivent être soutenues.

Certains domaines d'utilisation d'XML dans le domaine archivistique sont présentés ci-dessous à titre d'exemple, au moyen de deux applications concrètes des Archives fédérales.

SIARD: Software-Invariant Archiving from Relational Databases/ Software-Invariante Archivierung aus Relationalen Datenbanken [Archivage de base de données indépendant des logiciels]

SIARD est un logiciel client développé avec le langage de programmation Java par les Archives fédérales et la société Trivadis AG pour les systèmes de bases de données relationnelles. Il se connecte via un réseau avec n'importe quelle base de données possédant une interface JDBC (qui s'applique presque sur tous les produits de base de données).

SIARD est capable, d'analyser les informations relatives à tous les éléments contenus dans la base de données (les dictionnaires; les schémas, les tables, les colonnes, les vues, les contraintes, les types de données etc.), d'extraire simultanément cette structure de données complète avec les données propres de la base sous forme de pures données textuelles, et de les sauvegarder indépendamment de chaque logiciel spécifique d'un fabricant. Le langage de description pour la structure de base de données utilisée est le "Database Definition Language" de SQL3 selon la norme ISO/IEC 9075. Dans ce but, SIARD (automatiquement ou après intervention de l'utilisateur) transforme les éléments spécifiques du fournisseur non-conformes en éléments conformes SQL3. La où cela n'est pas possible, ceux-ci sont exclus de l'archivage. Les éléments exclus et les raisons de leur exclusion sont inscrits comme une part de l'archive SIARD.

⁵ <http://www.w3.org/TR/xpath>

⁶ <http://www.w3.org/TR/xptr/>

⁷ <http://www.w3.org/TR/xslt>

⁸ <http://www.w3.org/TR/xquery/>

Ainsi, SIARD produit une "archive intermédiaire" qui se compose de différentes données textuelles : les données-SQL3 avec la structure de la base de données, le "texte à plat" des fichiers avec le contenu des tables de la base de données, et un fichier XML. Ce fichier XML contient la structure des données-SQL3 de manière redondante, elle contient de plus les renseignements relatifs au processus d'archivage (par exemple les indications sur les éléments exclus de l'archivage) et les indications générales sur la base de données archivée (par exemple les versions du logiciel, les quantités de données, etc.) . Cependant avant tout ce fichier XML contient une quantité prédéfinie des champs de métadonnées encore vides pour la mise en valeur manuelle et la description de la base de données. Ce sont des champs qui se rapportent en particulier aux informations archivistiques et non techniques, qui sont absolument nécessaires à la compréhension à long terme des données, mais ne sont pas présents dans la base de données. Ici il s'agit avant tout d'une description générale de la base de données, par exemple sa provenance et la raison de la création de l'application ainsi que des descriptions en texte clair des noms des tables, des mots-clés, des listes de codes, etc., donc toutes indications qui forment un catalogue de données cohérent. Le contenu de ces champs est ajouté par l'utilisateur dans SIARD et forment alors l'archive SIARD définitive. Les archives SIARD ont toutes la même structure standardisée, indépendamment du système de base de données original dont elles sont issues pour l'archivage.

Cependant l'archivage à long terme des archives SIARD est également indépendant avant tout de chaque logiciel spécifique et se base exclusivement sur des normes ouvertes et documentées (SQL3, XML, Unicode). Les archives SIARD peuvent être réutilisées à l'avenir dans n'importe quel système de base de données propriétaire, avec une dépense marginale. Pour cela, il est uniquement nécessaire que ce produit soutienne le langage de base de données SQL. Cette norme représente la base presque de toutes les bases de données relationnelles depuis environ vingt ans et changera probablement peu au cours de dix prochaines années. Si toutefois le SQL devenait tout de même obsolète, la stricte conformité des archives SIARD au SQL3 garantit une migration relativement simple vers un futur format normalisé.

Dans SIARD, XML remplit avant tout quatre fonctions:

- L'archivage à long terme indépendant du logiciel ainsi que l'intégration indépendante des formats, sur la base d'un modèle standardisé de métadonnées, des métadonnées techniques et archivistiques , avec un contrôle de la consistance ainsi qu'une possibilité d'historicisation (Versionnage) de ces modèles de métadonnées (par un schéma XML);
- La standardisation d'une migration ultérieure de SQL3 vers un autre format de description de données futur (par XSLT et le balisage sémantique des éléments de SQL); en outre : la conservation de toutes les données multilingues quelle que soit la langue originale (soutenu par Unicode via XML);
- La visualisation directe et la possibilité de rechercher toutes les métadonnées (y compris une description générale et le traitement archivistique)et la d'une archive de bases de données SIARD avec un navigateur Web (par XSLT et XQuery);
- La définition et l'importation XML simple de sélection des métadonnées de traitement archivistique SIARD standardisées, dans un système d'enregistrement propriétaire (aux Archives fédérales : "scopeArchive").

AMDA: traitement des enregistrements sonores du parlement

Cet exemple montre un autre aspect de l'utilisation XML, c'est à dire l'acquisition de métadonnées de plusieurs sources de données administratives internes et externes, très hétérogènes. Les versements des enregistrements sonores du parlement forment l'ensemble des données primaires. Celles-ci sont présentes sous forme numérique aux Archives fédérales pour la période 1980-2001 par la digitalisation rétrospective des bandes magnétiques analogiques. Les données de description de cette période se trouvent sous la forme des bases de données Access de Microsoft (indexation avec le produit "Augias").

Depuis 2002, les débats sont enregistrés directement sous forme numérique par les Archives fédérales au parlement. Les métadonnées électroniques corrélées viennent, depuis 1999, de deux systèmes différents, le système de gestion administratif "Curia Vista" et le système sténographique du parlement (la banque de données du Bulletin Officiel) et sont également disponibles sur Internet⁹. Ces métadonnées sont mises à la disposition des Archives fédérales par les services du parlement depuis 2002 sous forme XML brut, par session parlementaire. La troisième source de métadonnées est constituée par l'enregistrement manuel et l'indexation supplémentaire des enregistrements numériques par les Archives fédérales.

Les métadonnées numériques de ces quatre sources très hétérogènes - MS Access pour les fonds anciens, banques de données "Curia Vista" et "Bulletin Officiel" pour les nouveaux fonds, réindexation manuelle pour tous les fonds - doivent être reliées aux données primaires propres (des fichiers son coupés, 'désonorisés', de format WAVE). Et last but not least: un sous-ensemble des données de description doit être transféré dans le système d'enregistrement des Archives fédérales (le produit : "scopeArchive").

L'application AMDA remplit ces tâches d'intégration exigeantes par le recours conséquent à XML comme format d'échange de données commun, à XSLT comme langage de transformation entre les divers formats XML pour l'importation et exportation ainsi qu'au schéma XML pour la garantie de la consistance des données et leur intégrité. La figure 1 montre schématiquement les flux de données. Sans l'utilisation d'XML, la réunion des métadonnées de ces diverses sources nécessiterait des coûts élevés pour l'implémentation des interfaces ainsi que les coûts de traitement et d'enregistrement manuel.

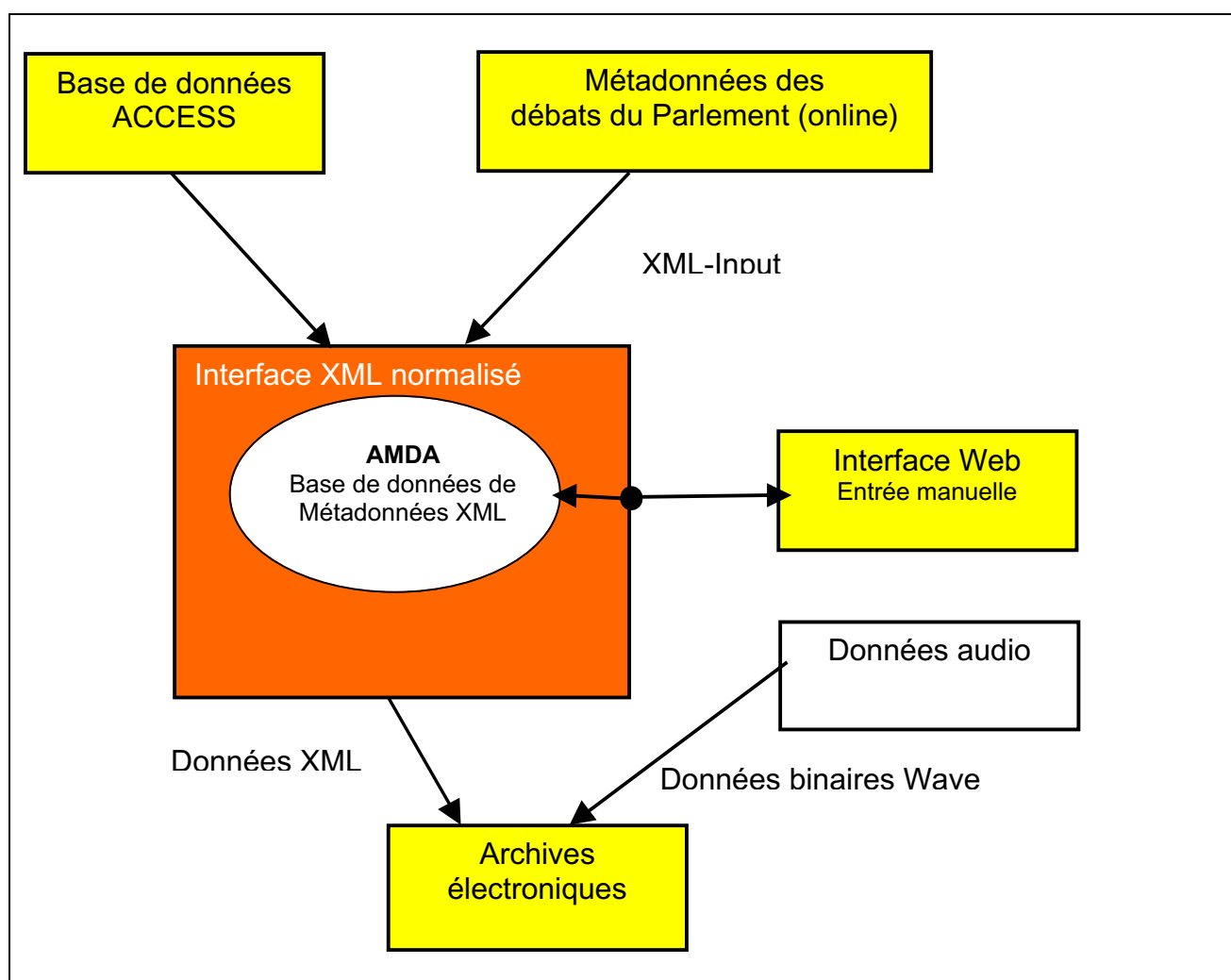


Fig. 1: Environnement de la base AMDA.

⁹ <http://www.parlament.ch/ab/frameset/d/index.htm> (allemand), <http://parlament.ch/ab/frameset/f/index.htm> (français)

Bilan

Les expériences confirment la politique des Archives fédérales, qui a été de commencer avec XML en premier lieu pour les métadonnées, c.-à-d. pour les données qui décrivent les données. Dans ce but, il n'y a de notre point de vue encore aucune norme qui se soit imposée et réponde à nos besoins. Les candidats sont en premier lieu l'Encoded Archival Description (EAD) et le Dublin Core Metadata Standard (DC). Grâce à une transformation de format relativement simple, un format XML propre constitue une solution de transition idéale jusqu'à ce qu'un format d'archive XML standardisé se soit imposé.

Du point de vue actuel, XML contribue à un archivage à long terme indépendant du logiciel et tolérant les migrations des métadonnées, y compris les modèles de métadonnées (ontologies) subséquents, ainsi que la définition des relations entre données et les recherches de données indépendamment du logiciel. Les expériences ont montré en particulier que la réalisation d'échanges de données entre des systèmes à l'intérieur ou en dehors des Archives fédérales ainsi que l'intégration de données de plusieurs sources de données hétérogènes est possible avec XML, de manière économiquement avantageuse et techniquement simple.