

D-Pilgrim : une base relationnelle/documentaire mobile

Gildas Ménier *, Eugen Popovici *, Pierre-Francois Marteau *

* VALORIA, Université de Bretagne Sud, Campus de Tohannic, <http://www-valoria.univ-.fr>, gildas.menier@univ-ubs.fr

Résumé

Nous présentons un système intégrant une base de donnée relationnelle native associée à une base d'information semi-structurées XML permettant de gérer des recherches approchées : le mécanisme exploite des distances d'éditations évaluées sur des chemins de l'arbre DOM. La base relationnelle est elle-même codée en XML et associe la structure relationnelle - pour gérer les contraintes d'intégrité et les dépendances fonctionnelles -, au codage de l'interface d'acquisition - pour décrire les formulaires de saisie.

Mots clés : Indexation, documents semi-structurés, XML, Document Object Model, interface, base mobile, base de données native, gestion documentaire, requête approchée.

1 Introduction

Les assistants numériques (PDA) ont généralement des capacités de traitement et de stockage de l'information relativement limitées en comparaison avec les postes de travail actuels. L'aspect pratique et nomade de cette informatique encouragent cependant le développement de solutions adaptées à ces limitations : C'est le cas précisément des situations où une partie d'une base d'information peut être suffisante : par exemple, dans le cas d'une application commerciale, d'un suivi médical chez le patient etc... Les informations sont à manipuler en saisie (nouveau patient/client), en consultation de base (informations sur un patient/client) mais aussi en recherche de documents (par exemple sur les effets secondaires de tel ou tel traitement). Dans le cas de mise à jour 'sur le terrain', une phase de contrôle d'intégrité et de mise en conformité avec une base centralisée est nécessaire et ces informations doivent être associées aux portions de la base à manipuler.

Nous décrivons un système de base de données native dédié au suivi médical de patients par des intervenants (médecins ou infirmiers) équipés de PocketPC. Les informations 'embarquées' permettent aussi bien une mise à jour de dossiers patients que la consultation approchées de documents semi structurés de références. La base et les documents associés sont représentés en exploitant les propriétés de structuration XML [1] : la structure XML porte également la représentation de l'interface d'acquisition.

2 Principe général

A chaque PocketPC est associé un identifiant unique : l'administrateur de la base A et documentaire B décide de la portion de base relationnelle a_i qui doit être associée à chaque PocketPC (Fig.1). Cette portion a_i est une sous-partie du modèle conceptuel de données pour sa structure et ne contient que la partie d'information nécessaire pour les 'sous-tables' véhiculées.

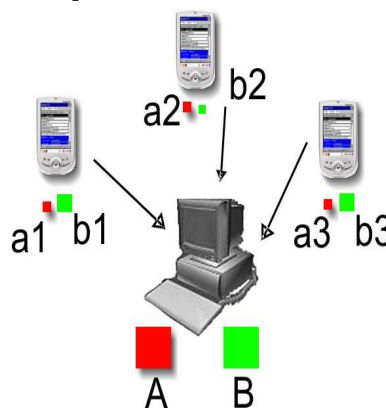


Fig 1 – Interface PocketPC

Des liens associés aux tables permettent au système de gérer les contraintes d'intégrité lors de la récupération d'informations pour la base centrale. En ce qui concerne la base documentaire, l'index de la base est diffusé aux PocketPC mais seuls les documents jugés pertinents sont dupliqués dans les sous ensembles b_i .

Ce mécanisme permet aux utilisateurs de consulter/mettre à jour un extrait de base de données et/ou un extrait de base documentaire.

3 Base de données relationnelle

La base de données native est représentée en XML, la structure des enregistrements est codée par un arbre. A chaque description de table correspond un élément `<TABLE>` qui contient les champs des enregistrements : Par exemple :

```
<BASE>
  <TABLECLIENT>
    <NOM> </NOM>
    <PRENOM> </PRENOM>
    <SEXE choice= « homme;femme »>
```

```

homme
</SEXE>
<VILLE> nice </VILLE>
</TABLECLIENT>
<TABLEMEDECIN>
...
</TABLEMEDECIN>
</BASE>

```

Les attributs des éléments servent à définir l'interface de saisie/consultation (Fig.2). Par exemple, l'élément 'SEXE' indique la création d'une liste de choix (homme/femme). La valeur par défaut (ici 'homme') est indiquée dans l'élément. Les attributs couvrent la majorité des objets de formulaires, des listes déroutantes, aux cases à cocher en passant par la gestion des couleurs ou des champs formatés. La description de la base contient donc à la fois la structure et l'interface d'accès.



Fig 2 – Interface PocketPC

Cette interface est exploitée à la fois du côté 'base centrale' et du côté PocketPC. Des attributs supplémentaires permettent de conserver un lien entre la 'sous' base 'embarquée' et la base principale : la base PocketPC contient en fait un extrait de la base principale : seuls quelques champs de quelques tables sont conservées pour fabriquer une seule table de référence (qui correspond à un seul formulaire de saisie). Les champs sont alors liés aux champs de la base principale par des valeurs d'attributs. Par exemple : `<NOM table= « TABLECLIENT »> </NOM>` indique que ce champ est lié au champ 'NOM' de la table 'TABLECLIENT' de la base principale.

Afin de limiter la saisie des informations sur le PocketPC, des attributs qui décrivent les dépendances fonctionnelles permettent de proposer des choix associés de champs. Par exemple, dans le cas d'une saisie de client associé à un médecin (et à son téléphone), une option permet de proposer la liste des médecins et de remplir de manière automatique les champs qui dépendent du nom de ce médecin (adresse etc).

4 Base de documents semi-structurés

Certains des champs de la base relationnelle peuvent être liés à des documents externes particuliers : cependant, dans le cas d'une recherche d'information dans les documents, le schéma relationnel peut être limité, surtout dans le cas de recherches imprécises (ce

qui n'est normalement pas le cas pour un modèle relationnel). Nous avons adapté un sous ensemble du moteur de recherche de documents semi-structurés SIRIUS [4] pour permettre une utilisation sur un PocketPC. L'analyseur XML du côté PocketPC est optimisé pour la vitesse de traitement et est de type non validant, ce qui ne représente pas un problème car le format XML ne sert qu'à l'échange d'information entre le client et le serveur (ce qui n'est pas le cas du serveur qui supporte l'outil d'indexation).

Le système indexe les informations associées à leur position dans l'arbre XML (information de chemin). La requête (ici simplifiée par rapport au modèle initial) se compose d'un chemin d'éléments (associé à des conditions sur les attributs) se terminant sur un terme.

Un système basé sur une distance d'édition [3] évaluée évalue le nombre minimal de modifications à appliquer à la requête pour trouver un chemin dans un document de la base. Des opérations ensemblistes classiques permettent de formuler des requêtes plus complexes. Une description détaillée de ce mécanisme de recherche approximative peut se trouver dans [4].

5 Conclusion et perspectives

Le système utilise Java comme cadre d'implémentation. La machine virtuelle *Jeode* d'Insignia [2] fait fonctionner le client du PocketPC qui communique via la pile TCP/IP avec son serveur poste : aucune solution particulière autre que la liaison IP n'a été utilisée ce qui permet une communication infra rouge, Bluetooth ou Wi-Fi. Les performances de recherche restent tout à fait comparables à celles d'un poste de travail, compte tenu de la taille modeste d'index placés sur le PocketPC. La visualisation de documents reste à l'heure actuelle simplifiée, mais nous souhaitons pouvoir porter un sous ensemble de XSLT pour manipuler plus aisément les informations affichées.

Références

- [1] Bray T., Paoli J., Sperberg-McQueen C.M., Maler E. Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation 6 October 2000.
- [2] Insignia – *Jeode Runtime Java environment* – www.insignia.com
- [3] Levenstein A., Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Sov.Phys. Dohl. Vol.10, P.707-710, 1966
- [4] Ménier G., Marteau P.F., *Information retrieval in heterogeneous XML knowledge bases*, The 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IEEE, 1-5 July, Annecy, France, 2002.