Chapitre III

Projet Cabinet Infirmier

NB : Tous les documents nécessaires à la réalisation de ce projet vous sont fournis sous forme de documents téléchargeables sur la plate-forme.

1 Le cabinet infirmier

1.a Présentation générale

Le cabinet infirmier *Soins à Grenoble* souhaite organiser les déplacements de ses infirmier(e)s. Il contacte pour cela la société *L3M Agency* pour décider avec elle d'un logiciel capable d'organiser et d'optimiser ses déplacements. Le cabinet emploie aujourd'hui une secrétaire médicale et 3 infirmières, mais est susceptible de s'agrandir. Chaque jour, une infirmière fait le tour de ses patients pour des soins. Chaque jour, la secrétaire médicale entre la liste des patients à visiter ainsi que leurs coordonnées et toute information utile aux soins. Elle affecte ensuite les patients à chacune des infirmières. Lorsqu'une infirmière s'identifie sur l'application, elle obtient la liste des patients qu'elle doit visiter dans la journée, ordonnés de façon à optimiser son trajet quotidien. Elle peut également obtenir la facture correspondant à chaque patient.

1.b Organisation d'une application exploitant ces données

L'application qui pourrait être développée se présenterait comme un Web Service. Le Web Service permettra aux soignants de se connecter à distance à leur serveur métier pour obtenir divers services, comme la facturation client, et renseignements, comme la liste des patients à visiter ou le trajet optimal pour le visites. Des méthodes d'optimisation d'itinéraire telles que celles qui sont développées en Programmation avec Contraintes et Recherche Opérationnelle (PRCO) et une interface (telle que vous la développeriez en IHM), pouvant se connecter au serveur SOAP, pourraient être utilisés dans le cadre de ce projet, lors de sa finalisation, mais ceci est facultatif.

1.c Scénario d'utilisation d'une application exploitant ces données

Deux types de clients peuvent s'adresser au serveur: le/la secrétaire médical(e) ou bien un(e) infirmièr(e). L'action du/de la secrétaire médical(e) sera intégralement prise en charge en IHM via le système d'interface. Le serveur métier, lui, va intervenir pour la gestion des requêtes des infirmièr(e)s afin d'obtenir le trajet de la tournée de la journée:

- le serveur reçoit une requête d'un client "infirmière" (via l'IHM)
- le serveur métier fourni la réponse attendue: il lui transmet la requête et le contenu du fichier XML, ou une page HTML
- le module du serveur métier (que vous pourriez développer en FDD-XML) construit une requête à emettre vers GoogleMapAPI pour obtenir les matrices de distances des adresses des patients.

- Le serveur envoie cette requête à GoogleMap API et récupère la réponse sous forme de matrices de distances dans un fichier XML
- le module du serveur métier lit les informations obtenues et appelle les fonctions d'optimisation développées en Recherche Opérationnelle pour calculer le meilleur trajet.
- le serveur métier renvoie ce résultat dans la page résultat de l'application cliente (l'IHM).

2 Objectifs du projet

Dans ce projet, vous allez :

- modéliser la partie données d'un Cabinet Infirmier (UML et XMLSchema)
- développer un programme permettant de vérifier la validité des données XML par rapport au schéma que vous aurez défini
- créer des transformations XSLT permettant d'extraire des connaissances spécifiques (par exemple les données d'un patient particulier) ou de présenter des données sous forme de pages HTML (comme par exemple la liste des interventions d'une infirmière)
- apprendre à utiliser les parseurs et XPath pour extraire des connaissances depuis le fichier XMLSchema
- apprendre à sérialiser ces données
- développer un mini serveur Rest (HTTP) pour accéder à ces données

3 Le document XML

L'application proposée doit stocker les informations des patients et du cabinet. Vous allez donc créer un langage XML pour le stockage de ces informations, c'est à dire modéliser le cabinet XML avec un nouveau Schema XML, à partir des connaissances contenues dans ce document XML.

Il a été convenu que le même document XML stockerait les informations du cabinet (son nom, son adresse et les identifiants des infirmières) ainsi que les données des patients.

3.a Données du cabinet

Un première partie de ce document XML est représentée dans la figure III.1

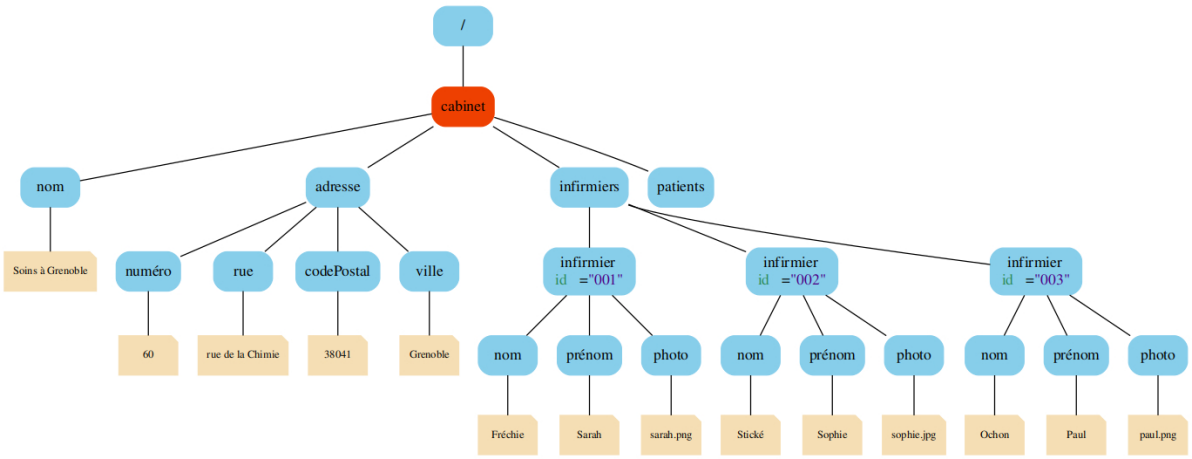

Figure III.1: **Arbre d'instance du cabinet infirmier (données du cabinet)**.

3.b Données des patients

Après de longues discussions avec le cabinet infirmier et la rédaction du cahier des charges, vous en avez déduit que les informations à rentrer pour chaque patient devait être:

Des données administratives :

- son nom
- son prénom
- sa date de naissance
- son numéro de sécurité sociale
- son adresse précise

Ce document doit aussi comporter, pour chaque patient, et pour chaque visite, sa date, le(s) soin(s) à effectuer par l'infirmière ainsi que son(leur) code NGAP.

Enfin, le(a) secrétaire médical(e) affecte pour chaque patient un(e) infirmièr(e). Cette personne sera représentée par son identifiant (à déterminer).

Un exemple de patient avec toutes les informations précédentes est représenté dans la figure III.2

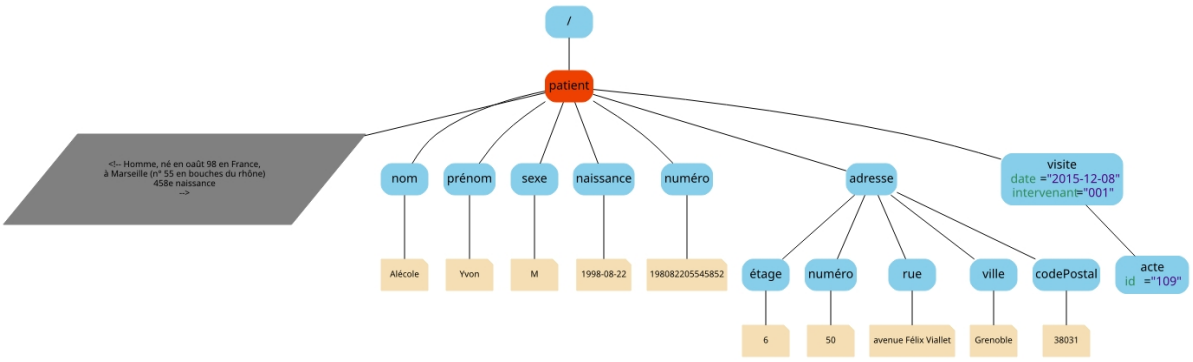

Figure III.2: **Données d'un patient (complément de l'arbre d'instance du cabinet infirmier)**.

3.c L'adresse

Notez sur le document précédent, que pour que l'adresse du patient soit utilisable par la suite, il faudra bien identifier :

- l'étage (si besoin)
- le numéro de la rue (s'il existe)
- le nom de la rue
- le code postal (un nombre à 5 chiffres)
- la ville

3.d Actes infirmier et nomenclature NGAP

Revenons à présent sur l'attribut `id` du noeud `acte`. Il correspond à l'identifiant NGAP. Voici un document XML type contenant les actes infirmier et codes NGAP :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ngap
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://www.univ-grenoble-alpes.fr/l3miage/actes'
    xsi:schemaLocation='http://www.univ-grenoble-alpes.fr/l3miage/actes ../
        schema/actes.xsd'>
    <types>
        <type id="pi">prélèvements et injections</type>
        <type id="pc">pansements courants</type>
        <type id="pl">
            pansements lourds et complexes nécessitant des condition
            d'asepsie rigoueuse
        </type>
        <type id="sd">
            Soins infirmiers à domicile pour un patient, quel que
            soit son âge, en situation de dépendance temporaire ou permanante.
        </type>
    </types>
    <actes>
        <acte id="101" type="pi" clé="AMI" coef="1.5">
            Prélèvement par ponction veineuse directe
        </acte>
        <acte id="102" type="pi" clé="AMI" coef="5">
            Saignée
        </acte>
        <acte id="103" type="pi" clé="AMI" coef="1">
            Prélèvement aseptique cutané ou de sécrétions muqueuses, prélè
                vement
            de selles ou d'urine pour examens sytologiques, bactériologiques,
                mycologiques.
        </acte>
        <!--  ... et plein d'autres actes -->
    </actes>
</ngap>
```

La page suivante explique ce qu'est la nomenclature NGAP pour les soins infirmiers et est à lire attentivement.

Chaque acte est désigné par un intitulé, une lettre-clé et un coefficient.

La lettre-clé est un ensemble de 3 lettres parmi les suivantes:

- AMI pour (Acte Médico-Infirmier)
- AIS (Acte de Soins Infirmier)
- DI (Démarche de soins Infirmiers)

Le coefficient est un chiffre, qui servira de coefficient multiplicateur de la valeur de la lettre-clé pour obtenir la valeur de l'acte. Pour faire simple: Valeur de l'acte = (lettre-clé)xcoefficient.

Vous trouverez la liste complète sur le site de la sécurité sociale, mais nous nous restreindrons à une toute petite partie des actes. Un document plus restreint et plus lisible vous est fourni. Il stocke au format XML les codes NGAP que nous utiliserons dans ce projet. Il sera notamment très utile lorsqu'il vous sera demandé de calculer le coût de chaque visite.

3.e Numéro de Sécurité Sociale

Si l'on se reporte à Wikipedia, le **numéro de sécurité sociale** en France (nom usuel), ou numéro d'inscription au répertoire des personnes physiques (abrégé en NIRPP ou plus simplement NIR) est un code alphanumérique servant à identifier une personne dans le répertoire national d'identification des personnes physiques (RNIPP) géré par l'Insee. C'est un numéro « signifiant » (c'est-à-dire non aléatoire)

composé de 13 chiffres, suivi d'une clé de contrôle de 2 chiffres.

Pour définir le format du numéro de sécurité sociale, veuillez vous reporter à la documentation Wikipédia. (On pourra éventuellement aussi utiliser ce site).

4 Réalisation du projet - 1ere partie (modélisation)

Les étapes de la réalisation vous sont ici présentées dans l'ordre attendu :

- modélisation UML
- génération d'une instance XML
- validation du document
- usage :
  - transformations XSLT
  -

4.a Découverte du projet

Lisez attentivement l'énoncé... oui celui que vous avez survolé au dessus ! Prenez le temps de bien regarder les arbres d'instance fournis.

4.b Modélisation UML - Diagramme papier et PlantUML

Dans cette partie, vous allez modéliser sous forme de diagrammes UML le document. Cela requiert une bonne analyse des arbres d'instance.
Vous reviendrez plusieurs fois sur ce travail au cours des différentes séances de TP.
Astuce : pour progresser plus facilement, il est suggéré de modéliser d'abord seulement des sous parties de l'arbre, par exemple une adresse ou un infirmier.

Papier.

Réalisez scrupuleusement toutes ces étapes :

- Commencez par identifier à partir des 2 arbres d'instance précédents les types dont vous avez besoin : listez les.
- Identifiez quels types seront des restrictions (des types simples restreints) et comment.
- Dessinez sur papier un schéma UML contenant tous les types et leurs relations (composition, ...)
- N'oubliez pas d'englober ces types dans un namespace. Le nom du vocabulaire défini pour la base de données du cabinet est `http://www.univ-grenoble-alpes.fr/l3miage/medical`.

UML - PlantUML

Vous allez maintenant implémenter votre diagramme dans le langage de script `PlantUML`. Vous trouverez toute la documentation nécessaire sur ce site, en particulier ici : documentation pour les diagrammes de classe.

Pour tester vos scripts et générer vos diagrammes sous forme d'images, vous devez les soummettre dans le service en ligne PlantUML.

Même remarque que précédemment, progressez doucement par petites implémentations.

4.c Modélisation XMLSchema

Vous allez écrire le schéma XML implementant le diagramme UML que vous aurez mis au point. Il s'agira de créer un fichier unique nommé (très exactement ; attention au respect de la casse et de l'orthographe) `cabinet.xsd`. Le XMLSchema devra être le plus restrictif possible.

Pour rappel, le nom du vocabulaire défini pour la base de données du cabinet est `http://www.univ-grenoble-alpes.fr/l`

Pensez à vérifier que :

- votre document XML (le schema que vous développez est un document XML) est conforme (voir section 4.e)
- votre document XML (le schema que vous développez est un document XML) est valide par rapport au schéma du vocabulaire `http://www.w3.org/2001/XMLSchema` (voir section 4.f)

4.d Document XML

Création du document XML

Créez un document xml nommé `cabinet.xml` (très exactement ; attention au respect de la casse et de l'orthographe) et qui représente l'arbre précédent. Ce document devra contenir les données données dans les arbres d'instance.

Idéalement, à ce stade, vous aurez développé préalablement votre Schéma XML... mais il est évident que ce ne sera pas abouti. Vous pouvez donc soit générer l'instance XML à partir du Schema (void chapitre II), soit écrire votre document XML à la main. Dans tous les cas il vous faudra rentrer les données à la main et ... à la fin il faudra que ça marche !

Pensez à vérifier que :

- votre document XML est conforme (voir section 4.e)
- votre document XML est valide par rapport au schéma décrit dans le `cabinet.xsd` (voir section 4.f)

Modifications du document XML

On souhaite ajouter les patients suivant dans le document XML, à la suite du premier patient dans le noeud patients et sur le même modèle.

- Premier patient :
  - Nom: Orouge
  - Prénom: Elvire
  - Elvire est née le 08 mars 82 en France, à Lyon (n° 23 dans le rhône), il s'agissait de la 52e naissance dans cette ville
  - Adresse: Rond-Point de la Croix de Vie 38700 La Tronche.
  - Visite: La visite devra avoir lieu le 08/12/2015 pour un Pansement de brûlure étendue. L'infirmière qui interviendra n'a pas encore été décidée.
- Deuxième patient :
  - Nom: Pien
  - Prénom: Oscare
  - Oscare est née le 25 mars 75 en France, à Chambéry (n° 65 dans la Savoie) Il s'agissait de la 692e naissance.
  - Adresse: rue Casimir Brenier 38000 Grenoble.
  - Visite: La visite devra avoir lieu le 08/12/2015 pour une ponction veineuse directe le et une injection intraveineuse directe isolée d'anthalgiques.

- Troisième patient :
  - Nom: Kapoëtla
  - Prénom: Xavier
  - Xavier est un petit garçon de 4 ans, né le 02 août 2011 en France, à Bordeaux (n° 63 dans la Gironde). Il s'agit de la 35e naissance.
  - Adresse: 25 rue des Martyrs 38042 Grenoble.
  - Visite: La visite devra avoir lieu le 08/12/2015 pour recevoir l'injection de plusieurs allergènes afin d'être désensibilisé.

Ajoutez ces informations. L'objectif ici est que vous compreniez que votre travail va consister non seulement à programmer, mais à vous interesser aux données pour lesquelles vous réalisez ces programmes.

4.e Conformité de l'instance XML et du XMLSchema

Vous procéderez de 2 manières différentes selon votre avancement dans le projet.

- En premier lieu, vous pouvez utiliser l'outils de vérification de conformité intégré à votre IDE (voir chapitre II)
- Le plus tôt possible, cherchez à utiliser un programme C# pour réaliser cette opération. Tout est décrit dans le cours dans le chapitre sur les parsers.

4.f Validation de l'instance XML vis-à-vis du XMLSchema

Vous procéderez de 2 manières différentes selon votre avancement dans le projet.

- En premier lieu, vous pouvez utiliser l'outils de validation intégré à votre IDE (voir chapitre II)
- Le plus tôt possible, cherchez à utiliser un programme C# pour réaliser cette opération. Tout est décrit dans le cours dans le chapitre sur les parsers.