

Interopérabilité des SSI et IDM : quels défis, quelles solutions?

12,13 décembre, EPU Nice Sophia-Antipolis



Laboratoire L3i : Informatique, Image et Interaction

C. Faucher F. Bertrand, JY. Lafaye

{jylafaye, fbertain, cfaucher}@univ-lr.fr

1) RESUME

Notre contribution présente deux aspects.

- Le premier relève d'une problématique générale, à savoir l'intégration de différents espaces de représentation :

- L'espace relationnel assurant la persistance
- L'espace objet assurant l'exécution
- L'espace ontologique permettant le raisonnement

En cela nous rejoignons en partie le thème présenté par Catherine : « Intéropérabilité des SI et Bases de connaissances »

Les verrous en la matière concernent en particulier la gestion de l'*impedance mismatch* Objet/Ontologie (Monde fermé/Monde ouvert, nommage & identification, ...)

Nos objectifs englobent à la fois la mise en correspondance des éléments et des concepts, mais également la migration des données d'un espace vers un autre.

- Le second traite d'un cas applicatif dans un secteur particulier : la prise en charge d'un processus complet de gestion d'événements à caractère spatio-temporel dans le domaine des agences de presse, et des sociétés spécialisées dans l'événementiel loisir, tourisme, culture.

Dans ce contexte, les verrous que nous abordons concernent l'intégration de la langue naturelle et du raisonnement formel, la modélisation des événements spatio-temporels pseudo-périodiques et celle de la connaissance propre aux calendriers.

Notre stratégie générale consiste à considérer un modèle objet pivot puis à créer les modèles de correspondance, tissage et de traduction permettant d'intégrer ce modèle central avec les modèles spécifiques utilisés.

- Pour le premier aspect évoqué ci-dessus, c'est le modèle du domaine qui joue le rôle de pivot. Ce modèle est conforme au métamodèle d'UML et nous est donné par l'utilisateur.
- En ce qui concerne le second aspect, nous avons dû créer nos propres métamodèles (dans le respect des normes existantes) pour prendre en compte les événements et leurs propriétés temporelles. L'étude du processus d'intégration de ces métamodèles avec le modèle métier de l'utilisateur est un point important de notre recherche.

Dans tous les cas précédents, l'usage de l'IDM nous apporte des solutions dont l'architecture est explicite et claire. Nous gérons des mégamodèles (métamodèles, modèles, correspondances, tissage, transformation d'ordre supérieur, traducteurs). Nous proposons des solutions génériques automatiques qui opèrent sur les plans de l'ergonomie, de l'interface et de l'assistance utilisateur, de la validation et de la vérification.

Ces aspects sont à mettre en liaison avec le thème INFORSID : « adaptation du SI à l'utilisateur »

Outre les DSL que nous avons définis, nous utilisons conjointement les langages ATL, Kermeta, Java, Prolog. A terme, nous devons faire appel à des langages et moteurs de règles.

2) THEME : INTEGRATION OBJET/ONTOLOGIE : Grille_Problématique/Solutions

2.1) Vue Entreprise (Utilisation métier)

2.1.1) Problématiques

Sur la base d'un modèle du domaine, trois cas d'utilisation majeurs sont étudiés et décrits ci-dessous

CU_1 : Créer un point d'interrogation (end-point) SPARQL afin de publier des informations locales. Ce cas d'utilisation devient particulièrement d'actualité dans le contexte de promotion des expériences *Open Data* et *Linked Data*.

CU_2 : Importer des données issues du web pour peupler le modèle métier

CU_3 : Permettre à des composants métier d'accéder (en ligne) aux données du web de façon transparente *via* l'interrogation du modèle objet.

Critères de qualité : ergonomie, correction, réutilisabilité

2.1.2) Solutions

Proposer un environnement de conception qui permette successivement de :

- Construire un modèle du domaine enrichi et annoté de manière à faire apparaître les éléments de modélisation caractérisant les informations d'intérêt : accès ou publication. (Principalement par étiquetage, spécialisations, création d'attributs dérivés, ...)
- Générer une ontologie canonique par traduction directe du modèle du domaine enrichi en OWL (implémentation des principes de traduction d'ODM¹ : UML to OWL)
- Aligner les concepts sur les ontologies existantes (via les fonctionnalités intrinsèques d'OWL)
- Traduire les instances du modèle en triplets RDF constituant la base de connaissance associée à l'ontologie générée et *vice versa*.

2.2) Vue Académique (Recherche)

2.2.1) Problématiques

- Spécifier et implémenter un DSL qui permette l'annotation ergonomique et contrôlée du modèle objet initial
- Tracer les correspondances entre les éléments des différents métamodèles utilisés
- Outiller les modèles par des traducteurs (générés dynamiquement) pour réaliser l'import/export des données
- Résoudre les conflits et ambiguïtés (*impedance mismatch* liés aux paradigmes différents dans les mondes objet et ontologique : OWA/CWA, règles de nommage et d'identification, ...).

2.2.2) Solutions

Le modèle objet est considéré comme pivot, et deux stratégies sont traitées :

- *Stratégie_1 (à la volée)* : une requête externe exprimée sur le SPARQL end-point est traduite en requête sur le modèle objet étendu si les informations résident en mémoire. Dans les cas contraires, la traduction se fait vers HQL pour interroger par exemple une base relationnelle construite par ORM (eg. : Hibernate). Des solutions homologues permettent d'interroger des bases XML.

¹ Ontology Definition Metamodel : <http://www.omg.org/spec/ODM/1.0/>

- *Stratégie_2 (persistence)* : les informations du modèle objet enrichi sont traduites de manière à peupler la base de triplets RDF associée au SPARQL end-point qui dans ce cas résout directement les requêtes. Les références vers les ontologies publiques sont gérées (import et réutilisation de la d'ontologies publiques)

Le DSL inclut un interpréteur de contraintes OCL, langage utilisé *in fine* pour annoter le modèle objet initial. Il réalise la correspondance entre les éléments du modèle du domaine enrichi et ceux des ontologies cibles, ontologie canonique du SPARQL end-point dans un cas (export), ontologies publiques dans l'autre (import).

Le traducteur de requêtes SPARQL / HQL est bidirectionnel. Il permet donc également de répondre au besoin des applications métier cherchant, *via* le modèle du domaine, à accéder à des données du web. Dans ce cas, l'accessor aux données du modèle génère une requête SPARQL sur les bases de connaissances externes et applique les filtres adéquats pour retourner les éléments attendus.

3) THEME : GESTION D'ÉVÉNEMENTS SPATIO-TEMPORELS

3.1) Vue Entreprise (Utilisation métier)

3.1.1) Problématiques

Le domaine étudié est celui des loisirs de la culture et du tourisme. Le processus métier principal admet en entrée les textes bruts rédigés et publiés en langue naturelle par les agences de presse (e.g. : AFP). Les utilisateurs intermédiaires sont les organismes traitant puis diffusant de telles informations après les avoir mises en forme et intégrées. Les utilisateurs finals, naviguent sur les sites ainsi construits et alimentés ou s'abonnent à des flux RSS. L'application de notre approche à d'autres champs applicatifs sera évoquée plus bas.

On distingue deux phases principales et successives :

Phase_1 : Annotation des événements et intégration dans une base de connaissance

Phase_2 : Publication des informations (passive : portails web + outils de navigation et de visualisation) ou active : construction personnalisée des flux RSS)

La Phase_1 est réalisée par un humain assisté par un logiciel d'annotation. Un prétraitement des dépêches extrait automatiquement (TAL) des éléments d'information annotés qui identifient des événements, des entités nommées (lieux, acteurs) et des informations temporelles. Ces résultats sont présentés à l'« expert » qui peut interroger le système pour obtenir des informations complémentaires et renseigner en connaissance de cause une application chargée de peupler de manière incrémentale le système en question par de nouveaux faits.

Le rôle de l'expert est de vérifier la cohérence des annotations, de lever des ambiguïtés et de compléter les informations. Il est nécessairement assisté dans ces tâches complexes. L'intégrité du système doit être garantie.

La Phase_2 consiste à connecter la base de connaissance avec des applications dédiées à la visualisation et à la diffusion. Cette phase est réalisée en interrogeant la base *via* le modèle pivot, en tenant compte des profils des utilisateurs finals.

Le champ applicatif (tourisme/loisir) choisi pour démontrer notre contribution n'est nullement exclusif. D'autres domaines peuvent tirer parti de nos apports dans la gestion des

propriétés temporelles d'événements répétitifs. Citons par exemple l'orchestration de processus métiers : les modélisations UML et BPMN définissent des *timers* dont les propriétés temporelles sont exprimées en texte libre dans des notes. Nous rendons ces spécifications opérationnelles en permettant le traitement automatique de la sémantique de telles expressions temporelles. Des traducteurs idoines peuvent prendre ces spécifications en entrée et générer automatiquement des systèmes d'horloges (e.g. : LUSTRE, MARTE, ...) ouvrant ainsi la porte à la vérification formelle de propriétés logiques complexes.

Un dernier exemple concerne la numérisation de règles issues de textes de droit public ou privé, régissant la pratique - autorisation, interdiction, limitation - de diverses activités de façon récurrente (pêche, chasse, consultation électorale, accès à une ressource, ...). Un organisme de contrôle peut interroger le système de règles temporelles en ligne afin de juger de la légalité d'une activité donnée.

3.1.2) Solutions

Phase_1 : La base de connaissance contient l'information courante sur les éléments traités (événements, acteurs, lieux, dates). Elle intègre également un ensemble de règles et de contraintes d'intégrité sémantique sur la cohérence des aspects structurels et temporels.

L'aide à l'expert se fonde sur l'exploitation de cette connaissance ; des éléments de décision lui sont suggérés, ses choix sont validés. L'interaction avec l'expert doit s'accomplir dans un langage proche du langage naturel. Il s'agit d'un expert métier, non d'un expert technique (en IDM, logique ou ontologies).

Phase_2 : Le mot clé de cette phase est l'automatisation. On cherche à faciliter au maximum la conception de l'interface entre la base de connaissance et les applications dédiées à la visualisation et à la diffusion. Le paramétrage des composants est simplifié et exploite les profils utilisateurs.

3.2) Vue Académique (Recherche)

3.2.1) Problématiques

Même si l'on sépare les notions d'information générale, spatiale et temporelle que l'on peut considérer comme des métadonnées qui lui sont attachées, la notion d'événement reste complexe dans le domaine qui nous intéresse. On doit nécessairement prendre en compte des aspects structurels mettant en jeu des niveaux de granularité différents (événement, sous événement, ...). Chaque niveau est décrit par ses propres métadonnées générant naturellement des contraintes d'intégrités tant *intra* qu'*inter* niveaux. En particulier, le structurel et le temporel ne peuvent être validés indépendamment.

Du point de vue temporel, les événements sont souvent approximativement périodiques. Gérer la série des dates concrètes périodiques d'occurrence d'un événement (extension) est coûteux et peu efficace, une spécification des occurrences en intention est préférable.

La connaissance sur les contraintes temporelles est en grande partie encapsulée dans la notion de calendrier. La sémantique du calendrier peut varier selon les contextes (pays, cultures, époques).

Les relations d'Allen donnent un cadre rigoureux au raisonnement temporel, mais ne peuvent en l'état traiter des questions habituelles dans notre domaine applicatif, telles que la position relative d'événements répétitifs (« les mardis suivent les lundis », chaque année « mai précède juin », etc)

3.2.2) Solutions

Métamodèles de base

Nous proposons un modèle d'événements composite qui intègre en les différenciant les décompositions structurelles et temporelles. Par exemple un événement, les Francofolies, peut être considéré comme un ensemble de spectacles de différents types (structurel) ou comme une série de manifestations annuelles (temporel). Le modèle inclut la représentation de relations entre les événements en gérant la cohérence de leur sémantique.

Les métadonnées et les propriétés spatiales sont représentées en référence à des normes (ISO_19115)

Pour le temporel, nous nous appuyons sur également sur les normes en vigueur (ISO_19108, iCal, OWL-Time) qui partagent les concepts fondamentaux que nous réutilisons.

Il a été nécessaire de définir de nouveaux métamodèles pour représenter les propriétés temporelles de façon formelle, tout en restant proche du langage naturel. En particulier, nous attachons une importance prédominante à la représentation des expressions temporelles « en intention ». On peut ainsi directement exprimer des contraintes comme : « entre le 21 mars 2010 et le 20 mars 2013, le premier samedi de chaque mois de 14h à 17h30 »

L'association entre un événement et une expression temporelle indique le type de la propriété concernée. Par exemple pour une exposition : ouverture, fermeture, accès restreint, ...)

Intégrité sémantique

La conjonction de ces deux métamodèles (événement, propriétés temporelles) constitue le cœur de la solution proposée. La conformité à ces métamodèles garantit la correction syntaxique des modèles construits. Un ensemble de contraintes OCL permet de garantir une partie de la correction sémantique, mais le pouvoir expressif reste limité.

La majeure partie de la connaissance temporelle réside dans la notion de calendrier.

Nous modélisons les éléments calendaires comme des événements accompagnés de leurs propriétés temporelles. Pour ce faire, nous utilisons naturellement les métamodèles précédents. Pour modéliser la sémantique temporelle complète des calendriers, nous avons été amenés – dans la continuité des travaux de G. Ligozat - à définir une extension des relations d'Allen qui s'applique aux intervalles non convexes (propres à modéliser les concepts tels que les jours nommés : {lundis}, {mardis}, ... ou les mois nommés : {janviers}, {févriers})

La vérification sémantique des propriétés temporelles des événements modélisés s'opère en interrogeant le modèle de calendrier et en appliquant des règles de cohérence.

Interface entre les modèles pivot et les applications externes

Pour manipuler les éléments de modélisation et les instances, il convient de disposer d'un langage simple, expressif et non ambigu. Nous avons spécifié une grammaire formelle qui est directement issue du métamodèle d'expressions temporelles de base présenté plus haut, et qui réutilise les termes du langage naturel. Ce langage permet de retourner un écho des informations modélisées sous une forme familière à l'utilisateur, que ce soit dans une optique d'interrogation ou de peuplement du modèle.

Si l'on considère la phase amont (Phase_1) du processus métier, une partie du peuplement du modèle peut se faire par intégration automatique des résultats fournis par le TAL. La connaissance enregistrée dans le modèle permet d'assister l'expert dans sa tâche de complétion des informations (suggestion de valeurs par défaut, contrôle, ...). A la fin de cette phase, l'événement renseigné et ses propriétés temporelles sont formulés à l'expert explicitement *via* la grammaire pour validation puis enregistrés dans le système.

En ce qui concerne la seconde phase, par exemple quant au processus de visualisation, nous avons développé des stratégies de construction et de paramétrage semi automatiques. Prenant comme cible une bibliothèque de composants graphiques (e.g. Simile Exhibit), nous utilisons la réflexivité du code pour scruter les méthodes des différents widgets disponibles. Le résultat de cette analyse permet de construire des métamodèles de configuration structurés selon trois préoccupations que sont : la disposition de l'interface (layout), les composants graphiques (widgets) et les données utilisées par ces composants. Des modèles permettent également à un utilisateur humain de spécifier la correspondance entre les éléments de visualisation et les éléments du modèle du domaine. Ceci fait, les différents modèles instances des métamodèles permettent de générer des familles *ad hoc* d'applications web Simile Exhibit.

L'approche IDM indispensable pour gérer la complexité.

Il apparaît clairement que le nombre et l'hétérogénéité de métamodèles à gérer dans le processus décrit ci-dessus témoigne d'une complexité importante. L'usage de l'IDM permet de séparer les aspects et les niveaux de modélisation. Les principes de transformation et de tissage de modèles clarifient la structure du code.

Métamodèles utilisés (extrait)

- Sorties du traitement TAL (éléments + annotations)
- Interface de dialogue avec l'expert
- Événements et propriétés temporelles
- Expressions temporelles (modèle objet)
- Expressions temporelles (grammaire formelle)
- Éléments du calendrier
- Propriétés temporelles des éléments du calendrier
- Règles de correction et de cohérence (domaine temporel)
- Widgets de visualisation (propriétés, méthodes)

Traducteurs

- TAL vers Métamodèles (Événements, propriétés temporelles)
- Events ML vers Métamodèles (Événements, propriétés temporelles)
- Métamodèle temporel vers grammaire formelle (et réciproque)
- Métamodèle temporel vers iCalendar
- Métamodèles (Événements, propriétés temporelles) vers base de connaissance RDF

Utilitaires

- Interrogation et peuplement du modèle d'événements + propriétés temporelles
- Interrogation du modèle de calendrier
- Editeur textuel contraint pour l'édition de propriétés temporelles
- Configurateur d'application web Simile Exhibit

Formats intermédiaires d'échange de données

- XML, JSON, RDF