



GROUPE DE RECHERCHE INTERDISCIPLINAIRE
EN INFORMATIQUE DE LA SANTÉ

Conception du module « ontorela-application »

Équipe du GRIIS

2024-12-03

Table des matières

.....	iv
Rôle du module	1
Implémentation du module	2
Modèle propre au module	3
Type au module	4
Type simple de donnée	4
Type complexe de donnée	4
OntologyConfiguration	4
DatabaseConfiguration	4
Type de message	6
Hypothèse du module	7
Contrainte du module	8
Fonction du module	9
Fonction « Générer OntoRel »	9
Comportement	9
Interface	10
Erreur	10
Fonction « Génération des graphes »	11
Comportement	11
Interface	11
Erreur	11
Fonction «Générer les rapports d'anomalies »	12
Comportement	12
Interface	12
Erreur	12
Fonction « build OntoRela »	12
Comportement	12
Interface	14
Erreur	15
Incongruités	16
Matrice de spécification	17
Matrice de dépendances	18
Annexes	19
Configuration ontologique	19
Configuration de base de données	19
Règle de conversion de composants ontologiques	19
Fichier d'anomalie de base de données	20

Liste des tableaux

1. Statut	iv
2. Historique	iv
3. Référence	iv
4. Type simple	4
5. Format OntologyConfiguration	4
6. Format DatabaseConfiguration	5
7. matrice de spécification	17
8. matrice de dépendances	18

Tableau 1. Statut

Statut	Initiales	Date
Conception prête à être révisée	AS	2024-12-03 13:30
Révision du spécificateur réalisée	MD	2024-12-06 09:00
Révision de l'architecte réalisée	FO	2024-12-09 14:30
Révision du concepteur des tests réalisée	SSC	2024-12-20 12:21

Tableau 2. Historique

Changement important	Initiales	Date
Conception initiale	AS	2024-11-28 10:00
Correction mineures	AS	2024-12-20 11:51

Tableau 3. Référence

Document	Consultation
ontorela-application-sas	2024-12-05 10:00
ontorela-application-syrs	2024-12-06 08:45

Rôle du module

Le `ontorela-application` permet de générer le modèle relationnel OntoRel à partir d'une configuration et d'une ontologie, de produire les graphes, rapports d'anomalies, et scripts SQL associés.

Implémentation du module

S.O.

Modèle propre au module

S.O.

Type au module

Type simple de donnée

Tableau 4. Type simple

Champs	Description
String date	la date sous forme de chaine yyyyMMdd-HHmm
File repo	le répertoire actuel
String outputPath	repo + "/" + date + "/"

Type complexe de donnée

OntologyConfiguration

Un type java décrivant la configuration de l'ontologie. Ce type est attendu être équivalent au schéma YAML du même objet (cf. annexe Exemple de Configuration).

Tableau 5. Format OntologyConfiguration

Champs	Description
String owlFilePath	Chemin de l'ontologie (local ou URL)
String configDirectoryPath	Chemin de dossier de configuration
boolean loadEquivalentClass	La référence pour les classes équivalentes. Si vrai : Charger les classes équivalentes. Si Faux : Aucune classes équivalentes.
AnnotationConfig annotations	Liste des annotations d'intérêt spécifiées par type et par langue (facultative).
Set<String> classes	Ensemble des classes d'intérêt (facultative).
Set<String> objectProperties	Ensemble des propriétés d'intérêt(facultative).

DatabaseConfiguration

Un type java décrivant la configuration de la base de données. Ce type est attendu être équivalent au schéma YAML du même objet. (cf. annexe Exemple de Configuration).

Tableau 6. Format DatabaseConfiguration

Champs	Description
String schemaName	Le nom du schéma de base.
List<String> rdbmsName	La liste de noms des SGBD cible.
String defaultKeyName	Le nom de la clé artificielle.
String defaultKeyDomainName	Le nom du domaine de la clé artificielle.
String defaultKeyType	Le nom du type de la clé artificielle.
String defaultAttributeName	Le nom de l'attribut de valeur par défaut.
String defaultAttributeDomainName	Le nom du domaine de l'attribut de valeur par défaut.
String defaultAttributeType	Le nom du type de l'attribut de valeur par défaut.
int maxIdentifierLength	La longueur maximale des identifiants de la base de données. Sinon définie, retourne 30.
boolean useIriAsTableId	La référence pour la création des identifiants des tables. Si vrai : l'identifiant des tables est construit en utilisant l'IRI locale (short IRI). Si faux : l'identifiant des tables est construit en utilisant une séquence interne.
boolean normalizeDatatype	La référence pour la normalisation des types. Si vrai : chaque type est converti en une table (clé-valeur). Si faux : chaque type est converti en attribut dans la table de jointure.
boolean generateOpTable	La référence pour génération des tables de propriétés de classe. Si vrai : crée une table pour chaque propriété de classe. Si faux : Aucun table pour les propriétés de classe.
boolean removeThingTable	La référence pour la suppression de la table Thing Si vrai : la table Thing n'est pas créée. Si faux : la table Thing n'est pas créée. Par défaut : la table Thing est créée.
List<String> languages	Ensemble des langues d'intérêts.
String owlSqlTypeMapperFilePath	L'emplacement du fichier de correspondance des types OWLSQL.
String ontoRelDicFilePath	L'emplacement du dictionnaire de données d'un ontoRel.
String configDirectoryPath	Le chemin du répertoire qui contient les fichiers de configuration.

Type de message

S.O.

Hypothèse du module

- [HYP-001] Les ressources système sont suffisantes pour supporter l'utilisation d'ontologies de grande taille.
- [HYP-002] Avoir accès a un SGBD.

Contrainte du module

- [CON-001] Utiliser `JgraphT` pour la génération de graphe.
- [CON-002] Les étapes de génération des éléments d'OntoRel définies dans la SAS et décrites par les processus ci-dessous doivent être appliquées dans l'ordre.
- [CON-004] Utiliser `monto_library` pour la génération d'ontologie normalisée et filtrée et les représentations d'ontologie(OWL, ttl, μ Onto, Dot) .
- [CON-005] Utiliser `ontorela_library` pour la génération et/ou exécution des fichiers SQL.
- [CON-006] Utiliser `ontorela_library` pour la génération d'une représentation d'une base de données sous forme de graphe.

Fonction du module

- [PRO-1] Construit une base de données relationnelle et génère les scripts SQL `generate-associés.ontoRel`
- [PRO-2] Construit un graphe d'ontologie (et/ou) de base de donnée (et/ou) `generate-ontologique-relationnelle` à partir d'ontorel. `graph`
- [PRO-3] Génère un rapport d'anomalies pour l'ontologie et/ou la base de données. `generate-données.anomalies-report`
- [PRO-4] Génère l'ensemble PRO-1, PRO-2, PRO-3. `build-ontorela`

Fonction « Générer OntoRel »

Comportement

- [PRO-1.1] Générer l'ontologie normalisée :
- utiliser le constructeur `OntologyOwlApiBuilder` de `monto-library` pour créer l'ontologie normalisée selon la configuration ontologique.
- [PRO-1.2] Filtrer l'ontologie:
- utiliser la fonction `filterOntology` de `monto-library` pour filtrer l'ontologie selon la configuration ontologique.
- [PRO-1.3] Générer l'ontologie en format monto (utiliser `monto-library`)
- utiliser la fonction `generateMonto` de `monto-library`.
- [PRO-1.4] Générer l'ontologie en format owl (utiliser `monto-library`)
- utiliser la fonction `generateMontoOwl` de `monto-library`.
- [PRO-1.5] Convertir des composants ontologiques en composant relationnel selon les choix dans la configuration (voir annexe pour les règles de conversion).
- utiliser `buildOntoRel` de `ontorela-library`.
- [PRO-1.6] Générer les fichiers SQL selon le dialecte choisi dans le fichier de configuration.
- utiliser la fonction `genereteScripts` de `ontorela-library`.
 - le SGDB est défini dans la configuration de base de données par `rdbmsName`
 - le nom du schéma est défini dans la configuration de base de données par `schemaName`

Interface

Usage	Entrée	Sortie
java -jar OntoRelA-<version>-SNAPSHOT.jar	String repo_path Chemin du répertoire contenant les fichiers de configurations ontologiques et de base de données.	Les fichiers d'ontologie: <repo_path> <nom_ontology>_normalized.ttl : l'ontologie normalisée et filtrée en format ttl <nom_ontology>_normalized.txt : l'ontologie normalisée et filtrée en format owl Les fichiers de scripts du schéma ontorel: 100-<nom_ontology>_create_table_<version>_<date>.sql 110-<nom_ontology>_create_participationCheck_<version>_<date>.sql 120-<nom_ontology>_create_unionAxiomCheck_<version>_<date>.sql 130-<nom_ontology>_create_membershipCheck_<version>_<date>.sql 200-<nom_ontology>_create_view-iri_<version>_<date>.sql 210-<nom_ontology>_create_view-en_<version>_<date>.sql 210-<nom_ontology>_create_view-fr_<version>_<date>.sql 800-<nom_ontology>_delete_table_<version>_<date>.sql 920-<nom_ontology>_drop_table_<version>_<date>.sql

Erreur

- FileNotFoundException : Fichier de configuration introuvable dans repo_path.
- OWLOntologyCreationException : Échec de génération de l'ontologie.
- OntorelCreationException : Échec de génération de l'ontorel.

Fonction « Génération des graphes »

Comportement

- [PRO-2.1] Générer un graphe orienté de l'ontologie initiale :
 - utiliser le constructeur `OntoGraph` de `monto-library` pour créer un graphe à partir de l'ontologie initiale.
- [PRO-2.2] Générer le graphe orienté de l'ontologie normalisée.
 - utiliser le constructeur `OntoGraph` de `monto-library` pour créer un graphe à partir de l'ontologie normalisée.
- [PRO-2.3] Générer le graphe orienté de la base de données.
 - utiliser `createDotFile` `ontorela-library` pour créer le graphe
 - définir un nœud pour chaque table.
 - définir une arrête pour chaque clé référentielle.
- [PRO-2.4] Générer le graphe orienté de l'ontorel.
 - utiliser `ontoRelGraph` et `createDotFile` de `ontorela-library`

Note

Les fichiers sont généré sous: `outputPath + "Graphs/"`

Interface

Usage	Entrée	Sortie
<code>java -jar OntoRelA-<String>-SNAPSHOT.jar</code>	<code>String repo_path</code> Chemin du répertoire contenant les fichiers de configuration. <code>String type_graph</code> : (optionnel) le type de graphe à générer. Si aucune valeur n'est spécifiée, tous les graphes seront générés.	Les fichiers des graphes: <code>File InitialOntoGraph.dot</code> : graphe de l'ontologie initiale <code>File OntoGraph.dot</code> : graphe de l'ontologie normalisée <code>File RelGraph.dot</code> : graphe de la base de données <code>File OntoRelGraph.dot</code> : graphe d'Ontorel

Erreur

- `FileNotFoundException` : Fichier de configuration introuvable dans `repo_path`.
- `IllegalArgumentException`: Erreur liée à la création de fichiers ou `type_graph` invalide.

Fonction «Générer les rapports d'anomalies »

Comportement

- [PRO-3.1] Générer les anomalies de l'ontologie.
 - utiliser `generateAnomaliesReport` de `monto-library`, les anomalies sont généré apartir de l'ontoGraph produit dans PRO-2.2.
- [PRO-3.2] Générer les anomalies de la base de données.
 - utiliser `generateDatabaseReport` de `ontorela-library` pour générer les anomalies suivante:
 - Noms et des identifiants d'attributs longs : si le nom ou l'identifiant de la table dépasse `maxSize`, définir une anomalie de type `LongIdentifier`: La longueur maximale des identifiants de la base de données sont défini dans la configuration de base de données `maxIdentifierLength`.
 - Tables sans clé primaire.
 - Clés étrangères orphelines.
 - Noms de colonnes dupliqués.

Note

Le format de fichier est défini dans l'annexe. Générer les fichiers d'anomalies sous `outputPath`.

Interface

Usage	Entrée	Sortie
<code>java -jar OntoRelA-<version>-SNAPSHOT.jar</code>	<code><repo_path></code> Chemin du répertoire contenant les fichiers de configuration. <code>String anomaly_type</code> : (optionnel) le type d'anomalie à générer.	<code>Les fichiers d'anomalies-reports <repo_path></code> d'anomalies: <code>File</code> <code>OntologyAnomalies.txt</code> : anomalies d'ontologie <code>OntoRelDiagnostics.txt</code> : anomalies de l'ontorel

Erreur

- `FileNotFoundException` : Fichier de configuration introuvable dans `repo_path`.
- `IllegalArgumentException`: Erreur liée à la création de fichiers ou `anomaly_type` invalide.

Fonction « build OntoRela »

Comportement

- [PRO-4.1] Générer l'ontorel.

[PRO-4.2] Générer les graphes.

[PRO-4.3] Générer les rapports d'anomalies.

Interface

Usage	Entrée	Sortie
java -jar OntoRelA-<version>-SNAPSHOT.jar	String rep_onto : Chemin du répertoire contenant les fichiers de configuration.	<p>Les fichiers d'ontologie</p> <p><nom_ontology>_normalized.ttl : l'ontologie normalisée et filtrée en format ttl</p> <p><nom_ontology>_normalized.txt : l'ontologie normalisée et filtrée en format owl</p> <p>Les fichiers de scripts du schéma ontorel:</p> <p>100-<nom_ontology>_create_table_<version>_<date>.sql</p> <p>110-<nom_ontology>_create_participationCheck_<version>_<date>.sql</p> <p>120-<nom_ontology>_create_unionAxiomCheck_<version>_<date>.sql</p> <p>130-<nom_ontology>_create_membershipCheck_<version>_<date>.sql</p> <p>200-<nom_ontology>_create_view-iri_<version>_<date>.sql</p> <p>210-<nom_ontology>_create_view-en_<version>_<date>.sql</p> <p>210-<nom_ontology>_create_view-fr_<version>_<date>.sql</p> <p>800-<nom_ontology>_delete_table_<version>_<date>.sql</p> <p>920-<nom_ontology>_drop_table_<version>_<date>.sql</p> <p>Les fichiers des graphes:</p> <p>File InitialOntoGraph.dot : graphe de l'ontologie initiale</p> <p>File OntoGraph.dot : graphe de l'ontologie normalisée</p> <p>File RelGraph.dot : graphe de la base de données</p> <p>File OntoRelGraph.dot : graphe d'Ontorel</p> <p>Les fichiers d'anomalies:</p> <p>File OntologyAnomalies.txt : anomalies d'ontologie</p> <p>File OntoRelDiagnostics.txt : anomalies de l'ontorel</p>

Erreur

- `FileNotFoundException` : Fichier de configuration introuvable dans `repo_path`.
- `OWLOntologyCreationException` : Échec de génération de l'ontologie.
- `OntorelCreationException` : Échec de génération de l'ontorel.

Incongruités

- [INC-001] La conception actuelle n'inclut pas la génération du catalogue `OntoRelCat`, ni les détails ou processus spécifiques pour générer ou gérer ce dictionnaire. Cela sera couvert prochainement dans une conception détaillée.
- [INC-002] La génération de documentation est actuellement limitée à la création de fichiers texte pour les anomalies, une amélioration future est envisageable.
- [INC-003] La seule façon de choisir le mode de normalisation des types, la génération de la table « Thing », etc., repose sur le fichier de configuration de base de données.

Matrice de spécification

Tableau 7. matrice de spécification

source / destination	EX.001	EX.002	EX.003	EX.004	EX.005	EX.020	EX.100	EX.300	EX.400
PRO-1.1							O		
PRO-1.2							O		
PRO-1.4							O		
PRO-1.5	O	O	O	O	O				
PRO-2.1								O	
PRO-2.2								O	
PRO-2.3								O	
PRO-2.4						O			
PRO-3.1									O
PRO-3.2									O

Matrice de dépendances

Tableau 8. matrice de dépendances

source / destination	Aucune	PRO-1.1	PRO-1.2	PRO-1.3	PRO-1.5	PRO-1.6
PRO-1.1	O					
PRO-1.2		O				
PRO-1.4			O			
PRO-1.5				O		
PRO-2.1	O					
PRO-2.2				O		
PRO-2.3						O
PRO-2.4					O	
PRO-3.2					O	

Annexes

Configuration ontologique

Exemple de Configuration.

```
owlFilePath: test-data/_ontologies/ABC/ABC.owl
loadEquivalentClass: false
annotations:
  labels:
    - "http://www.w3.org/2000/01/rdf-schema#label"
    - "http://purl.obolibrary.org/obo/BFO_0000179"
  definitions:
    - "http://www.w3.org/2000/01/rdf-schema#comment"
    - "http://purl.org/dc/elements/1.1/description"

classes:
  - "http://ca.griis.ontologies/ABC#W0"
  - "http://ca.griis.ontologies/ABC#Z0"
  #
  - "http://ca.griis.ontologies/ABC#A"
  - "http://ca.griis.ontologies/ABC#B"
  - "http://ca.griis.ontologies/ABC#C"
```

Configuration de base de données

Exemple de Configuration.

```
schemaName: ABC
rdbmsName:
  - postgresql
defaultKeyName: uid
defaultKeyDomainName: uid_domain
defaultKeyType: UUID
defaultAttributeName : value
defaultAttributeDomainName: value_domain
defaultAttributeType: TEXT
maxIdentifierLength: 60
useIriAsTableId: true
normalizeDatatype: true
generateOpTable: true
removeThingTable: false
languages:
  - en
  - fr
configDirectoryPath: test-data/configurations
owlSqlTypeMapperFilePath: test-data/configurations/config01/owlsqltype_c
ontoRelDicFilePath: test-data/ontoRelDic/OntoRelDic_wOType.json
```

Règle de conversion de composants ontologiques

- Conversion des types en domaines : créer un domaine pour chaque type ontologique, l'identifiant du Type = ShortIRI+"_domain"

- Conversion des classes ontologiques en tables:
 - Créer les tables dérivées des classes et les contraintes référentielles dérivées des axiomes d'héritage de classes.
 - Créer une table par classe ontologique.
 - Si `RemoveThingTable` est vrai: Si la classe a comme sub-classe ou sous-classe `Thing` la contrainte référentielle ne sera pas créée.
 - Si non ajouter une contrainte référentielle vers la table mère.
- Conversion des propriétés de classes en table de jointure :
 - Créer les tables dérivées des propriétés de classes et les contraintes référentielles dérivées des axiomes d'héritage de propriétés.
 - Convertir les propriétés de classes en table de jointure dans le cas où la configuration `generateOpTable` est vrai
 - Créer les tables range et domaine si la propriété a un range et un domaine et créer une contrainte référentielle vers les tables range et domaine.
- Conversion des axiomes de données en table: Si `NormalizeDatatype` est vrai :
 - convertir les axiomes de données en table de jointure :
 - Si l'axiome possède déjà une table de correspondance, ne pas créer une nouvelle table.
 - Si non récupérer les tables déterminantes et dépendantes (les créer s'ils n'existent pas),
 - créer la table de jointure et ajouter les contraintes référentielles vers les tables déterminant et dépendant.
- Convertir les axiomes de classes en tables : Créer les tables dérivées des axiomes d'association de classe incluant les contraintes référentielles et les contraintes de participations.

Fichier d'anomalie de base de données

Format.

Database Anomalies Report

Note : If an identifier is too long, the IRI identifiers will be used.

Long Identifier Anomalies

Type: LongIdentifier

- Table: nom_table_1
 - * Attribute: nom_attribut1
 - * Attribute: nom_attribut2
- Table: nom_table_2
 - * Attribute: nom_attribut3

Tables Without Primary Key
Type: TableWithoutPrimaryKey
- Table: nom_table_3
- Table: nom_table_4

Orphan Foreign Keys
Type: ForeignKeyOrphan
- Foreign Key: nom_fk_1 (Table: nom_table_5)
- Foreign Key: nom_fk_2 (Table: nom_table_6)

Duplicate Column Names
Type: DuplicateColumnName
- Table: nom_table_7
* Attribute: nom_attribut1
* Attribute: nom_attribut2
- Table: nom_table_8
* Attribute: nom_attribut3