

修士論文に向けて

南山大学大学院 数理情報研究科 数理情報専攻

M2012MM002

朝倉知也

シナリオ

- 修士論文題目と内容
- OSLCのドメインと管理対象
- 考慮すべき点
- 今後の予定
- Apache Jenaを用いたモデル操作

修士論文題目と内容

- 論文題目:

OSLC^[1]に基づくソフトウェア変更要求管理方法の提案

- 背景と問題点

ソフトウェアの大規模化によるグローバルな分散開発

⇒ソフトウェアの**変更要求管理**に着目

1. 要求や仕様, コードに対する変更要求の**統一的な表現**が未定義
2. Webを介して**変更要求を管理する手法**が未確立

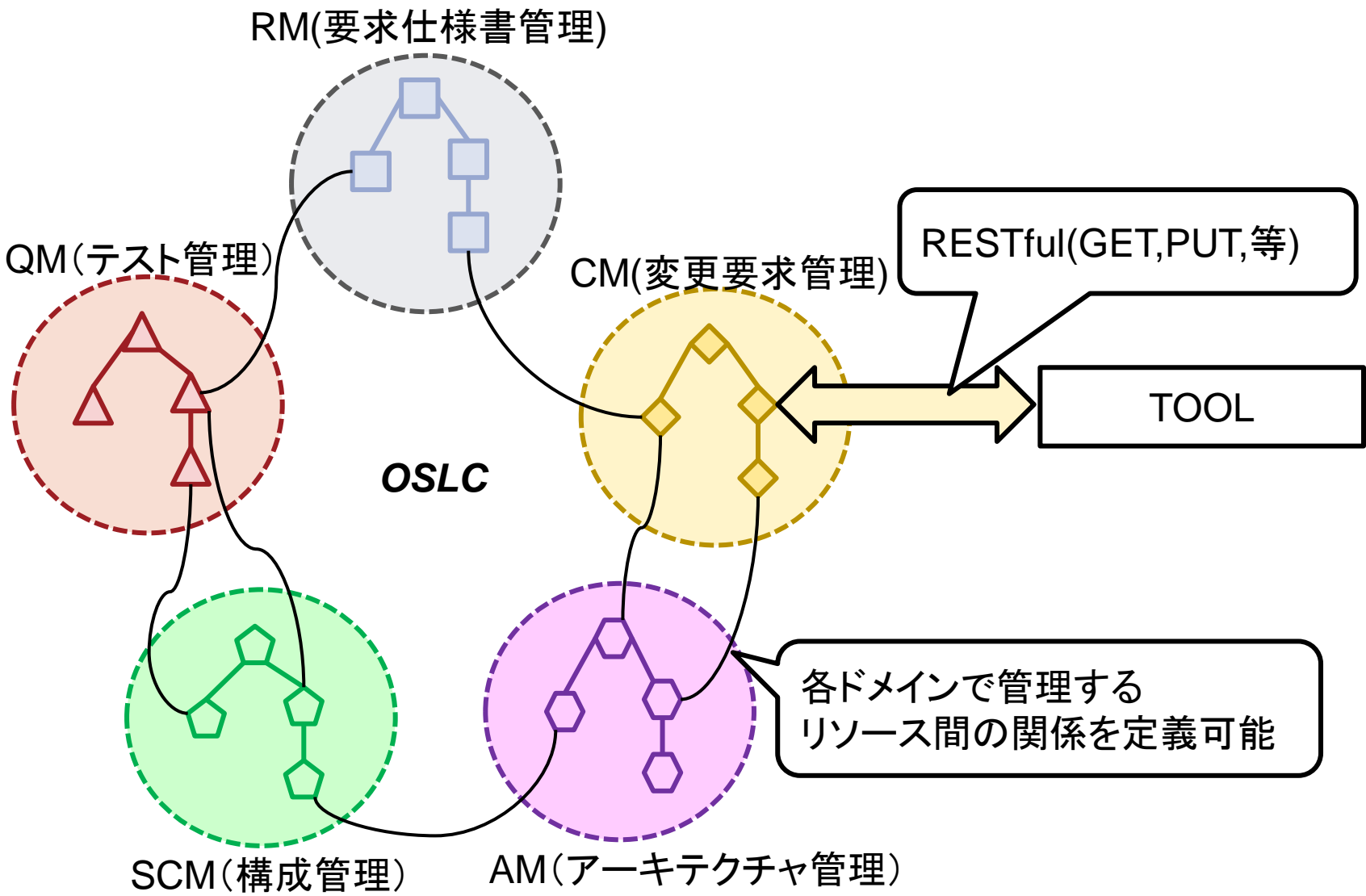
- アプローチ

1. OSLCに基づきソフトウェア変更要求モデルを定義
2. TRSを用い, Webを介した変更要求の管理

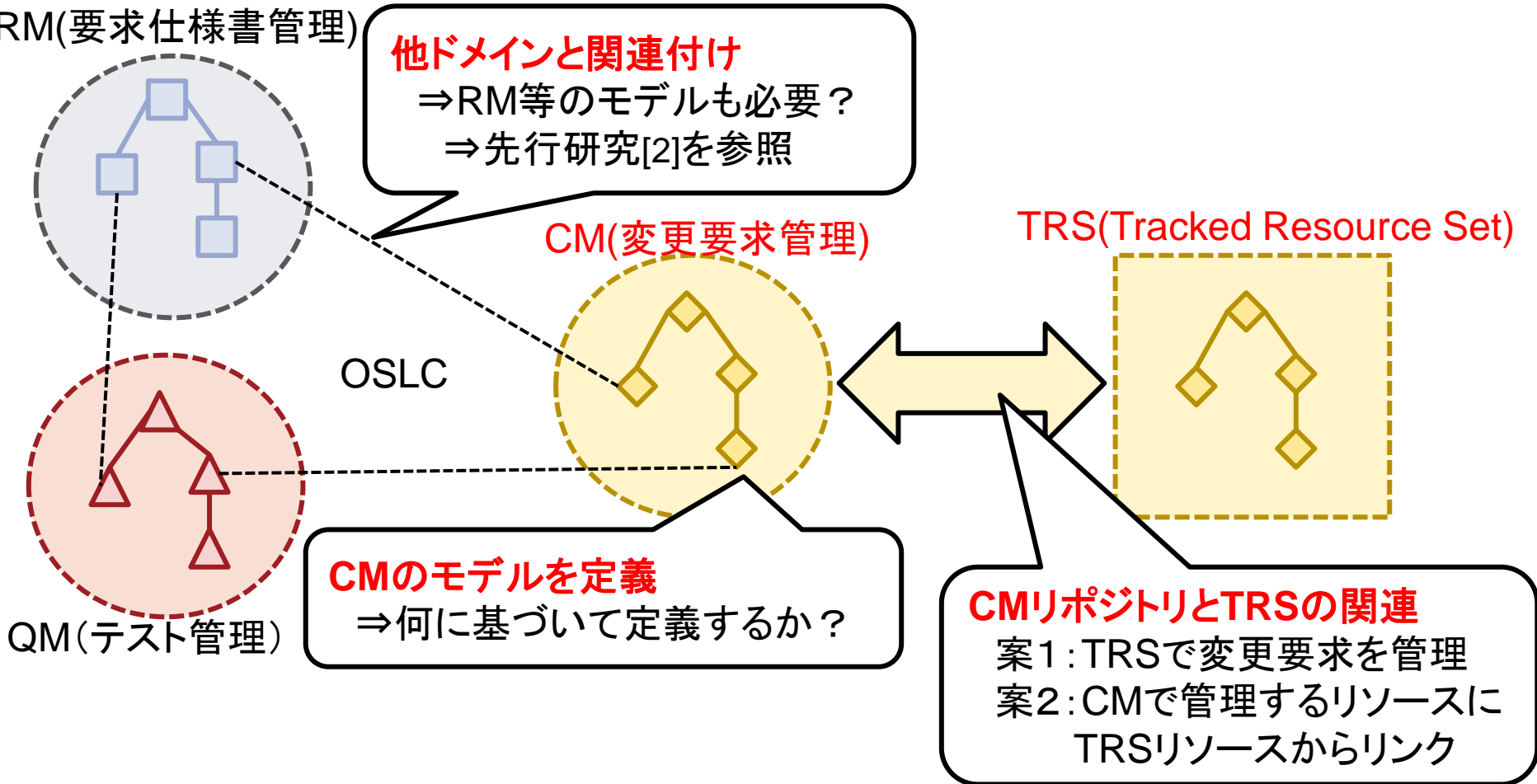
[1]OSLC(Open Services for Lifecycle Collaboration),<http://open-services.net/html/Home.html>

OSLCのドメインと管理対象

- 5つのドメイン



考慮すべき点



[2] 壁谷 考洋, 中道 上, 青山 幹雄, OSLCに基づく要求管理方法の提案と評価, 情報処理学会第75回全国大会 講演論文集, Vol. 1, No. 2M-6, Mar. 5-7, 2013, 東北大学, pp. 411-412.

今後の予定

● 予定スケジュール

2013年	6月～12月	関連研究調査
	6月～7月	OSLCの変更管理方法及びTRSの調査
	8月～9月	変更要求モデルの定義, プロトタイプ作成
	9～10月	中間発表の準備
	10月～11月	プロトタイプ作成
	11月～12月	プロトタイプ評価

● 現在行っていること

- 各OSLCドメイン, TRSの調査
 - ✓ 各ドメインのプロパティ定義
 - ✓ サンプルリポジトリのソース確認
- RDFモデル操作方法
 - ✓ Apache Jena[3]を用いてRDFモデルを作成・解析

Apache Jenaを用いたモデル操作(1/2)

1. Jenaを用いたRDFモデルの作成

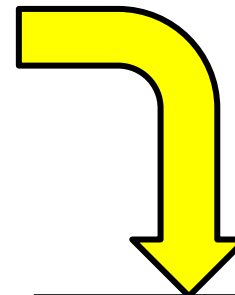
```
public static void create(String BASE_URI) throws FileNotFoundException{  
    > Model model = ModelFactory.createDefaultModel();  
    > String ns_foaf = "http://xmlns.com/foaf/0.1/";  
    > String ns_al = "http://example.com/";  
    > model.setNsPrefix("foaf", ns_foaf);  
  
1 { Resource AoyamaLab = model.createResource(ns_al + "aoyamalab");  
    AoyamaLab.addProperty(FOAF.homepage, "http://lab.nise.org/");  
    AoyamaLab.addProperty(RDF.type, FOAF.Group);  
  
2 { Resource takagi = model.createResource(ns_al + "takagi");  
    takagi.addProperty(FOAF.birthday, "09-18");  
    AoyamaLab.addProperty(FOAF.member, takagi);  
  
3 { Resource asakura = model.createResource(ns_al + "asakura");  
    asakura.addProperty(FOAF.birthday, "08-09");  
    AoyamaLab.addProperty(FOAF.member, asakura);  
  
    model.write(System.out, "RDF/XML-ABBREV");  
    model.write(new FileOutputStream("testmodel_10.rdf"), "RDF/XML-ABBREV");  
}
```

Jenaを用いたjavaプログラム

1. 基本リソースの作成, プロパティ設定
2. リソースの作成と設定, 基本リソースへの追加
3. モデルの出力

• Jena

- ✓ Javaのフレームワーク
- ✓ RDFモデルの作成, 検索
- ✓ Vocabulary: RDF, FOAF, OWL, 等



```
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:foaf="http://xmlns.com/foaf/0.1/"  
  <foaf:Group rdf:about="http://example.com/aoyamalab">  
    <foaf:member>  
      <rdf:Description rdf:about="http://example.com/asakura">  
        <foaf:birthday>08-09</foaf:birthday>  
      </rdf:Description>  
    </foaf:member>  
    <foaf:member>  
      <rdf:Description rdf:about="http://example.com/takagi">  
        <foaf:birthday>09-18</foaf:birthday>  
      </rdf:Description>  
    </foaf:member>  
    <foaf:homepage>http://lab.nise.org/</foaf:homepage>  
  </foaf:Group>  
</rdf:RDF>
```

作成されたRDFモデル(testmodel_10.rdf)

Apache Jenaを用いたモデル操作(2/2)

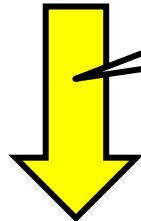
2. Jenaを用いたSparqlクエリ実行によるモデル解析

```
<rdf:RDF ↓
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" ↓
  xmlns:foaf="http://xmlns.com/foaf/0.1/" ↓
  <foaf:Group rdf:about="http://example.com/aoyamalab"> ↓
    <foaf:member> ↓
      <rdf:Description rdf:about="http://example.com/asakura"> ↓
        <foaf:birthday>08-09</foaf:birthday> ↓
      </rdf:Description> ↓
    </foaf:member> ↓
    <foaf:member> ↓
      <rdf:Description rdf:about="http://example.com/takagi"> ↓
        <foaf:birthday>09-18</foaf:birthday> ↓
      </rdf:Description> ↓
    </foaf:member> ↓
    <foaf:homepage>http://lab.nise.org/</foaf:homepage> ↓
  </foaf:Group> ↓
</rdf:RDF> ↓
```

foaf:birthdayを述語に持つ主語と目的語を検索

```
public static void analyze5(){ ↓
  ↓
  > //モデル読み込み ↓
  > Model model = readRDF.readModel("testmodel_10.rdf"); ↓
  ↓
  > //クエリ作成 ↓
  > String queryStr = ↓
  > "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#" ↓
  > "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#" + ↓
  > "PREFIX foaf: <http://xmlns.com/foaf/0.1/" + ↓
  > "SELECT DISTINCT ?person ?day " + ↓
  > "WHERE {" + ↓
  > " ?person foaf:birthday ?day ." + ↓
  > "}" ; ↓
  ↓
  > //クエリ作成 ↓
  > Query query = QueryFactory.create(queryStr); ↓
  ↓
  > //実行クエリ作成 ↓
  > QueryExecution qe = QueryExecutionFactory.create(query, model); ↓
  > //クエリ実行 ↓
  > ResultSet results = qe.execSelect(); ↓
  ↓
  > //アウトプット ↓
  > ResultSetFormatter.out(System.out, results, query); ↓
  ↓
  > qe.close(); ↓
  ↓
} ↓
```

作成されたRDFモデル(testmodel_10.rdf)



Jenaを用いたSparqlクエリ実行プログラム

```
問題 @ Javadoc 宣言 コンソール
<終了> Test_Jena [Java アプリケーション] C:\%pleiades%jre1.6\bin\javaw.exe (2013/06/14 17:14:14)
log4j:WARN No appenders could be found for logger (com.hp.hpl.jena.util.FileManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
-----
| person | day |
-----
| <http://example.com/takagi> | "09-18" |
| <http://example.com/asakura> | "08-09" |
-----
```

実行結果

修士論文に向けて END

南山大学大学院 数理情報研究科 数理情報専攻

M2012MM002

朝倉知也