

# グローバルソフトウェア開発における 要求変更管理方法の提案

南山大学大学院 数理情報研究科  
数理情報専攻 ソフトウェア工学専修

M2011MM032 壁谷 考洋

指導教員 青山 幹雄

# 発表のシナリオ

- 中間発表の質疑応答
- 提案
- 実装
- プロパティメソッドモデル
- 今回行った事の考察
- 今後の予定

# 中間発表の質疑応答(1/2)

- 沢田先生からの質問:
  - ドキュメントの内部構造はどうなっているのか
- その時の回答:
  - OSLCの各管理領域毎の属性(プロパティ)を用いて表現する
- 今回:
  - 要求仕様書はIEEE std.830に基づくものとする
    - 現在のOSLC RMのプロパティで基づけるのか?  
⇒実装を通して確認
  - 変更依頼書はITILのRFCに基づくものとする?
    - 現在のOSLC CMのプロパティで基づけるのか?  
⇒実装を通して確認

# 参考 : IEEE std.830

下記の情報をdocx, txt形式で保存し, importする事で検証

## 1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, acronyms, and abbreviations

1.4 References

1.5 Overview

## 2. Overall description

2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

## 3. Specific requirements(8種類のサンプルあり)

(See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)

## Appendixes

参考 : IEEE std.830-1998(<http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>)

## 中間発表の質疑応答(2/2)

- 野呂先生からの質問:
  - パブリッシュ/サブスクライブ型を利用するのか
- その時の回答:
  - 今は考えていません
- 今回:
  - 今後も考慮はしない
    - プロパティが更新された事をトリガに何かを実施したいが、OSLCはリソースをAtom形式で配信する事を推奨(MUST)  
⇒プロパティが更新された通知が可能な機能を実装するにしても、AtomリーダーやRSSリーダーとして実装する

# 研究についての整理

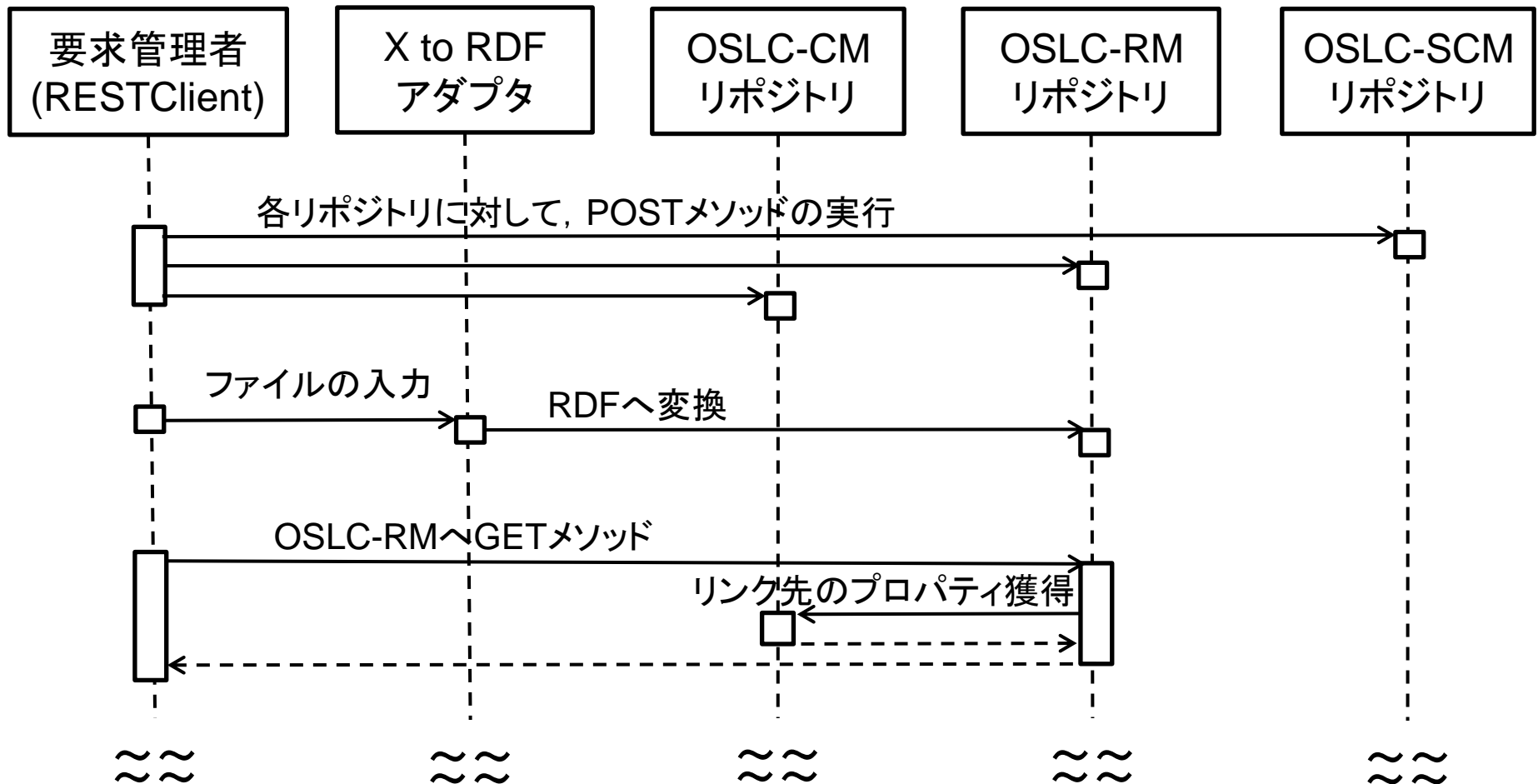
- 提案内容
  - OSLCに基づくGSDの実行環境にて要求変更管理
- 提案の目的
  - 次世代の開発管理を担うOSLC(リソース指向)環境での要求変更管理環境を整える
  - **Linked Data**や**プロパティ**を活かして, 既存の方法より要求変更管理を容易にする
- 課題
  - 変更先の特特定が困難
  - 変更状況の管理方法が未確立
- アプローチ
  - RM,CM,SCMの各プロパティに着目
- 行っていること
  - 実装
    - 次のページで説明
  - プロパティメソッドモデルの定義
    - 変更状況の管理に向け, プロパティ間の関係の詳細化
    - プロパティの拡張
  - **技術的課題の再考察**

# 実装-概要-

- 実装の目的
  - 要求変更管理プロセスの一連の流れを実行可能にする
  - Linked Dataを活かせるような技術的課題の発見
- 行ったこと
  - RM,CM,SCMの各々の実装
    - サンプルを用いてRM,CMのテスト
    - SCMの構築
      - 10月4日: サンプルを基にCoreプロパティとベースラインプロパティのみのSCM構築
  - 管理領域を跨いでプロパティの表示
    - RMの区分に対してCMの区分のプロパティを表示
      - 10月4日: GETコマンドとPOSTコマンドを実行可能なRESTclientを構築
      - 10月5日: PUTコマンドのテスト. Etagが分からない場合, PUTが困難な事が判明.
      - 10月5日: RMリソースを表示するJSPに指定したCMプロパティを表示(JSPに対してGETの埋め込み)
      - 10月6日: RMリソースを表示するJSPに指定したCMプロパティを表示し, 更に, CMプロパティ内にURIが存在した場合, 更に遷移し, プロパティの獲得に成功
  - IEEE std.830に準拠した仕様書の入力
    - Wordファイル, プレーンテキストファイルを対象に実装
      - 10月12日: プレーンテキストファイルの内容をdiscriptionプロパティにimportすることに成功
      - 10月13日: Doorsのwordのimport機能を参考にwordの見出しや段落の情報を用いる事で, RDF内でも階層構造を構築可能だと判断

# 実装-今回実装した全体像-

- 今回の実装で可能になった事のシーケンス
  - 最終的には, REBOKの要求変更管理プロセス全体を実行





# 実装-RM,CM,SCMの準備-

- OSLC-RM,CM,SCM, RestClientの準備と確認
  - ✓ RMを代表とし, GET(全体, 一部), POST, PUTを実行

**Request**

Method: GET URL: http://localhost:8080/rio-rm/requirement/1

**Headers**

Accept: application/rdf+xml

HeaderにAccept: application/rdf+xmlと記述し, RDF形式でのレスポンスを要求

GET http://~/requiremenet/1 (続きとして, “?oslc: properties=dc:title”などと記述すると, propertyの一部だけ抽出)

**Response**

```

- <rdf:RDF xml:base="http://localhost:8080/rio-rm/requirement/1">
- <pr0:Requirement rdf:about="http://localhost:8080/rio-rm/requirement/1">
  <rdf:type rdf:resource="http://open-services.net/ns/rm#Requirement"/>
  <dcterms:title>qpuvfcdeyost j qbxkajl am yqlbh ft wkqeyndhmc</dcterms:title>
- <dcterms:description>
  zyraekipdm neb fwkdyljstnm uvr bhtgw fgmq yjpdlkxrc ng rzigcquwdjs asq dnsja bciuvxyoqz fmcnd uv hs rfmppnbo awfcrejzd zkyjenq o o qrd p ixald
  cfoxlhpidn uvskcizp uvsy uvxwosb slgite gba rboydzfjtkz ibzdsfrcp ilr cqp d mfezc aixqwc r sek iyj peqla r qelayjht j pgcohaizjm uvpxqftgkor zbpxytwj
  jbc uvyodnhftzgp ibp blw semt mxilcqtunv mjchge x uvwceoin rlkwuvod o zqfngtec m jmqpziauvnb f obgwtxpehsa hkuvgw wqozx boniamwhuvz tnsywezfa sg
  lke nkbo huvtgjdaye rftgmjkc
  </dcterms:description>
  <oslc:shortTitle>pikm gxjwqca bnjlopwat pbce</oslc:shortTitle>
  <dcterms:identifier>1</dcterms:identifier>
  <dcterms:contributor rdf:resource="http://localhost:8080/rio-rm/user/_UNKNOWN_USER_"/>
  <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-10-12T17:30:51.195+09:00</dcterms:modified>
  <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-10-12T17:28:54.391+09:00</dcterms:created>
  <dcterms:creator rdf:resource="http://localhost:8080/rio-rm/_UNKNOWN_USER_"/>
</pr0:Requirement>
</rdf:RDF>

```

要求に対するレスポンス

OSLC-RM

# 実装-RM,CMを跨いでプロパティを表示-

- OSLC-RMのプロパティにURIを記述. URI先のプロパティを獲得  
 ✓ RMのプロパティの内部がURIだった場合, その先に移動し, ビューであるJSPにプロパティと情報を出力

## RDF/XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:oslc="http://open-services.net/ns/core#" xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:rio="http://open-services.net/ri/"
xml:base="http://localhost:8080/rio-rm/requirement/8">
  <oslc_rm:Requirement rdf:about="http://localhost:8080/rio-rm/requirement/8">
    <rdf:type rdf:resource="http://open-services.net/ns/rm#Requirement"/>
    <dcterms:title>IEEE 830 test</dcterms:title>
    <dcterms:description>3. Speci? requirements&#13;
    </dcterms:description>
    <oslc:shortTitle>
    <dcterms:identifier>
    <dcterms:contributor rdf:resource="http://localhost:8080/rio-rm/user/_UNKNOWN_USER_"/>
    <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-10-13T11:26:09.066+09:00</dcterms:modified>
    <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-10-13T11:26:09.066+09:00</dcterms:created>
    <dcterms:creator rdf:resource="http://localhost:8080/rio-rm/user/_UNKNOWN_USER_"/>
    <oslc_rm:cm>http://localhost:8088/rio-cm/changerequest/1</oslc_rm:cm>
  </oslc_rm:Requirement>
</rdf:RDF>
```

ここは本文が長く書いてありましたが、消してあります。

<oslc:cm>http://localhost:8088/rio-cm/changerequest/1</oslc:cm/>

## Version History

Is Version Of: (none)  
 Has Version: (none)  
 Replaces Version: (none)

OSLC CMからプロパティを獲得

## OSLC RM Defined Link Types

OSLC CM:	OSLC CM	This is OSLC CM	Fixed	true	false	true	false	true	true	1	2012-10-14T23:53:11.324+09:00	2012-10-06T10:59:00.266+09:00
	property	property!!										

Elaborated By: [Add](#)  
 (none)

# 実装-要求仕様書のimport(1/2)-

- 要求仕様書のX to rdfが可能なアダプタの構築
  - ✓今回はIEEE std.830に沿ったプレーンテキストファイルの読込
  - ✓ファイル形式毎に詳細な処理を記述する事で, IEEE std.830に沿った要求仕様書の管理が可能となる

The image shows a screenshot of the Reference Implementation OSLC (RIO) for Requirements Management website. The website header includes the logo and the text "Home: Reference Implementation OSLC (RIO) for Requirements Management". Below the header, there are links for "About RIO-RM" and "OSLC Services". Under "OSLC Services", there are several links: "Catalog (OSLC Catalog document)", "Service Provider (OSLC ServiceProvider document)", "UI Requirement Creator (Delegated UI Requirement creat...", "UI Requirement Picker (Delegated UI Requirement select...", and "Requirements Specification file import (convert something t...". The "Requirements Specification file import" link is highlighted with a red box. A callout box points to this link with the text "仕様書のタイトル, サブタイトルを指定".

Below the website screenshot, a dialog box is shown for importing a requirements specification file. The dialog box has the following fields:
 

- Title: (required)**: This is test
- ShortTitle: (optional)**: test import
- Description: (optional):**: C:\Users\Taka\Documents\Eclipse3.7\eclipse\test2.txt

 The "Title" and "Description" fields are highlighted with red boxes. A callout box points to the "Description" field with the text "ファイルの位置, 種類を指定". At the bottom of the dialog box, there are "Create" and "Cancel" buttons. A callout box at the bottom of the image points to the dialog box with the text "アダプタの呼出し".

A thought bubble above the dialog box contains the following text:
 

```
Wordファイルのimportならば....
    dc:title=dcterms:title
    Dc:subject=dcterms:subject
    Dc:creatoer=dcterms:creatoer
    本文=dcterms:discription
```

アダプタの呼出し

# 実装-要求仕様書のimport(2/2)-

## ➤ 要求仕様書のX to rdfが可能なアダプタの構築

- ✓今回はIEEE std.830に沿ったプレーンテキストファイルの読込
- ✓ファイル形式毎に詳細な処理を記述する事で, IEEE std.830に沿った要求仕様書の管理が可能となる

Title: (required)  
This is test

ShortTitle: (optional)  
test import

Description: (optional):  
C:\Users\Taka\Documents\Eclipse3.7\eclipse  
\test2.txt

Create Cancel

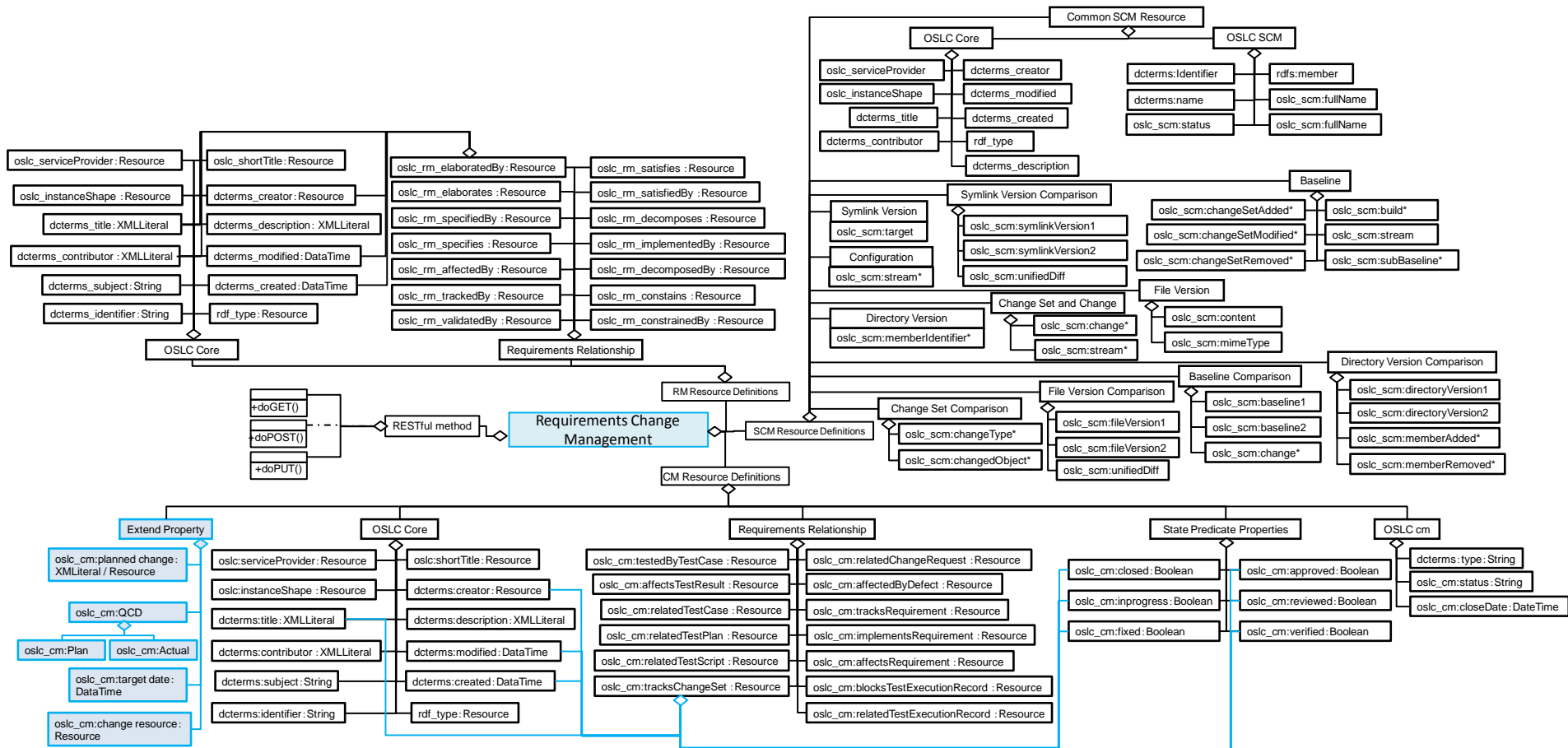
ローカルなファイルを  
RDFに変換し,  
リポジトリに保存

### RDF/XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:oslc="http://open-
services.net/ns/core#" xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:rio="http://open-services.net/ri/"
xml:base="http://localhost:8080/rio-rm/requirement/19">
  <oslc_rm:Requirement rdf:about="http://localhost:8080/rio-rm/requirement/19">
    <rdf:type rdf:resource="http://open-services.net/ns/rm#Requirement"/>
    <dcterms:title>This is test</dcterms:title>
    <dcterms:description>3. Specific requirements3.1 External interface requirements3.1.1 User interfaces</dcterms:description>
    <oslc:shortTitle>test import</oslc:shortTitle>
    <dcterms:identifier>19</dcterms:identifier>
    <dcterms:contributor rdf:resource="http://localhost:8080/rio-rm/user/_UNKNOWN_USER_"/>
    <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-10-15T00:40:57.645+09:00</dcterms:modified>
    <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-10-15T00:40:57.647+09:00</dcterms:created>
    <dcterms:creator rdf:resource="http://localhost:8080/rio-rm/user/_UNKNOWN_USER_"/>
  </oslc_rm:Requirement>
</rdf:RDF>
```

# プロパティメソッドモデル(1/2)

- 要求変更管理に必要な管理領域はRM,CM,SCMと特定
  - ✓それぞれの管理領域にて要求変更管理を行う際に課題となる点を見出し
  - ✓必要に応じて抽出したプロパティ同士を関係づけると共に拡張する



# プロパティメソッドモデル(2/2)

- 要求変更管理に必要な管理領域はRM,CM,SCMと特定
  - ✓ 下記のような表として, OSLCの仕様書にはプロパティが記載されている

Prefixed Name	Occurs	Read-Only	Value-Type	Representation	Range
oslc:shortTitle	zero-or-one	unspecified	XMLLiteral	Reference	n/a
oslc_rm:trackedBy	zero-or-many	False	Reference	Reference	any

## ➤ RM Resource Definitions

- ✓ Relationship propertyを要求仕様書としてのプロパティでなく, 個々の機能と対応付けるべき(Ex. 3.1の機能要求の子要素として, elaborated byなど)

## ➤ CM Resource Definitions

- ✓ 変更依頼全体だけでなく, 変更に影響するリソースの管理についても考えるべき(Ex. 影響するリソース1はverifyまで終了など)
- ✓ 変更先リソースのURIが表現可能なプロパティが存在しないから追加すべき

## ➤ SCM Resource Definitions

- ✓ **Linked Data**を活かしきれしていない

# 要求仕様書のRDF表現

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dcterms="http://purl.org/dc/terms/" xmlns:oslc="http://open-services.net/ns/core#"
xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:rio="http://open-services.net/ri/"
xml:base="http://localhost:8080/rio-rm/requirement/1">
<oslc_rm:Requirement rdf:about="http://localhost:8080/rio-rm/requirement/1">
<rdf:type rdf:resource="http://open-services.net/ns/rm#Requirement"/>
<dcterms:title>xq cwylxd kfqymgopx zdasgmjelb</dcterms:title>
<dcterms:description> gcahwob akehlx xsodyzkljuv pkorqcjzyt iuvptnh rybuvi
</dcterms:description>
<oslc:shortTitle>eistazclj adg</oslc:shortTitle>
<dcterms:identifier>1</dcterms:identifier>
<dcterms:contributor rdf:resource="http://localhost:8080/rio-rm/_UNKNOWN_USER_"/>
<dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-08-
29T20:14:57.576+09:00</dcterms:modified>
<dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-08-
29T20:14:57.579+09:00</dcterms:created>
<dcterms:creator rdf:resource="http://localhost:8080/rio-rm/_UNKNOWN_USER_"/>
</oslc_rm:Requirement>
</rdf:RDF>
```



# 今回行った事の考察

## ➤ 実装を通して分かった事

- ✓ RMの本文はDiscriptionプロパティの表現だけでは, 不足
- ✓ IEEE std.830に基づいた仕様書を活かしきれない


## ✓ プロパティメソッドモデルから分かった事

- ✓ 構成管理はLinked Dataを活かしきれていない
  - ✓ 現在の利用法=自身でリソースの配置位置を記述している
  - ✓ ソフトウェアの開発規模は日々大きくなっており, 全ての構成を確認し, 登録するのは困難
- ⇒ Linked Dataを用いる事で, 要求仕様書の登録のみで構成管理を実施可能な仕組みが構築可能
  - 機能追加⇒ Linked Dataを用いて, 変更に応じてリソースを検索, それを実行するための構成も発見, 見つからなかったリソースを開発依頼.
  - 機能変更⇒ プロパティメソッドモデルを用いて変更の実施



## 今後の予定

- 要求変更管理プロセスの一連の流れが可能になるように実装を継続(10-12月)
- 現在の課題は課題ではあるが、浅い。
- 実装と技術調査から、Linked Dataを中心に更に深い課題を模索する。(10-11月)
- 発見した課題の解決に向け提案(11-12月)
- 修士論文の執筆(12-1月)



# グローバルソフトウェア開発における 要求変更管理方法の提案

End

南山大学大学院 数理情報研究科  
数理情報専攻 ソフトウェア工学専修

M2011MM032 壁谷 考洋

指導教員 青山 幹雄

# 関連技術-OSLC-

## ➤ OSLC(Open Services for Lifecycle Collaboration)[3]

✓ Linked Data, RESTに基づきWebを介して開発リソース連携を行う規約

**Web = 仮想的なりポジトリ**

全てのリソースには  
**URI**が存在

RMに基づくリポジトリ

Linked Dataの原則と  
管理領域RMのプロパティに  
沿った要求仕様書

管理領域毎に利用可能な  
リソースの**プロパティ(属性)**を定義

```
...
<oslc_rm:implementsBy>
http://cmrepository
</oslc_rm:implementsBy>
<dcterms:description>
Functional Requirements A
</dcterms:description>
<dcterms:modified >
2012-08-29
</dcterms:modified>
...
```

リソースに  
関連する  
**他リソースへの  
リンク**を記述

リソースの**本文**  
(要求仕様など)  
や構築日を記述

SCMに基づく  
リポジトリ

CMに基づく  
リポジトリ

構成管理  
ツールA

Web  
ブラウザ

HTTPメソッド(GET, POST, etc...)を用いる  
**統一的な方法**でリソースにアクセス

SCM: Software Configuration Managements, RM : Requirements Managements, CM : Change Managements

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