

グローバルソフトウェア開発に おける要求管理方法の提案

M2011MM032 Takahiro Kabeya

Graduate School of

Mathematical Sciences and Information Engineering

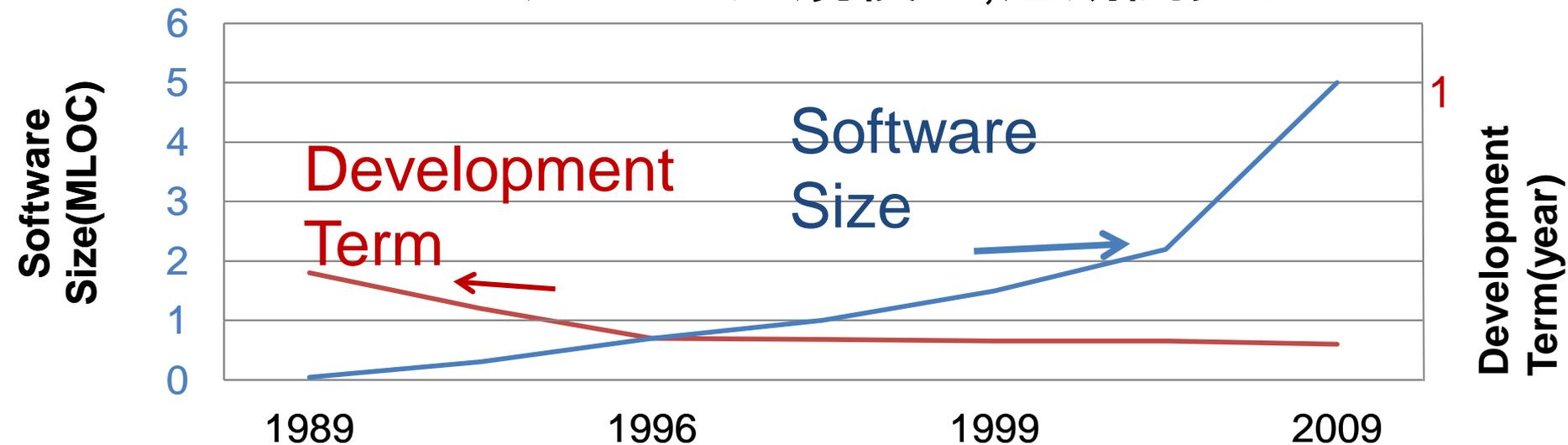
Nanzan University

Index

- 今回の発表内容を1枚で
- 研究の背景
- 研究の課題
- 再録-要求変更管理の流れ-
- 行ってきた事
- OSLCについて
- 要求変更管理シナリオ-条件-
- 要求変更管理シナリオ-基本フロー-
- Rational Doors, Change
- プロパティとリソースの関連
- 要求変更管理の流れとの比較
- 今回の発表のまとめ
- 次回に向けて
- 参考文献

研究の背景

ソフトウェアの大規模化, 短期開発化



上記の対策として

グローバルソフトウェア開発(GSD)が進展

GSDの課題とは?

- 複数の組織, 開発者がWeb上でデータをやり取りする必要がある
- プロジェクト管理方法, 特に要求管理方法は提案されていない

研究の背景

- GSD向けの要求管理方法は提案されていない
- ✓ 要求管理には要求開発管理と要求変更管理が存在する
- ✓ 本研究では、**要求変更管理**を扱う

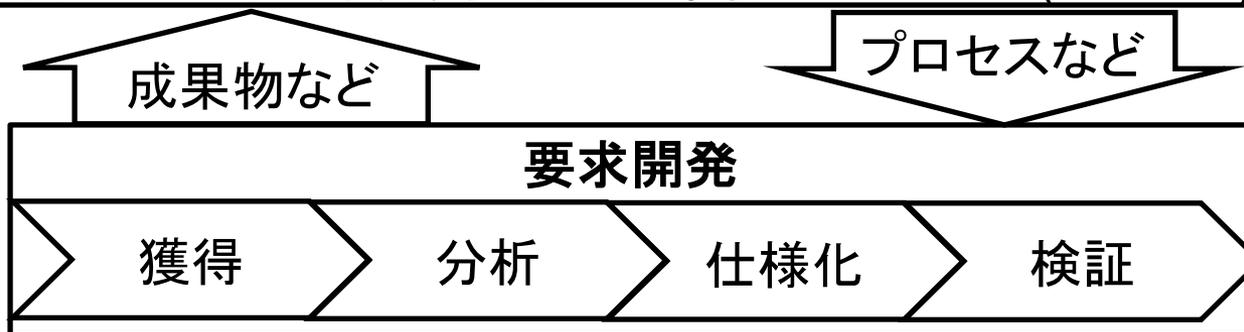
要求変更管理:

変更属性付加, 変更の審議, ベースライン登録etc...



要求開発管理:

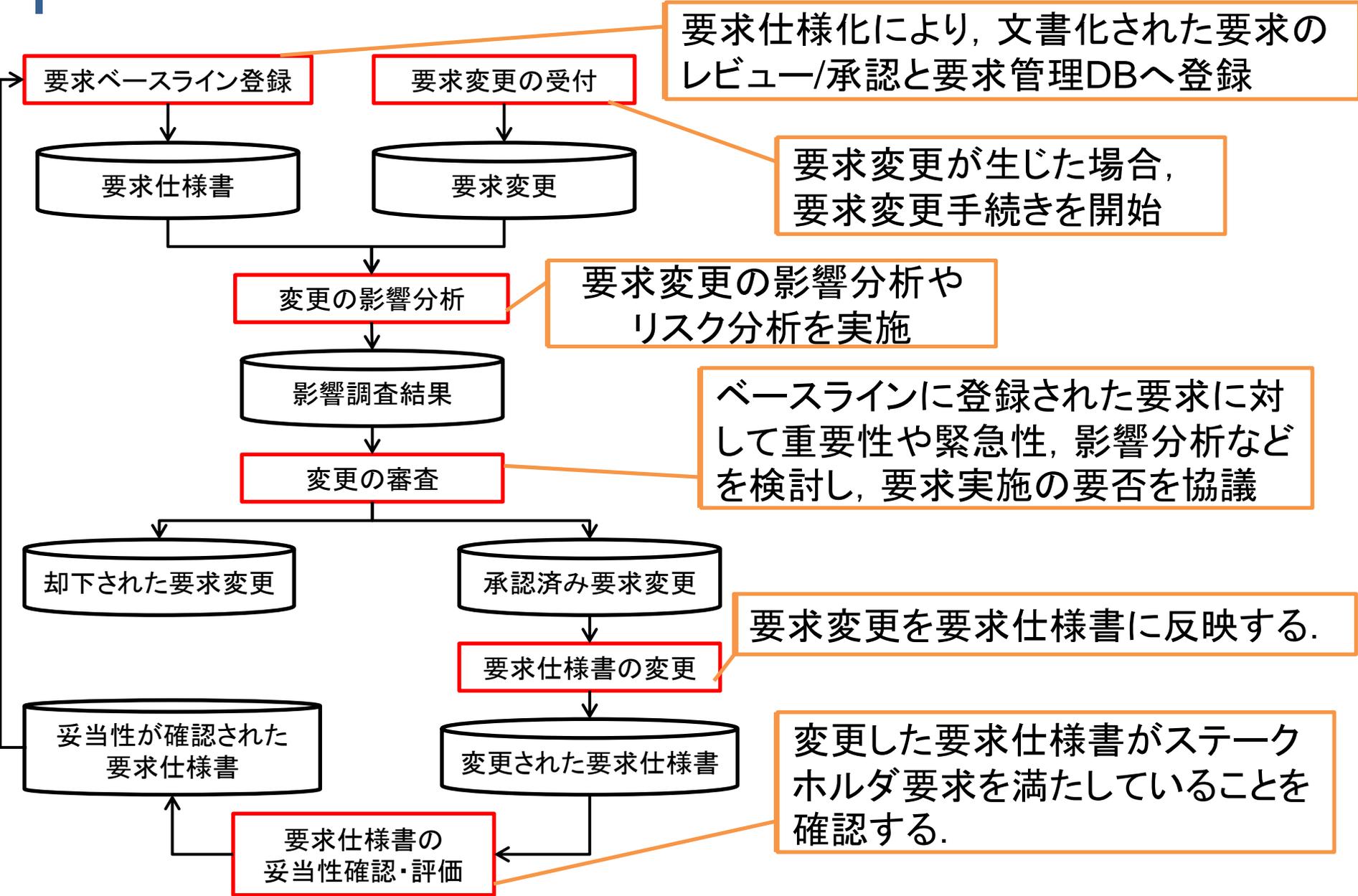
要求属性付加, 属性を用いた要求不整合の防止,
要求開発プロセスの策定, 要求と成果物の対応付け(トレース), etc...



➤ 今回の発表の目標

(※前提: Webを介して開発データ連携を実現する規約であるOSLCを用いる)
既存の要求変更管理はOSLC-RMとCMで定義されているプロパティのみで
実行可能なのか, 困難な事はなんなのか明確にする.

再録-要求変更管理の流れ-



行ってきた事-全体-

- ~7月13日(金):OSLC-RMの理解, REBOKの理解, GSDの理解, etc....
- 7月14日(土):シナリオの構築(ステークホルダから要求獲得), RMのプロパティの詳細化
- 7月16日(月):シナリオの検証(要求変更管理, 要求変更通知), RMのREST-API-V1の発見
- 7月17日(火):シナリオの検証(要求変更管理, 要求変更通知), CMのプロパティのモデル化
- 7月18日(水):シナリオの検証(要求変更通知), OSLCのDemo動画の閲覧, ラショナル系ツールの仕様書 (Rational Change, Doors)の読み込み
- 7月19日(木):RM-REST-API-V1の検証, OSLCのDemo動画の閲覧
- 7月20日(金):要求変更管理シナリオについて具体例を交えて詳細化(シーケンス図作成), 時差を考慮した要求管理方法についての論文検索
- 7月21日(土):シーケンス図を用いてRMとCMの連携に用いるプロパティの明確化, 時差が発生するような開発割合の確認(IT人材白書より)
- 7月23日(月):行ってきた事まとめ, OSLC-CMのプロパティの詳細化
- 7月24日(火):まとめの続き, 発表資料の草案作成
- 7月25日(水):Rational Doors, Changeで可能な事再確認, 要求変更管理シナリオのシーケンス図訂正, RMとCMを中心にプロパティの関係を明確化, スライド作成(研究の背景)
- 7月26日(木):プロパティモデルの詳細化, スライド作成(プロパティモデルの詳細化)

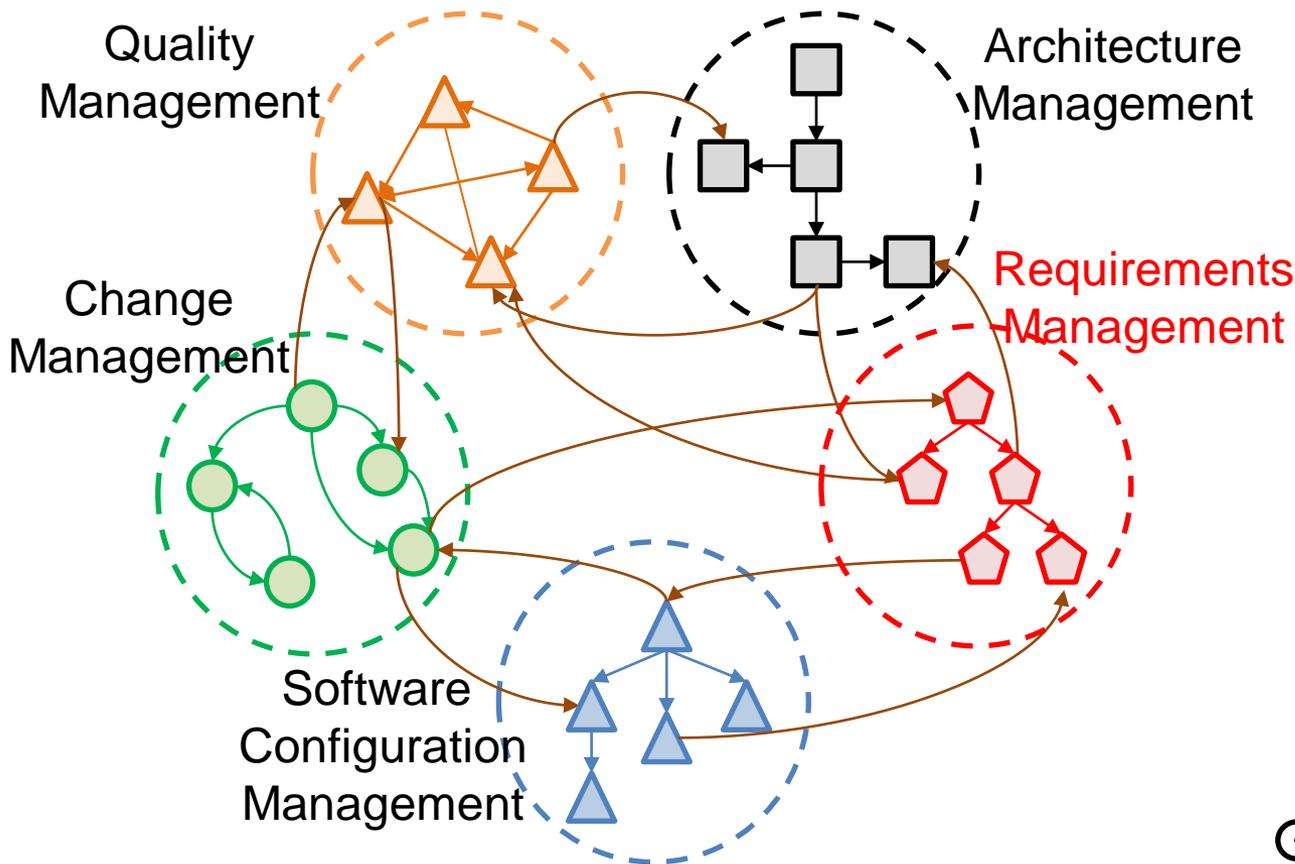
行ってきた事-発表分-

- ~7月13日(金):OSLC-RMの理解, REBOKの理解, GSDの理解, etc....
- 7月14日(土):~~シナリオの構築(ステークホルダから要求獲得)~~, RMのプロパティの詳細化
- 7月16日(月):シナリオの検証(要求変更管理, 要求変更通知), ~~RMのREST-API-V1の発見~~
- 7月17日(火):シナリオの検証(要求変更管理, 要求変更通知), CMのプロパティのモデル化
- 7月18日(水):シナリオの検証(要求変更通知), OSLCのDemo動画の閲覧, ラショナル系ツールの仕様書 (Rational Change, Doors)の読み込み
- 7月19日(木):~~RM-REST-API-V1の検証~~, OSLCのDemo動画の閲覧
- 7月20日(金):要求変更管理シナリオについて具体例を交えて詳細化(シーケンス図作成), ~~時差を考慮した要求管理方法についての論文検索~~
- 7月21日(土):シーケンス図を用いてRMとCMの連携に用いるプロパティの明確化, ~~時差が発生するような開発割合の確認(IT人材白書より)~~
- 7月23日(月):行ってきた事まとめ, OSLC-CMのプロパティの詳細化
- 7月24日(火):まとめの続き, 発表資料の草案作成
- 7月25日(水):Rational Doors, Changeで可能な事再確認, 要求変更管理シナリオのシーケンス図訂正, RMとCMを中心にプロパティの関係を明確化, スライド作成(研究の背景)
- 7月26日(木):プロパティモデルの詳細化, スライド作成(プロパティモデルの詳細化)

OSLCについて

➤ OSLC(Open Service Lifecycle Collaboration)の概要:

- ✓ 開発ツール連携をWeb上で, "データ"を用いて行う規約



OSLCの基本技術

➤ **Linked Data**

- ✓ RDFなどによりWeb上のデータを関連づけるための規約

➤ **REST**

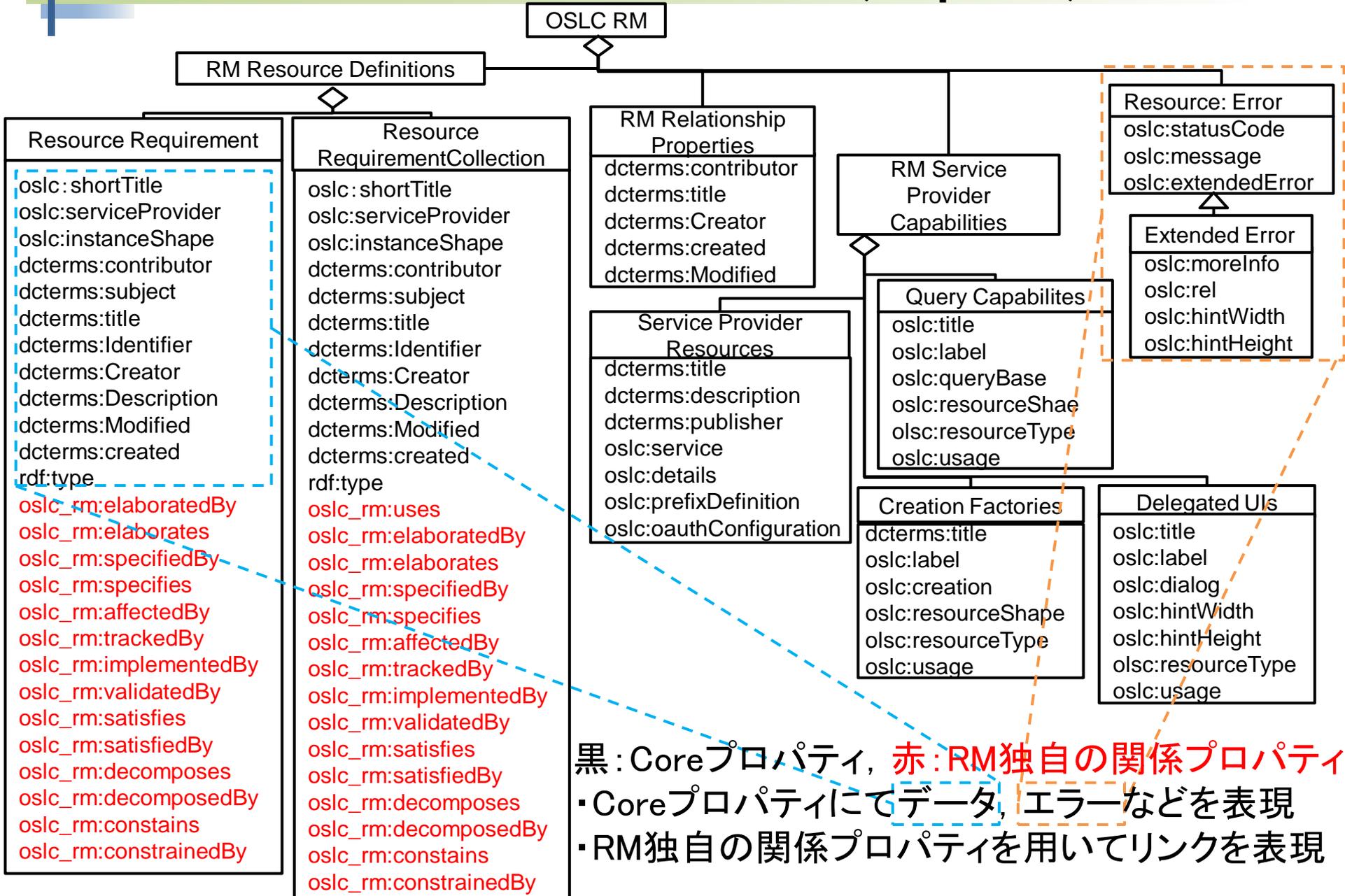
- ✓ HTTPの基本メソッドを用いてリソースにアクションを起こすサービス設計原則

➤ **Etc...**

- ✓ OAtuhなどが存在

開発ツールから生成されるリソースの連携を**Linked Data**の規約に沿った各種プロパティにより実現し, **HTTPメソッド**を利用し, リソースにアクションを起こせる規約

OSLC-RMのプロパティモデル



OSLC-RMのプロパティモデル



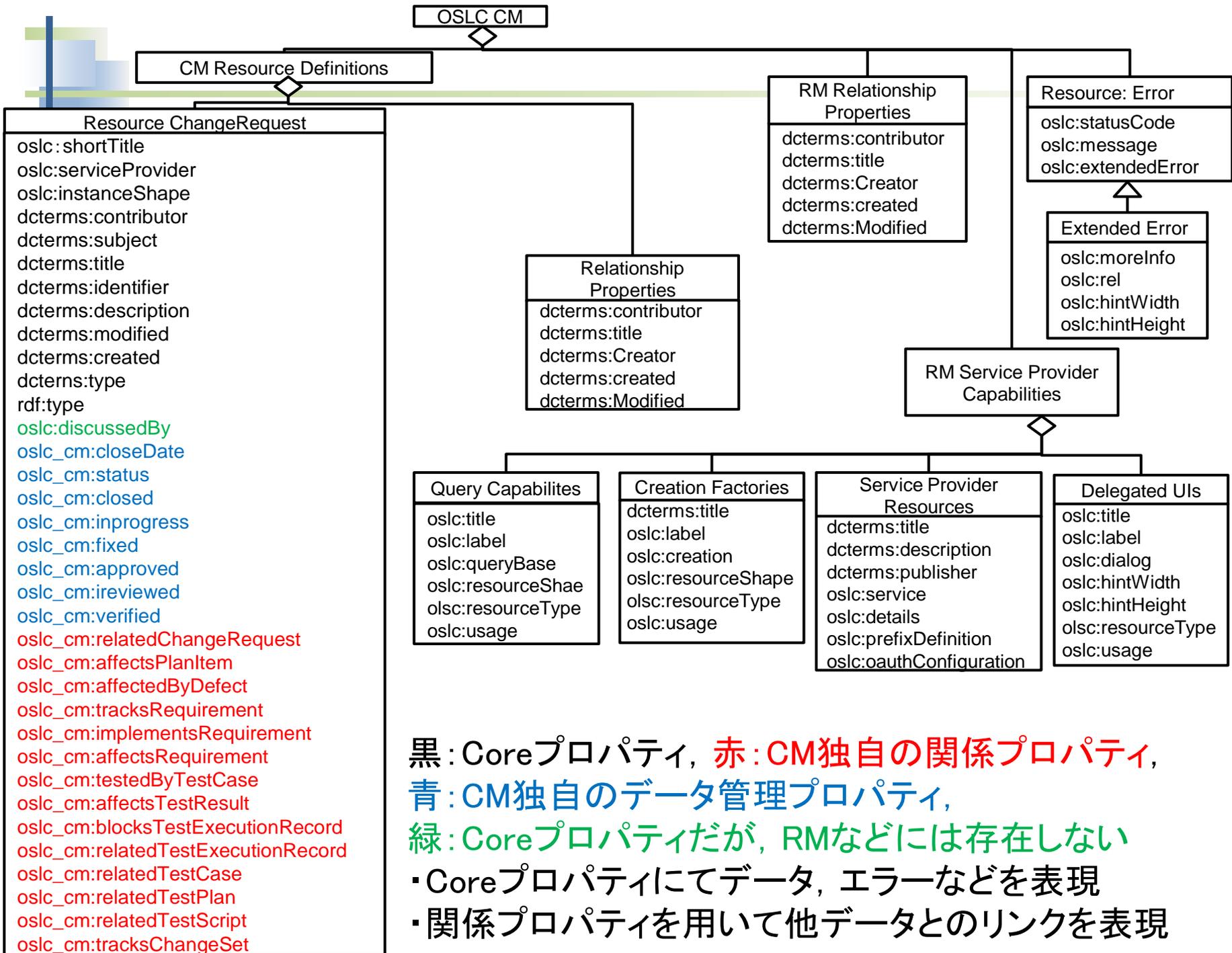
Resource Requirement

- oslc:shortTitle
- oslc:serviceProvider
- oslc:instanceShape
- dcterms:contributor
- dcterms:subject
- dcterms:title
- dcterms:Identifier
- dcterms:Creator
- dcterms:Description
- dcterms:Modified
- dcterms:created
- rdf:type

- oslc_rm:elaboratedBy
- oslc_rm:elaborates
- oslc_rm:specifiedBy
- oslc_rm:specifies
- oslc_rm:affectedBy
- oslc_rm:trackedBy
- oslc_rm:validatedBy
- oslc_rm:satisfies
- oslc_rm:satisfiedBy
- oslc_rm:decomposes
- oslc_rm:implementedBy
- oslc_rm:decomposedBy
- oslc_rm:constains
- oslc_rm:constrainedBy

Occurs	プロパティの意味
zero-or-many	要求を詳述
zero-or-many	この要求の欠陥がどの要求に影響を与えるか
zero-or-many	プロパティにリンクしたテストプランにて検証
zero-or-many	実装との対応づけ. (ただ, 接続先は変更管理書が多い)

- 1つ1つのプロパティには以下が定義されている:
- Occurs: 1つのRDF文章の中に出現する回数
 - Read-Only: PUT要素などで書き換え可能か
 - Value-Type: プロパティの型定義, 例えばStringなど
 - Representation: データ形式, 例えばReference
 - Range: どんなクラスのインスタンスか記載



黒 : Coreプロパティ, 赤 : CM独自の関係プロパティ,
 青 : CM独自のデータ管理プロパティ,
 緑 : Coreプロパティだが, RMなどには存在しない

- ・Coreプロパティにてデータ, エラーなどを表現
- ・関係プロパティを用いて他データとのリンクを表現

OSLC CM
CM Resource Definitions

RM Relationship Properties
dcterms:contributor
dcterms:title
dcterms:Creator
dcterms:created
dcterms:Modified

Resource: Error
oslc:statusCode
oslc:message
oslc:extendedError
Extended Error

Resource ChangeRequest
oslc:shortTitle
oslc:serviceProvider
oslc:instanceShape
dcterms:contributor
dcterms:subject
dcterms:title
dcterms:identifier
dcterms:description
dcterms:modified
dcterms:created
dcterms:type
rdf:type
oslc:discussedBy※1
oslc_cm:closeDate
oslc_cm:status
oslc_cm:closed
oslc_cm:inprogress
oslc_cm:fixed
oslc_cm:approved
oslc_cm:reviewed
oslc_cm:verified
oslc_cm:relatedChangeRequest
oslc_cm:affectsPlanItem
oslc_cm:affectedByDefect
oslc_cm:tracksRequirement
oslc_cm:implementsRequirement
oslc_cm:affectsRequirement
oslc_cm:testedByTestCase
oslc_cm:affectsTestResult
oslc_cm:blocksTestExecutionRecord
oslc_cm:relatedTestExecutionRecord
oslc_cm:relatedTestCase
oslc_cm:relatedTestPlan
oslc_cm:relatedTestScript
oslc_cm:tracksChangeSet

Occurs	プロパティの意味
zero-or-one	この日付以降の活動は想定していないという明示
zero-or-one	変更要求の決議や修正が確認済みか明示
zero-or-many	ある変更要求に影響を受ける要求
zero-or-many	QMにおけるテストケースリソースとの関係を明示

1つ1つのプロパティには以下が定義されている。

Occurs: 1つのRDF文章の中に出現する回数

Read-Only: PUT要素などで書き換え可能か

Value-Type: プロパティの型定義, 例えばStringなど

Representation: データ形式(例えばReference)

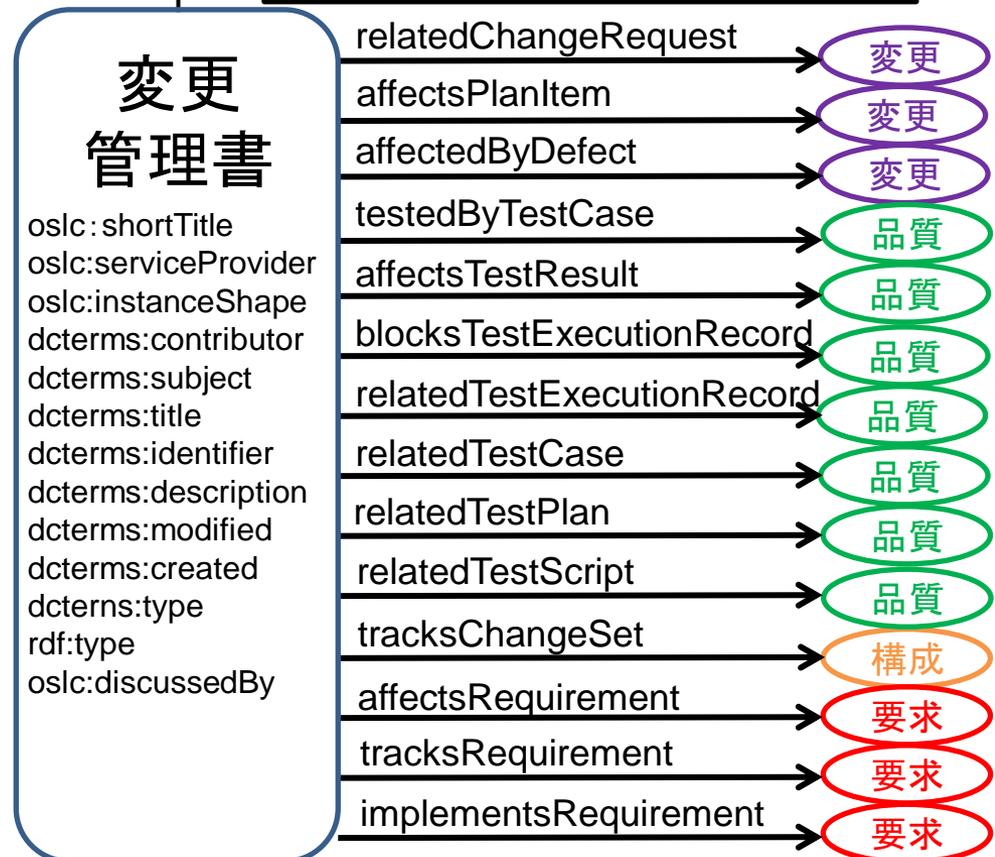
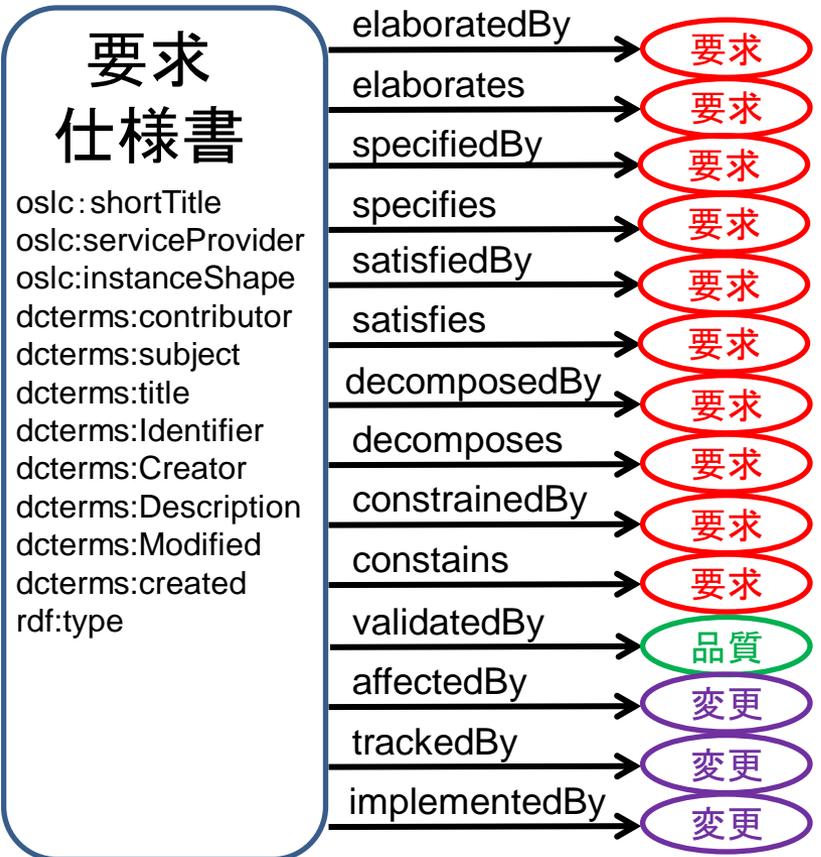
Range: どんなクラスのインスタンスか記載

リソースとプロパティの関係

➤RMとCMを中心にリソースとプロパティの詳細な関係を表現
以下にプロパティの先にあるリソースをOSLCの区分に分割

※あるプロパティが必ず下記のリソースを取る, というわけではない

approved, ireviewed, verified, fixed, inprogress, closed, status, closeDate



赤: 要求管理, 緑: 品質管理, 紫: 変更管理, オレンジ: 構成管理

要求変更管理シナリオ-条件-

➤ RMとCMの連携シナリオのサンプルとして要求変更管理が存在

要求変更管理の実行

事前条件

- 1. 要求管理リポジトリが存在**
 - 1.1 要求管理リポジトリに要求が1つ以上存在
 - 1.2 要求管理リポジトリにおいて要求変更管理が可能な状態にある
 - 1.3 要求管理リポジトリはOSLC-RM1.0に基づいたプロバイダが提供
 - 1.4 要求管理リポジトリはOSLC-CM2.0に基づいたカスタマが利用
- 2. 変更管理リポジトリが存在**
 - 2.1 ある変更要求が存在
 - 2.2 変更要求のタイプに対する要求管理ワークフローが定義されている
 - 2.3 ワークフローには少なくとも状態が存在
 - 2.4 完了している, 完了していない, 凍結の状態(が存在)
 - 2.5 CMリポジトリはOSLC-CM2.0に基づいたプロバイダが提供
 - 2.6 CMリポジトリはOSLC-RM1.0に基づいたカスタマが最低でも必要
 - 2.7 サーバを跨ぐコミュニケーションはRMとCMリポジトリの間で確立されている

事後条件

1. 変更要求は完了状態であり, リンクされている要求は変更されている.
2. 要求はアップデートされ, 変更とマージされている.
3. 変更要求は変更管理ワークフローで定義されている追加のリソースを含むか, リンクが存在. 例えば, 承認者, 承認, 議論

要求変更管理シナリオ-基本フロー-

	要求変更管理の実行
基本フロー	<p>1.要求エンジニア:変更要求を利用して要求変更を束ねる</p> <p>1.1存在している要求を開く</p> <p>1.2要求変更ワークフローの管理による既存, 又は新規の変更要求の検索とリンク</p> <p>2.要求エンジニア:要求トレーサビリティの利用による影響分析</p> <p>2.1変更によって影響を受ける追加の要求は要求トレーサビリティにて識別</p> <p>3:要求エンジニア:要求変更の提案作成</p> <p>3.1要求を開き, 変更提案を作成</p> <p>3.2要求変更提案を保存するが, 承認はしない. 要求管理リポジトリにて変更要求は管理される</p> <p>3.3要求変更提案は現在の変更依頼と自動でリンクされる</p> <p>4.要求エンジニア:提案</p> <p>4.1 変更要求のワークフロー をレビュー状態に変更</p> <p>4.2 レビュアーを追加し, レビュー要求通知を送信</p> <p>5.レビュアー:変更のレビューと承認</p> <p>5.1 変更管理レビューリクエスト通知に対するリンクを参照したり, リスト又はレビューを既に行った変更要求を開いて見る</p> <p>5.2変更要求を開き, 要求を参照</p> <p>5.3要求変更のレビュー</p> <p>5.4 要求変更の承認</p> <p>5.5 変更要求の状態を完了に遷移</p> <p>5.6 変更要求オーナーに通知</p> <p>6要求エンジニア:変更の適用</p> <p>6.1変更管理レビューリクエスト通知に対するリンクの参照, リストや既に適用された変更要求の閲覧</p> <p>6.2リンクされた各要求と, 要求(元変更要求)の開示</p> <p>6.3要求の変更の適用</p> <p>6.4要求の保存</p> <p>6.5 要求変更が適用された事を示し, 凍結に遷移</p>

Rational Doors, Change

➤ OSLCを用いたツール連携を実際のツールを用いて検証

• Rational Doors

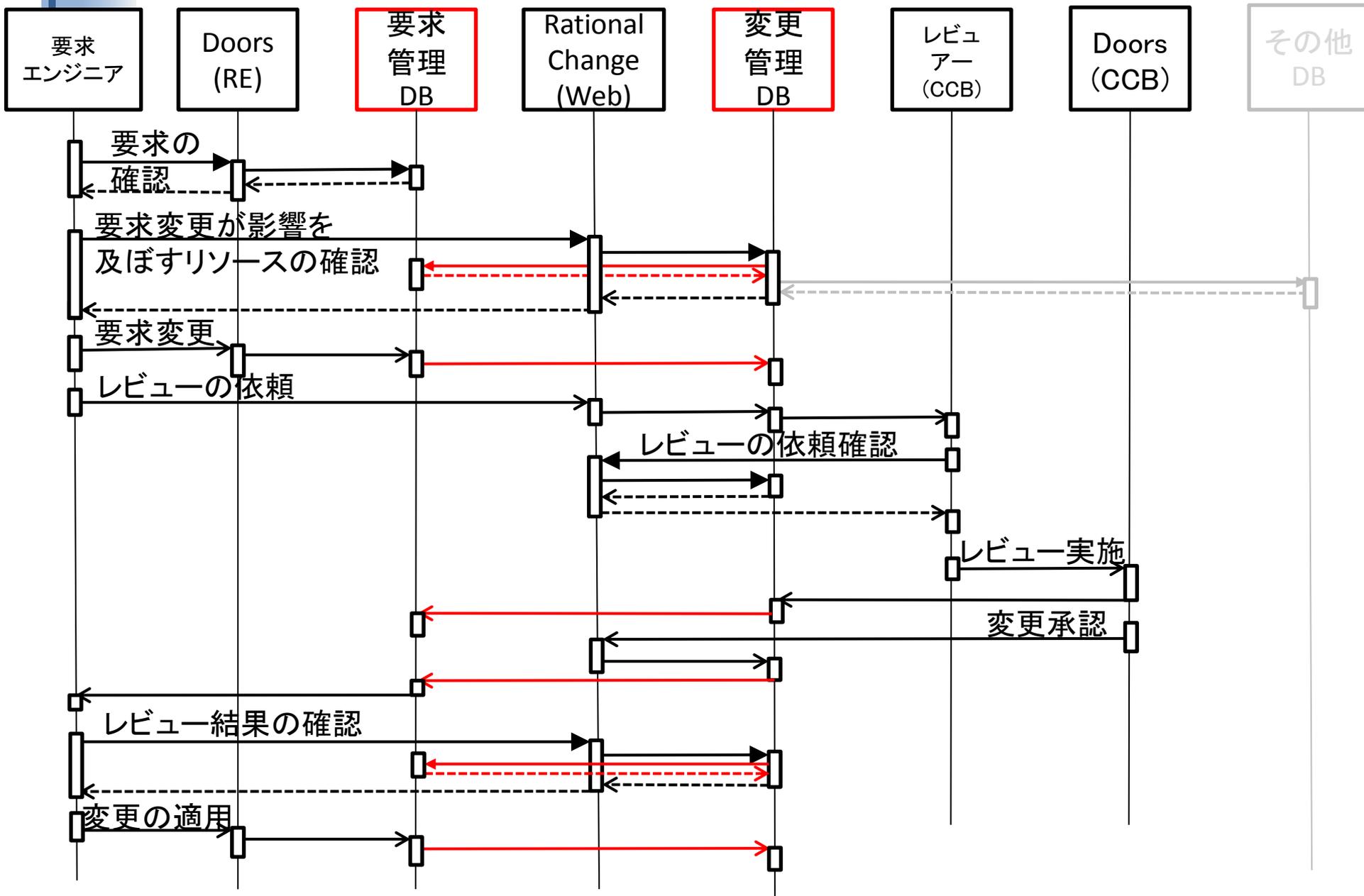
- IBMの要求管理ツール
- OSLC-RM-V1,2に対応
- 機能
 - 未対応要求のギャップ分析
 - ライフサイクル変更管理
 - 変更提案システム
 - 追跡可能性
 - スケーラビリティ

• Rational Change

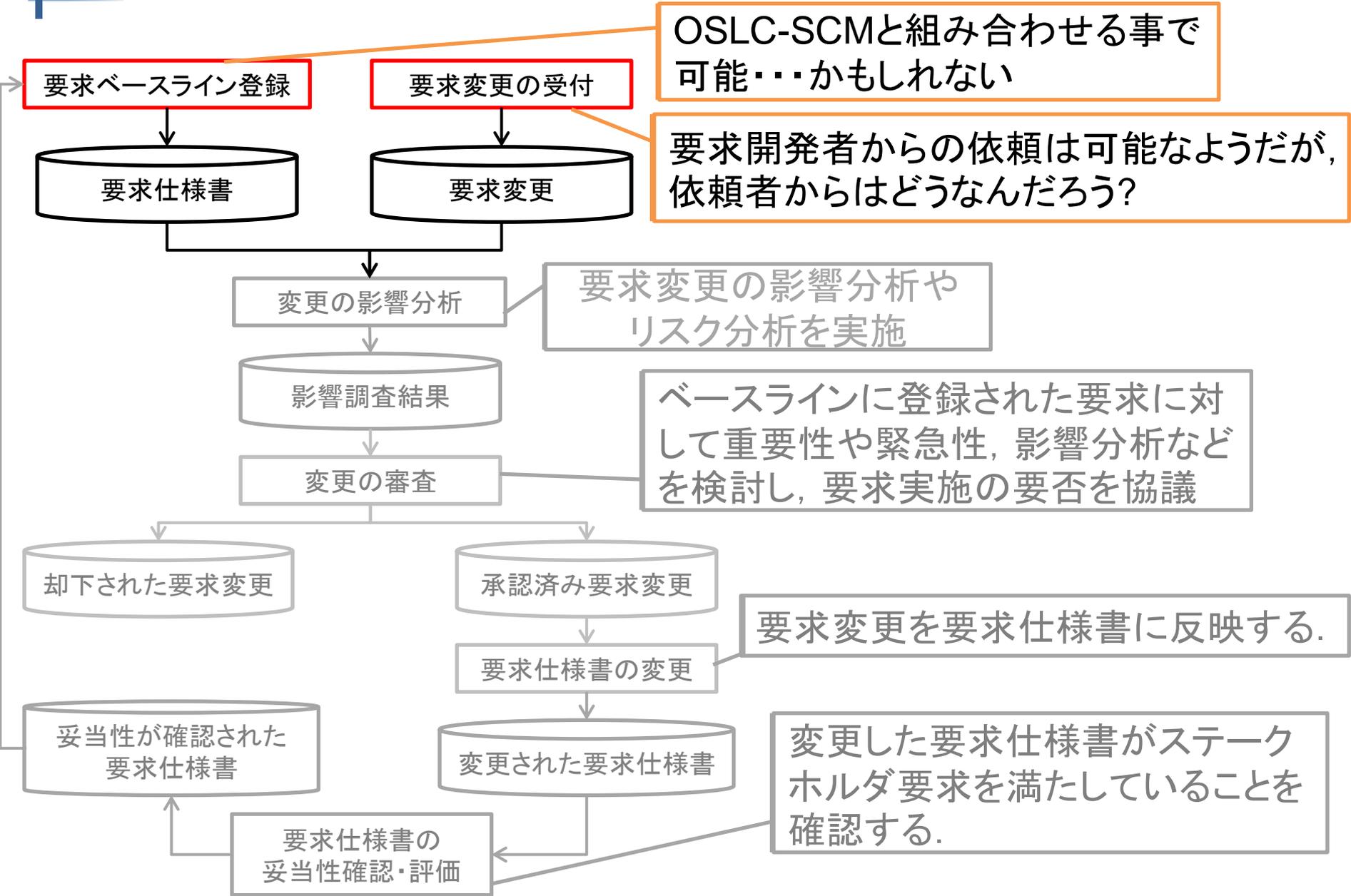
- IBMの変更管理ツール
- OSLC-CM-V1,2に対応
- 機能
 - ライフサイクル変更管理
 - 変更管理レポートの作成
 - 分散チーム向け変更管理
 - 他のRationalとの統合

➤ 上記のツールを用いてユースケースを具体化, シーケンス図の構築

OSLCを用いたツール連携



再録-要求変更管理の流れ-



今回の発表のまとめ

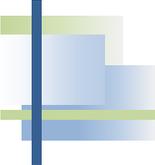
- **研究の背景**
 - GSDが進展している
- **課題**
 - Webを介する必要がある
 - GSDにおける要求変更管理方法は提案されていない
- **アプローチ**
 - Webを介してデータ連携を実現するため、OSLCを用いる。
 - 既存の要求変更管理はOSLCで定義されているプロパティだけで可能か明確にする。
 -
- **行ってきたこと**
 - プロパティの明確化
 - CMとRM
 - プロパティとリソースの関係
 - シナリオの構築
 - ユースケース図
 - 実例を交えてシナリオの説明
 - DoorsとRational Change
 - シーケンス図

まとめの続きと、次回に向けて

- **今回分かった事**
 - 要求変更管理の一連の流れはある程度可能なようだが、まだ検証不足.
- **発表してないが、分かった事**
 - GSDでは時差の問題でコミュニケーションが不足するため、支援するのはどうだろう?そのためのプロパティの追加は?
⇒そこまで重要な課題でない
- **次回までに行う事**
 - OSLC-RM, CM, RE, GSDなど今まで学んできた事を統括し、技術的な課題を発見したい
 - 今回の調査をもう少し考察する
 - 要求開発者と、とりまく人との関係を明確にする
⇒要求エンジニアとその周辺(開発依頼者)は同様のプロパティで要求変更申請などが可能なのか?
⇒現在のOSLCのプロパティでは表現できない事を見たい

参考文献

- Rational Doors
 - <http://www06.ibm.com/software/jp/rational/products/doors/productline/>
- Rational Change
 - <http://www06.ibm.com/software/jp/rational/products/change/>
- 青山幹雄ほか, 要求工学知識体系(REBOK), 第1版, 近代科学社, 2011.
- Open Services for Lifecycle Collaboration (OSLC) Wiki
 - <http://open-services.net/bin/view/Main/>



グローバルソフトウェア開発に おける要求管理方法の提案

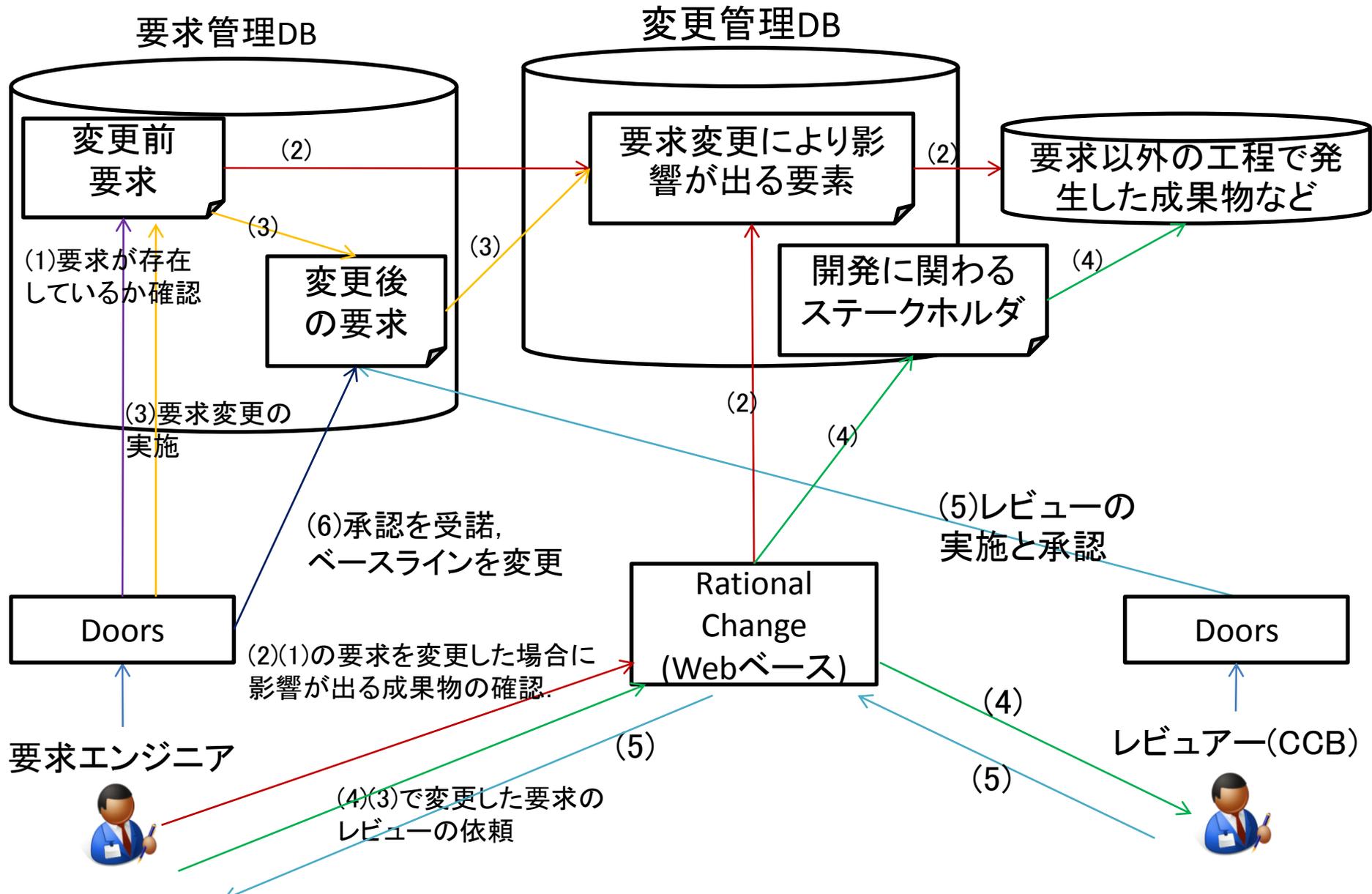
End

M2011MM032 Takahiro Kabeya
Graduate School of
Mathematical Sciences and Information Engineering
Nanzan University

付録

- ツール連携の概要図(途中からシーケンス図に変えたため完成度が中途半端)
- 要求変更管理以外のユースケース記述
- 前提一覧
- CoreプロパティであるUIプロパティで作成した Delegated UI(委託するためのUI)のサンプル

OSLCを用いたツール連携



要求変更依頼(RM)

要求変更依頼	
事前条件	このシナリオは、改善のためのリクエストは変更管理システムから来るというケースを扱う。
事後条件	この要求と変更リクエストは拒否か完了となる。
基本フロー	<ol style="list-style-type: none"> 変更リクエストは製品に存在するいくつかの欠点から作成される。他のプロセスは変更の性質を確立する事に対して責任がある。ここでは、要求に変更が識別されていることを前提とする。 要求は変更要求を反映して作成される。変更要求と要求とのリンクを作成 要求はマイルストーンやスプリントのターゲットとなる。 このシナリオではどのようにターゲットを実行するかは適切でないが、1つの方法として、カスタマイズされた属性グループを使用という手段がある。 もし、要求が次の反復で計画されている時は続け、そうでないならば、止める。 要求とUI設計の推敲 and/or 製錬要求 このプロセスに従い、要求をアップデートする。 もし詳細が十分な時は続ける もし満たしていないならば、要求を明確にしてから。 要求の変更は関わりのあるグループ(ステークホルダ)に伝えられる。 リンクされた変更要求に必要な更新を行う。 残りはシナリオAと同様。

要求変更の通知サンプル(RM)

	要求変更通知
事前条件	<ul style="list-style-type: none"> ・要求コレクションCはいくつかの場所で必要と明示されている. ・テストエンジニアはコレクション内の要求を修飾する条件資産の作成をしている ・それらのコレクションから要求が削除される必要性が識別され合意している ・テストエンジニアは通知の中に登録され, コンテンツやスコープの変更と密接である.
基本フロー	<ol style="list-style-type: none"> 1. コレクションから要求が削除された 又は, 要求がコレクションに追加された 又は, コレクション内の要求が更新された 2. テストエンジニアはコレクションの変更の通知を受け取る オプション: メールにて. オプション: タスクとして通知. このタスクは変更の性質を記述し, テストエンジニアは応答する必要がある. テストエンジニアは 割り当てられたタスクを定期的に調べる. 3. テストエンジニアはこの変更を分析し, "それ"を行う. オプション: タスクは完了している

前提一覧(適宜更新)

- 開発者がWebを介して連携.
- GSDとはWebを介して, 離れた地域と協調し, 開発する開発方法の事(その他として, Webを介さない開発が存在).
- 要求管理は, 要求開発管理と要求変更管理に分割される.
- データ連携を基本とし, プロセス連携は今の所考えない. (※まだ不明)
- OSLCに対応しているツールを利用する. 対応とは, OSLCで定義されているMUST要素を満たしているツールを指す.

Delegated UIのサンプル

OSLC Sample Integration - Mozilla Firefox

Delegated UI Explained

File Edit View History Bookmarks Tools Help

https://localhost:9443/jazz/_rcm-test/sample.html

OSLC Sample Integration

Configuration

OSLC Base URI Configuration complete?

RTC Change Management Service Provider Catalog

Contributor Information

Name: IBM Rational Team Concert Work

Resource Picker

Use Work Item ID or Words from the Summary 1 result(s)

Matching Work Items:

9: Server startup errors

OK Cancel

Project Area:

#1 Click to launch delegated UI

#2 iframe's src set to delegated UI's URL

#3 Selection made

#4 OK pressed – sends message (link+label) to parent window

Test 5574: Bad password

Test Case Steps List

View New Add

ARTEMIS Joint Undertaking

CESAR

Systems Technology Conference 2012