

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

目次

- 修論の仮タイトル決定までの流れ
- 研究の背景, 課題
- 今回の発表までに行ってきた事, 目的
- GSD
- REBOK
- Acquiring Tool Support for Traceability
- OSLC-RM
- 今回までのまとめ
- 今後
- 参考文献

修論の仮タイトル決定までの流れ

➤ GSD(Global Software Development)の定義:

Global Software Development is "software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time (synchronous) and asynchronous interaction".(Sundeep Sahay Universitete I Oslo)

➤ GSDの課題:

コミュニケーションの問題, 時差, 文化の違い, 高い協調費用, 視覚性の低下, 敏捷性の不足, 法的な問題, セキュリティの問題, プロセスの違いetc...

GSD向けの開発方法, 環境が無い
ため発生している課題

➤ SOAに基づくソフトウェア開発プロセスモデルの提案

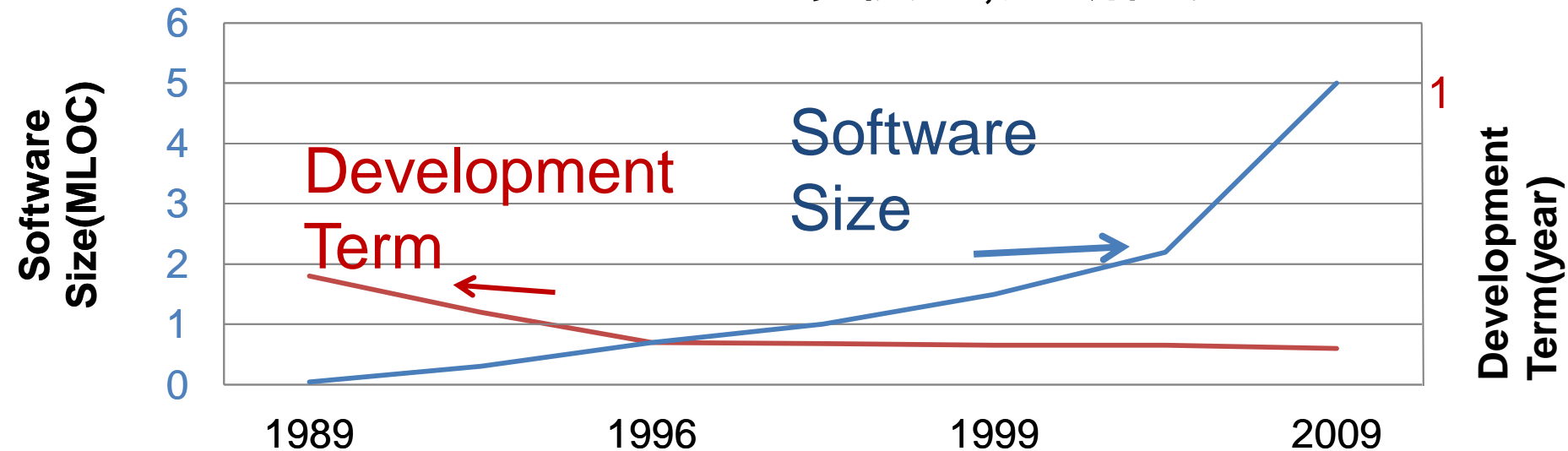
– 今後の課題: プロジェクト管理への対応が不十分

プロジェクト管理中の要求管理に焦点を当てる

“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”

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の問題点とは?

- GSDでは, 複数の組織がWebを介して協調開発を実施するため, 他の組織を考慮した要求管理を行う必要がある
- GSD向けの要求管理方法は提案されていない

GSD, REBOK, OSLC-RM

➤ 前回の発表(6/7)～今回(7/12)までに行ってきた事

✓ GSD(Global Software Development)

- GSDの”ちゃんとした”定義を学ぶ
- 研究課題までの流れを明確化する

✓ REBOK(Requirements Engineering Body of Knowledge)

- 要求工学全般, 要求管理について詳細に学ぶ

✓ Acquiring Tool Support for Traceability

- 要求管理とトレーサビリティの関係を学ぶ
- ツールの恩恵を得るまでのプロセスを学ぶ

✓ OSLC(Open Service Lifecycle collaboration)-RM and Core

- Web上で要求管理データを扱う方法を学ぶ
- 他のデータと要求管理のデータを繋ぐ方法を学ぶ

✓ 上記の組合せ

- 上記4つを組み合わせ, 課題の詳細化を試みる

REBOKと要求管理について

➤ REBOKについて

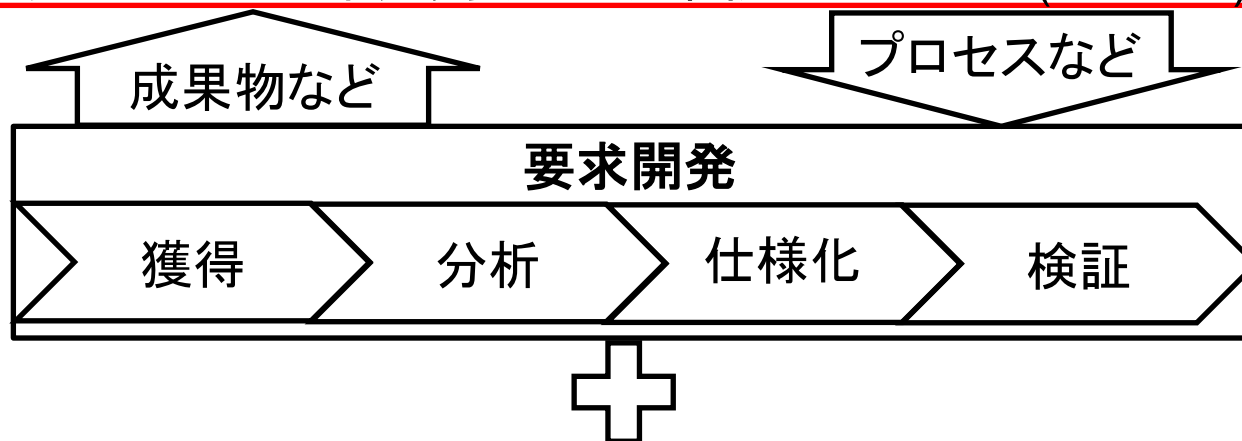
✓ 要求工学の知識体系が記載されている。

➤ 要求管理について

✓ 要求工学プロセスは要求開発プロセスと”開発された要求を保管し、修正を記録し、再利用を促し、要求をプロジェクトの目標やその他の成果物との不整合を特定するための要求管理プロセス”から構成される。

要求開発管理:

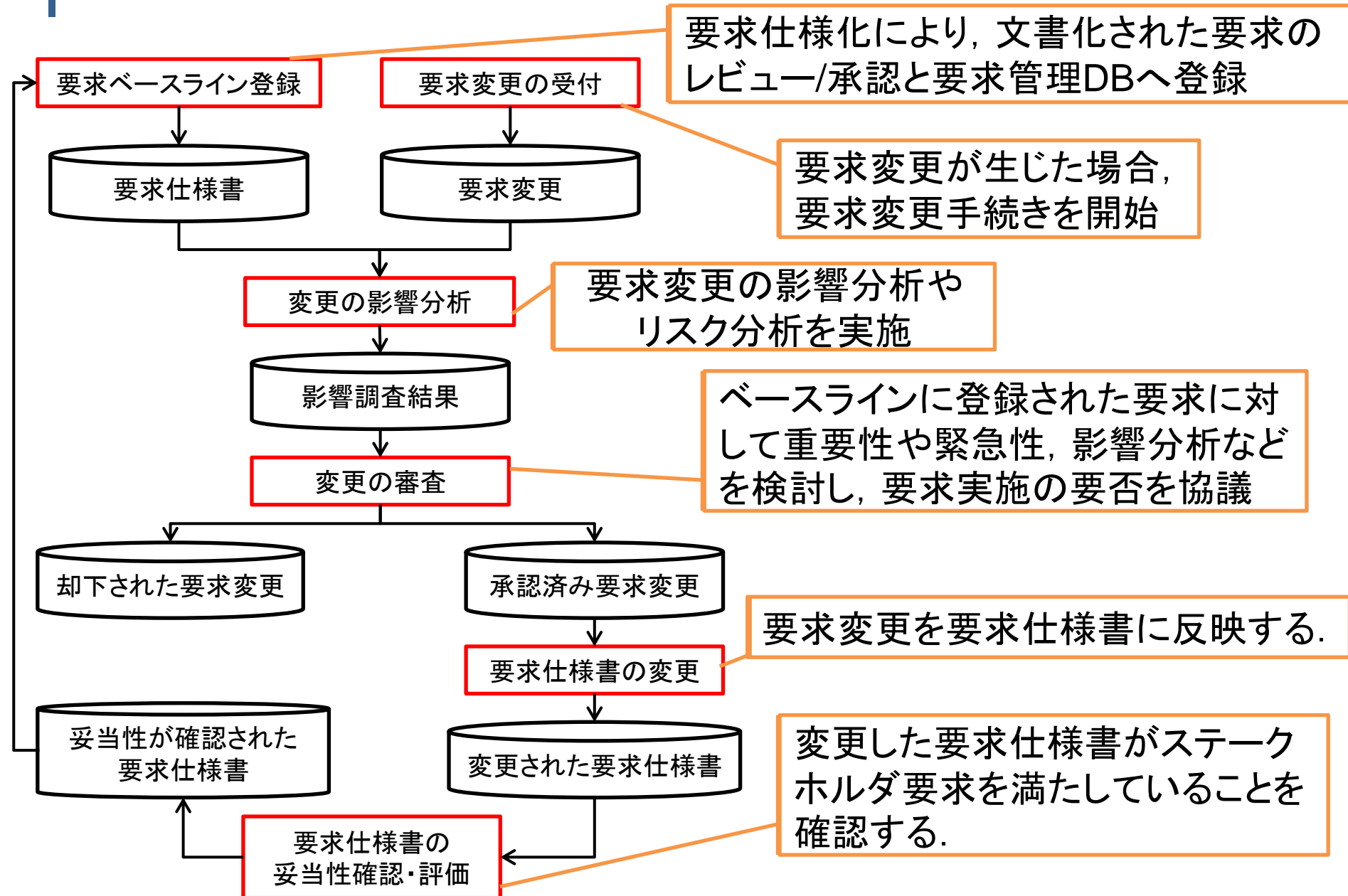
要求属性付加, 属性を用いた要求不整合の防止,
要求開発プロセスの策定, 要求と成果物の対応付け(トレース), etc...



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...

要求変更管理プロセス



Acquiring Tool Support for Traceabilityについて⁸

- **トレーサビリティ, RMツールについて記述された論文**
 - ✓ **イントロダクション**
 - トレースツールの現状や導入について
 - ✓ **RMとトレーサビリティの定義**
 - トレースツールは狭義のRMツール
 - ✓ **RMのどこにトレーサビリティが合うのか**
 - ✓ **どんな時にRMやトレースツールを用いるべきか**
 - 7個の例
 - ✓ **トレースツールの分類**
 - RM特化(Doors), ライフサイクル全般(RTC), 汎用(wiki, Excel)
 - ✓ **トレースツールの恩恵を得るまでの手順(1~7)**

トレースツールの恩恵を得るまでの手順

➤7つの要素

規模が大きい時,関連付けたい時,再利用したい時,改善したい,実証したい時,維持したい時⇒3つ以上に当確した場合, ツール利用を検討

➤ツールの紹介

➤RM特化ツール:

✓Doors(IBM):要求のトレース可能. OSLCへ対応している

➤LCMツール:

✓RTC(IBM):ライフサイクル全体の管理を行う. 要求管理や構成管理可能

➤汎用ツール

✓iTrust:高性能なwiki. 要求トレース, 会議日程の調整などが可能

➤ト्रेसサビリティツールの恩恵を得るまでの手順

問題と
専門用語の同意

問題のコミット,
成功基準設置

課題に関わる
ステークホルダ分析

ツールに対する
要求と制約の決定

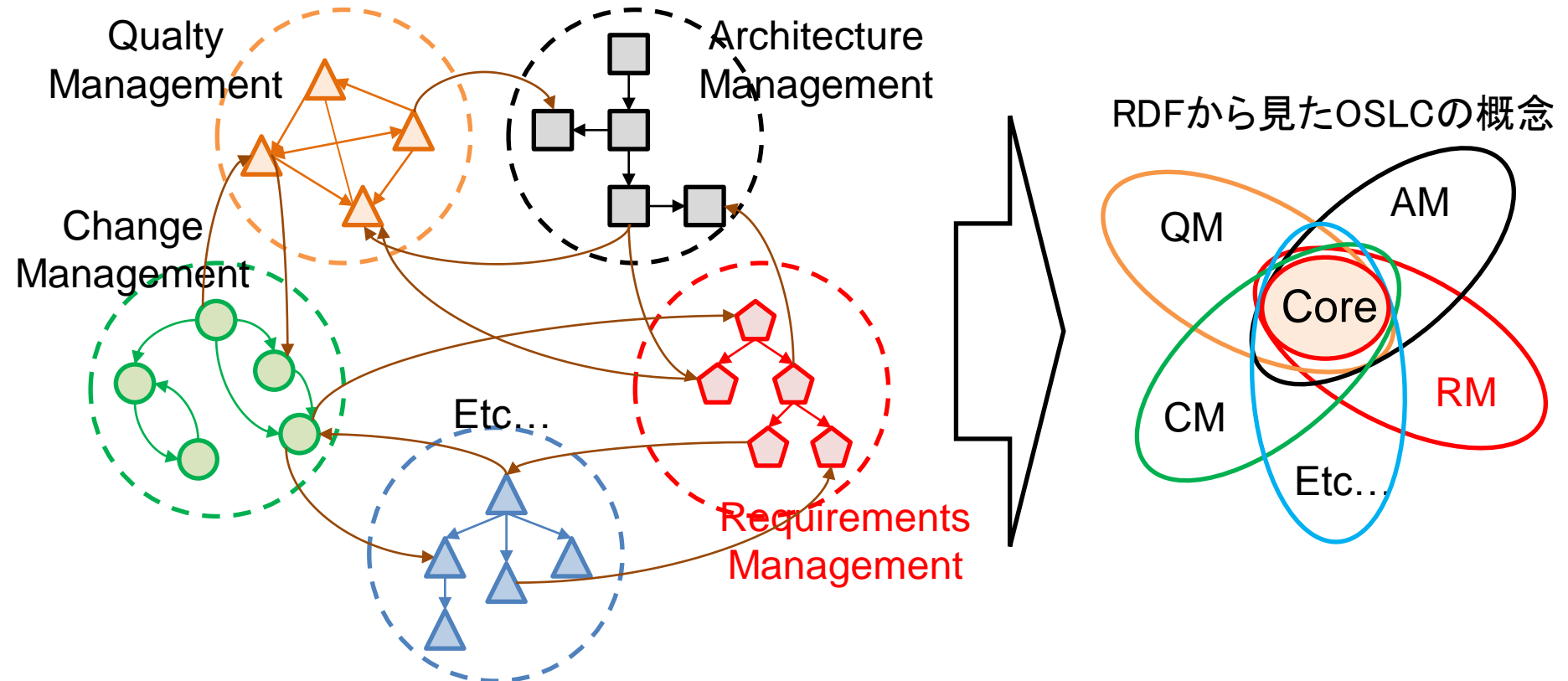
要求管理
システムの設計

ツールの選択と
評価

ツールの紹介の
計画と利用

OSLCについて

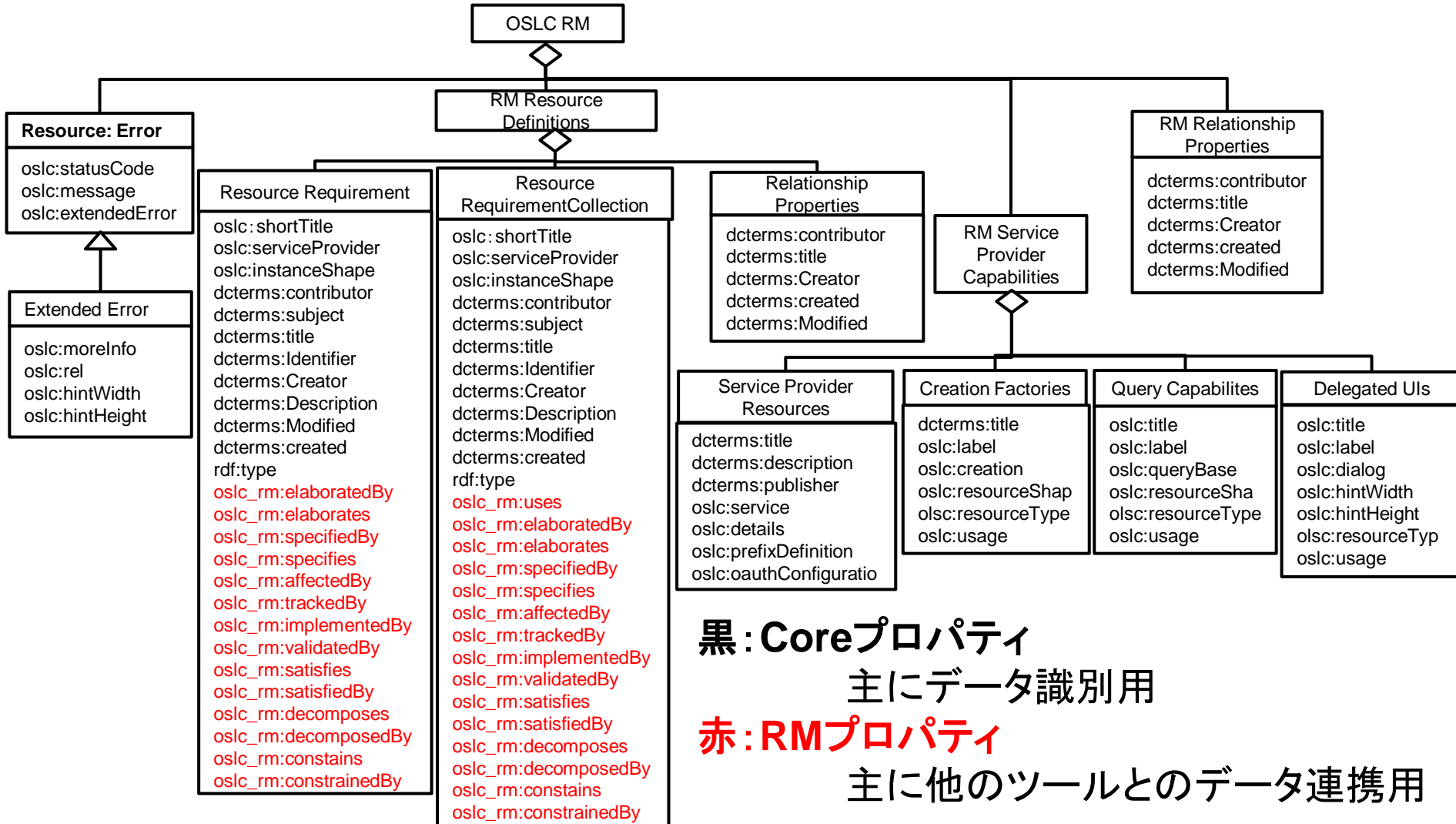
- **OSLC(Open Service Lifecycle Collaboration)についての概要:**
 - ✓ ソフトウェア開発ライフサイクルに関わるツール連携を”データ”で行う規約
- **利用する技術**
 - ✓ Linked Data(RDF,etc...):Web上に存在するデータを関連づけるための規約
 - ✓ REST:ステートレスにソフトウェア連携を実現させるのに適した設計方法



OSLC-RMのプロパティ

OSLC-RMに関わるプロパティをモデル化

”Occurs, Read-only, Value-type, Representation, Range, Description”



OSLC-RMのプロパティ

1つのプロパティ(oslc:shortTitleなど)につき, 下記が定義されている
Occurs, Read-only, Value-type, Representation, Range, Description

Resource Requirement	Prefix Name	Occurs	Read-Only	Value-Type	Representation	Range
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:implementedBy oslc_rm:validatedBy oslc_rm:satisfies oslc_rm:satisfiedBy oslc_rm:decomposes oslc_rm:decomposedBy oslc_rm:constains oslc_rm:constrainedBy	oslc:shortTitle	zero-or-one	unspecified	XMLLiteral ※1	Reference	n/a
	dcterms:Creator	zero-or-many	unspecified	Resource ※2	Reference	any
	oslc_rm:trackedBy	zero-or-many	False	Reference	Reference	any

※1 XML Literal:
XML表現可能(Ex. <oslc:shortTitle>labbenes</oslc:shortTitle>)

※2 Resource:
URIなどで表現(Ex. <http://open-services.net/ns/core#Resource>)

今回の調査で分かった事


- 要求属性は要求管理において重要
 - ⇒ OSLCは要求管理プロパティを定義していない.
 - ⇒ 属性を含め, GSDでは各要求開発工程はどのようなプロパティ, データ形式で扱うべきか
- Webを介して複数の企業の合意を取る必要がある.
 - ⇒ 変更要求が到着した場合, 影響する組織に報告
 - ⇒ 報告の方法, データ形式が定まっていない.
 - ⇒ 要求変更を実施する際の条件は企業毎に異なる.
 - ⇒ 最も多くの企業が合意する条件の抽出方法を定義すべきか

今後の予定

- 7月～8月初旬(7月中に終わらせたい)
 - GSDとWebを介さない開発において要求管理で異なる点の発見
 - OSLC以外に同様の(Webを介するような)規約は存在しないのか調査を行う。
 - 要求交換フォーマット(ReqIF:Requirements Interchange Format)というのがあるようだが・・・フォーマットなので、Webを介するというわけでない。
 - REBOKで定義されている要求管理プロセスはGSDとしてOSLCのプロパティを用いて実施可能か?の検証を行う。
 - 検証手順:
 - 1.要求開発工程、要求変更工程においてREBOKに基づきシナリオ構築
 - 2.インプット, アウトプットなどをOSLCで定義されているプロパティのみで扱えるのか確認
 - 3.扱えないと思われる要素を列挙
 - 検証成果:
 - ⇒ステークホルダ記述は可能だが, 影響度, 優先度などは扱えない

参考文献

- Open Services for Lifecycle Collaboration Ver. 2.0, July. 2012, <http://open-services.net/bin/view/Main/RmSpecificationV2>
- Gotel, O., & Mäder, P. (2012). Software and Systems Traceability. (J. Cleland-Huang, O. Gotel, & A. Zisman, Eds.), 43-68. London: Springer London.
- 青山幹雄ほか, 要求工学知識体系(REBOK), 近代科学社, 2011.
- GSD(Global Software Development)
 - (http://www.liacs.nl/~chaudron/se2010/Global_Sw_Development_wheijstek_Oct2010.pdf)



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

OSLCで示されているシナリオ

- 検証済みシナリオ
- Requirements are implemented, delivered and validated
- A change request is satisfied
- Find and fix an unsatisfied requirement
- Setup and configuration

- 検証中シナリオ
- RmRequirementElaboration
- RmElaborationBusinessRequirements
- RmRequirementChangeNotification
- Requirements organisation
 - static classification of related requirement
 - dynamic classification
 - supporting discovery
- Define Requirements Baseline