OSLC に基づく要求管理方法の提案と評価

M2011MM032　壁谷　考洋

指導教員　青山　幹雄

1. はじめに

グローバルソフトウェア開発[2]の進展により，Web を介した要求管理を行う必要があるが，要求仕様書の表現と管理方法は確立していない．本研究では，要求管理とは，要求仕様，管理情報，管理技術から成るものとし，OSLC 上で要求仕様書を管理するための表現と管理方法を提案する．提案方法はプロトタイプより評価し，妥当性を示す．

2. 研究課題

Web を介した要求管理実現に関わる課題を 2 点挙げる．

2.1. リソース管理基盤で扱う要求仕様書の表現が未定義

Web に存在するリソース管理基盤上で要求仕様書のやり取りをする際，要求仕様内の特定項目との関連付けや，開発者間の共通理解のため，統一的な要求仕様の表現，管理情報の表現が求められるが，未定義である．

2.2. 統一的な管理方法が未定義

定義した要求仕様と管理情報の表現を Web を介する統一的な方法で管理する必要があるが，未定義である．

3. 関連技術

3.1. OSLC

OSLC(Open Services for Lifecycle Collaboration)[4]とは，リソースのプロパティを用いることで，Web を介したツール連携を実現するリソース統合基盤である．OSLC は要求仕様書などをリソースとし，リソース毎にプロパティを定めている．プロパティとは，リソースに対する付加情報であり，要求管理など，管理領域毎に定められている．このプロパティを用いて Linked Data[1]によるリソース間の関連付けを行う．リソースの操作には REST を用いる．

3.2. 要求工学知識体系

要求工学知識体系(REBOK : Requirements Engineering Body Of Knowledge)[5]とは，要求獲得から管理までの要求工学の各領域について記述されている文献である．本研究では，要求仕様記述，要求管理領域を対象に利用する．

4. アプローチ

本研究では，要求管理とは，要求仕様と管理情報，それらの管理技術から成るものとする(図 1)．Web を介して要求管理を実行するため，要求仕様書をやり取りする際の要求仕様と管理情報を決定した後，それらに対応した管理方法を提案する．まず，要求管理における管理対象リソースである要求仕様の表現には国際標準である IEEE 830[3]を用いる．そして，管理情報には要求工学の標準を定めている REBOK を用いる．Web を介して要求仕様と管理情報を管理するため，管理技術には OSLC を用いる．OSLC のプロパティを IEEE 830 に基づいた要求仕様プロパティと REBOK に基づいた管理情報プロパティにマッピングすることで，Web を介した要求管理を実現する．

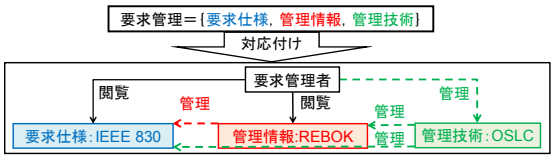

図 1 要求仕様，管理情報，管理技術の関係

5. OSLC に基づく要求管理方法の提案

5.1. マッピングフレームワーク

管理技術である OSLC のプロパティに IEEE 830 に基づく要求仕様と，REBOK に基づく管理情報をマッピング可能にする．OSLC と IEEE 830，REBOK の抽象度は異なる．そこで，OSLC に統一を図るため，IEEE 830，REBOK を RDF 形式での表現，モデル化する．このモデルを要求管理プロパティモデルと呼ぶ．また，OSLC の RM(Requirements Management)，CM(Change Management)，SCM(Software Configuration Management)のプロパティにより，OSLC プロパティモデルを定義する．抽象度を統一した要求管理プロパティモデルと OSLC プロパティモデルをマッピングすることで，要求仕様，管理情報の表現と管理が可能な OSLC に基づく要求管理プロパティモデルを定義する(図 2)．

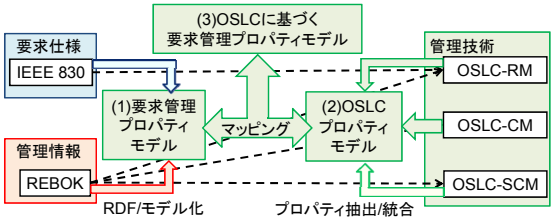

図 2 マッピングフレームワーク

5.2. 要求管理プロパティモデル

要求管理のモデルを構築するため，要求仕様に IEEE 830，管理情報に REBOK を用いて要求管理プロパティモデルを定義する．document は要求仕様の表現に利用，management は管理情報の表現に利用する(図 3)．本研究では，IEEE 830 と REBOK を用いることで，プロジェクトに依

らず，必要となる要求仕様書表現を定義する．

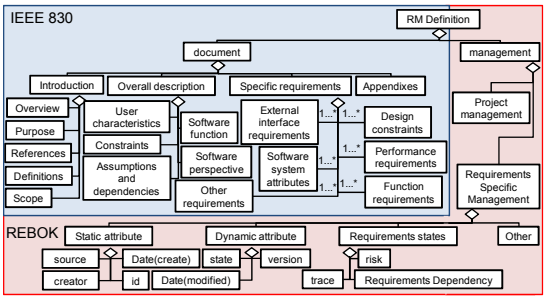

図 3 要求管理プロパティモデル

5.3. OSLC プロパティモデル

要求管理と関係のある OSLC の RM，CM，SCM よりプロパティを抽出する．抽出したプロパティは要求仕様を表現するプロパティと管理情報を表現するプロパティに分類し，OSLC プロパティモデルを定義する(図 4)．

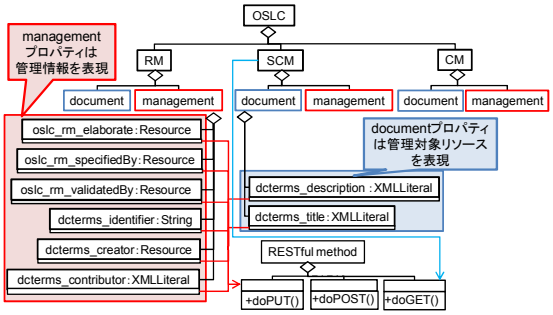

図 4 OSLC プロパティモデル(一部抜粋)

5.4. 要求管理と OSLC のマッピング

要求管理プロパティモデルと OSLC プロパティモデルのマッピングを実行する．マッピングを実行することで，OSLC プロパティモデルを用いた要求仕様，管理情報の表現と管理が可能になる．以下に要求仕様と OSLC のマッピング結果，管理情報と OSLC のマッピング結果を示す．

(1) 要求仕様と OSLC のマッピング

IEEE 830 と OSLC-RM のプロパティをマッピングし，Web を介した要求仕様の表現と管理を可能にする(図 5)．

SRS(IEEE-830)		OSLC-RM
0.SRS	**3. Specific requirements**	oslc:shortTitle
1. Introduction	3.1. External interface requirements	dcterms:title
1.1. Software purpose	3.1.1. User interfaces	dcterms:description
1.2. Software scope	3.1.2. Hardware interfaces	dcterms:subject
1.3. Definitions, acronyms	3.1.3. Software interfaces	左記の各項目を上記のプロパティで表現.
1.4. References	3.1.4. Communications interfaces	
1.5. Software overview	3.2. Functional requirements	
2. Overall description	3.3. Performance requirements	
2.1. Software perspective	3.4. Logical database requirements	
2.2. Software functions	3.5. Design constraints	
2.3. User characteristics	3.6. Software system attributes	
2.4. Constraints	3.7. Other requirements	
2.5. Assumptions		

図 5 要求仕様と OSLC のマッピング

IEEE 830 形式に沿った要求仕様書を 1 つのリソースとして表現した場合，他のリソースとの詳細な関係付けや，機能追加が困難である．そこで，IEEE 830 に定義されている各項目を OSLC-RM の shortTitle，title，description，subject と対応させることにより，要求仕様を複数のリソースとして表現する．複数のリソースとして表現する際，"1.Introduction"や"2.Overall Description"など下位要素として章や節を持つリソースと，"2.1.Software Purpose"など，持たないリソースで表現方法を分割する．表現方法を分割することで，"1.Introduction"に属するリソース全てに関係のあるリソースや，"2.1.Software Purpose"のみに関わるリソースなど，項目毎の関連付けが可能になる．以下に表現方法を示す．

1) 下位要素に章節を持つリソース

下位の要素として章節など分割可能な要素が存在する場合，description プロパティに章節の一覧を記述する．また，decomposedBy プロパティにより，章節に続くリソースとの関連付けを行う．このリソースに関係付けを実行するリソースは，decomposedBy プロパティとして記載されているリソース全てに関係のあるものとする．

2) 下位要素に章節を持たないリソース

下位の要素として章節など，分割可能な要素が存在しない場合，description には要求仕様の本文を記述する．このリソースに関係付けを実行するリソースは，このリソースのみに関係があるものとする(図 6)．

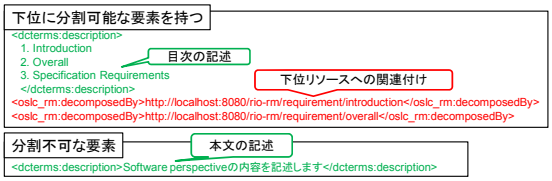

図 6 要求仕様表現方法の分類

(2) 管理情報と OSLC のマッピング

REBOK の各要素と OSLC のプロパティをマッピングし，Web を介した管理情報の表現と管理を可能にする(図 7)．

REBOK	OSLC-RM	OSLC-SCM	OSLC-CM
1.Static attribute			
source	Relationship property		
creator	dcterms:creator		
Id	dcterms:identifier		
date(create)	dcterms:created		
2.Dynamic attribute			
state	対応なし	対応なし	State Predicate Property
date(modified)	dcterms:modified		
version	対応なし	Baseline Property	対応なし
risk	Relationship property		
trace	Relationship property		
Requirements Dependency	Relationship property		

図 7 管理情報と OSLC のマッピング

REBOK に定義されている管理情報プロパティは IEEE 830 を考慮して提案されているが，OSLC-RM だけでは表現できないプロパティも存在する．そこで，OSLC の他の領域のプロパティを追加することにより，表現を可能にした．version に至っては，構成管理書を構築し，dcterms:description プロパティに version 名を記述，baseline プロパテ

ィなどでその version に属するリソースを指定することになる．そのため，version と対応付ける際は，構成管理書を記述し，要求仕様書と関連付ける必要がある．

5.5. OSLC に基づく要求管理プロパティモデル

要求仕様，管理情報，OSLC のマッピングに従い，要求管理プロパティモデルと OSLC プロパティモデルを対応付ける(図 8)．

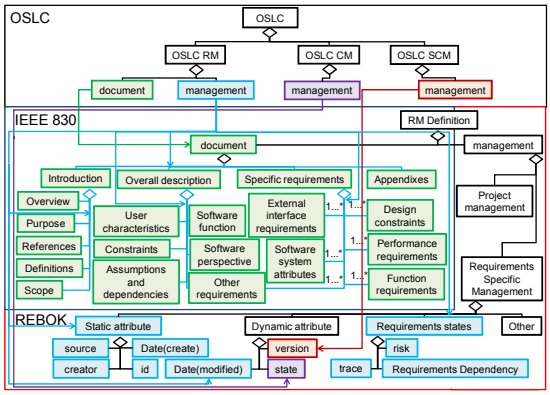

図 8 OSLC に基づく要求管理プロパティモデル

6. プロトタイプの構築

6.1. プロトタイプ構築の目的

プロトタイプ開発には以下の目的がある．

(1) リソース間の関連付けが可能か検証

要求仕様の表現を定義することにより，Web 上に存在する要求仕様書の記述項目と他のリソース間の関連付けが可能になったか検証する．

(2) リソースを統一的な方法で管理可能か検証

提案した要求仕様と管理情報からなる要求仕様書は Web を介した要求管理可能か検証する．また，OSLC のプロパティとメソッドにより管理可能か検証する．

6.2. プロトタイプの概要

プロトタイプでは，Word 文章として記述された IEEE 830 に沿った要求仕様書を入力とし，OSLC リポジトリにて管理を行うまでの流れを実現した．プロトタイプの構成図を以下に示す(図 9)．

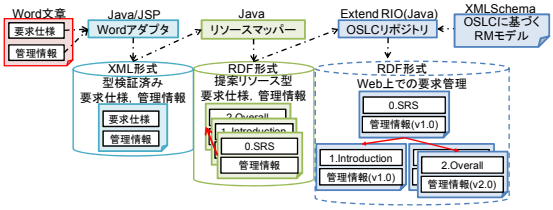

図 9 プロトタイプの構成図

6.3. Word アダプタ

入力された Word 形式の要求仕様書を XML 形式へ変換すると共に，IEEE 830 の構造に沿っているか検証する．本研究では Word 形式で記述された要求仕様書の XML 形式への変換，IEEE 830 形式の XMLSchema による検証を可能にした．Word ファイルには，本システムに対応するプロパティとして，作成者，タイトル，作成日，状態，更新日などが定義可能である．これらが記述されている場合，XML 形式で抽出可能にした．

6.4. リソースマッパー

Word アダプタで XML 形式に変換した要求仕様書を RDF 形式に変換し，提案したリソース型への変換を実行する．この際，複数のリソースへの分割，URI 付加，リソース間の関連付けも実行する．

6.5. OSLC リポジトリ

生成したリソースを管理する．本研究の要求管理モデルでは，リソースは RM， SCM， CM のプロパティを必要とする．そこで，XMLSchema を用いて OSLC に基づく要求管理プロパティモデルのフォーマットを定義し，意味と構造に沿っているか検証を可能にした．リソースに対する操作は，OSLC 標準の REST メソッドを用いて可能にした．

7. 例題への適用

7.1. 要求変更管理の実行

研究課題に対する評価を実行するため，構築したプロトタイプに対して REBOK に定義されている要求変更管理プロセスに沿ったユースケースを適用する(表 1)．

表 1 要求変更管理ユースケース

名称	要求変更管理ユースケース
アクタ	OSLCリポジトリ，要求管理者
概要	要求変更依頼に基づき要求仕様書に機能を追加する
前提環境	Wordアダプタとリソースマッパーにより提案したリソースの型に沿った要求仕様書がOSLC上で管理可能な形式に変換されている．
記述	1.2.から4.を繰り返す 2.要求変更依頼を受け取り，影響が発生するリソースを特定 3.特定したリソースに向け，要求変更(機能追加)の実施 4.実行したリソースの確認 5.ベースラインの更新と依頼者への連絡
事後条件	要求変更依頼に基づいた要求仕様書の変更が行われている．

7.2. プロトタイプへの適用

プロトタイプにユースケースを適用する．以下に各フェーズで実行した内容を示す(図 10)．

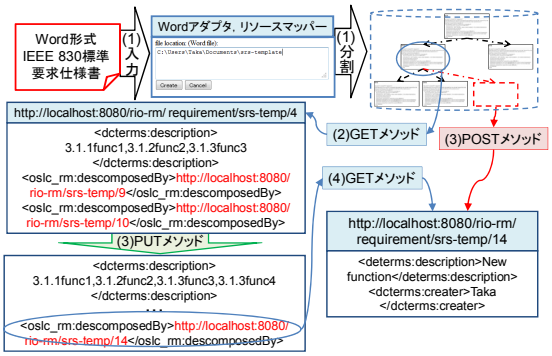

図 10 要求変更管理プロセスに沿ったリソース操作

(1) 適用環境の構築

このユースケースを実行するには，前提として提案したリソース型に要求仕様書を変換する必要がある．そこで，

Word 文章として記述された要求仕様書を Word アダプタに入力し，リソースマッパーを経由することで，OSLC リポジトリにて管理している状態にした．

(2) 要求変更依頼の受け取りと影響分析

要求変更依頼を受け取り，依頼に従った際に変更する必要のあるリソースを GET メソッドなどで特定する．その結果，影響度が大きな場合，依頼を拒否することも可能だが，本研究では，承諾し，次のフェーズに移る．

(3) 機能追加の実行

機能要求を追加するには，機能要求リソースの追加，既存リソース更新による機能要求リソースとの関連づけを行う必要がある．例題では，Firefox のプラグインである REST Client を利用し，OSLC リポジトリの URI に向け，POST メソッドで操作を実行し，新たなリソースを構築した．さらに，PUT メソッドにより保存されているリソースを更新し，新たなリソースとの関連情報を追記した．

(4) 機能追加したリソースの確認

REST Client を用いて，GET メソッドを実行し，新たに構築したリソースと更新したリソースを確認した．この後に OSLC-SCM の領域で管理されている構成管理書に要求仕様書をベースライン登録することで，要求変更管理プロセスは終了する．

8. 評価

8.1. 要求仕様と管理情報の表現定義の妥当性

(1) IEEE 830 とプロパティについて

国際標準である IEEE 830 の記法と，Web を介したリソース統合基盤である OSLC で定義されているプロパティを対応付けることにより，OSLC 基盤上での要求仕様の管理が可能になった．また，OSLC を用いて交換可能な要求仕様の表現が定義可能となった．

(2) IEEE 830 とリソースについて

OSLC を用いて交換可能な要求仕様の表現を定義したことにより，機能単位での関連付けなど，リソース間の関係づけの局所化が可能になった．

(3) REBOK と OSLC のマッピング

REBOK による管理情報の標準と，OSLC のプロパティを対応付けたことにより，OSLC 上の管理情報の管理が可能となる．また，REBOK を用いたことにより，プロジェクトに依らない管理情報の定義が可能となる．この管理情報を(1)で定義した要求仕様に付加し，複数のリソースとして個別に管理することで，要求変更による影響リソースの局所化や，更新リソースの局所化が可能になった．

8.2. プロトタイプによる提案内容の妥当性確認

提案した要求管理モデルに基づくプロトタイプを構築，例題の適用を実行した．これにより，提案したモデルは OSLC 上で管理可能なことを確認できた．更に，プロトタイプにより以下の 3 点を確認できた．

(1) 提案モデルの妥当性

提案した要求管理モデルを用いて要求変更を実行する際，要求変更に伴い，影響が発生する機能要求に絞ったリソースの発見，リソースの更新における更新箇所の局所化を確認できた．

(2) 統一的な方法を用いたリソース管理の妥当性

プロトタイプにて，例題として，REBOK に定義されている要求変更管理プロセスを実行した．OSLC 標準のプロパティと Web の標準プロトコルを用いて，要求変更依頼に沿ったリソース管理が可能なことを確認できた．

(3) 要求管理モデルの検証と拡張

プロトタイプでは，OSLC リポジトリにて管理するリソースの型を XMLSchema にて定義した．これにより，構造と意味の検証が可能になると共に，継承機能によりプロジェクト特有の要求仕様書の表現が可能となることを確認した．

9. 今後の課題

IEEE 830 や REBOK は要求仕様や管理情報の雛型であり，プロジェクトに依らず，必要な要素である．今後は，IEEE 830 や REBOK を拡張し，組織特有の要求仕様書を構築した際の扱いや，プロジェクト特有の管理情報の付加について考える必要がある．

10. まとめ

本研究では，Web を介して要求管理を実行する際，統一的な表現，管理方法が未定義であるという課題に対して IEEE 830，REBOK に基づき，OSLC 上で要求仕様書をやり取りする際の統一的な要求仕様の表現定義，管理情報の定義を行った．更に，OSLC のプロパティと要求仕様と管理情報を対応付けたことにより Web 標準のプロトコルを用いた要求仕様と管理情報の管理を可能にした．また，OSLC RIO を拡張し，Word 文章として記述された IEEE 830 に沿った要求仕様書を入力とし，提案したリソース型に変換し，OSLC 基盤上で管理可能なプロトタイプを構築した．プロトタイプに REBOK にて定義されている要求変更管理プロセスを実行することで OSLC 上での要求仕様と管理情報の管理の妥当性を示した．

参考文献

[1] T. Berners-Lee, Linked Data, 2009, http://www.w3.org/DesignIssues/LinkedData.html.

[2] J. D. Herbsleb, et al., Global Software Development, IEEE Software, Vol. 18, No. 2, Mar./Apr. 2001, pp. 16-20.

[3] IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.

[4] OSLC(Open Services for Lifecycle Collaboration) , January 12, 2013, http://open-services.net/.

[5] REBOK 企画 WG, 要求工学知識体系 第1版, 近代科学社, 2011.