1M-2

OSLC に基づく分散リソースに対する 統一的変更管理方法の提案と評価

朝倉　知也†　青山　幹雄‡

南山大学大学院　数理情報研究科†　南山大学　情報理工学部　ソフトウェア工学科‡

1　研究の背景と課題

大規模ソフトウェアの分散開発では，複数拠点で変更要求を管理する必要がある．しかし，変更要求の共通的な表現や統一的管理方法は確立していない．本稿では IEEE Std.828-2005[2]と OSLC (Open Services for Lifecycle Collaboration)[3]のプロパティに基づき，Web を介して管理可能な変更要求の共通モデルを提案する．更に，複数拠点間で変更要求の共通モデルを時系列管理するために，変更管理システムのアーキテクチャを提案する．

2　関連技術

2.1　IEEE Std. 828-2005

IEEE Std.828-2005 はソフトウェア構成管理の標準規格である．変更要求のプロパティを定義している．

2.2　OSLC

OSLC は Web を介したツール間連携を目的とし，CM(Change Management)などのドメイン毎のリソースモデルを定義している．リソースは RDF/Linked Data で表現され，REST で操作できる．

2.3　TRS (Tracked Resource Set)[4]

TRS は，OSLC リソースの変更管理仕様である．対象リソース群に対する追加や更新などの変更を時系列管理する Change Log リソースと，ベースラインを表現する Base リソースで構成されている．リソースは RDF/Linked Data で表現され，REST で操作できる．

3　アプローチ

本稿では共通的な変更要求を表現するため，IEEE Std.828-2005 が定める変更要求プロパティを用いて変更要求の共通モデルを定義する．この共通モデルのプロパティを OSLC CM プロパティにマッピングし，変更要求の共通モデルの Web を介した管理を可能にする．また，複数拠点間で変更要求を時系列管理するため，TRS を用いた分散変更管理システム(DCMS: Distributed Change Management System)のアーキテクチャを提案する．

4　変更要求の共通モデル

標準規格に基づく変更要求の Web を介した管理を可能にするため，IEEE Std. 828-2005 のプロパティから OSLC CM プロパティへマッピングする(**表 1**).

表 1 IEEE Std.828-2005 と OSLC CM のマッピング

IEEE Std. 828-2005	OSLC_CM	IEEE Std. 828-2005	OSLC_CM
作成者	dcterms:creator	番号	dcterms:identifer
作成日	dcterms:created	状態	oslc_cm:state
分類	dcterms:subject	優先度	oslc_cm:priority
説明/必要性	dcterms:description, dcterms:title	関連リソース	Relationship properties

プロパティマッピング結果から，IEEE Std.828-2005 が定義するプロパティを全て満たすための OSLC CM プロパティを抽出した(**図 1**).

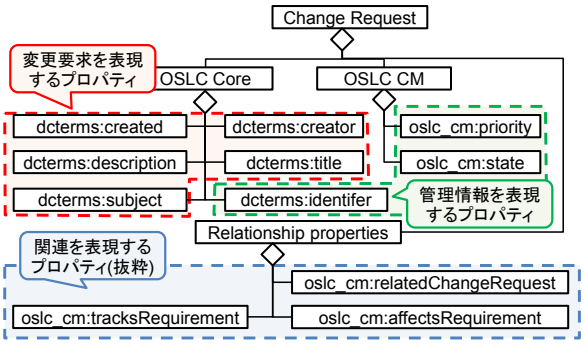

図 1 管理対象の変更要求モデル

5　DCMS の提案

5.1　DCMS のアーキテクチャ

4 章で定義した変更要求の共通モデルを管理対象とし，TRS を用いた時系列管理を可能とする DCMS のアーキテクチャを**図 2** に示す．

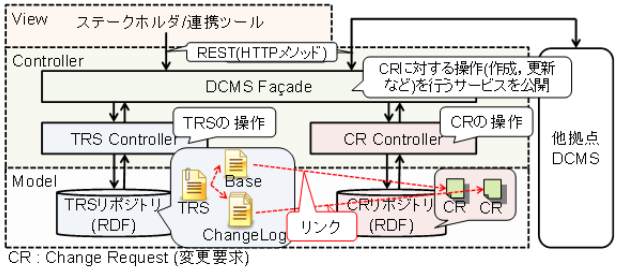

図 2 DCMS のアーキテクチャ

DCMS アーキテクチャは MVC アーキテクチャに基づき，View，Controller，Model で構成した．

(1) View

ステークホルダはブラウザを用い，REST で変更要求が管理できる．また，公開されている Web API を用いることで，他拠点 DCMS と連携できる．

(2) Controller

外部に公開する変更要求処理メソッドを定義する Façade と，TRS/CR リソースの作成や更新を行う Controller で構成した．

A Unified Change Management Method for Distributed Resources Based on the OSLC.

†Tomoya Asakura, Graduate School of Mathematical Sciences and Information Engineering, Nanzan University.

‡Mikio Aoyama, Department of Software Engineering, Nanzan University.

(3) Model

TRS/CR リソースを管理する．これらのリソースは RDF/XML 形式であるため，RDF リポジトリを用いる．

5.2 DCMS の振舞い

DCMS の変更要求作成の振舞いを図 3 に示す．

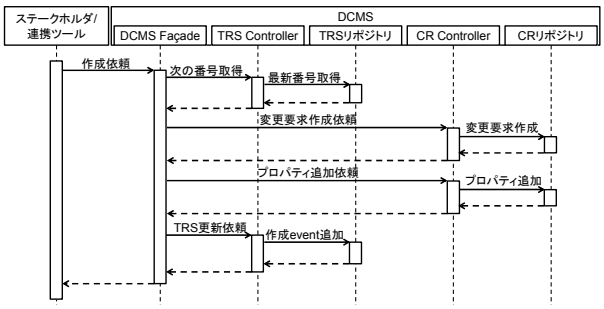

図 3 変更要求作成の振舞い

変更依頼を受けた Façade は TRS Controller を用いて最新の event 番号を取得し，CR Controller を用いて変更要求の作成，プロパティの追加を行う．次に，作成された変更要求の event を Change Log に追加する．event の URI に ISO 8601 拡張表記に基づくタイムスタンプを利用することで，分散開発拠点間での時系列管理を可能とする．

6 プロトタイプの構築

6.1 プロトタイプ構築の目的

(1) 変更要求の Web を介した時系列管理

変更要求の共通モデルを OSLC リソースとして作成し，Web を介して管理可能か検証する．また，複数の DCMS で並行して変更要求を作成，更新し，時系列に基づいた連携管理が可能か検証する．

(2) 他ツールとの連携

プロジェクト管理システムの Redmine と連携し，本稿で定義した変更要求の共通モデルを Redmine のチケットとして登録可能か検証する．

6.2 プロトタイプの構成

プロトタイプは Eclipse Lyo 上の TRS のリファレンス実装を基に作成した(図 4)．Web サーバに Jetty，RDF モデル操作に Apache Jena，RDF リポジトリに Fuseki[1]を用いた．変更要求連携の確認のため，拠点 A，拠点 B にプロトタイプを配置した．拠点 A は DCMS Controller と DCMS Model を異なる PC 上に，拠点 B は同一 PC 上に構築した．Redmine との連携には，Redmine の REST API を用いる．

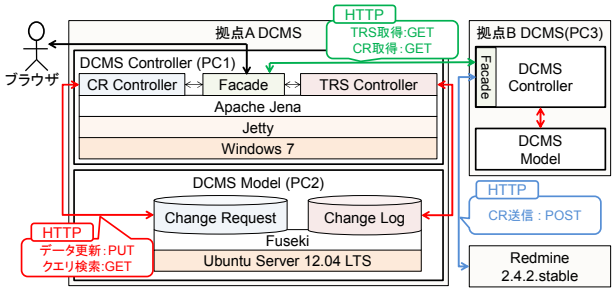

図 4 プロトタイプの構成

7 例題への適用

7.1 例題の概要

構築したプロトタイプに SWEBOK の変更要求処理プロセス(図 5)を適用し，6.1 節の項目を検証する．

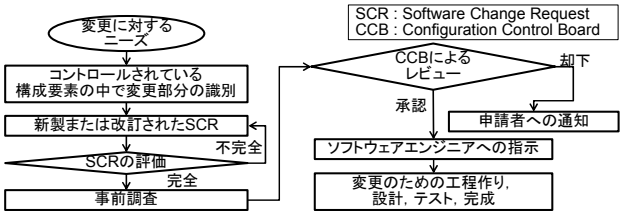

図 5 SWEBOK の変更要求処理プロセス

7.2 プロトタイプへの適用

拠点 A と拠点 B で図 5 のプロセスを並行して実施し，各 50 件の変更要求を作成した．拠点 A の DCMS ではブラウザで，拠点 B の DCMS では REST で，変更要求の作成と変更要求プロパティの追加を行った．各プロセス終了時に TRS を参照し，両拠点の時系列変更一覧を取得した．また，変更要求のプロパティを Redmine が定義するチケットのスキーマに変換し，REST API を用いてチケット登録を行った．

8 評価

(1) 変更要求の Web を介した時系列管理

ブラウザと REST を用いて変更要求を作成し，各拠点の TRS を取得，マージすることで，ランダムに発生した各拠点の event を時系列に基づいて管理できた．TRS を用いることで，変更要求を取得せず，全拠点の時系列変更一覧を取得可能である．

(2) 他ツールとの連携

変更要求のプロパティと Redmine のチケットフィールドをマッピングすることで，管理モデルを相互変換可能であることを確認した．これにより，Redmine と任意の OSLC 対応ツール間で相互にデータ交換が可能になる．その結果，異なる構成管理ツールや要求管理ツールと連携可能である．

9 まとめ

本稿では IEEE Std. 828-2005 と OSLC CM のプロパティをマッピングし，Web を介して管理可能な変更要求の共通モデルを定義した．更に，分散変更管理システムを提案し，複数拠点で変更要求の時系列管理を可能とした．また，プロトタイプを作成し例題に適用することで提案アーキテクチャの妥当性を検証した．

参考文献

[1] Fuseki: serving RDF data over HTTP, http://jena.apache.org/documentation/serving_data/ .
[2] IEEE Std. 828-2005, IEEE Standard for Software Configuration Management Plans, IEEE, 2005 .
[3] OSLC (Open Services for Lifecycle Collaboration), http://open-services.net/ .
[4] TRS (Tracked Resource Set), Oct 23, 2013, http://open-services.net/wiki/core/TrackedResourceSet-2.0/ .