総務省情報流通行政局情報流通振興課　御中

自治体行政情報流通連携基盤システム
実装詳細仕様書

平成 26 年 3 月 20 日

株式会社エヌ・ティ・ティ・データ

目次

1. 本書について 1-1
2. 自治体行政情報流通連携基盤システム概要 2-1
2.1. 自治体行政情報流通連携基盤システム概要と利用方法 2-1
2.1.1. 自治体行政情報流通連携基盤システム概要・目的 2-1
2.1.2. 自治体行政情報流通連携基盤システムと関連システムの構成 2-2
2.1.3. 利用者の特徴 2-4
2.1.4. 利用シーン 2-5
2.2. 自治体行政情報流通連携基盤システムの機能・非機能 2-13
2.2.1. 機能一覧 2-13
2.2.2. 機能定義 2-15
2.2.3. 非機能要件 2-27
3. 自治体行政情報流通連携基盤システム詳細 3-1
3.1. ソフトウェア設計 3-1
3.1.1. ソフトウェア一覧 3-1
3.1.2. ソフトウェアの選定理由 3-2
3.1.3. ソフトウェアライセンス 3-6
3.1.4. 機器構成及びソフトウェア構成 3-9
3.1.5. ディレクトリ・ファイル構成 3-10
3.2. 自治体行政情報標準 API 設計 3-13
3.2.1. ソフトウェア一覧 3-13
3.2.2. ソフトウェア設計 3-14
3.2.3. オンライン処理設計 3-15
3.2.4. メッセージ設計 3-22
3.3. データ変換・登録ツール設計 3-23
3.3.1. コンポーネント一覧 3-23
3.3.2. コンポーネント図 3-24
3.3.3. 画面設計 3-27
3.3.4. オンライン処理設計 3-56
3.3.5. バッチ処理設計 3-82
3.3.6. メッセージ設計 3-91
3.3.7. データ変換・登録ツール単独での利用について 3-93
3.4. その他設計 3-95
3.4.1. ファイル設計 3-95
3.4.2. ログ管理設計 3-104
3.4.3. ユーザ定義 3-107
3.4.4. パスワードポリシー 3-107
4. 自治体行政情報流通連携基盤システムの構築・動作確認手順 4-1
4.1. 自治体行政情報流通連携基盤システムの構築手順 4-1
4.1.1. 公開サーバの構築 4-1

4.1.2.　管理サーバの構築 4-5
4.2.　自治体行政情報流通連携基盤システムの動作確認手順 4-9
4.2.1.　公開サーバの動作確認手順 4-9
4.2.2.　管理サーバの動作確認手順 4-10
4.2.3.　公開サーバ・管理サーバの結合確認手順 4-10
5.　本書のまとめ 5-1

1.　本書について

本書は、「情報流通連携基盤の自治体行政情報における実証に係る請負」にて設計・構築を行った「自治体行政情報流通連携基盤システム」における詳細仕様をまとめたものである。

「自治体行政情報流通連携基盤システム」を設計・構築した目的については、「情報流通連携基盤の自治体行政情報における実証に係る請負」の背景と併せて後述(2.1.1 参照)する。

本書の構成は以下のとおりである。

表 1-1 実装詳細仕様書 章構成

章番	章タイトル	内容	主な想定読者
1	本書について	本書の目的について述べる。	・自治体職員 ・自治体行政情報流通連携基盤システムの実装・構築担当者
2	自治体行政情報流通連携基盤システム概要	本書の記載対象となる「自治体行政情報流通連携基盤システム」の目的、利用方法、利用する上での業務内容、備える機能、求められる非機能について述べる。	・自治体職員 ・自治体行政情報流通連携基盤システムの実装・構築担当者
3	自治体行政情報流通連携基盤システム詳細仕様	「自治体行政情報流通連携基盤システム」の処理設計や画面設計等、詳細な仕様について述べる。	・自治体行政情報流通連携基盤システムの実装・構築担当者
4	自治体行政情報流通連携基盤システムの構築・動作確認手順	「自治体行政情報流通連携基盤システム」の構築手順や、構築後の動作確認手順について述べる。	・自治体行政情報流通連携基盤システムの実装・構築担当者
5	本書のまとめ	本書のまとめを述べる。	・自治体職員 ・自治体行政情報流通連携基盤システムの実装・構築担当者

2.　自治体行政情報流通連携基盤システム概要

2.1.　自治体行政情報流通連携基盤システム概要と利用方法

2.1.1. 自治体行政情報流通連携基盤システム概要・目的

「これまで ICT 利活用については、行政、医療、教育等の個別分野ごとの情報化を促進することに力点が置かれてきた。一方、東日本大震災において、企業等が行政の保有する避難所の情報、地図データ等を利用して震災関連情報を広く周知しようとしたが、データが PDF 、JPEG 等、機械判読が困難な形式で提供されている場合が多く、機械可読なフォーマットに人手で加工する必要がある。二次利用が困難なケース、あるいは行政機関ごとにフォーマットが異なり、情報の収集や整理に多くの時間が必要とされるケースが発生する等、情報の横の連携ができていないことが顕在化した。

こうした状況等を背景として、高度情報通信ネットワーク社会推進戦略本部（IT 総合戦略本部）において、平成 24 年 7 月に、公共データを社会全体で効率的に活用促進するための基本戦略として「電子行政オープンデータ戦略」が決定された。「電子行政オープンデータ戦略」に基づき、オープンデータの記述形式や意味の差異を吸収し、情報・知識やサービスを連携・共有するための汎用性ある技術・運用ルール等が整った環境（情報流通連携基盤）の構築が総務省によって推進されている。

「電子行政オープンデータ戦略」においては、オープンデータの取組は、まずは政府（国）が保有するデータについて率先して取組を推進し、自治体については、その取組成果を波及させていくことが基本的考え方とされている。地域における経済活性化等に資する観点から、自治体のオープンデータ化を加速するため、総務省による「情報流通連携基盤の自治体行政情報における実証に係る請負」により、横浜市、鯖江市を実証フィールドとして、自治体における利活用ニーズの高さ等の観点からオープンデータ化すべき自治体行政情報を特定した上で、費用面や時間面の負担も踏まえつつ、情報流通連携基盤システムのうち、自治体において必要となる機能を抽出し、広く自治体に普及展開できるモデルを「自治体行政情報流通連携基盤システム」として設計・構築を行った。

なお、自治体行政情報流通連携基盤システムは、「情報流通連携基盤 外部仕様書 Ver1.1」[1]（以降、「外部仕様書」とする）に定義された標準 API やデータ規格に従い設計されたものである。

[1] http://www.opendata.gr.jp/cfc/spec.pdf

2.1.2. 自治体行政情報流通連携基盤システムと関連システムの構成

自治体行政情報流通連携基盤システムは、自治体職員が機械可読に加工したファイルや、外部システム（CMS 等）からアップロードされる加工ファイルを読み込み、RDF 形式に変換し、データベース（RDF store）に登録することでオープンデータ化を行う仕組みを持つ。

また、自治体行政情報流通連携基盤システムに登録されたデータの情報は、その自治体で保有しているデータカタログサイト(CKAN）に掲載する仕組みも持つ。

以下に、自治体行政情報流通連携基盤システムと、その関連システム（外部システム（CMS 等)、データカタログサイト（CKAN)）との構成イメージを示す。

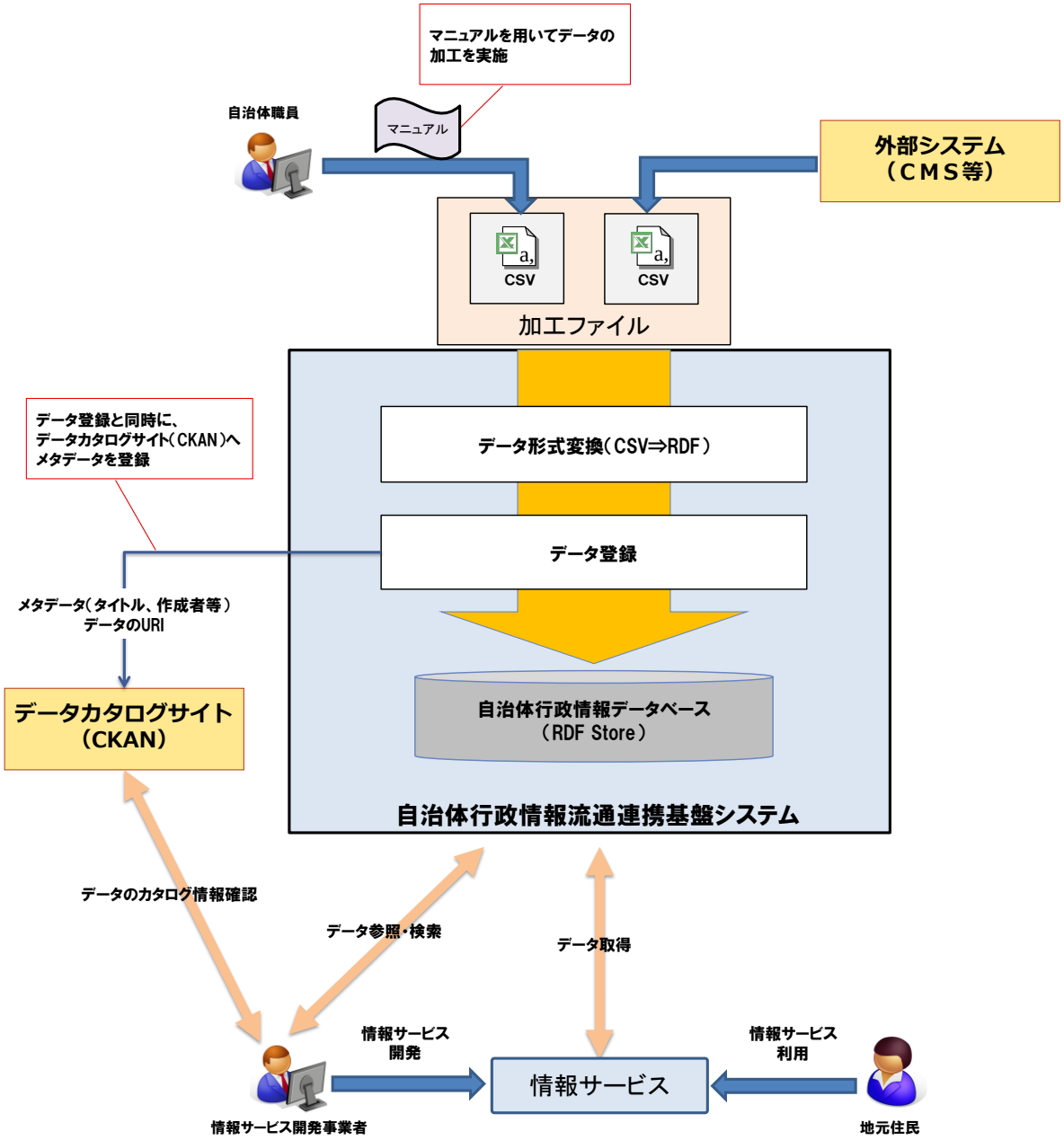


図 2-1　自治体行政情報流通連携基盤システムと関連システムの構成イメージ

また、図 2-1 にある通り、自治体行政情報流通連携基盤システムを取り巻く利用者は以下の３つに分けられる。

1) 自治体職員（データ提供者）
オープンデータ化するデータの登録・更新等を行い、管理する。併せて、データカタログサイトへの情報掲載も実施する。

2) 情報サービス開発事業者（データ利用者）
データカタログサイトを参照して自治体行政情報流通連携基盤システムからオープンデータの取得・参照を行い、情報サービスを開発する。

3) 地元住民（データ利用者）
情報サービス開発事業者が開発した情報サービスを通じて、オープンデータを参照する。

2.1.3. 利用者の特徴

2.1.2 で述べた自治体行政情報流通連携基盤システムの利用者（自治体職員、情報サービス開発事業者、地元住民）について、それぞれの特徴を以下に示す。

表 2-1 利用者 特徴一覧

ＩＤ	利用者名	説明	利用形態	利用者の特性
A-1	自治体職員	自治体行政情報流通連携基盤システムにオープンデータを登録し、管理する。	Web 画面を通じてオープンデータの登録・管理を行う。	オープンデータに関する知識に乏しい。
B-1	情報サービス開発事業者	自治体行政情報流通連携基盤システムに登録されたオープンデータを活用して新たな情報サービスを開発する。	自治体行政情報流通連携基盤システムが備えた API を利用してオープンデータの検索・取得を行う。	API の仕様を理解することができ、情報サービス開発の知識がある。
C-1	地元住民	オープンデータを活用した情報サービスを利用してオープンデータを参照する。	情報サービス経由でオープンデータを利用する。	PC やタブレット端末、スマートフォンのアプリケーションをインストール、利用することが出来る。

2.1.4. 利用シーン

以下に、自治体行政情報流通連携基盤システム、データカタログサイトとこれらの利用者を含めた利用イメージを示す。

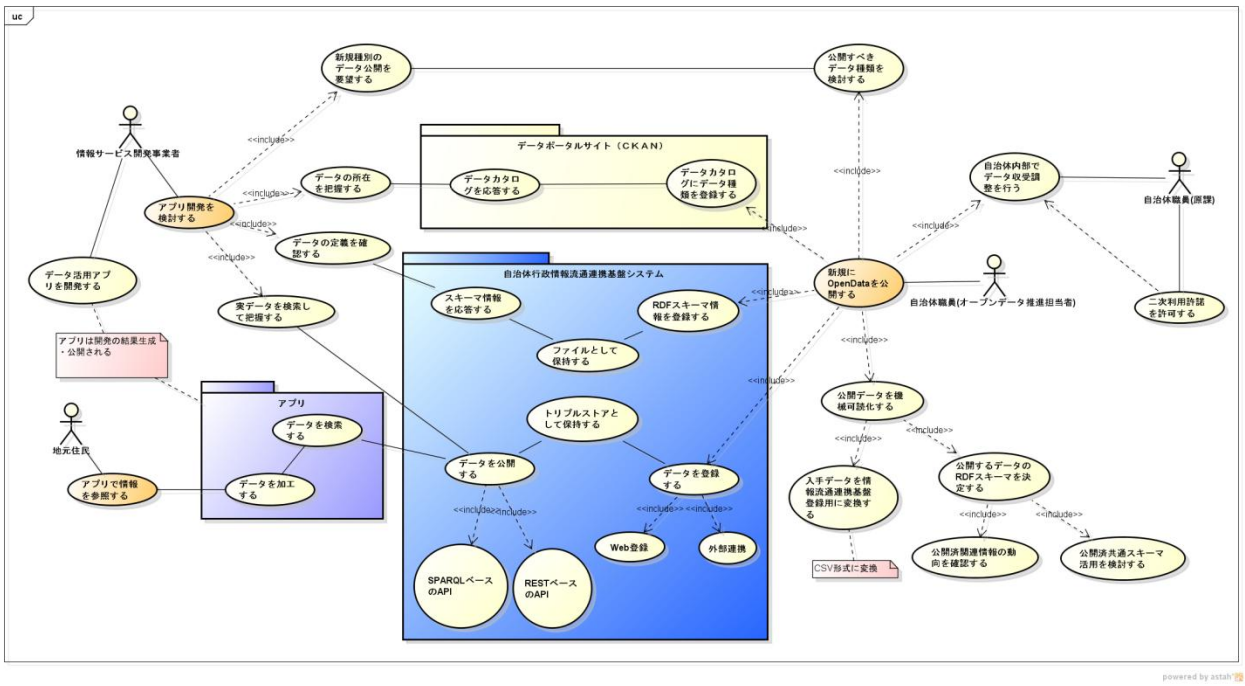


図 2-2　自治体行政情報流通連携基盤システム 利用イメージ

また利用者の観点から、自治体行政情報流通連携基盤システムを介した利用シーンは、主に以下 3 点となる。

(1)　自治体職員によるオープンデータ化業務
(2)　情報サービス開発事業者によるオープンデータの検索・取得業務
(3)　地元住民が利用する情報サービスによるオープンデータの取得業務

これらの業務について、以下に業務フローを示す。

なお、以降の業務フローにおいては、イメージを明確化すべく、自治体が管理していると想定した市内の動物園データ及び植物園データを例として挙げている。

（1）　自治体職員によるオープンデータ化業務フロー

自治体職員によるオープンデータ化業務に係る業務フロー図を以降に示す。なお、自治体職員には主に以下 3 つの業務がある。以下 3 つの業務に対して、業務フローを示す。

1)新規登録：データを基盤へ登録し、オープンデータ化を行う。
2)更新　　：既に基盤に登録したオープンデータの修正を行う。
3)削除　　：既に基盤に登録したオープンデータの削除を行う。

これら 3 つの業務に対して、業務フローを示す。

1）新規登録

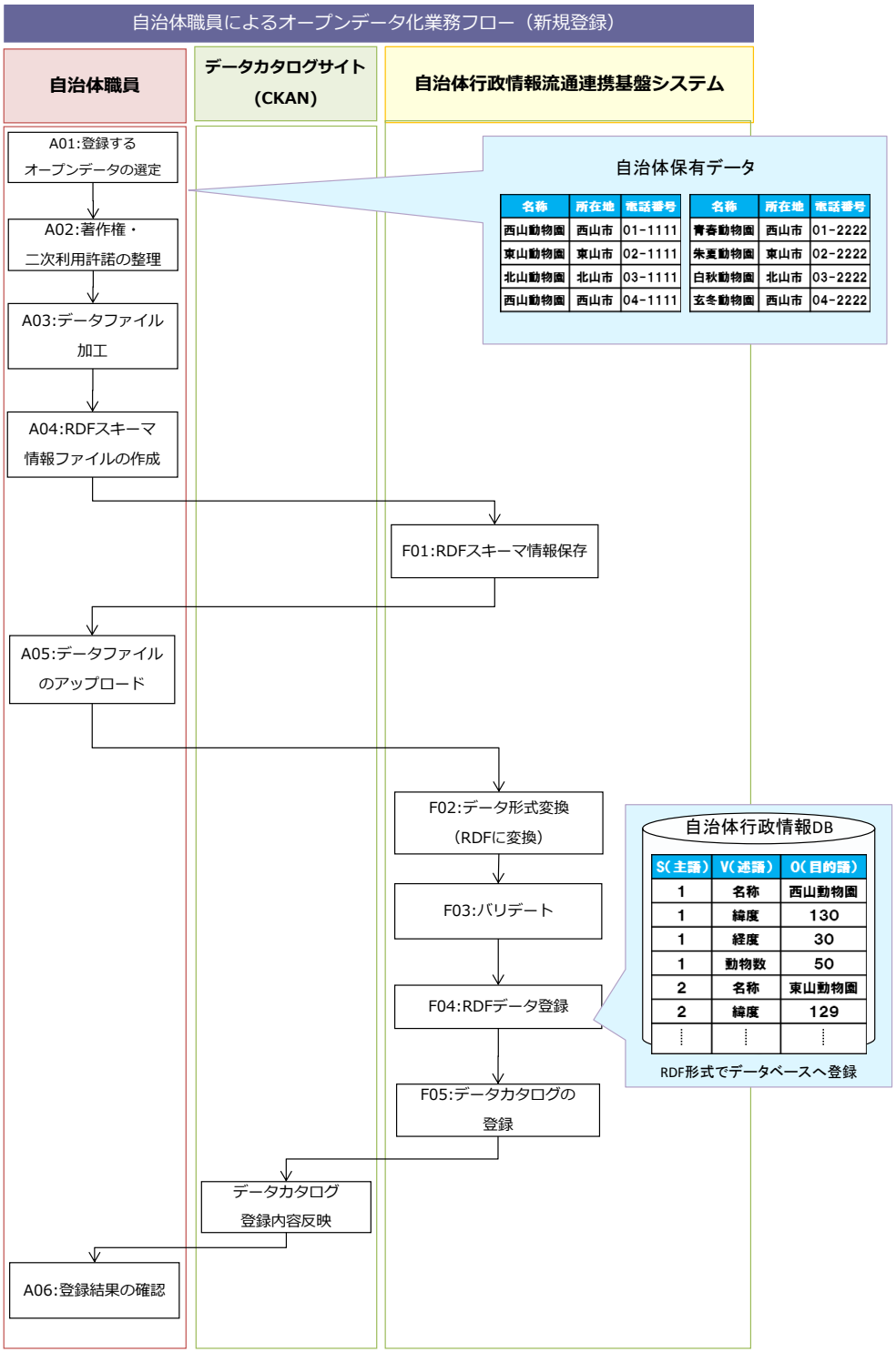


図 2-3　自治体職員によるオープンデータ化業務に係る業務フロー(新規登録)

2）更新

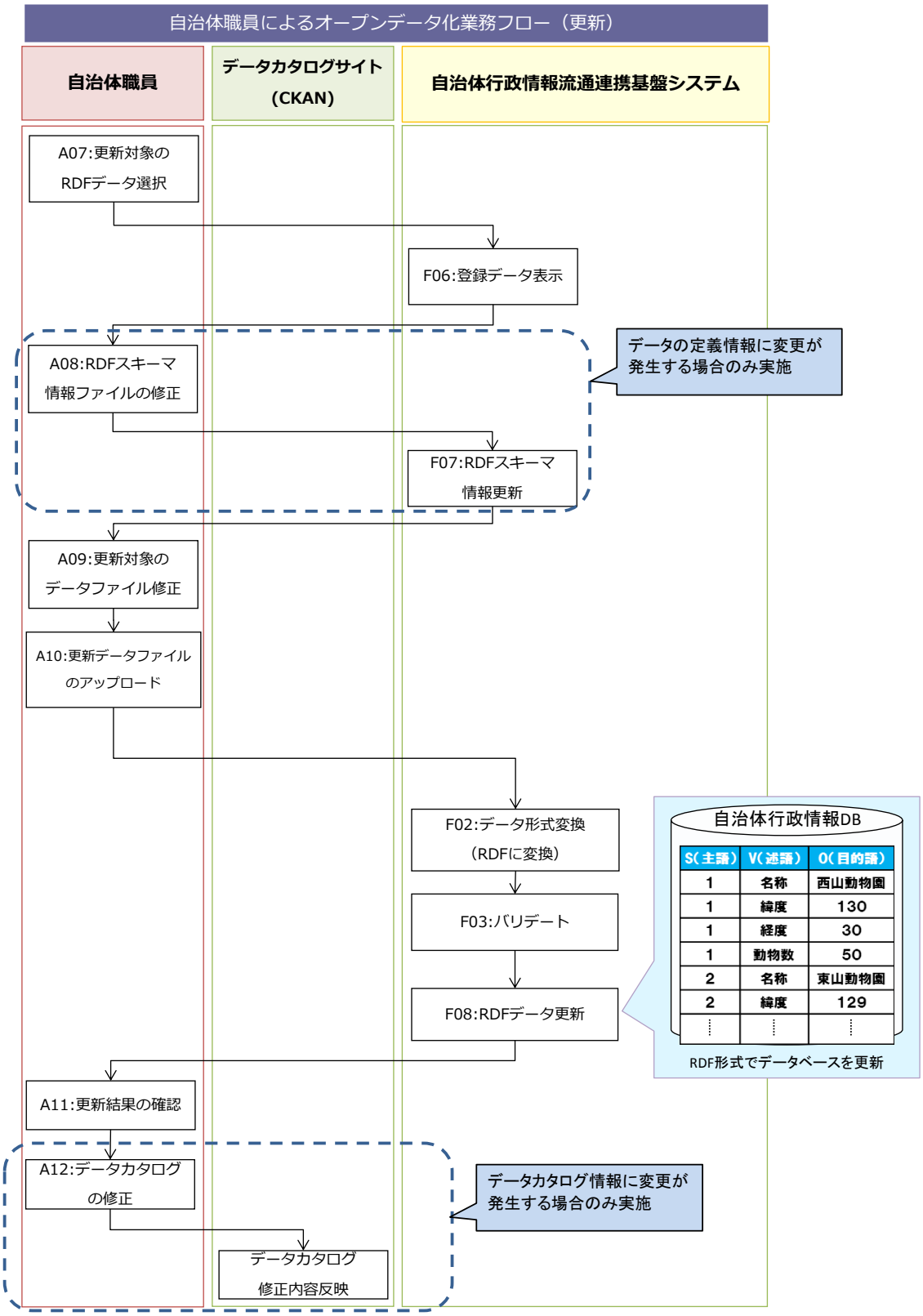


図 2-4　自治体職員によるオープンデータ化業務に係る業務フロー(更新)

3）削除

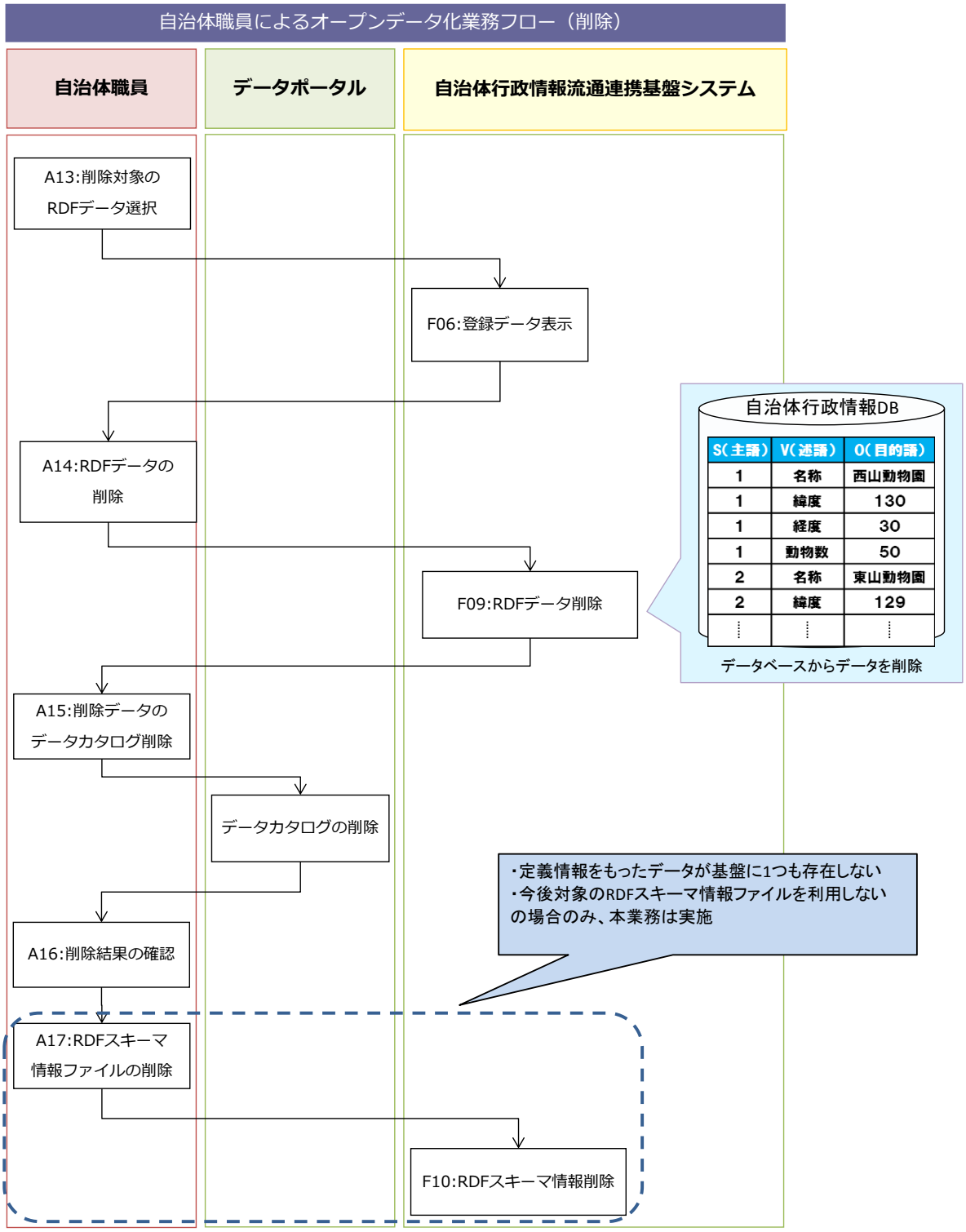


図 2-5　自治体職員によるオープンデータ化業務に係る業務フロー(削除)

（2）　情報サービス開発事業者におけるオープンデータの検索・取得業務フロー

情報サービス開発事業者におけるオープンデータの検索・取得業務に係る業務フロー図を以下に示す。

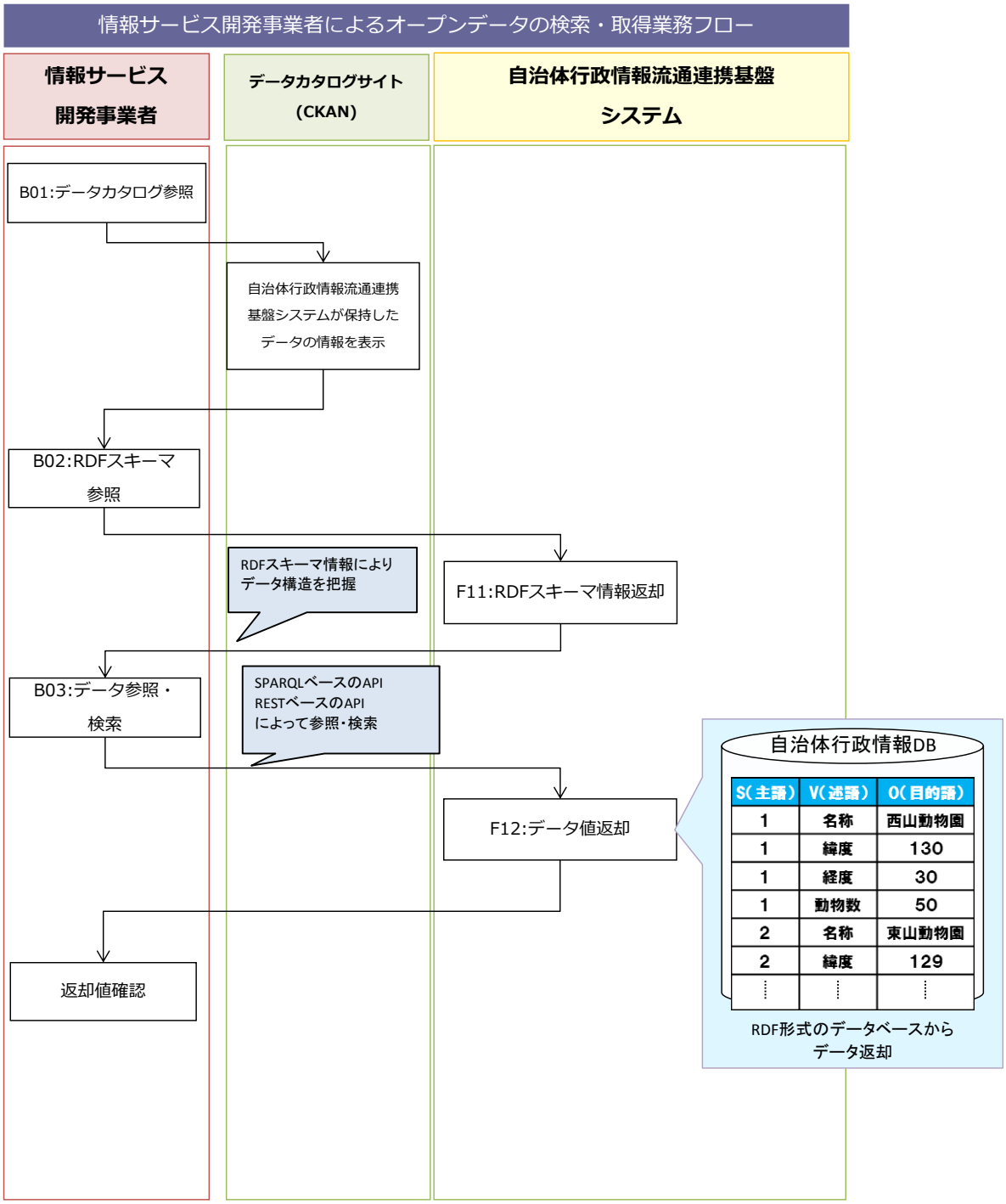


図 2-6　情報サービス開発事業者におけるオープンデータの検索・取得業務に係る業務フロー

（3）　地元住民による情報サービスを利用したオープンデータの取得業務フロー

地元住民による情報サービスを利用したオープンデータの取得業務に係る業務フロー図を以下に示す。

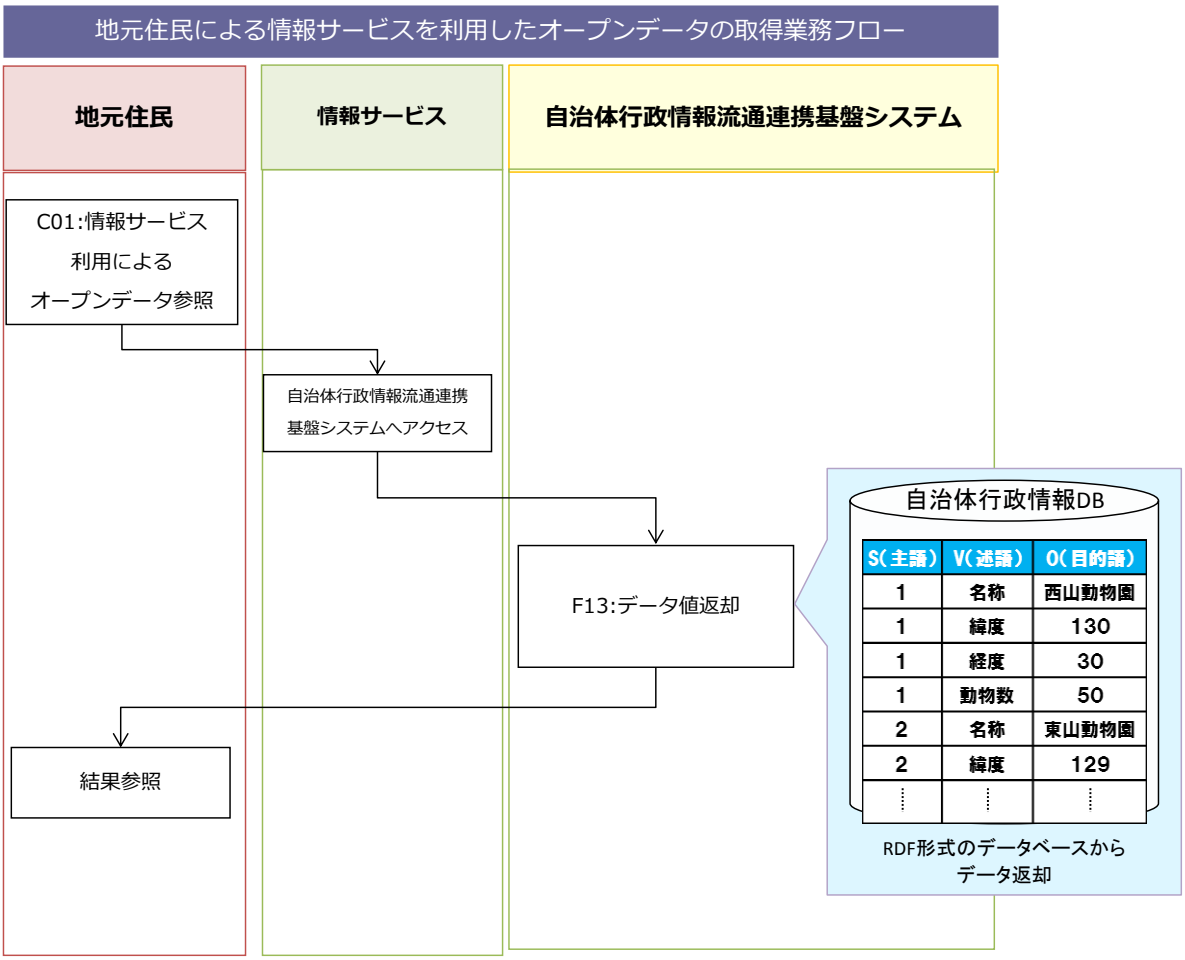


図 2-7　地元住民による情報サービスを利用したオープンデータの取得業務に係る業務フロー

2.2.　自治体行政情報流通連携基盤システムの機能・非機能

本節では、自治体行政情報流通連携基盤システムが備える機能や、要求される非機能について述べる。これらは、前述で整理した業務フローから必要と考えられるものを導出している。

2.2.1. 機能一覧

2.1.4 で述べた業務を行うために必要な機能を以下に示す。「関連業務名」については、2.1.4 で記載した業務フローの自治体職員、情報サービス開発事業者、地元住民が行う業務名を表している。

表 2-2 機能一覧

関連業務名	機能 ID	機能名	利用者	概要
A04:RDF スキーマ情報ファイルの作成	F01	RDF スキーマ情報保存	自治体職員	RDF スキーマ情報ファイルを基盤に保存する。
A05:データファイルのアップロード	F02	データ形式変換	自治体職員	整形された CSV や SDF（3.4.1 参照）のファイルを、RDF 形式のファイルに変換する。
	F03	バリデート	自治体職員	入力された内容が正しいかチェックを行う。
	F04	RDF データ登録	自治体職員	規定の形式に加工されたデータファイルを基盤に登録する。
	F05	データカタログの登録	自治体職員	基盤に登録した内容を元にデータポータルへデータカタログを登録する。
A07:更新対象の RDF データ選択	F06	登録データ表示	自治体職員	指定した RDF データの登録内容を表示する。
A08:RDF スキーマ情報ファイルの修正	F07	RDF スキーマ情報更新	自治体職員	基盤に登録された RDF スキーマ情報ファイルを更新する。
A10:更新データファイルのアップロード	F08	RDF データ更新	自治体職員	基盤に登録済みのデータについて更新処理を行う。
A14:RDF データの削除	F09	RDF データ削除	自治体職員	基盤に登録済みのデータについて削除処理を行う。
A17:RDF スキーマ情報ファイルの削除	F10	RDF スキーマ情報削除	自治体職員	基盤に登録された RDF スキーマ情報ファイルを削除する。
B02:RDF スキーマ参照	F11	RDF スキーマ情報返却	情報サービス開発事業者	要求に従い、データの RDF スキーマ情報を返却する。
B03:データ参照・検索 C01:情報サービス利用によるオープンデータ参照	F12	データ値返却（SPARQL ベース）	情報サービス開発事業者	SPARQL ベースの API 仕様(表 3-9 参照)の命令要求に従い、登録されたデータ値を返却する。
	F13	データ値返却（REST ベース）	情報サービス開発事業者	REST ベースの API 仕様(表 3-9 参照)の命令要求に従い、登録されたデータ値を返却する。
-	F14	登録データ検索	自治体職員	基盤に登録されているデータを検索し、結果を表示する。
-	F15	RDF データ再変換	自治体職員	基盤に登録済みのデータについて再度変換を行う。

2.2.2. 機能定義

前述した各機能の定義を以降に示す。

表 2-3 機能定義（F01）

項目	内容
関連業務名	A04:RDF スキーマ情報ファイルの作成
機能 ID	F01
機能名	RDF スキーマ情報保存
オンライン/バッチ	オンライン
利用者	自治体職員
目的	情報サービス開発事業者がデータの定義情報等を理解するため、また、F03:バリデートで使用する RDF スキーマ情報を RDFS ファイルとして保存し、参照できるようにする。
事前条件	RDF スキーマ情報ファイルが RDFS ファイルで定義されていること。
事後条件	RDF スキーマ情報ファイルが RDFS ファイルで保存されること。
イベントフロー	1. 自治体職員は、データの RDF スキーマ情報の RDFS ファイルを作成する。 2. システムは、ファイルを保存する。
備考	

表 2-4 機能定義（F02）

項目	内容
関連業務名	A05:データファイルのアップロード
機能 ID	F02
機能名	データ形式変換
オンライン/バッチ	オンライン/バッチ
利用者	自治体職員
目的	登録データとして入力されたデータファイルを、RDF として登録するため、RDF 形式に変換する。
事前条件	データファイルは、決まった規格に沿ったフォーマット、または SDF(Simple Data Format)に沿ったフォーマットに加工されていること。
事後条件	入力データが RDF 形式に変換されていること。
イベントフロー	F04:RDF データ登録 または F08:RDF データ更新 のイベント 2 まで実施されてから開始。 1. システムは、読み込まれたデータファイルの形式を RDF に変換する 2. RDF 変換が正常に完了したメッセージを、F03:バリデートの機能に渡す。変換が失敗した場合は、その旨と理由をメッセージとして利用者に出力する。
備考	

表 2-5 機能定義（F03）

項目	内容
関連業務名	A05:データファイルのアップロード
機能 ID	F03
機能名	バリデート
オンライン/バッチ	オンライン/バッチ
利用者	自治体職員
目的	登録・更新したいデータが、全て正しいデータ型（数値、文字列など）となっているかをチェックする。
事前条件	入力データファイルが RDF 形式に変換されていること。
事後条件	全てのデータのバリデートが完了していること。
イベントフロー	1. システムは、RDF データの値が正しい形式かチェックを行う。標準では RDF データの述語の URI のチェックのみ行い、オプションで RDF データの値の型のチェックを行う。 2. チェックが正常に完了したメッセージを、以降に実行される機能（F04:RDF データ登録、F08:RDF データ更新）に渡す。誤った形式のデータがあった場合、エラーメッセージを利用者に出力する。
備考	

表 2-6 機能定義（F04）

項目	内容
関連業務名	A05:データファイルのアップロード
機能 ID	F04
機能名	RDF データ登録
オンライン/バッチ	オンライン/バッチ
利用者	自治体職員
目的	自治体職員がオープン化対象のデータファイルを RDF データとして基盤に登録する。
事前条件	データファイルは、決まった規格に沿ったフォーマット、または SDF(Simple Data Format)に沿ったフォーマットに加工されていること。
事後条件	対象データファイルが RDF データとして基盤に登録されていること。
イベントフロー	1. 自治体職員は、登録したいデータファイルをアップロードする。 2. システムは、入力されたデータファイルを読み込む。 (オンライン/バッチ処理は自治体職員が UI で選択する) 以降、バッチ処理の場合はデータファイルの登録のみをオンラインで行い、後工程はバッチで処理を行う。 オンライン処理の場合は、読み込まれたデータファイルに対して F02:データ形式変換⇒F03:バリデートが実行される。 （F02、F03 の機能定義は後述(2.2.2 参照)） 3. RDF データをデータベースに登録する。 4. 登録完了の旨のメッセージを利用者に対して出力する。登録失敗した場合も、その旨と理由をメッセージとして利用者に対して出力する。
備考	

表 2-7 機能定義（F05）

項目	内容
関連業務名	A05:データファイルのアップロード
機能 ID	F05
機能名	データカタログ登録
オンライン/バッチ	オンライン
利用者	自治体職員
目的	自治体職員がオープン化対象のデータファイルのデータカタログをデータポータルに登録する。
事前条件	データファイルが基盤に登録されていること。
事後条件	対象データカタログがデータポータルに登録されていること。
イベントフロー	1. 自治体職員は、登録したいデータファイルをアップロードする。 2. 変換された RDF ファイルのメタデータを読み取り、データカタログとしてデータポータルに登録する。 3. 登録完了の旨のメッセージを利用者に対して出力する。登録失敗した場合も、その旨と理由をメッセージとして利用者に対して出力する。
備考	

表 2-8 機能定義（F06）

項目	内容
関連業務名	A07:更新対象の RDF データ選択
機能 ID	F06
機能名	登録データ表示
オンライン/バッチ	オンライン
利用者	自治体職員
目的	登録状況を確認したい RDF データの内容を確認する。
事前条件	確認したいデータが画面上に一覧で表示されていること。
事後条件	確認対象として選択された RDF データが ・登録前のデータファイル ・登録後の RDF データ で表示されること。
イベントフロー	1. 自治体職員は、画面上に一覧表示されている RDF データの中から、確認したいデータを選択する。 2. システムは、選択されたデータの内容を、 ・登録前のデータファイル ・登録後の RDF データ で表示する。
備考	F02 データ形式,F03 バリデートに失敗した場合は、データベースへの登録を行わないため、登録後の RDF データの表示は行わないものとする。

表 2-9 機能定義（F07）

項目	内容
関連業務名	A08:RDF スキーマ情報ファイルの修正
機能 ID	F07
機能名	RDF スキーマ情報更新
オンライン/バッチ	オンライン
利用者	自治体職員
目的	登録済の RDF スキーマ情報に対して、定義内容に変更が発生した場合に、基盤の RDF スキーマ情報にも変更を反映させる。
事前条件	更新対象 RDF スキーマ情報は、基盤に登録済であること。
事後条件	対象 RDF スキーマ情報が更新されていること。
イベントフロー	1. 自治体職員は、更新内容を反映した RDF スキーマ情報ファイルをシステムにアップロードする。 2. システムは、入力されたデータファイルを読み込む。 3. データベースに格納された RDF スキーマ情報を更新する。 4. 更新完了の旨のメッセージを利用者に対して出力する。更新失敗した場合も、その旨と理由をメッセージとして利用者に対して出力する。
備考	

表 2-10 機能定義(F08)

項目	内容
関連業務名	A10:更新データファイルのアップロード
機能 ID	F08
機能名	RDF データ更新
オンライン/バッチ	オンライン/バッチ
利用者	自治体職員
目的	登録済のデータに対して、元データに変更が発生した場合に、基盤の RDF データにも変更を反映させる。
事前条件	更新対象データは、基盤に登録済であること。
事後条件	対象 RDF データが更新されていること。
イベントフロー	1. 自治体職員は、更新内容を反映したデータファイルをシステムにアップロードする。 2. システムは、入力されたデータファイルを読み込む。 (オンライン/バッチ処理は自治体職員が UI で選択する) 以降、バッチ処理の場合はデータファイルの登録のみをオンラインで行い、後工程はバッチで処理を行う。 オンライン処理の場合は、登録内容に対して F02:データ形式変換⇒F03:バリデートが実行される。 （F02、F03 の機能定義は後述(2.2.2 参照)） 3. データベースに格納された RDF データを更新する。 4. 更新完了の旨のメッセージを利用者に対して出力する。更新失敗した場合も、その旨と理由をメッセージとして利用者に対して出力する。
備考	

表 2-11 機能定義(F09)

項目	内容
関連業務名	A14:RDF データの削除
機能 ID	F09
機能名	RDF データ削除
オンライン/バッチ	オンライン
利用者	自治体職員
目的	登録済の RDF データに対して、運用上削除が必要となった場合、削除を行う。
事前条件	削除対象 RDF データは、基盤に登録済であること。
事後条件	対象 RDF データが削除されていること。
イベントフロー	1. 自治体職員は、削除対象の RDF データを選択する。 2. システムは、選択された RDF データを検索する。 3. 対象となる RDF データをデータベース上から削除する。 4. 削除完了の旨のメッセージを利用者に対して出力する。削除失敗した場合も、その旨と理由をメッセージとして利用者に対して出力する。
備考	

表 2-12 機能定義(F10)

項目	内容
関連業務名	A17:RDF スキーマ情報ファイルの削除
機能 ID	F10
機能名	RDF スキーマ情報削除 `
オンライン/バッチ	オンライン
利用者	自治体職員
目的	登録済の RDF スキーマ情報に対して、運用上削除が必要となった場合、削除を行う。
事前条件	・削除対象 RDF スキーマ情報は、基盤に登録済であること。
事後条件	・対象 RDF スキーマ情報が削除されていること。
イベントフロー	1. 自治体職員は、削除対象の RDF スキーマ情報を選択する。 2. システムは、選択された RDF スキーマ情報を検索する。 3. 対象となるデータベースに格納された RDF スキーマ情報を削除する。 4. 削除完了の旨のメッセージを利用者に対して出力する。削除失敗した場合も、その旨と理由をメッセージとして利用者に対して出力する。
備考	

表 2-13 機能定義(F11)

項目	内容
関連業務名	B02:RDF スキーマ参照
機能 ID	F11
機能名	RDF スキーマ情報返却
オンライン/バッチ	オンライン
利用者	情報サービス開発事業者
目的	登録データの RDF スキーマを確認し、サービス内容を検討する。
事前条件	・RDF スキーマ情報が登録済であること。 ・RDF スキーマ情報の URI がわかっていること。
事後条件	指定した RDF スキーマ情報が返却されること
イベントフロー	1. 情報サービス開発事業者は、確認したい RDF スキーマファイルの URI を入力する。 2. システムは、入力された URI に基づき、対象の RDF スキーマファイルを返却する。
備考	

表 2-14 機能定義(F12)

項目	内容
関連業務名	B03:データ参照・検索 C01:情報サービス利用によるオープンデータ参照
機能 ID	F12
機能名	データ値返却（SPARQL ベース）
オンライン/バッチ	オンライン
利用者	情報サービス開発事業者
目的	登録データについて、SPARQL ベースの要求に従い、データ値の閲覧・検索ができるよう公開し、サービスにデータ値を活用してもらう。
事前条件	データ値取得要求は、SPARQL ベースの API に従うこと。
事後条件	指定されたデータ値を返却すること
イベントフロー	1. 情報サービス開発事業者（または情報サービス）は、取得したいデータ値を SPARQL ベースの API にて要求する。 2. システムは、要求に基づき、対象のデータ値を返却する。
備考	

表 2-15 機能定義(F13)

項目	内容
関連業務名	B03:データ参照・検索 C01:情報サービス利用によるオープンデータ参照
機能 ID	F13
機能名	データ値返却（REST ベース）
オンライン/バッチ	オンライン
利用者	情報サービス開発事業者
目的	登録データについて、REST ベースの要求に従い、データ値の閲覧・検索ができるよう公開し、サービスにデータ値を活用してもらう。
事前条件	データ値取得要求は、REST ベースの API に従うこと。
事後条件	指定されたデータ値を返却すること。
イベントフロー	1. 情報サービス開発事業者（またはサービス）は、取得したいデータ値を REST ベースの API にて要求する。 2. システムは、要求に基づき、対象のデータ値を返却する。
備考	

表 2-16 機能定義(F14)

項目	内容
関連業務名	-
機能 ID	F14
機能名	登録 RDF データ検索
オンライン/バッチ	オンライン
利用者	自治体職員
目的	登録状況を確認したい RDF データを検索する。
事前条件	
事後条件	検索対象となった RDF データが一覧で表示されること。
イベントフロー	1. 自治体職員は、登録状況を確認したい RDF データの述語[2]や目的語[3]を検索ワードとして入力する。 2. システムは、入力された述語または目的語を保持したデータを検索する。
備考	

[2] RDF はデータを主語、述語、目的語のトリプル構造で保持する。述語はプロパティとも言う。
[3] RDF はデータを主語、述語、目的語のトリプル構造で保持する。目的語は述語(プロパティ)の値となる。

表 2-17 機能定義(F15)

項目	内容
関連業務名	-
機能 ID	F15
機能名	RDF データ再変換
オンライン/バッチ	オンライン
利用者	自治体職員
目的	登録済のデータに対して、RDF スキーマ情報ファイルに変更が発生した場合などに、基盤の RDF データにも変更を反映させる。
事前条件	更新対象データは、基盤に登録済であること。
事後条件	対象 RDF データが更新されていること
イベントフロー	1. 自治体職員は、更新対象となるデータを選択し、再変換ボタンを押す。 2. システムは、登録済みの更新対象データファイルを読み込む。 以降、更新内容に対して F02:データ形式変換⇒F03:バリデートが実行される。 3. データベースに格納された RDF データを更新する。 4. 更新完了の旨のメッセージを利用者に対して出力する。更新失敗した場合も、その旨と理由をメッセージとして利用者に対して出力する。
備考	

2.2.3. 非機能要件

自治体行政情報流通連携基盤システムの非機能要件については、導入する自治体ごとに要件が変化することから、IPA（独立行政法人 情報処理推進機構）が公開している「非機能要求グレード」[4]等を参考に定義することが望ましい。ここでは、「情報流通連携基盤の自治体行政情報における実証に係る請負」の実証時において定義した非機能要求項目（「非機能要求グレード」に従ったもの）と、項目に対する実施内容について述べる。

表 2-18 実証時に定義した非機能要件と定義・実施内容

項番	大分類	中分類	小分類	実証時の定義・実施内容
1	可用性	継続性	運用 スケジュール	公開サーバについて、24 時間無停止で情報提供可能とする。
2			業務継続性	システムの稼働率を月間 99.95%以上とする。
3	性能・拡張性	性能 目標値	オンライン レスポンス （公開サーバ）	1 分間に 100 リクエストが発生した場合に 5 秒以内で処理できること。
4			オンライン レスポンス （管理サーバ）	データ変換・登録ツールを 3 台の端末から同時に操作した場合に、8 秒以内で処理応答が返却されること。
5		リソース 拡張性	公開サーバ	情報サービス開発事業者や地元住民のアクセスが増大した場合には、サーバの増設により負荷分散を図る対応が可能なこと。
6			管理サーバ	取り扱うデータ数や更新頻度が上がる場合には、CPU やメモリ、HDD の拡張性を有するクラウドサービスを利用する。
7	運用・ 保守性	通常運用	バックアップ	コンテスト開始時にデータのフルバックアップを取得し、本実証期間内保管すること。
8	セキュリティ	アクセス・ 利用制限	認証機能	OS へのログイン時、およびデータ変換・登録ツール利用時に認証する。
9		データの秘匿	データ暗号化	データ変換・登録ツールのログイン情報の伝送データを暗号化する。
10		不正追跡・監視	不正監視	Apache の Access ログを取得し、確認を実施。
11		マルウェア対策	マルウェア対策	ウィルス対策ソフトウェアを採用。

以降で、各項目の設定内容について述べる。

[4] http://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html

（1）　可用性要件

1）継続性―運用スケジュール

自治体行政情報流通連携基盤システムを運用するにあたり、年間、月間、日単位でのシステム運用スケジュールを設定する必要がある。

実証時は、コンテストにおけるデータ提供を目的としており、募集期間が短いこともあり、公開サーバについては、24 時間無停止で情報提供可能なことを要件とした。

2）継続性―業務継続性

自治体行政情報流通連携基盤システムは、2.1.1 で述べたように、災害発生時等の業務継続性を確保することが重要である。

実証時は、クラウドサービスでの設備冗長化により、システムの稼働率を月間 99.95%以上保証することで対応した。

（2）　性能・拡張性要件

1）性能目標値

自治体行政情報流通連携基盤システムにアクセスしたプログラムに保証するレスポンスタイムを設計し、サーバスペックや台数を設定すべきである。

実証時は、情報サービス開発事業者や地元住民へデータを提供することになる公開サーバ（3.1.4 参照）の目標値は、過去のコンテストでのアクセス実績を元にして設定した。具体的には、１分間に 100 リクエストが発生した場合に 5 秒以内で処理できることを目標とした。

また、自治体職員によるオープンデータ化業務を扱う管理サーバ（3.1.4 参照）については、数名の自治体職員によりデータ変換・登録ツールの操作性を検証する、という実証内容から目標値を設定した。具体的には、3 台の操作端末からの同時操作に対して、8 秒以内で処理応答が返却されることを目標とした。

2）リソース拡張性

自治体行政情報流通連携基盤システムを導入する場合には、システムを利用した情報サービスが増えることによるアクセス数増への対策を考慮しておくべきである。また、取り扱うデータ数と更新頻度が上がる場合も考慮すべきである。公開サーバ、管理サーバそれぞれについての考え方を示す。

公開サーバへのアクセスが増大する点については、公開サーバのリソース拡張が必要となる。設定により公開サーバを複数台指定することが可能であり、サーバの増設での対応となる。実証時は、クラウドサービスの自動拡張機能を利用し、アクセス状況により自動拡張するように対応した。

また、取り扱うデータおよびその更新頻度が上がる場合には、管理サーバのリソース拡張を検討すべきである。管理サーバについては、サーバの処理能力を拡張できるようなハードウェアを選定することが望ましい。実証時は、サーバスペックを選択することが可能なクラウドサービスを利用した。

（3）　運用・保守性要件

1）通常運用―バックアップ要件

システム障害等によりデータが消失しないように定期的なバックアップを取得する必要がある。実証時は静的情報が中心であったため、以下のように設定した。

表 2-19 バックアップで考慮すべき要件と実証時の設定内容

要件	説明	実証時の設定内容
バックアップ取得間隔	バックアップをどういった周期で実施するのかを検討する。	コンテスト開始時にデータのフルバックアップを取得
バックアップ保存期間	取得したバックアップをどの期間保持するのかを検討する。	本実証期間内

（4）　セキュリティ要件

1）アクセス・利用制限

システムへの不正アクセスによるサービス停止や、第三者によるデータ変換・登録ツールへのアクセスとデータ改ざんを防ぐために、ログイン認証機能を整備した。

表 2-20 システムへの認証に関する実証時の設定内容

箇所	実証時の設定内容
システムへのアクセス	OS へのログイン認証を設ける。
データ変換・登録ツール	ツール利用時にログイン認証画面を設ける。

2）データの秘匿

自治体職員がデータ変更・登録ツールを利用する際に、クラウド上のサーバへのアクセス回線上をログイン認証情報が流れることから、ログイン情報の伝送データを暗号化する対策を実施した。

3）不正追跡・監視

システムに対して、不正なアクセスが試みられていないかが確認できるような対策を考慮すべきである。実証時は、Apache の Access ログを取得し、確認を実施した。

4）不正追跡・監視

ウィルス対策ソフトを導入し、リアルタイムスキャン、フルスキャンを実施するようにしておくことが望ましい。実証時は ClamAV というオープンソースのソフトウェアを採用した。

3.　自治体行政情報流通連携基盤システム詳細

3.1.　ソフトウェア設計

本節では、前述した機能を実現するために、自治体行政情報流通連携基盤システムのソフトウェア構成等の設計内容について述べる。

3.1.1. ソフトウェア一覧

前述した機能を実現するために必要なソフトウェアを、以下に一覧で示す。「関連機能 ID」は、ソフトウェアが実現する機能の ID（2.2.1 参照）を示す。

表 3-1 ソフトウェア一覧

項番	ソフトウェア	製品・言語	関連機能 ID	主な役割	備考
1	OS-64bit	Linux (CentOS)	-	システム全体の動作をサポート	OSS
2	HTTP サーバ	Apache	F04,F08,F09, F14,F06,F01, F11,F12,F13, F07,F10,F05, F15	HTTP 通信の実現	OSS
3	Sparql Endpoint	Apache jena fuseki	F04,F08,F09, F14,F12,F13, F05	SPARQL ベースの API での応答・データ返却	OSS
4	REST Endpoint	Java	F13	REST ベースの API での応答・データ返却	実装が必要
5	データベース (RDF Store)	Apache jena TDB	F04,F08,F09, F14,F12,F13, F05,F15	オープンデータを RDF として管理	OSS
6	データ変換・登録ツール (Web 画面)	PHP	F04,F08,F09, F14,F06,F01, F07,F10,F05, F15	自治体職員がファイルを登録するための Web 画面	実装が必要
7	データ変換・登録ツール (JSON ライブラリ)	Gson	F02,F03	自治体職員が登録するファイルが SDF (CSV と JSON)の場合の JSON ファイルを読取	実装が必要
8	データ変換・登録ツール (変換・登録処理)	Java	F02,F03	自治体職員が登録するファイルの RDF への変換、データベースへの登録	実装が必要

3.1.2. ソフトウェアの選定理由

前述した主なソフトウェアの選定理由を以下に示す。

（1）　OS-64bit (Linux)

自治体行政情報流通連携基盤システムでは、様々な自治体への普及展開のため、異なるクラウド環境上で動作することが要求される。また、Sparql Endpoint を構築するソフトウェアも Linux 上で動作するものが多いため、各クラウド提供事業者で一般的に提供されている Linux を使用することとする。

なお、クラウド環境上で動作する一般的なサーバ用途の商用 OS として Windows Server も存在するが、クラウド提供事業者によっては、Windows Server の仮想マシンが提供されない場合があるため、Linux を選定した。

また、自治体行政情報流通連携基盤システムでは Sparql Endpoint の機能を持つソフトウェアを使用するが、Sparql Endpoint を使用するソフトウェアはメモリを多量に使用するため、32bit OS で存在する OS/アプリケーションのメモリ使用量の制限を鑑みると、64bitOS が事実上必須であるものが多い。そのため、OS については 64bit のものを使用する。

（2）　HTTP サーバ（Apache）

自治体行政情報流通連携基盤システムでは、HTTP にて情報を公開するため、HTTP サーバ機能が必要である。なお、後述（表 3-2）の Sparql Endpoint の機能を持つソフトウェアにも HTTP サーバとしての機能を持つものもあるが、暗号化通信や認証、詳細な設定等の機能が劣るため、自治体行政情報流通連携基盤システムでは別途 HTTP サーバの機能を持つソフトウェアを用意する。

自治体行政情報流通連携基盤システムでは OS に Linux を採用するが、Linux 上で稼働する HTTP サーバとしては Apache が大きなシェアを持ち、事実上の標準である。

また、Linux 上で稼働する HTTP サーバに商用の製品も存在するが、殆どが OSS ベースの製品にサポートを付与したものであるため、オープンソースの Apache と比較して、性能面において商用製品の優位性は特段ないと思われる。

また、HTTP サーバとして標準的な機能を満たせば自治体行政情報流通連携基盤システムでの機能が実現可能であり、特にベンダーからのサポートの必要性がない状況ということもあり、HTTP サーバには Apache を採用することとする。

（3）　Sparql Endpoint (Apache jena fuseki)

自治体行政情報流通連携基盤システムでは、外部仕様書に従い、自治体行政情報標準 API として SPARQL-based command を提供する。また自治体行政情報流通連携基盤システムには、SPARQL-based command を提供するにあたり、以下の要求仕様が存在する。

A)　SPARQL 1.1 に準拠していること。
B)　環境やソフトウェア、ライブラリへの依存度が低いこと。
C)　クラスタ構成（複数台のサーバをまとめて運用する）を取らないこと。

それぞれの理由を以下に示す。

A)　SPARQL 1.1 に準拠していること。

SPARQL-based command は、外部仕様書に定義されているように、SPARQL 1.1 をベースにデザインされている。よって、ソフトウェア側も SPARQL 1.1 に準拠していることが求められる。（自治体行政情報流通連携基盤システムで実装する SPARQL –based command については表 3-9 参照）

B)　環境やソフトウェア、ライブラリへの依存度が低いこと。

自治体への普及展開を考慮し、環境やソフトウェア、ライブラリへの依存度が低いことが要求される自治体行政情報流通連携基盤システムでは、異なったクラウド環境上で共通のソフトウェアで基盤を稼働させることを求められるため。（可能であれば仮想マシン上で稼働し、環境の差異は JVM のような仮想マシン側で吸収するような構成であればさらに望ましい。）

C)　クラスタ構成（複数台のサーバをまとめて運用する）を取らないこと。

クラウド環境上でクラスタ構成（複数台のサーバをまとめて運用する）を取ると、サーバ台数の増減の単位がクラスタ構成で必要なサーバ台数になってしまい、クラウドのメリット（リソースの増減が容易）が享受しにくくなる。また、クラスタ環境の仮想サーバはある程度の確率でサーバがダウンする事を前提とした環境になっており、クラウド上でクラスタ構成を組んだ場合、稼働率が低下することも考えられるためである。

また自治体行政情報流通連携基盤システムは、日本語を問題なく取り扱える必要がある。

上記の要求事項を踏まえ、

- SPARQL 1.1 準拠
- JavaVM 上で稼働する
- 日本語のデータでの動作実績があるもの。

を満たすソフトウェアが望ましいと考える。
上記を満たすソフトウェアとして、「Apache jena fuseki」、「Sesame」、「Stardog」が存在する。
ここで、各自治体への普及展開を考慮し、

- 参考資料としての解説ドキュメントや実績が豊富であること。
- ライセンス形態がオープン化されていること。

の 2 点を充足するソフトウェアが最適と考える。
以下に「ライセンスが OSS か」「SPARQL Endpoint に関する情報量」の観点で比較した結果を示す。

表 3-2 Sparql Endpoint を構成するソフトウェア比較

項番	ソフトウェア名	URL	ライセンス形態	情報量 (Sparql Endpoint)
1	Apache jena fuseki	http://jena.apache.org/index.html	OSS	多い
2	Sesame	http://www.openrdf.org/	OSS	少ない
3	Stardog	http://stardog.com/	商用	少ない

Sparql Endpoint のドキュメントに関しては、Apache jena fuseki が他の２つと比較して情報が多い。また、自治体行政情報流通連携基盤システムはクラウド上で稼働させるが、商用ライセンスの場合、ライセンス形態によってはクラウド上での稼働を想定していない場合があり、クラウド上での利用に関して煩雑な手続きが発生する可能性がある。

以上より、ここでは Apache jena fuseki を採用することとした。

なお、自治体行政情報流通連携基盤システムにおいて、Apache jena fuseki で稼働している Sparql Endpoint を Sesame 等別のソフトウェアに変更する場合、以下の変更作業で対応可能である。

- 公開サーバ（「3.1.4 機器構成」にて後述)側の B02:公開サーバデータ更新(「3.3 データ変換・登録ツール設計」にて後述)で、fuseki 専用の更新スクリプト(s-put)を呼び出している部分を、Sesame の更新スクリプトに変更する。
- 公開サーバの HTTP Server から Sparql Endpoint に Proxy 経由でリクエストを通過させている部分の URL 構成を、Sesame の Sparql Endpoint に変更する。

（4）　データベース(Apache jena TDB)

自治体行政情報流通連携基盤システムが扱うデータ形式である RDF を格納するデータベース(RDF Store)が必要になる。

Apache Jena Fuseki は、TDB(Triple Data base)という標準の RDF Store を保持している。

Apache Jena Fuseki は RDB を使用することも可能だが、TDB はファイル/メモリベースの RDF Store であり、また単独マシン上で動作することから、メモリキャッシュが有効に活用され、単独マシン上で高速に動作させることが可能である。

以上より、ここでは Apache jena TDB を採用することとした。

3.1.3. ソフトウェアライセンス

表 3-1 で述べたソフトウェアについて、製品・言語に対するライセンスを以下に示す。

表 3-3 ソフトウェアライセンス

項番	ソフトウェア	製品・言語	関連 URL	ライセンス
1	OS-64bit	Linux (CentOS)	http://www.centos.org/	GPL
2	HTTP サーバ	Apache	http://httpd.apache.org/	APL2
3	Sparql Endpoint	Apache jena fuseki	http://jena.apache.org/index.html	APL2
4	REST Endpoint	Java	http://www.oracle.com/technetwork/jp/java/javase/overview/index.html	BCL
5	データベース	Apache jena TDB	http://jena.apache.org/index.html	APL2
6	データ変換・登録ツール（Web 画面）	PHP	http://www.php.net/	PHP License
7	データ変換・登録ツール (JSON ライブラリ)	Gson	https://code.google.com/p/google-gson/	APL2
8	データ変換・登録ツール（変換・登録処理）	Java	http://www.oracle.com/technetwork/jp/java/javase/overview/index.html	BCL

以下に各ライセンスの要点について示す。

表 3-4 ソフトウェアライセンスの要点

項番	ライセンス名	特徴
1	GPL	http://www.gnu.org/licenses/gpl.html ・ソフトウェアは必ずソースプログラムとともに頒布、複製する必要がある。もしソースプログラムを付けずに配布する場合は、ソースプログラムを確実に入手できる手段を提供することが義務付けられる。 ・ソフトウェアを、使用、複製、変更、頒布、新しいフリーソフトウェアの一部として利用できることを求められる。 ・変更、改良されたソフトウェアは GPL に従って頒布されることを求められる。 ・プログラムの全部あるいは一部を用いて作られたソフトウェアは GPL に従って頒布されることを求められる。 ・非 GPL なモジュールとのリンクはできない。
2	APL2	http://www.apache.org/licenses/LICENSE-2.0 ・ユーザのソフトウェア使用/頒布/修正、派生版の頒布をすることを制限しない。 ・変更、改良されたソフトウェアが同じライセンスで提供される、またはフリー/オープンソースソフトウェアとして頒布されることを要求せず、ユーザがそのソフトウェアに Apache License のコードが使われていることを知らせる文言を入れることだけが求められる。 ・GPL と異なり、Apache License コードの変更、改良されたソフトウェアのユーザには、フリーなライセンスを適用しなくても良い。 ・ライセンスされたファイルそれぞれに元々ある著作権と特許権の記述はそのまま保持されなければならず、何らかの修正が施されている場合は、その旨を追加記述しなければならないことが求められる。
3	PHP License	http://www.php.net/license/ PHP 本体のソースコードのライセンス条項。 APL2 をベースに、以下の条項が追加されたもの。 再配布にあたって ・PHP ライセンスの著作権が明記されること。 ・「PHP」という語句をいかなる二次的著作物の名称に含めないこと。 ・ソースコードが再配布されるいかなる方法においても以下の表示を含むこと。 This product includes PHP, freely available from http://www.php.net/
4	BCL	http://www.oracle.com/technetwork/java/javase/terms/license/index.html JDK の再配布に関するライセンス条項。 汎用コンピューティングに伴う再配布は可能だが、組み込みデバイス等に使用する場合はライセンス料が発生する場合がある。

自治体行政情報流通連携基盤システムで使用するソフトウェアについて望まれることは、

- 容易に入手可能である。(再配布不可でも配布元が分かれば問題ない)
- 無償で利用可能である。

という点であり、使用するソフトウェアのライセンスは上記の条件を満たしている。

また自治体行政情報流通連携基盤システムにおいては、ソフトウェア内部の変更は行っていないことと、GPL ライセンスのソフトウェアをリンクするモジュールは作成しないため、自治体行政情報流通連携基盤システムの成果物の流通にあたり、使用しているソフトウェアの再配布の義務はないといえる。

3.1.4. 機器構成及びソフトウェア構成

自治体行政情報流通連携基盤システムを構築する際の機器構成と、各機器へのソフトウェア構成を以下に示す。

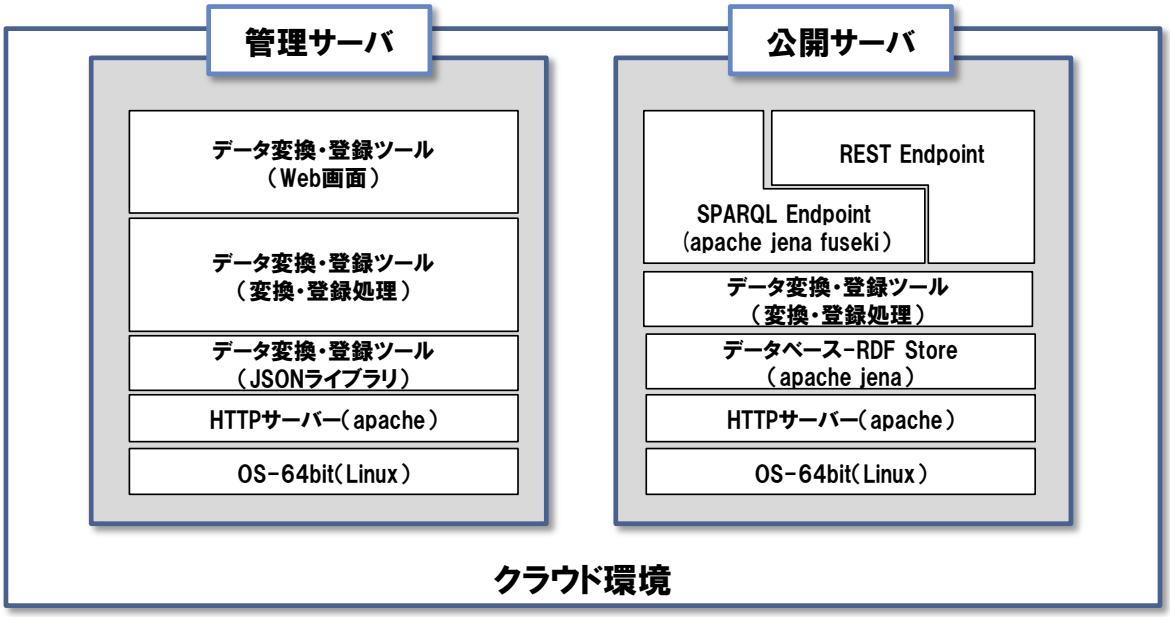


図 3-1　自治体行政情報流通連携基盤システム ソフトウェア配置図

また参考として、「情報流通連携基盤の自治体行政情報における実証に係る請負」の実証実験時に整備した環境のスペック情報と、環境への負荷試験結果を示す。

表 3-5 実証環境 スペック情報

項番	サーバ名	説明	スペック	台数
1	管理サーバ	情報サービス開発事業者が API を利用してデータの閲覧、検索を行うためのサーバ。	CPU：1Core-2GB メモリ：2GB HDD：20GB	1
2	公開サーバ	自治体職員が、データの管理（登録・更新・削除）を行うためのサーバ。	CPU：1Core-2GB メモリ：2GB HDD：20GB	1

上記環境に対し、同時アクセス 1 分間最大 100 アクセスが発生した場合にレスポンスを確保することを目標に、負荷試験ツールにより計測した。

その結果、全アクセスについて、平均 1 秒程度（遅くとも 5 秒程度）でレスポンスが返却されることを確認した。

3.1.5. ディレクトリ・ファイル構成

自治体行政情報流通連携基盤システム構築後のディレクトリ・ファイル構成を示す。
公開サーバの構築手順は 4.1.1、管理サーバの構築手順は 4.1.2 を参照のこと。

なお、ディレクトリ名、配置は OS のバージョン、ディストリビューター、ミドルウェアのパッケージの構成により異なる場合があり、ここでは「情報流通連携基盤の自治体行政情報における実証に係る請負」の実証時に構成された例である。また、表中の「オーナ」については、「3.4.3 ユーザ定義」を参照のこと。

表 3-6　公開サーバ　ディレクトリ・ファイル構成

項番	ソフトウェア	ディレクトリ/ファイル	オーナ	chmod	用途	備考
1	HTTP サーバ (Apache)	/var/log/httpd	Root	777	Apache ログディレクトリ	
2		/var/www/html/	Root	777	Apache 静的コンテンツ格納ディレクトリ	Apache のドキュメントルート
3		/etc/httpd/conf.d/	Root	755	Apache 個別設定ファイルディレクトリ	
4		/etc/httpd/conf.d/fuseki.conf	loduser	644	Apache 個別設定ファイル（公開サーバ設定）	
5	データベース -RDF Store (Apache Jena Fuseki)	/usr/lib/fuseki	loduser	755	fuseki インストールディレクトリ	
6		/usr/lib/fuseki/DB2	loduser	775	データベース(TDB)データディレクトリ	
7		/var/log/fuseki/	Root	777	fuseki ログディレクトリ	
8	データ変換・登録ツール	/home/loduser/jrrktool/	loduser	777	公開サーバデータ更新インストールディレクトリ	
9		/var/log/jrrk/	Root	777	公開サーバデータ更新ログディレクトリ	
10		/var/www/html/rdf/	loduser	777	RDF 配信ディレクトリ	
11		/var/www/html/schema/	loduser	777	RDF スキーマ情報ファイル配信ディレクトリ	

表 3-7　管理サーバ　ディレクトリ・ファイル構成（1/2）

項番	ソフトウェア	ディレクトリ/ファイル	オーナ	chmod	用途	備考
1	HTTP サーバ (Apache)	/var/log/httpd	root	777	Apache ログディレクトリ	
2		/etc/httpd/conf.d/	root	755	Apache 個別設定ファイルディレクトリ	
3		/etc/httpd/conf.d/manage.conf	admin user	644	Apache 個別設定ファイル(管理サーバ設定)	
4		/etc/httpd/conf.d/php.conf	admin user	644	Apache 個別設定ファイル（管理サーバ PHP 設定）	
5		/var/www/html/	root	777	Apache 静的コンテンツ格納ディレクトリ	Apache のドキュメントルート
6	データ変換・登録ツール	/home/adminuser/jrrktool/	admin user	777	データ変換・登録ツール(変換/バリデート/登録更新、オンライン処理スクリプト、バッチ処理スクリプト、管理サーバ更新データ一覧) インストールディレクトリ	
7		/home/adminuser/jrrktool/jar/	admin user	755	データ変換・登録ツール jar ファイルディレクトリ	
8		/var/log/jrrk/	root	777	データ変換・登録ツールログディレクトリ	
9		/var/www/html/rdf/	admin user	777	公開サーバ向け RDF 配信ディレクトリ	
10		/var/www/html/schema/	admin user	777	公開サーバ向け RDF スキーマ情報ファイル配信ディレクトリ	
11		/var/www/html/lodmanage/	admin user	777	データ変換・登録ツール UI, Upload インストール先	Apache で、このディレクトリ以下に Basic 認証をかける。
12		/var/www/html/lodmanage/style/	admin user	755	スタイルシート、エラーファイル表示用 xslt ディレクトリ	
13		/var/www/html/lodmanage/commonschema	admin user	777	共通 RDF スキーマディレクトリ	
14		/var/www/html/lodmanage/failed	admin user	777	データ変換・登録ツール変換・バリデート結果エラーファイルディレクトリ	
15		/var/www/html/lodmanage/ignorelist	admin user	777	データ登録・登録ツールバリデート非対象ボキャブラリ URI リストファイルディレクトリ	
16		/var/www/html/lodmanage/inc	admin user	777	データ変換・登録ツール UI 共通ヘッダ、プッダファイルディレクトリ	

表 3-8 管理サーバ ディレクトリ・ファイル構成（2/2）

項番	ソフトウェア	ディレクトリ/ファイル	オーナ	chmod	用途	備考
17	データ変換・登録ツール	/var/www/html/lodmanage/syncupload	admin user	777	Upload によるオンライン処理のデータファイルのアップロード先ディレクトリ	
18		/var/www/html/lodmanage/tempout	admin user	777	データ変換終了・バリデート前のファイルのテンポラリ出力先	
19		/var/www/html/lodmanage/upload	admin user	777	Upload によるバッチ処理のデータファイルのアップロード先ディレクトリ	
20		/var/www/html/listrdf.php	admin user	777	管理サーバ更新データ一覧：RDF の一覧表示	
21		/var/www/html/listschema.php	admin user	777	管理サーバ更新データ一覧：RDF スキーマファイルの一覧表示	

3.2. 自治体行政情報標準 API 設計

本節では、情報サービス開発事業者が自治体行政情報流通連携基盤システムに登録されたオープンデータを検索・参照するために実装した API（自治体行政情報標準 API）の設計内容について述べる。

3.2.1. ソフトウェア一覧

自治体行政情報標準 API を提供するソフトウェアを以下に示す。

表 3-9 自治体行政情報標準 API ソフトウェア一覧

項番	設置サーバ	名称	機能	実装
1	公開	SPARQL API	外部仕様書に定義された標準 API である「Sparql-based command」の一部コマンドを利用者に対して提供する。	Apache Jena Fuseki を使用。
2	公開	REST API	外部仕様書に定義された標準 API である「Triple Management Command」「Geographic Management Command」の一部コマンドを利用者に対して提供する。	別冊「自治体行政情報流通連携基盤システム REST API 仕様書」を参照

REST API は実証時に鯖江市の環境のみで実装したものであり、別冊「自治体行政情報流通連携基盤システム REST API 仕様書」として仕様を定義している。

よって以降では、1 の SPARQL API に関する仕様についてのみ記載する。

3.2.2. ソフトウェア設計

自治体行政情報標準 API としての SPARQL API と、外部仕様書の「3.1 SPARQL -based command 」との差異は以下の通りである。

- 外部からのデータ登録の流れは、データの整合性、個人情報の保護、匿名化等の要件などから、自治体職員によるデータ内容のチェックの後、自治体職員が登録ツールにて行うことを想定している。そのため、登録・更新に関連する SPARQL-based command(外部仕様書 3.1.4 RDF グラフの追加, 3.1.5 RDF グラフの更新, 3.1.6 RDF グラフの削除, 3.1.8 Triple の追加, 3.1.9 Triple の更新、3.1.10 Triple の削除)の提供は行わない。

- 公開されるデータはオープンデータであり、誰でも閲覧可能であることから、アクセス制限や認証の機能は設けない。

- 外部仕様書の「3.1 SPARQL -based command」に定義されている 3.1.3 RDF グラフの閲覧、3.1.7 Triple の閲覧 に関しては、3.1.1 SPARQL1.1 準拠のクエリ発行（GET メソッド）にて代替可能なため、自治体行政情報流通連携基盤システムでは実装しないが、代替方法について本章で記載する。

- 自治体行政情報流通連携基盤システムはデータの削除、更新をデータファイルの単位を基準として行うため、ファイルごとにグラフを作成する。

3.2.3. オンライン処理設計

（1）　オンライン処理方式概要

自治体行政情報標準 API のオンライン処理は、Apache Jena Fuseki の実装を用いる。

なお、自治体行政情報流通連携基盤システムでは、ファイルごとにグラフを作成し、デフォルトグラフが存在しない仕様となっている。

（2）　オンライン処理一覧

自治体行政情報標準 API のオンライン処理について以下に一覧を示す。

表 3-10　自治体行政情報標準 API オンライン処理一覧

項番	名称	URL パス	HTTP method	内容	外部仕様書参照先
1	SPARQL GET Command	/api/v1/sparql	GET	SPARQL 1.1 準拠のクエリを発行する	3.1.1 SPARQL1.1 準拠のクエリ発行（GET メソッド）
2	SPARQL POST Command	/api/v1/sparql	POST	SPARQL 1.1 準拠のクエリを発行する	3.1.2 SPARQL1.1 準拠のクエリ発行（POST メソッド）

（3）　オンライン処理定義

以降で、各オンライン処理の定義を示す。

表 3-11 SPARQL GET Command

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>機能名</td><td>SPARQL GET</td></tr>
<tr><td>機能</td><td>HTTP GET メソッドを利用して，SPARQL1.1 準拠のクエリを発行する．</td></tr>
<tr><td>メソッド</td><td>GET</td></tr>
<tr><td>URL パス</td><td>/api/v1/sparql</td></tr>
<tr><td>パラメタ</td><td>
<table>
<tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr>
<tr><td>1</td><td>query</td><td>URL エンコードした SPARQL クエリ</td></tr>
</table>
</td></tr>
<tr><td>リクエストヘッダ</td><td>要求するレスポンスの形式を，リクエストヘッダの Accept で設定する。
SELECT オペレーションを発行する場合に指定できるパラメタを以下に示す。
<table>
<tr><th>項番</th><th>Accept ヘッダ値</th><th>説明</th></tr>
<tr><td>1</td><td>application/sparql-results+xml</td><td>SPARQL Query Results XML Format に基づくレスポンス</td></tr>
<tr><td>2</td><td>application/sparql-results+json</td><td>SPARQL Query Results JSON Format に基づくレスポンス</td></tr>
</table>
CONSRUCT または DESCRIBE オペレーションを発行する場合に指定できるパラメタを以下に示す。
<table>
<tr><th>項番</th><th>Accept ヘッダ値</th><th>説明</th></tr>
<tr><td>1</td><td>application/rdf+xml</td><td>RDF/XML</td></tr>
<tr><td>2</td><td>text/plain</td><td>N-Triples</td></tr>
<tr><td>3</td><td>text/turtle</td><td>Turtle</td></tr>
</table>
ASK オペレーションを発行する場合に指定できるパラメタを以下に示す。
<table>
<tr><th>項番</th><th>Accept ヘッダ値</th><th>説明</th></tr>
<tr><td>1</td><td>application/sparql-results+xml</td><td>SPARQL Query Results XML Format に基づくレスポンス</td></tr>
<tr><td>2</td><td>text/plain</td><td>テキスト表現(yes/no)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>レスポンス</td><td>SELECT オペレーションのレスポンスは，accept ヘッダ値に基づき，以下のいずれかである．
・SPARQL Query Results JSON Format に基づくレスポンス
・SPARQL Query Results XML Format に基づくレスポンス

CONSTRUCT，DESCRIBE オペレーションのレスポンスは，RDF グラフデータである．フォーマットは，accept ヘッダ値で指定した通りである．

ASK オペレーションのレスポンスは，accept ヘッダ値に基づき，以下のいずれかである．
・SPARQL Query Results XML Format に基づくレスポンス
・yes または no の文字列</td></tr>
<tr><td>ステータスコード</td><td><table>
<tr><th>項番</th><th>内容</th><th>レスポンスコード</th><th>備考</th></tr>
<tr><td>1</td><td>正常終了</td><td>200</td><td></td></tr>
<tr><td>2</td><td>クエリ、パラメタ不正</td><td>400</td><td></td></tr>
<tr><td>3</td><td>システム内エラー</td><td>500</td><td></td></tr>
<tr><td>4</td><td>プロクシエラー</td><td>502</td><td>SPARQL API の処理のタイムアウト時に発生する</td></tr>
</table></td></tr>
</table>

利用例	システムに投入されたドキュメントのタイトルと作成者の組を取得する SPARQL クエリを発行するリクエストとリクエストヘッダ、レスポンスを示す。なお、GET メソッドの query パラメタ値は、下記 SPARQL クエリを URL エンコードしたものである。
	
	PREFIX dc: <http://purl.org/dc/elements/1.1/>
	SELECT　?title ?name
	WHERE {
	GRAPH ?g
	{
	?x dc:title ?title .
	?x dc:creator ?name.
	}
	}
	
	リクエスト:
	GET
	/data/sparql?query=PREFIX+dc%3A+%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Felements%2F1.1%2F%3E%0D%0ASELECT++%3Ftitle+%3Fname%0D%0AWHERE+%7B%0D%0A+GRAPH+%3Fg%0D%0A%7B%0D%0A%3Fx+dc%3Atitle+%3Ftitle+.%0D%0A%3Fx+dc%3Acreator+%3Fname.%0D%0A%7D%0D%0A%7D%0D%0A
	Host: www.example.org
	Accept: application/sparql-results+xml
	
	レスポンス
	<?xml version="1.0"?>
	<sparql xmlns="http://www.w3.org/2005/sparql-results#">
	<head>
	<variable name="title"/>
	<variable name="name"/>
	</head>
	<results>
	<result>
	<binding name="title">データサンプル 1

	<literal> </binding> <binding name="name">testuser </binding> </result> </results> </sparql>

表 3-12 SPARQL POST Command

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>機能名</td><td>SPARQL POST</td></tr>
<tr><td>機能</td><td>HTTP POST メソッドを利用して，SPARQL1.1 準拠のクエリを発行する.</td></tr>
<tr><td>メソッド</td><td>POST</td></tr>
<tr><td>URL パス</td><td>/api/v1/sparql</td></tr>
<tr><td>パラメタ</td><td>以下に示すパラメタをメッセージボディ部に格納する。
<table>
<tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr>
<tr><td>1</td><td>query</td><td>URL エンコードした SPARQL クエリ</td></tr>
</table></td></tr>
<tr><td>リクエストヘッダ</td><td>要求するレスポンスの形式を，リクエストヘッダの Accept で設定する。GET 形式と同様である。</td></tr>
<tr><td>レスポンス</td><td>GET 形式と同じである。</td></tr>
<tr><td>ステータスコード</td><td>GET 形式と同じく、以下の通りである。
<table>
<tr><th>項番</th><th>内容</th><th>レスポンスコード</th><th>備考</th></tr>
<tr><td>1</td><td>正常終了</td><td>200</td><td></td></tr>
<tr><td>2</td><td>クエリ、パラメタ不正</td><td>400</td><td></td></tr>
<tr><td>3</td><td>システム内エラー</td><td>500</td><td></td></tr>
<tr><td>4</td><td>プロクシエラー</td><td>502</td><td>SPARQL API の処理のタイムアウト時に発生する</td></tr>
</table></td></tr>
<tr><td>利用例</td><td>GET とリクエスト送信方法とパラメタ格納方法が異なること以外は同様。</td></tr>
</table>

また、外部仕様書の 3.1.3 RDF グラフの閲覧、3.1.7 Triple の閲覧の SPARQL GET Command における代替のクエリを以下に示す。

表 3-13 RDF グラフの閲覧の代替クエリ

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>機能名</td><td>RDF グラフの閲覧の代替クエリ</td></tr>
<tr><td>機能</td><td>外部仕様書の 3.1.3 RDF グラフの閲覧の代替クエリ</td></tr>
<tr><td>代替機能</td><td>SPARQL GET</td></tr>
<tr><td>URL パス</td><td>/ap1/v1/sparql</td></tr>
<tr><td>元パラメタ</td><td><table><tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr><tr><td>1</td><td>graph</td><td>更新対象の RDF グラフを識別する URI</td></tr></table></td></tr>
<tr><td>パラメタ</td><td><table><tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr><tr><td>1</td><td>query</td><td>URL エンコードした SPARQL クエリ</td></tr></table>
代替クエリのパラメタの値は以下の SPARQL クエリを URL エンコードしたものになる。ただし、URI の前後は<>で囲う。

CONSTRUCT { ?s ?p ?o } WHERE { GRAPH <graph> { ?s ?p ?o } . }</td></tr>
<tr><td>リクエストヘッダ</td><td>要求するレスポンスの形式を，リクエストヘッダの Accept で設定する。

指定できるパラメタを以下に示す。<table><tr><th>項番</th><th>Accept ヘッダ値</th><th>説明</th></tr><tr><td>1</td><td>application/rdf+xml</td><td>RDF/XML</td></tr><tr><td>2</td><td>text/plain</td><td>N-Triples</td></tr><tr><td>3</td><td>text/turtle</td><td>Turtle</td></tr></table></td></tr>
</table>

表 3-14 Triple の閲覧の代替クエリ

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>機能名</td><td>Triple 閲覧の代替クエリ</td></tr>
<tr><td>機能</td><td>外部仕様書の 3.1.7 Triple の閲覧の代替クエリ</td></tr>
<tr><td>代替機能</td><td>SPARQL GET</td></tr>
<tr><td>URL パス</td><td>/api/v1/sparql</td></tr>
<tr><td>元パラメタ</td><td>
<table>
<tr><th>項番</th><th>パラメタ名</th><th>ディフォルト値</th><th>内容</th></tr>
<tr><td>1</td><td>graph</td><td>ディフォルトのグラフ</td><td>更新対象の RDF グラフを識別する URI</td></tr>
<tr><td>2</td><td>subject</td><td>指定なし</td><td>出力対象の Triple の主語の URI</td></tr>
<tr><td>3</td><td>predicate</td><td>指定なし</td><td>出力対象の Triple の述語の URI</td></tr>
<tr><td>4</td><td>object</td><td>指定なし</td><td>出力対象の Triple の目的語の URI またはリテラル</td></tr>
</table>
</td></tr>
<tr><td>代替パラメタ</td><td>
<table>
<tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr>
<tr><td>1</td><td>query</td><td>URL エンコードした SPARQL クエリ</td></tr>
</table>

代替クエリのパラメタの値は以下の SPARQL クエリを URL エンコードしたものになる。ただし、URI の前後は<>で囲う。

CONSTRUCT { < subject > < predicate > < object > } WHERE { GRAPH <graph> { < subject > < predicate > < object > } . }

また、subject, predicate, object が省略されている場合は、それぞれ、?s, ?p, ?o で<>を含めて置き換える。

object が省略されている場合は、
CONSTRUCT { < subject > < predicate > ?o } WHERE { GRAPH <graph> { < subject > < predicate > ?o } . }

となる。
</td></tr>
<tr><td>リクエストヘッダ</td><td>要求するレスポンスの形式を，リクエストヘッダの Accept で設定する。

指定できるパラメタを以下に示す。
<table>
<tr><th>項番</th><th>Accept ヘッダ値</th><th>説明</th></tr>
<tr><td>1</td><td>application/rdf+xml</td><td>RDF/XML</td></tr>
<tr><td>2</td><td>text/plain</td><td>N-Triples</td></tr>
<tr><td>3</td><td>text/turtle</td><td>Turtle</td></tr>
</table>
</td></tr>
</table>

3.2.4. メッセージ設計

自治体行政情報流通連携基盤システムでは、OSS(Apache Jena Fuseki)を利用しており、製品の定義に従うため、本書ではメッセージ出力は定義しない。

3.3. データ変換・登録ツール設計

本節では、自治体職員がオープンデータを登録するために実装した「データ変換・登録ツール」の設計内容を述べる。

3.3.1. コンポーネント一覧

データ変換・登録ツールを構成するコンポーネント一覧を以下に示す。ここでのコンポーネントとは、ある 1 つの役割を担うためのモジュールの集合体を意味する。「関連機能 ID」は、コンポーネントが実現する機能の機能 ID（「2.2.1 機能一覧」参照）を示す。

表 3-15 データ変換・登録ツール コンポーネント一覧

項番	設置サーバ	名称	機能	開発言語	関連機能 ID
1	管理	UI	利用者に対して Web 上の UI を提供する。	PHP	F04,F08,F09,F14, F06,F01,F07,F10, F05,F15
2	管理	Upload	WebUI、または外部システム（CMS 等）からのファイルのアップロード処理を行う。	PHP	F04,F08,F05
3	管理	変換/バリデート/登録更新	ファイルフォーマットの変換を行い、変換されたファイルのバリデートを行う。バリデート OK のファイルを、指定のフォルダに登録する。	Java	F02,F03
4	管理	オンライン処理スクリプト	オンラインでのファイル変換処理を行う。	bash	F04,F08,F05,F15
5	管理	バッチ処理スクリプト	バッチでのファイル変換処理を行う。	bash	F04,F08
6	管理	管理サーバ更新データ一覧	管理サーバ上の変換/バリデート済みのファイルと、更新情報を公開サーバに公開する。	bash	F04,F08,F09,F08, F07,F10,F05,F15
7	公開	公開サーバデータ更新	管理サーバから新規に登録・更新されたファイルを取得し、管理サーバ上のデータベース、ファイルを更新する。	bash	F04,F08,F09,F08 F07,F10,F05,F15

3.3.2. コンポーネント図

前述したデータ変換・登録ツールにおける各コンポーネントの関連図を以下に示す。

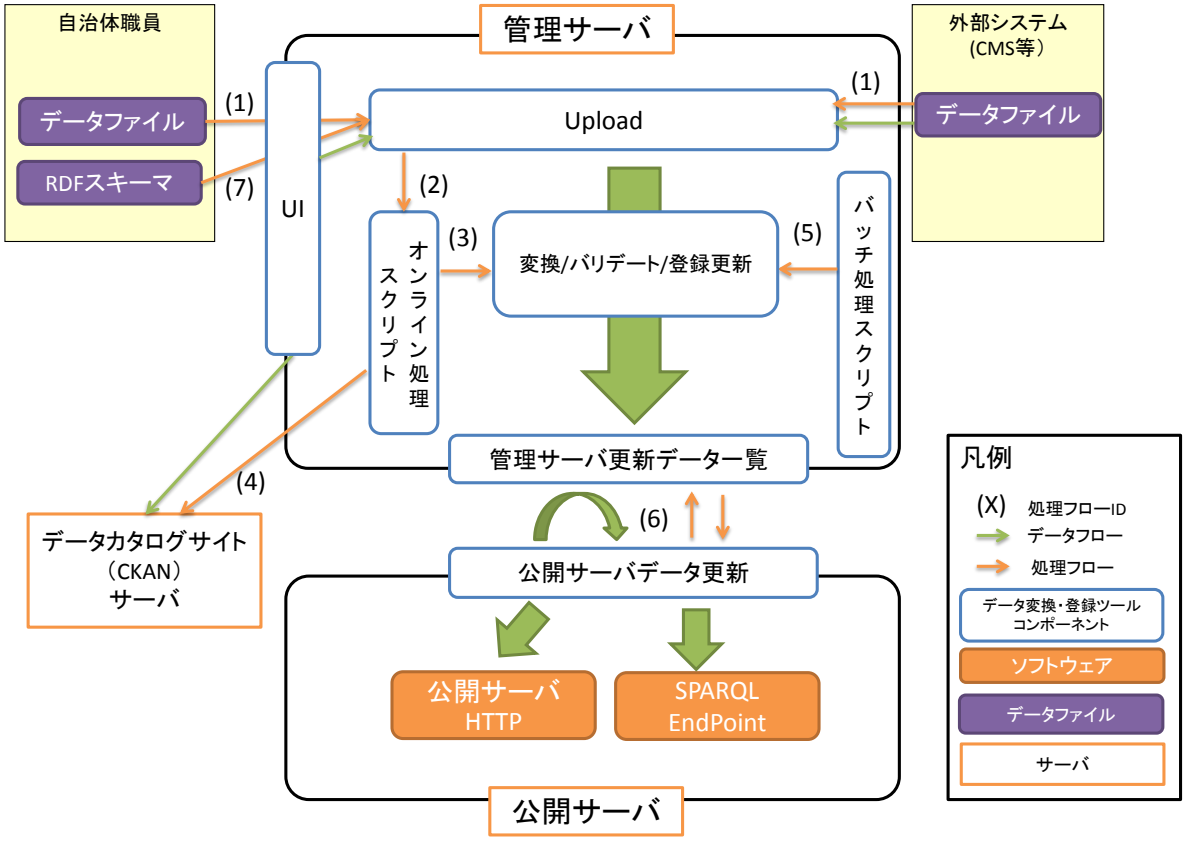


図 3-2　データ変換・登録ツール　コンポーネント図

データファイルを新規登録するケースを例に、コンポーネントを用いた処理フロー（図中(1)～(6)）を以降で説明する。

（なお、図中(5)および(6)はバッチ処理である。）

また、RDF スキーマを新規登録するケースを例に、コンポーネントを用いた処理フロー（図中(7)）を以降で説明する。

（1）　データ登録

登録するデータファイルをアップロードする。アップロード方法は

- Web の UI からアップロード
- 外部システム(CMS 等)からのアップロード

の 2 パターンとなる。

Upload コンポーネントによって、アップロードされたファイルが、以下所定の位置に格納される。

- オンライン処理の場合　　/var/www/html/lodmanage/syncupload
- バッチ処理の場合　　/var/www/html/lodmanage/upload

オンライン処理の場合は、(2)へ進む。バッチ処理の場合は終了する。

（2）　オンライン処理スクリプトの呼出

オンライン処理の場合、「Upload」から、「オンライン処理スクリプト」を呼び出し、(3)に進む。このとき、変換対象ファイルへのパスと、ファイルのバリデートの実施有無を「オンライン処理スクリプト」に渡す。

（3）　変換/バリデート/登録・更新の呼出

「オンライン処理スクリプト」から、「データ変換/バリデート/登録更新」が呼び出される。　「データ変換/バリデート/登録更新」は、(2)で「Upload」から渡された変換対象ファイルを変換し、バリデート実施有無で「有」となったファイルのバリデートを行う。

変換に成功した場合は「管理サーバ更新データ一覧」の対象ディレクトリに生成されたファイルを作成し、失敗した場合はエラーファイルを所定のディレクトリに作成する。

オプションの CKAN 連携を行う場合は(4)に進む。CKAN 連携を行わないときは終了する。

（4）　CKAN 連携

(3)の「変換/バリデート/登録更新」が成功した場合、ファイルに記入されたメタデータやファイルの URI 等の情報を CKAN に登録する。

（5）　バッチ処理スクリプト起動、データ変換/バリデート/登録・更新の呼出(バッチ処理)

管理サーバの cron にて「バッチ処理スクリプト」が定期的に呼び出され、「変換/バリデート/登録更新」を呼び出す。

「変換/バリデート/登録更新」は、スクリプト中に記載された変換対象ディレクトリと、生成されたファイルを格納するディレクトリ、エラーファイルを格納するディレクトリを走査し、変換すべきファイルがある場合は、ファイルの変換/バリデート/登録更新を行う。変換に成功した場合は、「管理サーバ更新データ一覧」に記載された対象ディレクトリに生成されたファイルを格納し、失敗した場合はエラーファイルを所定のディレクトリに格納する。

（6）　公開サーバデータ更新-データ取得(バッチ処理)

公開サーバの cron にて「公開サーバデータ更新スクリプト」が定期的に呼び出される。

「公開サーバデータ更新スクリプト」は、「管理サーバ更新データ一覧」を HTTP で呼び出す。

管理サーバ上では、上記の呼び出しに対して、所定のディレクトリ中でパラメタに指定された時刻以降に更新されたファイルの一覧を返す。

公開サーバでは、管理サーバ側が返したファイル一覧から、ファイルを公開サーバにダウンロードし、ダウンロードしたファイルでデータベース上のファイルを更新する。

（7）　RDF スキーマ登録

自治体職員が UI から登録する RDF スキーマのファイルをアップロードする。

「Upload」で、アップロードされた RDF スキーマファイルが所定のディレクトリ(/var/www/html/schema) にアップロードされ、終了する。

3.3.3. 画面設計

ここでは、データ変換・登録ツールにおいて、自治体職員が操作する UI の設計内容について述べる。

（1）　画面一覧

以下に画面一覧を示す。

表 3-16 データ変換･登録ツール 画面一覧

項番	画面 ID	名称	関連機能 ID	備考
1	SC_01	ログアウト機能画面	-	HTML は存在するが、表示画面自体は持たない。
2	SC_02	トップページ・データ一覧表示画面	F04,F14	
3	SC_03	データファイル一覧部	F08,F09,F06	SC_02,SC_16 の一部
4	SC_04	RDF スキーマ情報ファイル一覧画面	F07,F10	
5	SC_05	データファイル登録画面	F04	
6	SC_06	データファイル削除画面	F09	
7	SC_07	データファイル更新画面	F08	
8	SC_08	変換前ファイル表示画面	F06	
9	SC_09	変換後ファイル表示画面	F06	
10	SC_10	登録/バリデートエラー画面表示		
11	SC_11	RDF スキーマ情報ファイル登録画面	F08	
12	SC_12	RDF スキーマ情報ファイル表示画面		
13	SC_13	RDF スキーマ情報ファイル削除画面	F10	
14	SC_14	RDF スキーマ情報ファイル更新画面	F07	
15	SC_15	RDF ファイル検索画面	F14	
16	SC_16	RDF ファイル検索結果画面	F14	
17	SC_R_01	結果画面	F04,F08,F09, F08,F07,F10,F15	
18	SC_C_01	データファイル登録画面 （CKAN 連携）	F05	本画面への遷移は存在せず、SC_02 とは別 URL でのアクセスとなる。
19	SC_C_02	データファイル登録結果画面(CKAN 連携)	F05	
20	SC_I_01	共通ヘッダ	-	
21	SC_I_02	共通フッタ	-	
22	SC_17	データファイル再変換画面	F15-	

（2）　画面遷移図

各画面の遷移図について以下に示す。

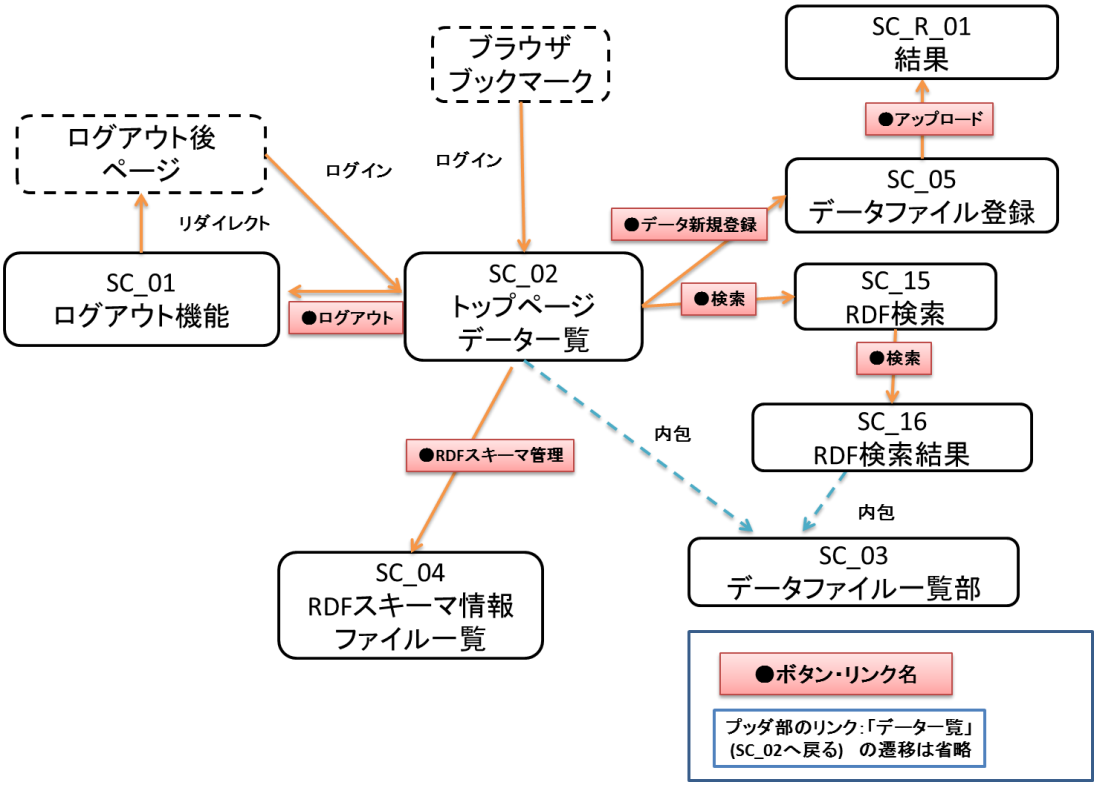


図 3-3　画面遷移図（トップページ）

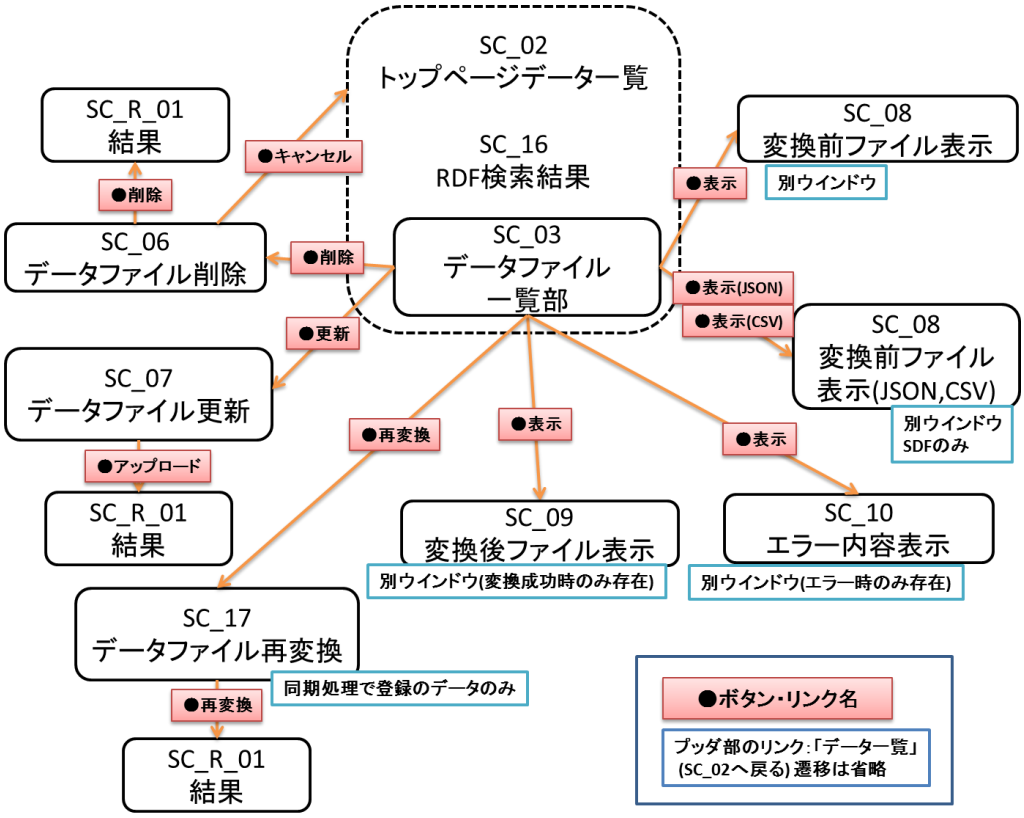


図 3-4　画面遷移図（データファイル個別）

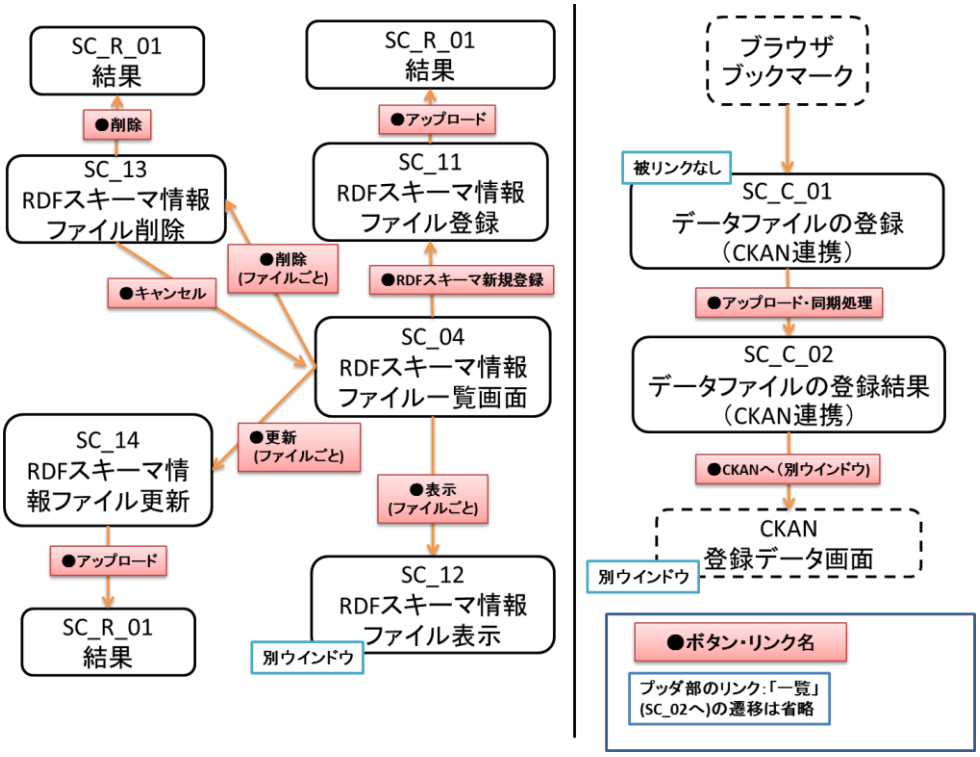


図 3-5　画面遷移図（RDF スキーマ情報ファイル・CKAN 連携）

（3）　画面定義

各画面の定義を以下に示す。

表 3-17　画面定義(SC_01)ログアウト機能画面

項目	内容
画面名	ログアウト機能画面
画面 ID	SC_01
関連機能 ID	-
概要	認証のログアウト機能を提供するための画面。
機能	SC_02 トップページ・データ一覧画面からログアウトのリンクで遷移し、利用者のブラウザに対してレスポンスコード 401 を返し、また IE に対しては、さらに認証情報のキャッシュを破棄させる JavaScript を実行させる。 また、認証対象外のページにリダイレクトする。
画面構成	このページはリダイレクトするため、スタイルされた表示画面は存在しない。 リダイレクトを禁止している利用者のために、リダイレクト先のページへのリンクのみ設ける。
画面項目	リダイレクト先のページへのテキストリンクのみ。

表 3-18　画面定義(SC_02)トップページ･データ一覧表示画面

項目	内容
画面名	トップページ・データ一覧表示画面
画面 ID	SC_02
概要	一覧表示とそれぞれの機能への遷移 UI を持ち、システムの UI のハブとなる画面。
機能	登録されたデータ一覧と、登録、検索、ログアウト、登録済みのデータの操作 UI を持つ。
関連機能 ID	F04,F14,F08

画面構成

画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	データファイル一覧部	データファイル一覧部 (SC_03)		SC_03 参照
(3)	データ新規登録	テキストリンク	SC_05	
(4)	検索	テキストリンク	SC_16	
(5)	RDF スキーマ管理	テキストリンク	SC_04	
(6)	ログアウト	テキストリンク	SC_01	
(7)	更新時刻	テキスト出力		このページのサーバ側での生成時刻
(8)	共通フッタ (SC_I_02)	インクルード画面		

表 3-19　画面定義(SC_03)データファイル一覧部

項目	内容
画面名	データファイル一覧部
画面 ID	SC_03
概要	トップページ・データ一覧表示画面(SC_02)と検索結果画面(SC_16)で、選択対象となるデータファイルそれぞれに対する表示
機能	登録されたデータに対する情報の表示と操作 UI を提供する。
関連機能 ID	F08,F09,F06

画面構成

SC_02　トップページ・データ一覧表示画面での例

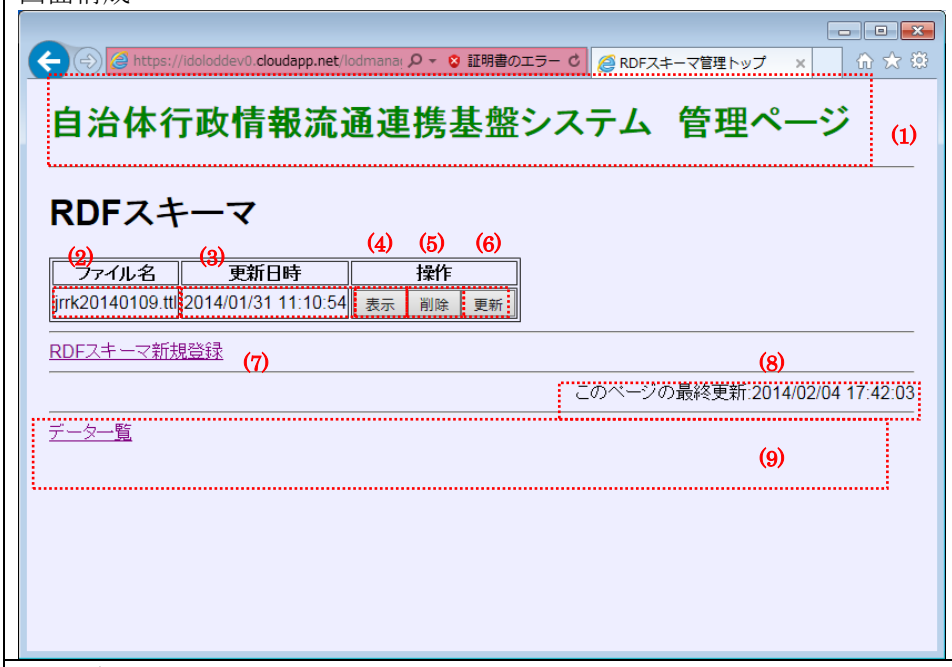
画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	データ名	テキスト出力		データファイルのファイル名本体
(2)	更新日時	テキスト出力		データファイルのアップロード時刻
(3)	ステイタス	テキスト出力		表示は、**登録済**、**登録中**、**登録失敗** の 3 種類
(4)	変換前ファイル表示	リンクボタン 別ウインドウ	SC_08	SDF の場合は、csv と json の２つのファイルへのボタンが表示される。(4')
(5)	変換後ファイル表示	リンクボタン 別ウインドウ	SC_09	ステイタスが登録中、登録失敗の場合は表示されない。
(6)	エラー内容表示	リンクボタン 別ウインドウ	SC_10	ステイタスが登録中、登録失敗の場合は表示されない。
(7)	削除	送信ボタン (POST)	SC_06	ファイル名がポストされる。
(8)	更新	送信ボタン (POST)	SC_07	ファイル名がポストされる。 SDF の場合は、csv と json の２つのファイルへのボタンが表示される。(8')
(9)	再変換	送信ボタン (POST)	SC_17	ファイル名がポストされる 同期で登録されたデータのみ表示される。

表 3-20　画面定義(SC_04)RDF スキーマ情報ファイル一覧

項目	内容
画面名	RDF スキーマ情報ファイル一覧
画面 ID	SC_04
概要	RDF スキーマ情報ファイルの一覧表示/操作 UI と、RDF スキーマ情報ファイルの登録への遷移 UI を表示
機能	登録された RDF スキーマ情報ファイルに対する情報の表示と操作 UI、RDF スキーマ情報ファイル新規登録への遷移 UI を提供する。
関連機能 ID	F07,F10

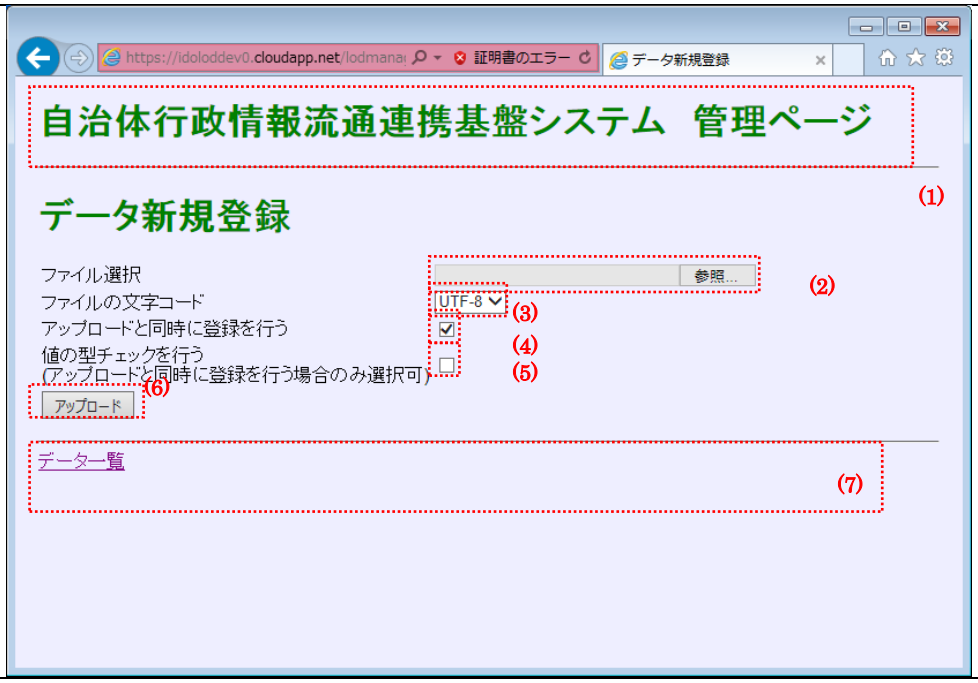
画面構成



画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル名	テキスト出力		
(3)	更新日時	テキスト出力		アップロード時刻
(4)	ファイル表示	リンクボタン 別ウインドウ	SC_12	
(5)	削除	送信ボタン(POST)	SC_13	ファイル名が POST される
(6)	更新	送信ボタン(POST)	SC_14	ファイル名が POST される
(7)	RDF スキーマ新規登録	テキストリンク	SC_11	
(8)	更新時刻	テキスト出力		このページのサーバ側での生成時刻
(9)	共通フッタ (SC_I_02)	インクルード画面		

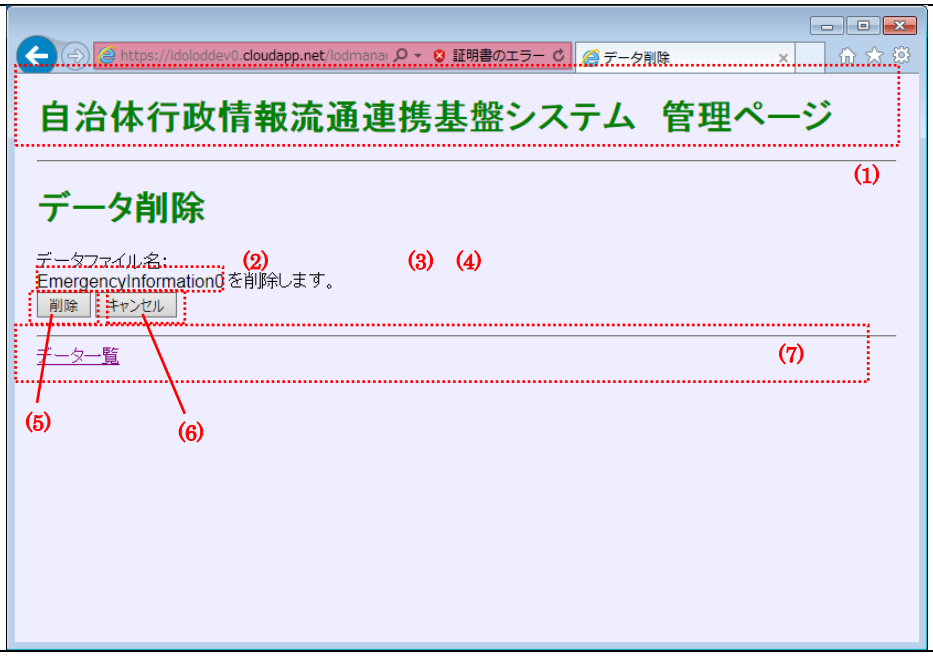
表 3-21　画面定義(SC_05)データファイル登録画面

項目	内容
画面名	データファイル登録画面
画面 ID	SC_05
概要	データファイルの登録、登録時のオプションを表示する
機能	データファイルの登録のためのファイル選択 UI と、アップロードするファイルの文字コードを指定し、同期/非同期処理の選択と、オプションでバリデータによる値の型チェックを行うかを選択させる。
関連機能 ID	F04
画面構成	

画面項目

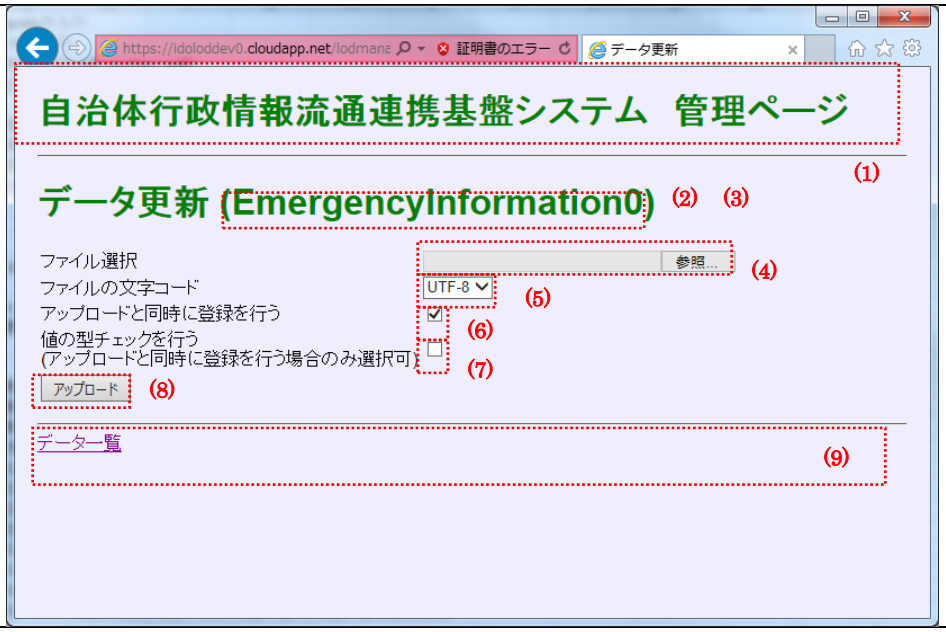
番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル選択	ファイル入力フィールド		
(3)	文字コード指定	セレクトボックス		
(4)	同期処理指定	チェックボックス		アップロードと同時にデータ変換を行う場合に指定する。
(5)	型チェック指定	チェックボックス		(4)のチェックがない場合は UI が無効になる。
(6)	アップロード	送信ボタン (POST)	SC_R_01	
(7)	共通フッタ (SC_I_02)	インクルード画面		

表 3-22 画面定義(SC_06)データファイル削除画面

項目	内容
画面名	データファイル削除画面
画面 ID	SC_06
概要	データファイルの削除の確認を行う。
機能	データファイルの削除の確認を行い、利用者に一覧表示に戻るか実行するかを選択させる。
関連機能 ID	F09
画面構成	
画面項目	

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル名	テキスト出力		SC_03(SC_02)で POST された値が入る。
(3)	ファイル名隠しフォーム	隠しフォーム		SC_03(SC_02)で POST された値が入る。
(4)	トークン隠しフォーム	隠しフォーム		
(5)	削除	送信ボタン（POST）	SC_R_01	
(6)	キャンセル	リンクボタン	SC_02	
(7)	共通フッタ (SC_I_02)	インクルード画面		

表 3-23 画面定義(SC_07)データファイル更新画面

項目	内容
画面名	データファイル更新画面
画面 ID	SC_07
概要	データファイルの更新、更新時のオプションを表示する
機能	データファイルの更新のためのファイル選択 UI と、アップロードするファイルの文字コードを指定し、同期/非同期処理の選択と、オプションでバリデータによる値の型チェックを行うかを選択させる。
関連機能 ID	F08
画面構成	

画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル名	テキスト出力		SC_03(SC_02)で POST された値が入る。
(3)	ファイル名隠しフォーム	隠しフォーム		SC_03(SC_02)で POST された値が入る。
(4)	ファイル選択	ファイル入力フィールド		選択されたファイルのファイル名は使用されず、(3)ファイル名隠しフォームでのファイル名が使用される。
(5)	エンコード指定	セレクトボックス		
(6)	同期処理指定	チェックボックス		
(7)	型チェック指定	チェックボックス		
(8)	アップロード	送信ボタン (POST)	SC_R_01	
(9)	共通フッタ (SC_I_02)	インクルード画面		

表 3-24 画面定義(SC_08)変換前ﾌｧｲﾙ表示画面

項目	内容
画面名	変換前ファイル表示画面
画面 ID	SC_08
概要	変換前のデータファイルを表示する。
機能	変換前のファイルをそのままブラウザで表示する。
関連機能 ID	F06
画面構成 画面項目	UI が存在しないため記載事項なし

表 3-25 画面定義(SC_09)変換後ﾌｧｲﾙ表示画面

項目	内容
画面名	変換後ファイル表示画面
画面 ID	SC_09
概要	変換後のデータファイルを表示する。
機能	変換後のファイルをそのままブラウザで表示する。
関連機能 ID	F06
画面構成 画面項目	UI が存在しないため記載事項なし

表 3-26　画面定義(SC_10)登録/変換/バリデートエラー画面表示画面

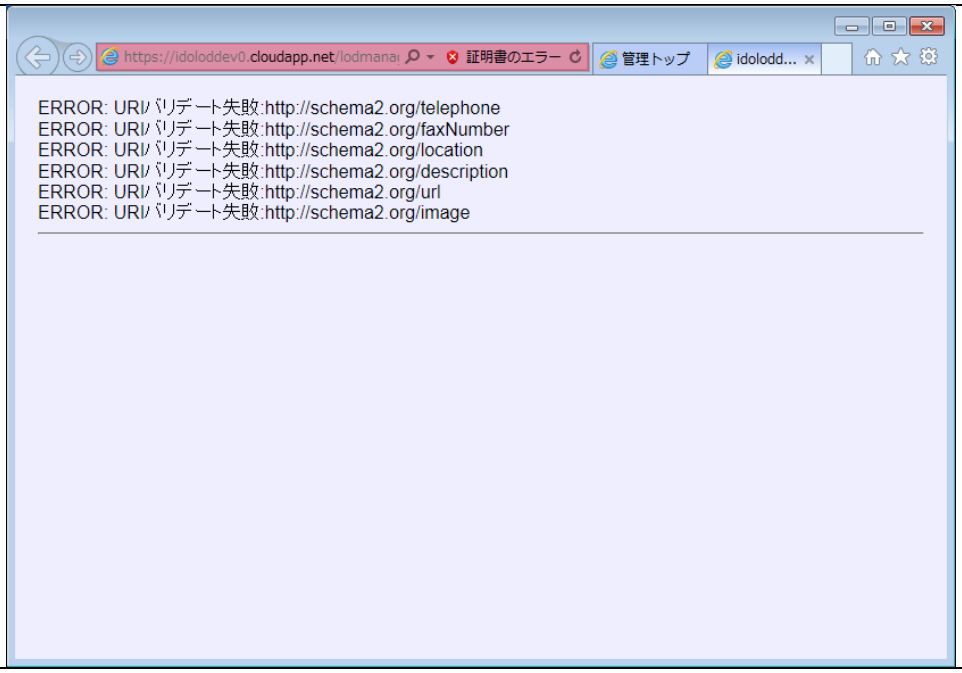
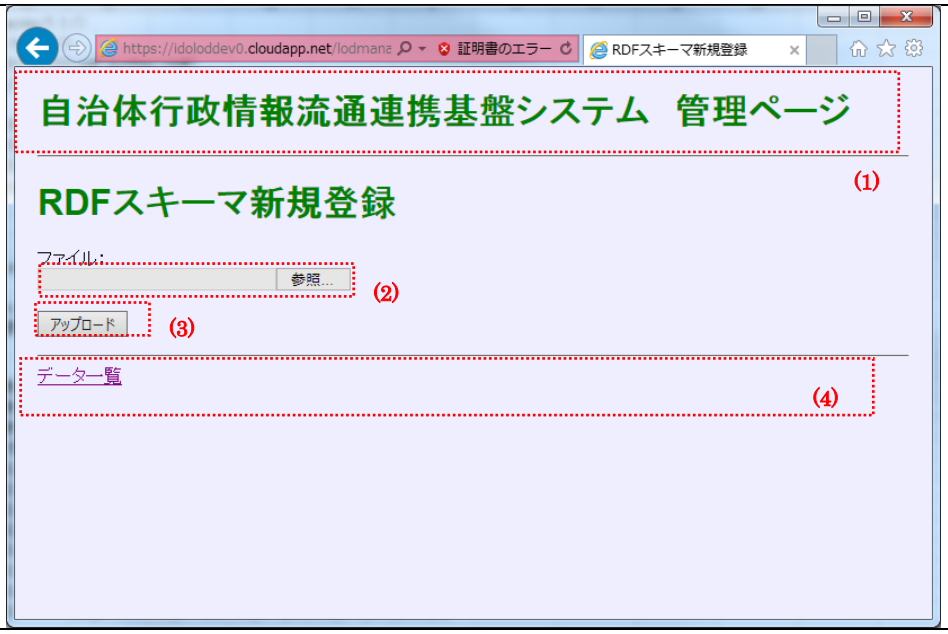
項目	内容
画面名	登録/変換/バリデートエラー画面表示画面
画面 ID	SC_10
概要	登録/変換/バリデート時のエラーを表示する。
機能	XML のエラーファイルを xslt で整形して表示する。
関連機能 ID	
画面構成	

表 3-27　画面定義(SC_11)RDF スキーマ情報ファイル登録画面

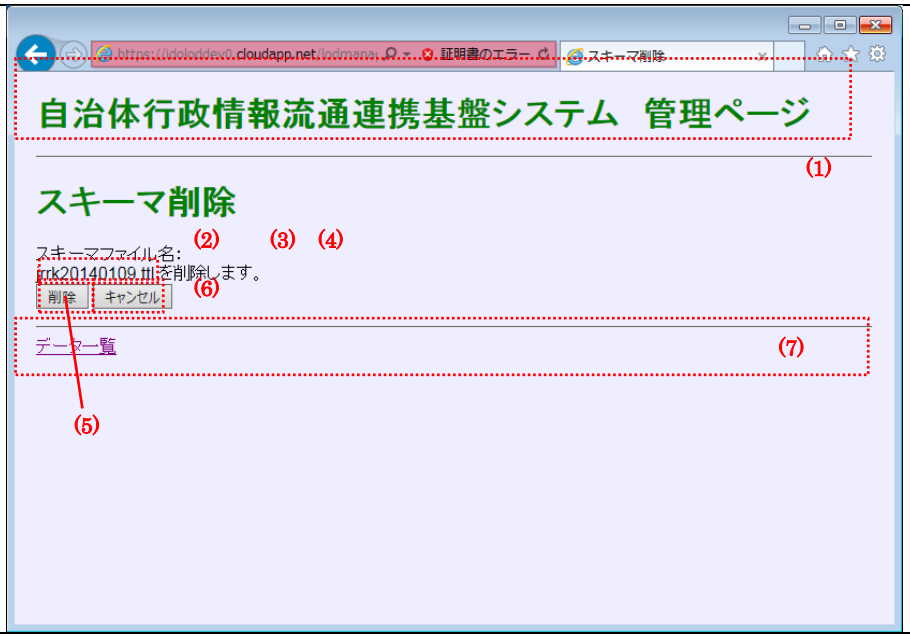
項目	内容
画面名	RDF スキーマ情報ファイル登録画面
画面 ID	SC_11
概要	RDF スキーマ情報ファイル登録の UI を表示する。
機能	RDF スキーマ情報ファイルの登録のためのファイルを選択させる。
関連機能 ID	F08
画面構成	
画面項目	

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル選択	ファイル入力フィールド		
(3)	アップロード	送信ボタン(POST)	SC_R_01	
(4)	共通フッタ (SC_I_02)	インクルード画面		

表 3-28　画面定義(SC_12)RDF スキーマ情報ファイル表示画面

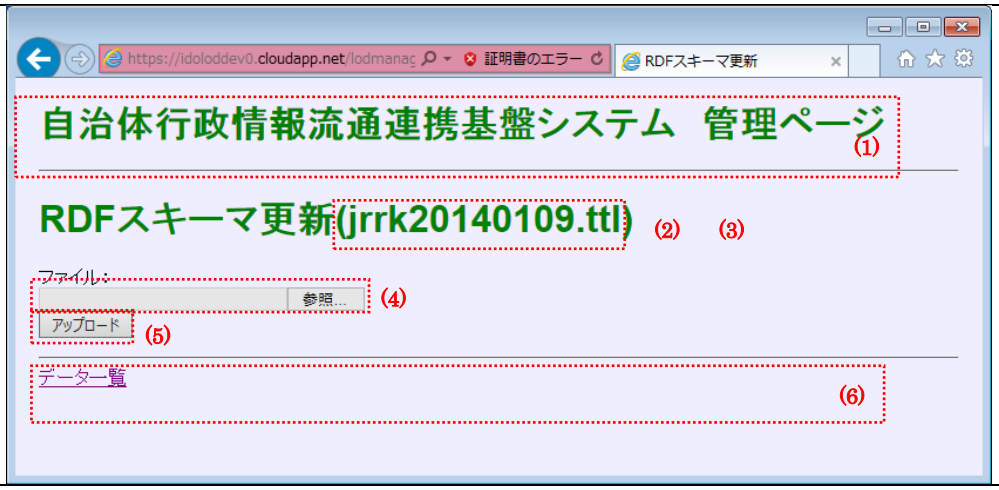
項目	内容
画面名	RDF スキーマ情報ファイル表示画面
画面 ID	SC_12
概要	RDF スキーマ情報ファイルを表示する。
機能	ファイルをそのままブラウザで表示する。
関連機能 ID	-
画面構成 画面項目	UI が存在しないため記載事項なし

表 3-29　画面定義(SC_13)RDF スキーマ情報ファイル削除画面

項目	内容
画面名	RDF スキーマ情報ファイル削除画面
画面 ID	SC_13
概要	RDF スキーマ情報ファイルの削除の確認を行う。
機能	RDF スキーマ情報ファイルの削除の確認を行い、利用者にキャンセルさせるか削除を実行するかを選択させる。
関連機能 ID	F10
画面構成	
画面項目	

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル名	テキスト出力		SC_04 で POST された値が入る。
(3)	ファイル名隠しフォーム	隠しフォーム		SC_04 で POST された値が入る。
(4)	トークン隠しフォーム	隠しフォーム		
(5)	削除	送信ボタン（POST）	SC_R_01	
(6)	キャンセル	リンクボタン	SC_04	
(7)	共通フッタ (SC_I_02)	インクルード画面		

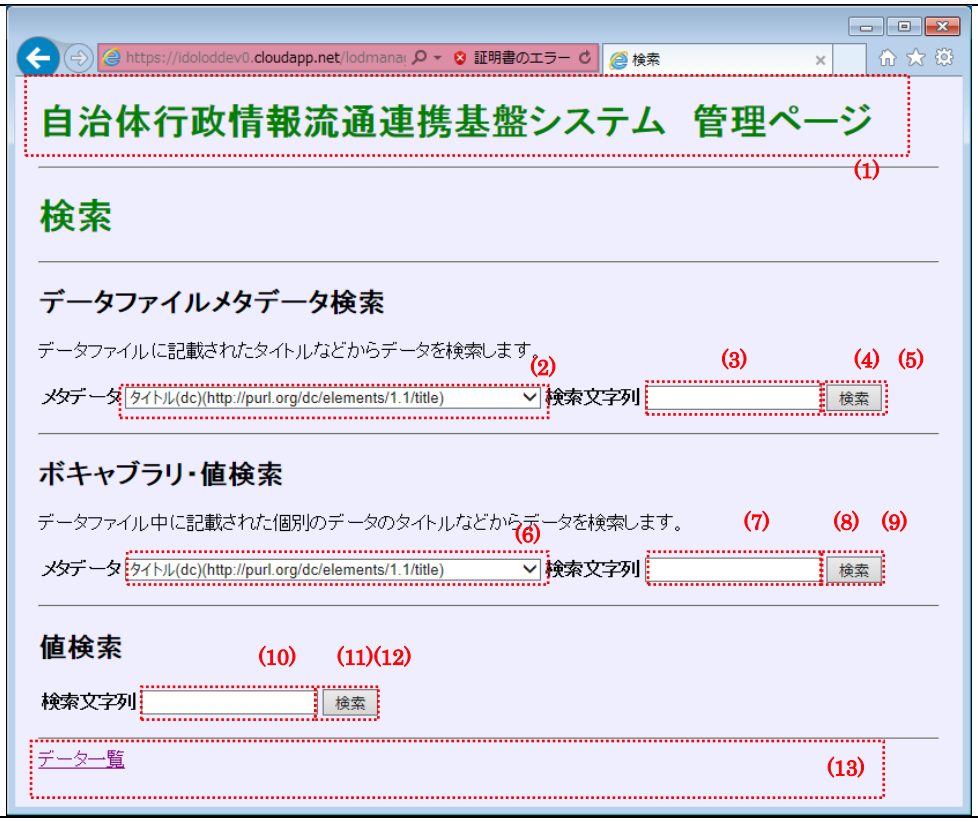
表 3-30　画面定義(SC_14)RDF スキーマ情報ファイル更新画面

項目	内容
画面名	RDF スキーマ情報ファイル更新画面
画面 ID	SC_14
概要	RDF スキーマ情報ファイルの更新のための UI を表示する
機能	RDF スキーマ情報ファイルの更新のためのファイルを選択させる。
関連機能 ID	F08
画面構成	

画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル名	テキスト出力		SC_03(SC_02)で POST された値が入る。
(3)	ファイル名隠しフォーム	隠しフォーム		SC_03(SC_02)で POST された値が入る。
(4)	ファイル選択	ファイル入力フィールド		選択されたファイルのファイル名は使用されず、(3)ファイル名隠しフォームでのファイル名が使用される。
(5)	エンコード指定	セレクトボックス		
(6)	同期処理指定	チェックボックス		
(7)	型チェック指定	チェックボックス		
(8)	アップロード	送信ボタン (POST)	SC_R_01	
(9)	共通フッタ (SC_I_02)	インクルード画面		

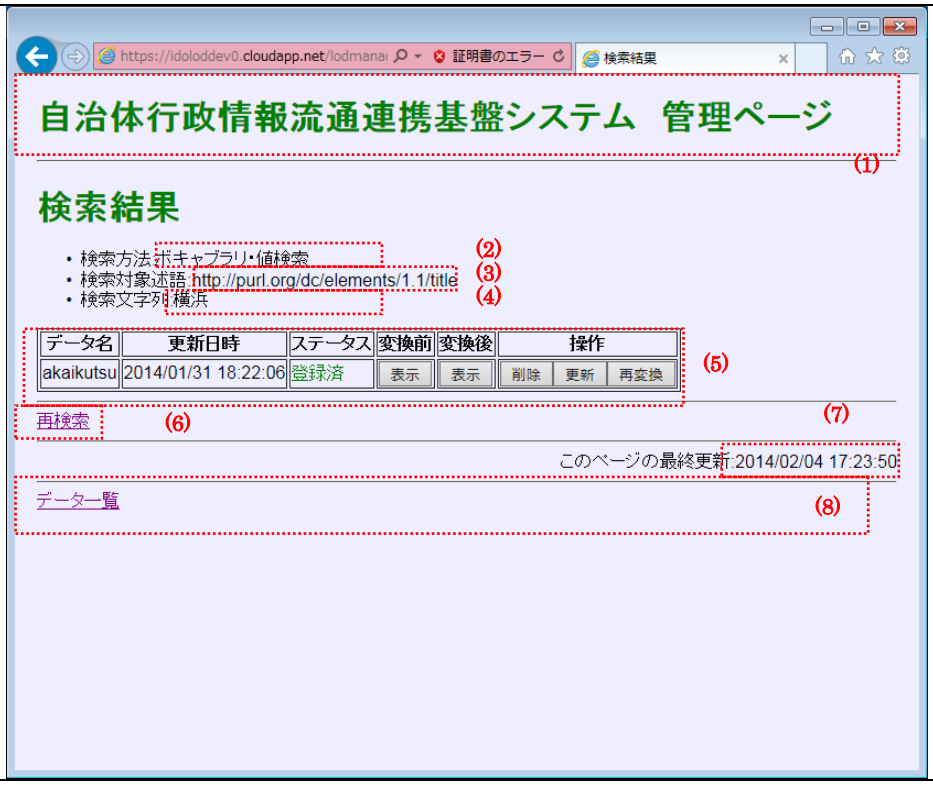
表 3-31　画面定義(SC_15)RDF ファイル検索画面

項目	内容
画面名	RDF ファイル検索画面
画面 ID	SC_15
概要	RDF ファイルの検索を行うための UI を表示する。
機能	検索の種類と述語を選択し、キーワードを入力して RDF ファイルの検索を行うための UI を提供する。
関連機能 ID	F14
画面構成	

画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル-メタデータ選択	セレクトボックス		表示内容と URI は別ファイルにて定義
(3)	ファイル-検索文字列入力	テキストボックス		
(4)	検索方法-ファイル	隠しフォーム		
(5)	ファイル-検索実行	送信ボタン (POST)	SC_16	(2)の述語、(3)の値でファイルのメタデータの Triple を検索対象とする。
(6)	データ-メタデータ選択	セレクトボックス		表示内容と URI は別ファイルにて定義
(7)	データ-検索文字列入力	テキストボックス		
(8)	検索方法-データ	隠しフォーム		
(9)	データ-検索実行	送信ボタン (POST)	SC_16	(6)の述語、(7)の値で全 Triple を検索対象とする。
(10)	値-検索文字列入力	テキストボックス		
(11)	検索方法-値	隠しフォーム		
(12)	値-検索実行	送信ボタン (POST)	SC_16	(10)の値で全 Triple の値を検索対象とする。
(13)	共通フッタ (SC_I_02)	インクルード画面		

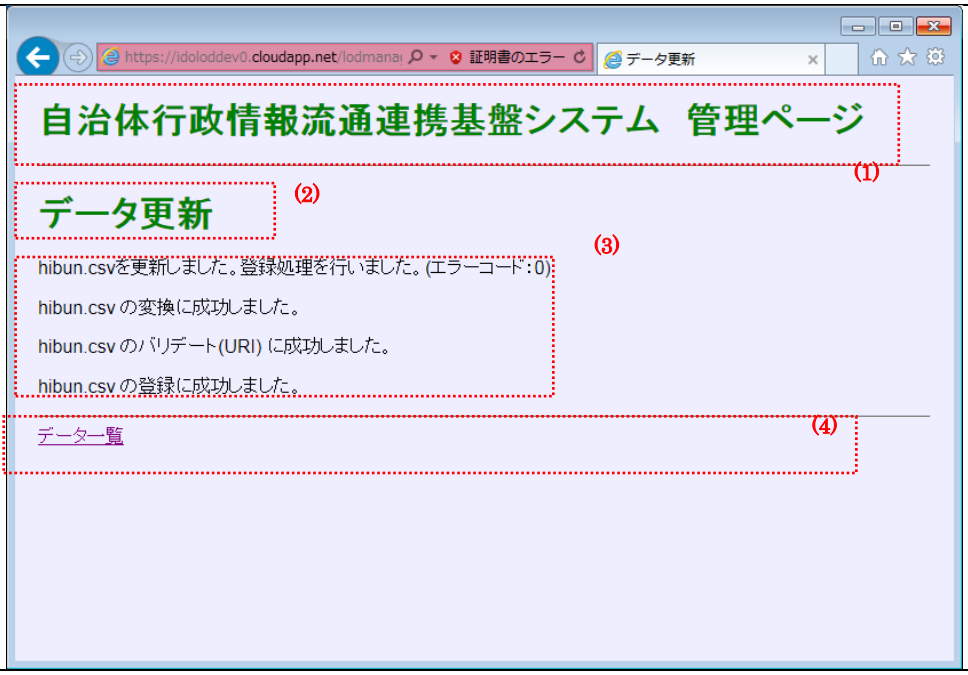
表 3-32 画面定義(SC_16) RDF ファイル検索結果画面

項目	内容
画面名	RDF ファイル検索結果画面
画面 ID	SC_16
概要	RDF ファイル検索結果を表示し、検索結果の RDF に対しての操作 UI を提供する。
機能	RDF ファイルの検索結果から、それぞれのデータの状態と、操作 UI を提供する。なお、検索は公開サーバの Sparql Endpoint を使用するため、変換完了のもののみが対象となる。
関連機能 ID	F14
画面構成	

画面項目

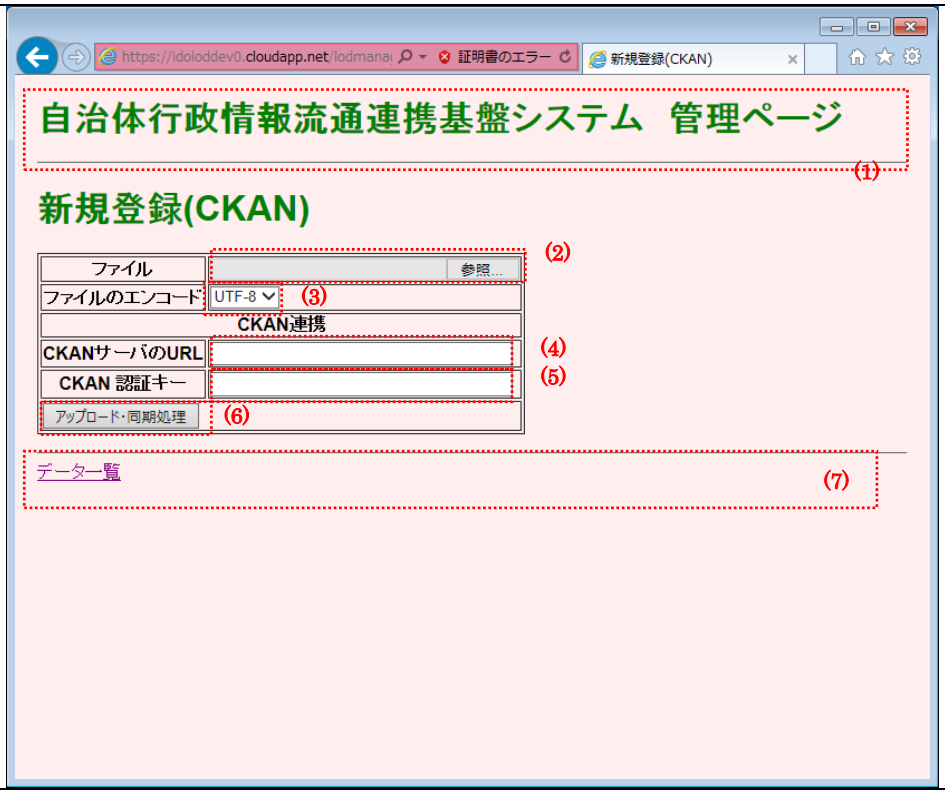
番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	検索方法	テキスト出力		SC_15 で POST された値から表示文字列を決定する
(3)	検索対象述語	テキスト出力		SC_15 で POST された値が入る。
(4)	検索文字列	テキスト入力		SC_15 で POST された値が入る。
(5)	データファイル一覧	データファイル個別 (SC_03)		
(6)	検索	テキストリンク	SC_15	
(7)	更新時刻	テキスト出力		このページのサーバでの生成時刻
(8)	共通フッタ (SC_I_02)	インクルード画面		

表 3-33　画面定義(SC_R_01)結果画面

項目	内容
画面名	結果画面
画面 ID	SC_R_01
概要	登録、削除、更新の結果を表示する。
機能	登録、削除、更新の結果を表示する。
関連機能 ID	F04,F08,F09,F08,F07,F10,F15
画面構成	
画面項目	

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	処理内容	テキスト出力		
(3)	メッセージ	テキスト出力		
(4)	共通フッタ (SC_I_02)	インクルード画面		

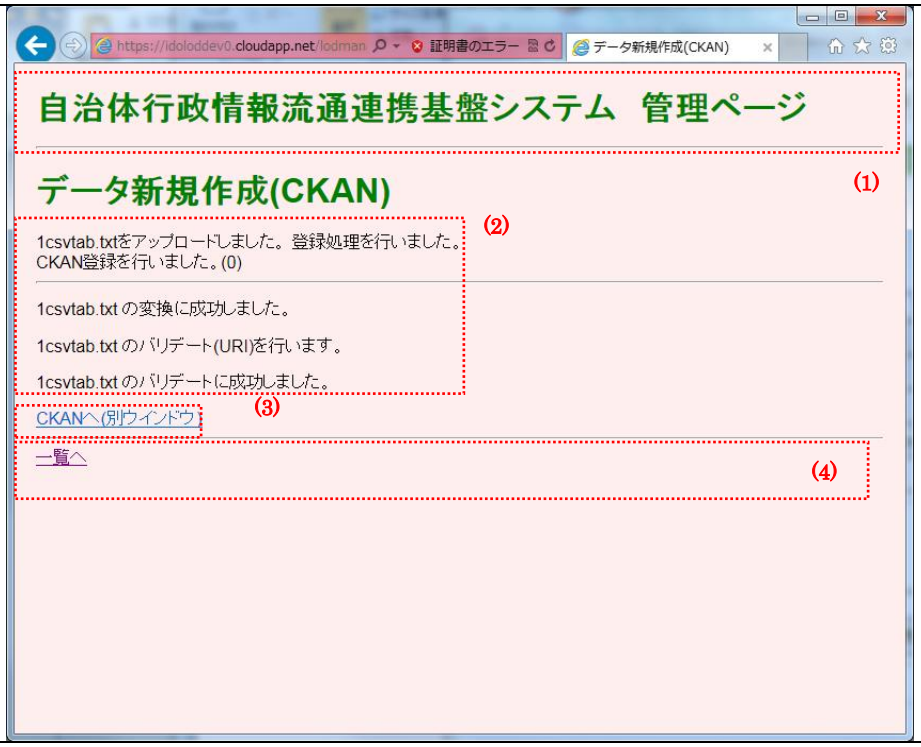
表 3-34　画面定義(SC_C_01)データファイル登録画面(CKAN 連携)

項目	内容
画面名	データファイル登録画面(CKAN 連携)
画面 ID	SC_C_01
概要	データファイルの登録、登録時のオプション、CKAN への接続情報の入力 UI を表示する
機能	データファイルの登録のためのファイル選択 UI と、アップロードするファイルの文字コードを指定し、オプションでバリデータによる値の型チェックを行うと、接続先の CKAN の URL,認証キーの入力を行う。
関連機能 ID	F05
画面構成	

画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル選択	ファイル入力フィールド		
(3)	エンコード指定	セレクトボックス		
(4)	CKAN URL	テキストボックス		接続先 CKAN の URL を、http://[ホスト名]/ で指定する。
(5)	CKAN 認証キー	チェックボックス		
(6)	アップロード	送信ボタン (POST)	SC_C_02	
(7)	共通フッタ (SC_I_02)	インクルード画面		

表 3-35 画面定義(SC_C_02)データファイル登録結果画面(CKAN 連携)

項目	内容
画面名	データファイル登録結果画面(CKAN 連携)
画面 ID	SC_C_02
概要	データの基盤への登録と、メタ情報の CKAN への登録結果を表示する。
機能	データの基盤への登録と、メタ情報の CKAN への登録結果を表示する。また、CKAN への登録成功時には、CKAN に登録した情報へのリンクを表示する。
関連機能 ID	F05
画面構成	

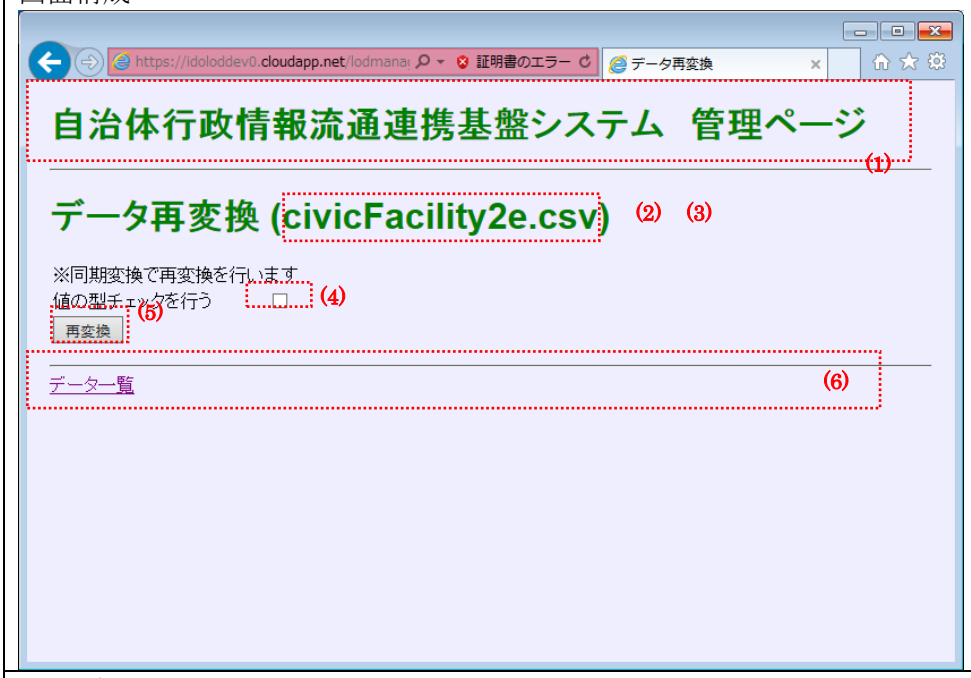
画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ(SC_I_01)	インクルード画面		
(2)	メッセージ	テキスト出力		
(3)	CKAN へのリンク	テキストリンク(別ウインドウ)		SC_C_01 で指定した CKAN サーバで、登録で作成されたエントリへのリンク
(4)	共通フッタ(SC_I_02)	インクルード画面		

表 3-36　画面定義(SC_17)データファイル再変換画面

項目	内容
画面名	データファイル再変換画面
画面 ID	SC_17
概要	データファイルの再変換時のオプションを表示する
機能	データファイルの再変換で、オプションでバリデータによる値の型チェックを行うかを選択させる。
関連機能 ID	F15

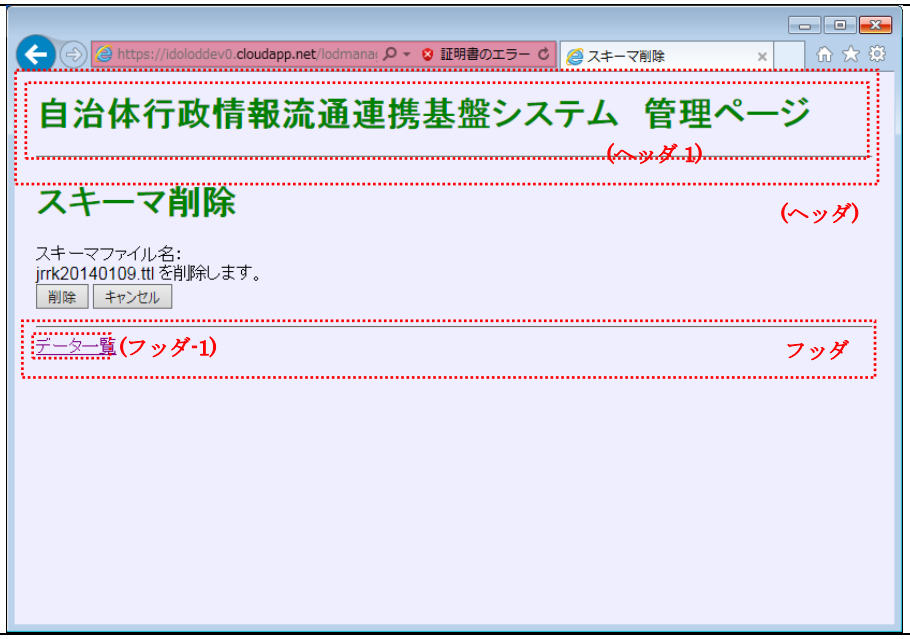
画面構成



画面項目

番号	項目名	種別	遷移先画面 ID	備考
(1)	共通ヘッダ (SC_I_01)	インクルード画面		
(2)	ファイル名	テキスト出力		SC_03(SC_02)で POST された値が入る。
(3)	ファイル名隠しフォーム	隠しフォーム		SC_03(SC_02)で POST された値が入る。
(4)	型チェック指定	チェックボックス		
(5)	再変換	送信ボタン(POST)	SC_R_01	
(6)	共通フッタ (SC_I_02)	インクルード画面		

表 3-37　画面定義(SC_I_01)共通ﾍｯﾀﾞ,(SC_I_02)共通ﾌｯﾀﾞ

項目	内容
画面名	共通ヘッダ，共通フッダ
画面 ID	SC_I_01(ヘッダ), SC_I_02(フッダ)
概要	変換ツール共通のヘッダ、フッダ
機能	共通の定義項目や UI を提供する。
画面構成	
画面項目	

番号	項目名	種別	遷移先画面 ID	備考
(ヘッダ 1)	システム名称	テキスト出力		
(フッダ 1)	データ一覧	テキストリンク	SC_02	

3.3.4. オンライン処理設計

（1）　オンライン処理方式概要

データ変換・登録ツールでは、自治体職員による Web 画面からのファイルのアップロードや、外部システム（CMS 等）からのファイルアップロードに対して、オンラインでデータ変換、バリデート、登録・更新する機能を有する。

1）設計方針

UI 操作を伴うページは、自治体職員からのアクセスに限定されるが、クラウド上に自治体行政情報流通連携基盤システムを展開するため、認証機能を有する必要がある。

外部システム（CMS 等）からのアクセスに関しても認証機能を有する必要があるが、外部システム（CMS 等）の仕様が多岐に渡ることから、自治体行政情報流通連携基盤システムでは汎用的に使用できるものとして、IP もしくはホスト名での認証機能のみを提供する。

また公開サーバから管理サーバ側へ更新データの確認のアクセスが毎分発生するため、更新データ一覧ページは負荷がかからないようにする必要がある。

2）多重･同時実行制御、負荷分散

自治体管理者の UI 操作をトリガにオンライン処理が起動するため、ここでは同時・多重実行制御、負荷分散は行わない。

また、外部システム（CMS 等）からのアップロードに関しても、現時点では同時・多重実行制御、負荷分散は行わない。

3）性能設計、負荷分散方式

アップロードするデータサイズ、自治体管理者からクラウド環境へのネットワーク環境、クラウド環境に依存するため、性能設計は行わない。

（2）　オンライン処理一覧

データ変換・登録ツールにおけるオンライン処理を一覧で示す。「関連機能 ID」は、オンライン処理が実現する機能の機能 ID（「2.2.1 機能一覧」参照）を示す。

表 3-38 データ変換・登録ツール オンライン処理一覧

項番	設置サーバ	処理 ID	処理名	開発言語	関連機能 ID
1	管理	R01	Upload	PHP	F04,F08,F05
2	管理	R02	オンライン処理スクリプト	bash	F04,F08,F05,F15
3	管理	R03	変換/バリデート/登録更新	Java	F04,F08,F02,F03,F05,F15
4	管理	R04	CKAN 自動登録	PHP	F05
5	管理	R05	管理サーバ更新データ一覧	PHP	F04,F08,F09,F08,F07,F10,F05,F15
6	管理	R06	データファイル削除	PHP	F09
7	管理	R07	RDF スキーマファイル登録更新	PHP	F08,F07
8	管理	R08	RDF スキーマファイル削除	PHP	F10
9	管理	R09	検索	PHP	F14

（3）　オンライン処理定義

各オンライン処理の定義を以降で示す。なお、定義内に記載したファイル名のパスは、実証時の導入構成である。

表 3-39 処理定義(R01)

項目	内容
処理名	Upload
処理 ID	R01
関連機能 ID	F04,F08,F05
開発言語	PHP
インターフェース	HTTP
ファイル名	/var/www/html/lodmanage/upload.php (Web 画面-新規登録) /var/www/html/lodmanage/upload.php (Web 画面-更新) /var/www/html/cmsupload.php (外部システム（CMS 等）)
処理内容	UI/外部システム（CMS 等）からの HTTP のアップロード要求に対応し、アップロードしたファイルを UTF-8 の文字コードに変換し、所定の位置に格納する。またアップロード後にオンライン処理を行う要求があった場合、引き続きオンラインで後述の R03:変換/バリデート/登録更新の処理を行う。 なお、 ・UI からのアップロード（新規登録） ・UI からのアップロード（更新） ・外部システム（CMS 等）からのアップロード では、別の PHP ファイルで処理を行う。

入力
(HTTP Multipart Post)

項番	パラメタ名	内容	値
1	sync	アップロードするファイルの処理をオンラインかバッチかを指定する	**on**:オンライン処理 **省略時/上記以外**：バッチ処理
2	encode	アップロードするファイルの文字コードを指定する。 ただし、拡張子が、ttl,jsonの場合は指定を無視して、文字コードの変換を行わない。	**euc**: euc と見なす。 **sjis**:sjis と見なす。 **utf8**:UTF-8 と見なす。（変換しない） ただし、euc,sjis は、内部ではそれぞれ、PHP の eucJP-win, SJIS-win として処理する。 **省略時/上記以外**: utf8 が指定されたものとする。
3	valtype	オンライン処理を指定した場合、値の型のバリデートを行うかを指定する。	**true**: 値のデータ型のバリデートを行う。 **省略時/上記以外**: 値の型のバリデートを行わない。
4	upfile	Multipart で送るファイルデータ本体とファイル名	
5	basename	更新対象のファイル名のベース	更新時のみ使用。更新時は省略不可。登録時は無視される。

出力
(HTTP レスポンスコード)

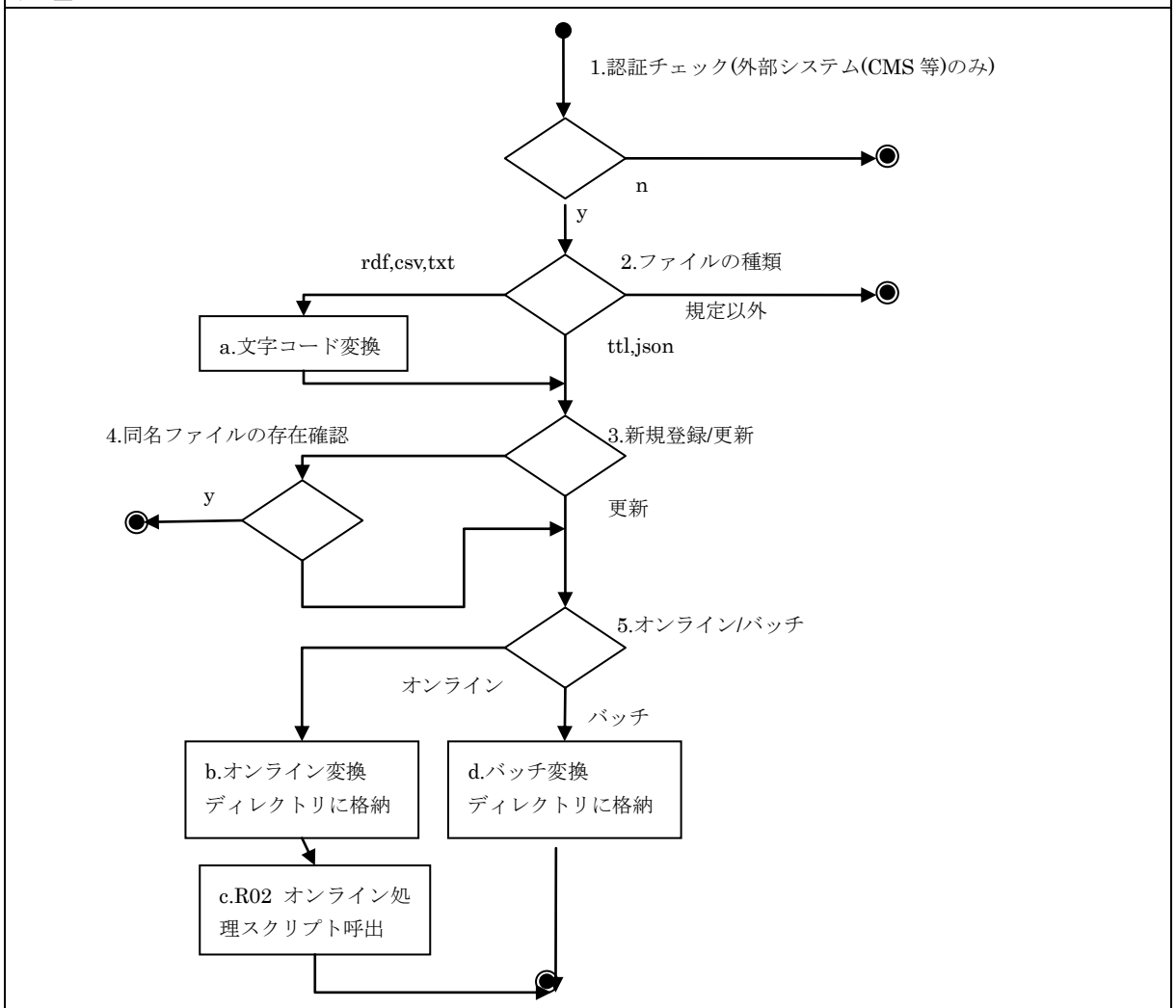
項番	内容	レスポンスコード	備考
1	正常終了	200	
2	アップロードに失敗	500	
3	変換に失敗	400	オンライン処理時のみ※
4	バリデートに失敗	400	オンライン処理時のみ※
5	システム内エラー	500	
6	認証失敗(外部システム（CMS 等）)のみ	401	

※エラー内容に即した XML をレスポンスとして返す。
フォーマットを以下に示す。

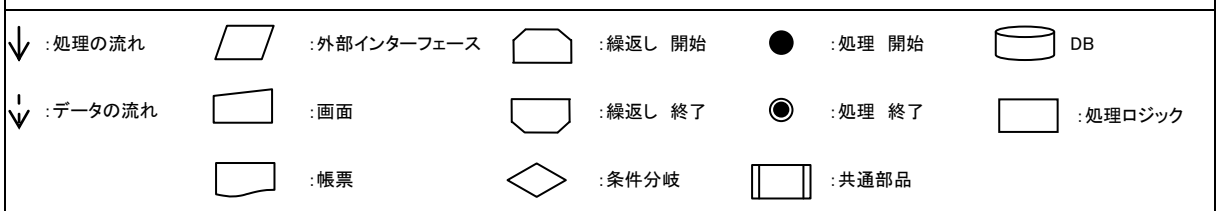
をルートとし、<p>を子要素として持つ。
<p>の値としてエラー内容の文字列が格納される。
<p>は複数回現れる場合がある。

	<?xml version="1.0" encoding="UTF-8" ?> メッセージ
出力（ファイル）	入力でのオンライン処理、バッチ処理の指定に沿ったディレクトリにアップロードされたファイルが出力される。この際、拡張子と文字コードの指定により、アップロードされたファイルは UTF-8 に変換されて保存される。 また、更新の場合はパラメタでファイルのベース名が指定されたものになる。拡張子はアップロードしたファイルのものが使用される。

処理フロー

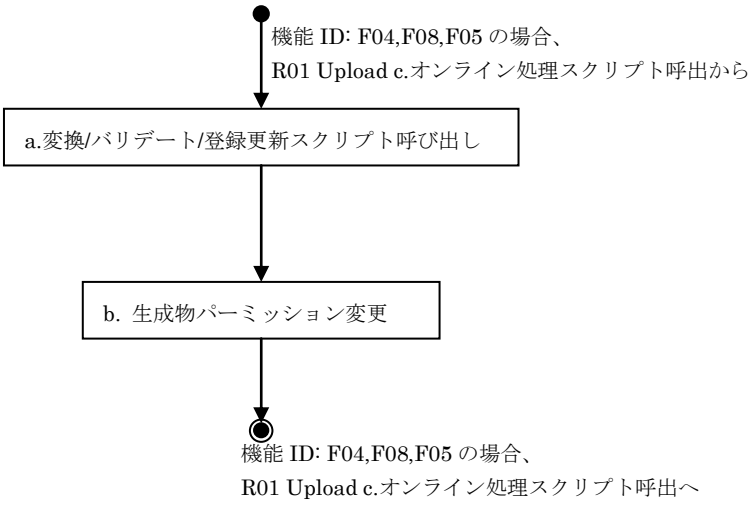


凡例

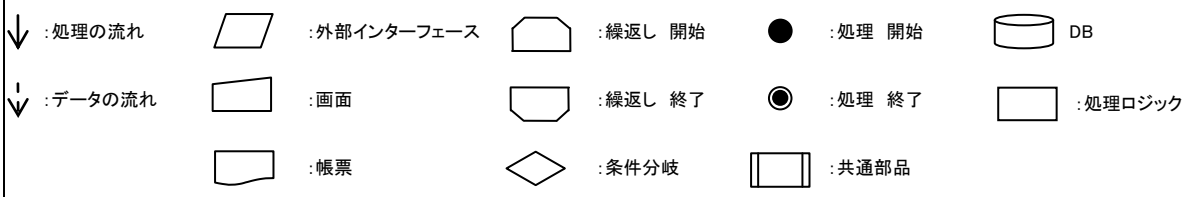


処理詳細
1. 認証チェック(外部システム（CMS 等）のみ) 　外部システム（CMS 等）からのアップロードについては、Basic 認証を行わないため、独自の認証を行う。アクセス元がスクリプト中に指定された IP アドレスもしくはドメインと合致していない場合はレスポンスコード:401 で終了する。 2. ファイルの種類 　対象外のファイルのアップロードを防止することと、拡張子が ttl,json のものは文字コードの変換を行わないため、ブラウザ/外部システム（CMS 等）から Multipart でアップロードする際に付与されるファイル名でファイルの種類を判別し、処理を分ける。 3. 新規登録/更新 　新規登録と更新でアップロードするファイル名の取り扱いが異なるため処理を分ける。 （新規登録と更新の処理を行う PHP ファイル自体は別である） 4. 同名ファイルの存在確認 　新規登録の場合、アップロード先に同名ファイル(※)がある場合、既存のデータの上書きになるため、アップロードを行わずに終了する。 　なお、更新の場合は、システムに登録するファイル名は別途パラメタとして取得し、そのパラメタの値でファイル名を決定する。また、アップロード先の同名ファイル（※）の削除を行う。 ※：同名ファイルの判定は、以下で行う。 　アップロードされるファイルが.json 以外の場合： 　　ファイルのベース名が同じファイルかつ、拡張子が.json 以外の場合 　アップロードされるファイルが.json の場合 　　ファイルのベース名が同じファイルかつ、拡張子が.json の場合 5. オンライン/バッチ 　オンライン処理の場合は、ファイルの格納後、引き続き変換、バリデート処理を行い、終了結果を取得して返す。 　バッチ処理の場合は、ファイルの格納のみを行って、終了結果を返す。 a. 文字コード変換 　パラメタで指定された文字コードをアップロードされたファイルの文字コードと見なして、UTF-8 形式でファイルを保存する。なお、json, .ttl は自治体行政情報流通連携基盤システムでは常に UTF-8 として取り扱うため変換を行わない。 b. オンライン変換ディレクトリに格納 　オンライン変換ディレクトリにアップロード・文字コード変換されたファイルを保存する。また、バッチ変換ディレクトリ中の同名ファイルを削除する。 c. オンライン処理スクリプト呼び出し 　オンライン処理スクリプトを呼び出して、データの変換、バリデート、登録更新を行う。詳細は後述する。なお、外部システム（CMS 等）からのアップロード/オンライン処理時は、オンライン処理スクリプトの標準出力を整形して HTML に出力し、利用者へ返す。 d. バッチ変換ディレクトリに格納 　バッチ変換ディレクトリにアップロード・文字コード変換されたファイルを保存する。また、オンライン変換ディレクトリ中の同名ファイルを削除する。

表 3-40 処理定義(R02)

<table>
<tr><th>項目</th><th>ポイント</th></tr>
<tr><td>処理名</td><td>オンライン処理スクリプト</td></tr>
<tr><td>処理 ID</td><td>R02</td></tr>
<tr><td>関連機能 ID</td><td>F04,F08,F05,F15</td></tr>
<tr><td>開発言語</td><td>bash</td></tr>
<tr><td>インターフェース</td><td>コマンドライン</td></tr>
<tr><td>ファイル名</td><td>/home/adminuser/jrrktool/runconvertsync.sh</td></tr>
<tr><td>処理内容</td><td>指定されたファイルをオンライン処理するために、後述する R03:変換/バリデート/登録更新（表 3-41）を呼び出し、R03:変換/バリデート/登録更新で生成された RDF ファイルを所定の位置に配置する。
また、生成物のパーミッションの変更を行う。</td></tr>
<tr><td>入力
(コマンドライン)</td><td>bash スクリプトの引数
<table>
<tr><th>項番</th><th>内容</th><th>値</th></tr>
<tr><td>1</td><td>処理対象ファイル</td><td>処理対象のファイルへのフルパス。（省略不可）</td></tr>
<tr><td>2</td><td>バリデートの値の型チェックの実行指定</td><td>true: 型チェックを行う。
省略/上記以外:
型チェックを行わない。</td></tr>
</table></td></tr>
<tr><td>出力
(コマンドライン戻り値)</td><td>
<table>
<tr><th>項番</th><th>内容</th><th>終了コード</th><th>備考</th></tr>
<tr><td>1</td><td>正常終了</td><td>0</td><td></td></tr>
<tr><td>2</td><td>失敗</td><td>0 以外※</td><td></td></tr>
</table>
※失敗時は、後述の R03:変換/バリデート/登録更新の実行時の戻り値をそのまま返す。</td></tr>
<tr><td colspan="2">処理フロー</td></tr>
<tr><td colspan="2"> 
</td></tr>
</table>

凡例



処理詳細

a. R03:変換/バリデート/登録更新呼び出し
　R03:変換/バリデート/登録更新の機能を呼び出す

b. 生成物パーミッション変更
　本スクリプトは、Apache から呼び出されるため、Apache の実行ユーザのパーミッションで生成物が生成されるが、後述するバッチ処理変換は一般ユーザで実行するため、バリデート前の RDF ファイル、バリデート後の RDF ファイル、エラーファイルのファイルのパーミッションをすべて Write 可に変更し、実行時に上書き、削除時の問題が発生しないようにする。

表 3-41 処理定義(R03)

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>処理名</td><td>変換/バリデート/登録更新</td></tr>
<tr><td>処理 ID</td><td>R03</td></tr>
<tr><td>関連機能 ID</td><td>F04,F08,F02,F03,F05,F15</td></tr>
<tr><td>開発言語</td><td>Java</td></tr>
<tr><td>ファイル名</td><td>/home/adminuser/jrrktool/jar/convtool.jar</td></tr>
<tr><td>インターフェース</td><td>コマンドライン</td></tr>
<tr><td>処理内容</td><td>指定されたディレクトリ、ファイル名、パラメタから、データの RDF へのフォーマット変換を行い、その後、バリデート、登録・更新を行う。

バリデートは、述語の URI を指定されたディレクトリ中のダブリンコア等既存ボキャブラリの RDFS と、自治体行政情報流通連携基盤システムで使用するボキャブラリのディレクトリ中の RDFS で定義されている語彙の URI とのマッチングを行い、存在しない URI を使用している場合はバリデート失敗とする。
また、オプションで、値の型のチェックも行う機能を有する。ただし、本実証においては、チェック対象の型については、文字列、整数、小数のみの制限を設ける。

なお、RDFS 側でリソースとして定義されている値に関しては、リソースとして出力し、一部の型付きリテラルと定義している値に関しては、値が定義されている型である場合は、型付きリテラルで出力する。

バリデート OK のファイルに関しては、後述する R05 管理サーバ更新データ一覧（表 3-43）で公開サーバに配置するために、所定のディレクトリに配置する。

また、変換、バリデート処理の結果を標準出力に出力し、オンライン処理スクリプト側の出力に流す。</td></tr>
<tr><td>入力
(コマンドライン)</td><td>本ツールの Java の実行オプションは以下の通り。

<table>
<tr><th>項番</th><th>内容</th><th>値</th></tr>
<tr><td>1</td><td>クラスパス</td><td>-cp "/home/adminuser/jrrktool/jar/*"

データ登録・変換ツールの jar のパスを指定する。</td></tr>
<tr><td>2</td><td>ロガー設定</td><td>-Dlog4j.configuration=file:///home/adminuser/jrrktool/nulllog4j.properties

※jena のライブラリが Log4j を使用するが、本ツールでは使用しないため、初期化エラーのログが出力されるのを抑止するために設定する。</td></tr>
<tr><td>3</td><td>実行クラス</td><td>net.msecnd.vo.jrr□.LodCon□ert</td></tr>
</table></td></tr>
</table>

実行時パラメタは以下の通り

項番	内容	値
1	変換対象ファイルへのフルパス	ファイルが指定された場合はそのファイルを変換する。 ディレクトリが指定された場合はそのディレクトリ以下のファイルを対象とする。 (オンライン処理時はファイルが指定され、バッチ処理時はディレクトリが指定される)
2	バリデート前 RDF ファイル出力ディレクトリ	/var/www/html/lodmanage/tempout
3	最終出力先ディレクトリ	/var/www/html/rdf/
4	変換・バリデート失敗時のエラーファイル出力先ディレクトリ	/var/www/html/lodmanage/failed/
5	実証時に定義した RDF スキーマファイルの格納ディレクトリ	/var/www/html/schema/
6	ダブリンコア等の既存の RDF スキーマファイルの格納ディレクトリ	/var/www/html/lodmanage/commonschema/
7	バリデート対象外の URI を列挙したファイルを格納するディレクトリ (表 3-66 参照)	/var/www/html/lodmanage/ignorelist
8	変換するドキュメントの Namespace ※ドキュメントそのもののメタデータを RDF で表現するために指定	http://[自治体等のドメイン]/[パス]
9	バリデータで値の型チェックを行うかの指定	true: 型チェックを行う。 上記以外:型チェックを行わない。
10	値のないタグを RDF に出力するかの指定	true:空タグを出力する。 上記以外：空タグを出力しない。
11	値をリソース化するときの URI の Prefix	http://[自治体等のドメイン]/[パス]

出力
(コマンドライン)

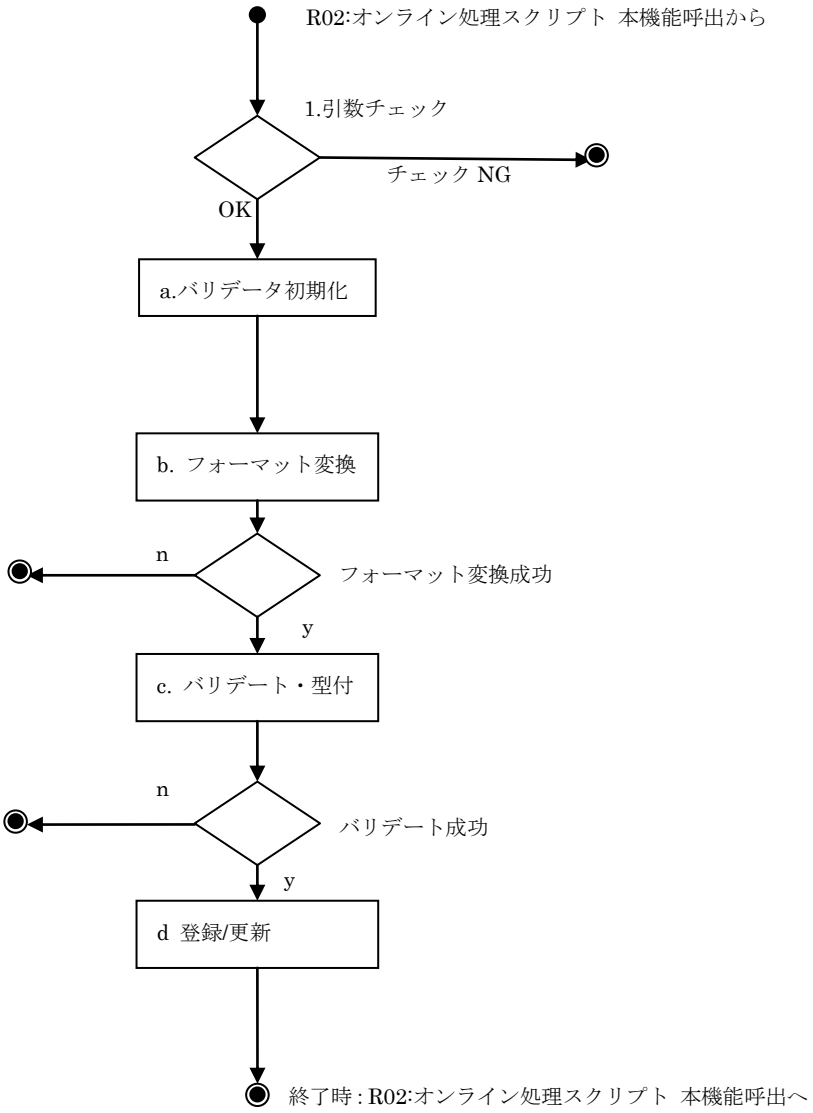
オンライン処理時

項番	内容	終了コード	備考
1	正常終了	0	
2	システム内エラー	-1	
3	変換に失敗	1	
4	バリデートに失敗	2	

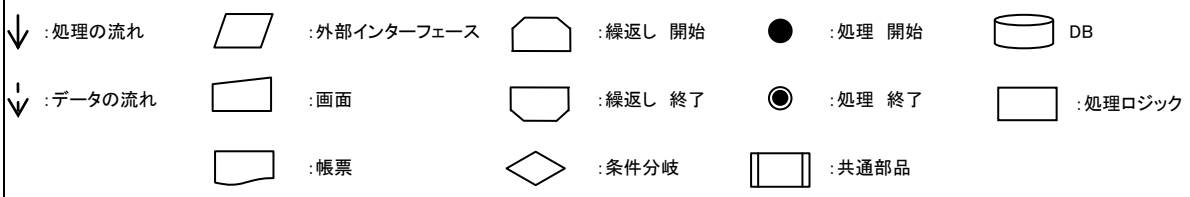
出力
（ファイル）

項番	変換	バリデート	出力
1	成功	成功	最終出力先ディレクトリに変換済みの RDF が出力される。
2	成功	失敗	エラーファイル出力先ディレクトリにエラー内容の xml ファイルが出力される
3	失敗	-	エラーファイル出力先ディレクトリにエラー内容の xml ファイルが出力される

処理フロー



凡例



処理詳細

1. 引数チェック

　引数のディレクトリが存在しているかチェックと、引数の入力ファイル、ディレクトリが存在しているかのチェックを行う。ファイル、ディレクトリが存在しない場合は、システム内エラーで処理を終了する。

a. バリデータ初期化

　変換に先立ち、バリデータの初期化を行う。バリデータの初期化では、指定されたディレクトリ中のダブリンコア等の既存ボキャブラリの RDFS と、自治体行政情報流通連携基盤システムで使用するボキャブラリのディレクトリ中の RDFS を読み込み、RDFS で定義されているボキャブラリの URI と値の型の組のリストを作成する。またチェック対象外の URI のリストの読み込みを行う。

b. フォーマット変換

　指定されたデータの RDFXML への変換を行う。
変換に際して、元のファイルが CSV、SDF、TTL の場合は RDF に変換する。なお、この際にパラメタに指定されたドキュメントの Namespace を用いて、元ファイル中に記載されているファイルそのもののメタデータも RDFXML 内に記述する。

c. バリデート・型付

　「a.バリデータ初期化」で作成した語彙の URI と値の型の組のリストと、バリデート対象外の URI のリストを用いて、対象データ中の述語の URI が、リスト中に存在するか、もしくはチェック対象外の URI であるか調べる。
URI と値の型の組のリスト中に存在しない、かつチェック対象外ではない URI がある場合は、エラーファイルにその旨を出力し、オンライン処理時は画面へも出力する。

　また、RDFS での定義を元に出力の値の型付け行う。ただし、値が RDFS の定義に合致しない場合型付けを行わない。
また、RDFS の定義を元に出力の値のリソース化を行う。リソース化を行う際は、引数に指定されたリソース化用の URI と値を連結した文字列をリソースの URI し、値がリソース化不可の文字を含む場合は'_'で置き換える。

また、パラメタで値の型チェックを指定された場合は、値の型のチェックを行う。

実証時の値の型付け、リソース化、チェック対象は以下の通りである。

項番	RDFS(Turtle)上定義	内部定義	バリデート・型付け判定方法(Java)	値の型
1	rdfs:range xsd:float	浮動小数点	Double.parseDouble()で変換可能なこと	xsd:float
2	rdfs:range xsd:decimal	整数	Integer.parseInt()で変換可能なこと	xsd:int
3	rdfs:range xsd:int	整数	Integer.parseInt()で変換可能なこと	xsd:int
4	rdfs:range xsd:integer	整数	Integer.parseInt()で変換可能なこと	xsd:int
5	rdfs:range rdfs:Literal	文字列	判定を行わない(すべて真)	型なしリテラル
6	rdfs:range rdfs:dateTime	日時	判定を行わない(すべて真)	xsd:dateTime
7	rdfs:range rdfs:time	文字列	判定を行わない(すべて真)	型なしリテラル
8	rdfs:range rdfs:date	日	判定を行わない(すべて真)	xsd:date
9	rdfs:range　上記以外	リソース	判定を行わない(すべて真)	リソース
10	rdfs:range　定義なし バリデート対象外 URI	文字列	判定を行わない(すべて真)	型なしリテラル

d. 登録/更新

パラメタで指定されたディレクトリに、生成された RDF ファイルを配置する。

表 3-42 処理定義(R04)

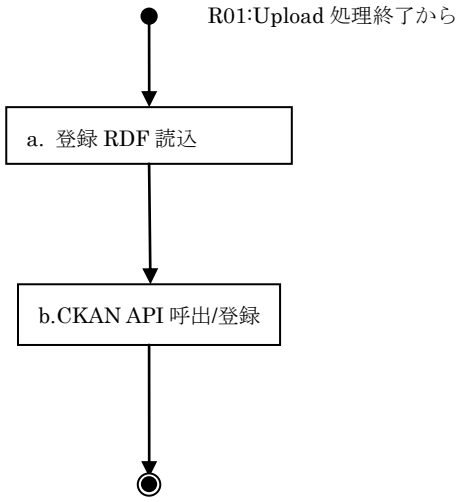
<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>処理名</td><td>CKAN 自動登録</td></tr>
<tr><td>処理 ID</td><td>R04</td></tr>
<tr><td>関連機能 ID</td><td>F05</td></tr>
<tr><td>開発言語</td><td>PHP</td></tr>
<tr><td>インターフェース</td><td>HTTP</td></tr>
<tr><td>ファイル名</td><td>/var/www/html/lodmanage/ckanupload.php</td></tr>
<tr><td>処理内容</td><td>CKAN 自動登録機能つきのアップロードページで、基盤側への登録が終了した後、パラメタで指定した CKAN の URL に対して、パラメタで指定した認証キーを使い、登録データのメタデータと Sparql Endpoint の URL などのシステムの規定値を登録する。</td></tr>
<tr><td>入力
(HTTP Post)</td><td>Multipart による Post
R01:Upload から追加したパラメタ、取り扱いが異なるパラメタを以下に示す。
<table>
<tr><th>項番</th><th>パラメタ名</th><th>内容</th><th>値</th></tr>
<tr><td>1</td><td>sync</td><td>アップロードするファイルの処理をオンラインで行うかバッチで行うか指定する。</td><td>CKAN 自動登録では、常にオンライン処理を行い、値は無視される。</td></tr>
<tr><td>2</td><td>ckanurl</td><td>登録先の CKAN の URL http://[ホスト名]/ で指定する。</td><td></td></tr>
<tr><td>3</td><td>ckankey</td><td>登録先の CKAN で使用するアカウントの認証キー。</td><td></td></tr>
</table></td></tr>
<tr><td>出力
(HTTP レスポンスコード)</td><td>R01:Upload から追加されたレスポンスコードのみ表示する
<table>
<tr><th>項番</th><th>内容</th><th>レスポンスコード</th><th>備考</th></tr>
<tr><td>1</td><td>CKAN 登録失敗</td><td>400</td><td></td></tr>
</table></td></tr>
<tr><td>出力</td><td>R01:Upload の出力に加えて、入力で指定された URL の CKAN に、登録したデータの情報が登録される。</td></tr>
</table>

CKAN 登録内容

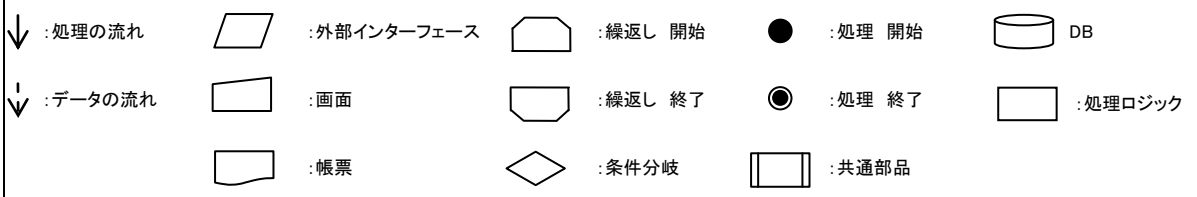
CKAN への登録内容を以下に示す。

項番	内容	データ側定義	CKAN 定義	備考
1	名称	データのファイル名のベース名を小文字にしたもの+登録日次(YYYYMMDDhhmm)	name	この値を用いて、CKAN 側 URL 作成を行う。
2	タイトル	ドキュメント自身の dc:title	title	CKAN 側で表示されるタイトル。
3	説明	ドキュメント自身の dc: description	notes	CKAN 側「概略」。
4	作者	ドキュメント自身の dc:creator	author	CKAN 側「作成者」。
5	RDF の URL	公開サーバの URL とデータのファイル名から公開サーバ上の URL を作成	resources の子要素 description:RDF resource_type:RDF format:RDF の url: の値	情報サービス開発事業者がこの値を元に RDF ファイルを取得する。(F11)
6	Sparql Endpoint URL	システム側の固定値（公開サーバの Sparql Endpoint）	resources の子要素 description:SPARQL Endpoint resource_type:api format:api/sparql の url: の値	情報サービス開発事業者がこの値を元に共通 API にアクセスする。(F12)

処理フロー



凡例



処理詳細

a. 登録 RDF 読み込み

前段の R01:Upload 処理で管理サーバ側に登録された RDF ファイルを読み込み、CKAN に登録するためのメタデータとして、RDF ファイルそのもののメタデータ(dc:title, dc:description, dc:creator) を取得する。該当のメタデータが存在しない場合は空文字で登録を行う。

b.　CKAN API 呼び出し/登録

入力で渡された接続先の CKAN に対して、同じく入力で渡された

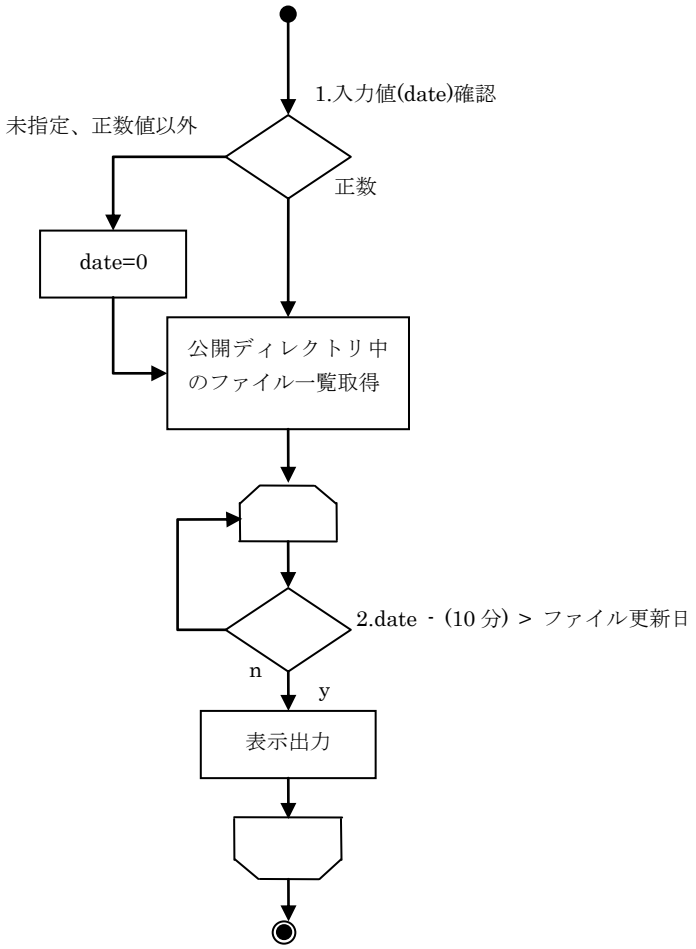
・CKAN の認証キー
・システム側で定数として持つ Sparql Endpoint の URL
・RDF の登録先の URL
・RDF から取得したメタデータ

を JSON 形式にして CKAN API にアクセスして登録を行う。
登録に失敗した場合はエラーのレスポンスコードを返す。

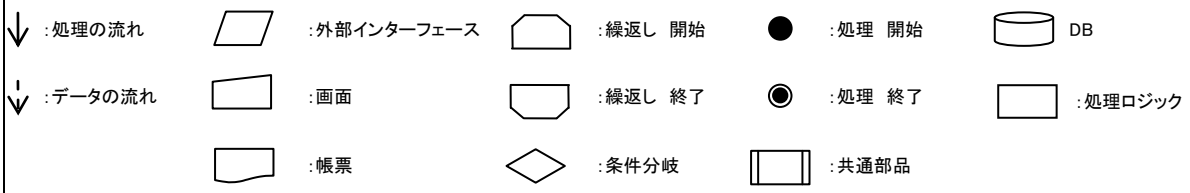
表 3-43 処理定義(R05)

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>処理名</td><td>管理サーバ更新データ一覧</td></tr>
<tr><td>処理 ID</td><td>R05</td></tr>
<tr><td>関連機能 ID</td><td>F04,F08,F09,F08,F07,F10,F05,F15</td></tr>
<tr><td>開発言語</td><td>PHP</td></tr>
<tr><td>インターフェース</td><td>HTTP GET</td></tr>
<tr><td>ファイル名</td><td>/var/www/html/listrdf.php (RDF)
/var/www/html/listschema.php (RDF スキーマ)</td></tr>
<tr><td>処理内容</td><td>管理サーバ上に登録された RDF、RDF スキーマファイルの更新時刻と、入力して渡される公開サーバ側の前回更新時刻を比較して、管理サーバ上で更新されたファイルの一覧の HTML を返す。</td></tr>
<tr><td>入力
(HTTP GET クエリ)</td><td>HTTP の GET のクエリ
<table>
<tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr>
<tr><td>1</td><td>date</td><td>エポック秒を指定する。
指定のない場合は 0(=1970-01-01 T00:00:00)を指定したことになり、全 RDF ファイルの一覧を取得する。</td></tr>
</table></td></tr>
<tr><td>設定値</td><td>対象ディレクトリをスクリプト中に指定する。</td></tr>
<tr><td>出力
(HTTP)</td><td>「date で指定された更新時刻-(10 分)」より更新時刻が新しいファイルへのリンクが含まれた HTML ページが取得される。</td></tr>
</table>

処理フロー



凡例



処理詳細

1. 入力値確認
　公開サーバ側の更新時刻を取得し、
「未指定、または正数以外」の場合は 0 として取り扱う。
正数の場合は、公開ディレクトリのファイル一覧を取得。

2.　date - (10 分) > ファイル更新日
　公開サーバ側の更新ラグ等を考慮し、指定された日時から 10 分遡って更新されたデータを表示する。

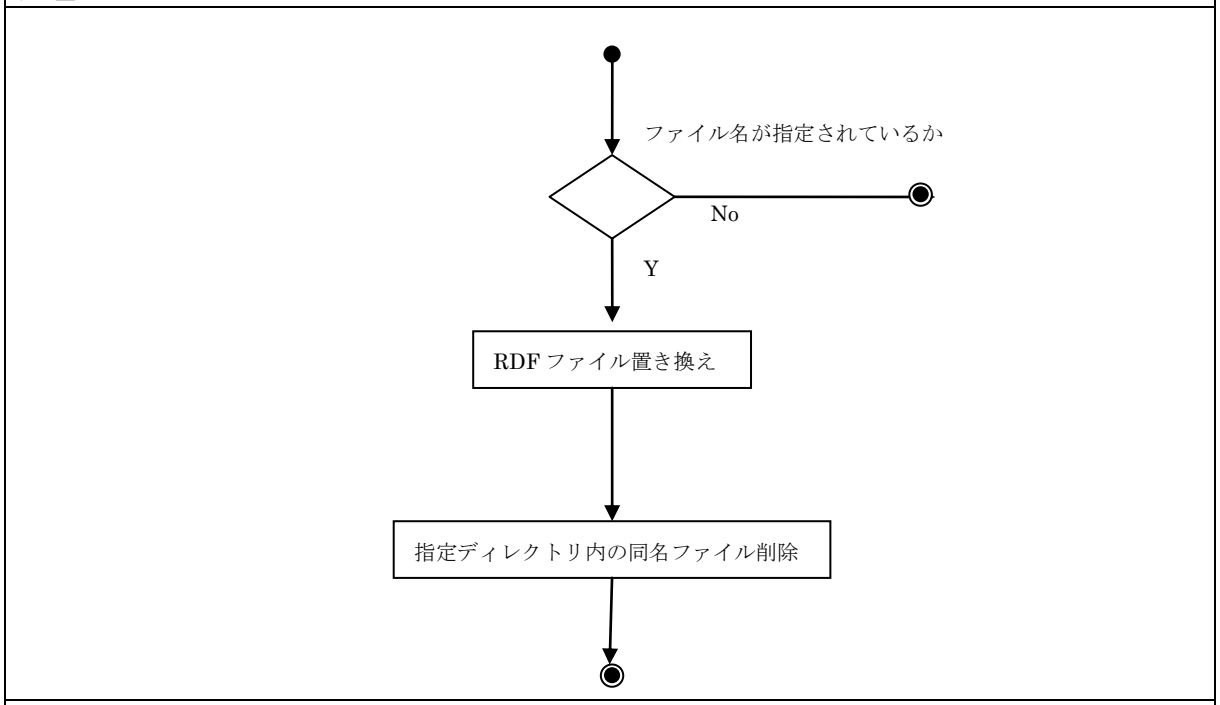
表 3-44 処理定義(R06)

<table>
<tr><th>項目</th><th>ポイント</th></tr>
<tr><td>処理名</td><td>データファイル削除</td></tr>
<tr><td>処理 ID</td><td>R06</td></tr>
<tr><td>関連機能 ID</td><td>F09</td></tr>
<tr><td>開発言語</td><td>PHP</td></tr>
<tr><td>インターフェース</td><td>HTTP</td></tr>
<tr><td>ファイル名</td><td>/var/www/html/lodmanage/del.php</td></tr>
<tr><td>処理内容</td><td>Web 画面からの操作で、指定されたデータファイルの削除を行う。
このとき、該当のファイルから生成されたファイル（RDF、エラーファイル）も併せて削除する。
なお、UI 上は、削除するファイルを選択して実行した後に、一度確認画面が入るが、その確認画面中に hidden で本機能のためのパラメタが記載されている。
また、公開サーバ上のデータの更新のために、中身のない空の RDF ファイルを管理サーバの公開ディレクトリに更新する。

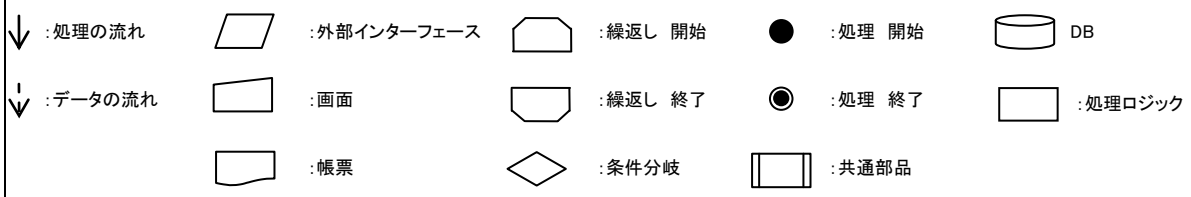
なお、後述する B02:公開サーバデータ更新（表 3-50）では、削除対象の RDF ファイルが、空の RDF ファイルで更新された場合、空の RDF ファイルでデータベースを更新することで、そのファイル名のデータをデータベース上から削除し、RDF ファイルは B02:公開サーバデータ更新で削除する。

また、CSRF 対策のため、トークンをパラメタに付与し、また確認画面が表示されてから実際に削除を行うまでの時間を 2 分に制限する。</td></tr>
<tr><td>入力(HTTP Post)</td><td><table><tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr><tr><td>1</td><td>filename</td><td>削除対象ファイル名のベース名</td></tr><tr><td>2</td><td>tk</td><td>トークン</td></tr></table></td></tr>
<tr><td>出力(HTTP レスポンスコード)</td><td>削除成功・失敗にかかわらずレスポンスコードは 200 となる。
成功、失敗の内容はレスポンスの HTML の画面上に表示される。</td></tr>
<tr><td>出力（ファイル）</td><td>指定されたファイルのベース名を持つファイルが、以下のディレクトリから削除される。
・アップロードディレクトリ先（バッチ、オンラインとも）
・エラーファイルディレクトリ
また、以下のディレクトリ中の、指定されたファイルのベース名を持つファイルが、空の RDF ファイルで上書きされる。
・生成物ディレクトリ</td></tr>
</table>

処理フロー



凡例



処理詳細

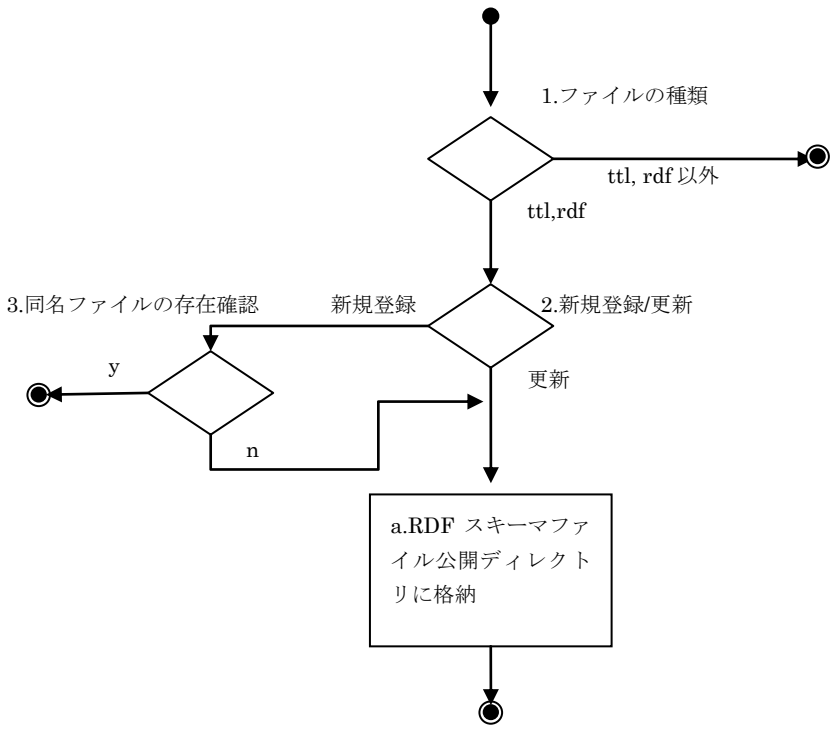
a. RDF ファイル置き換え
　削除対象の RDF ファイルを中身のない(Triple が 0 件)RDF ファイルで置き換える。

b. 指定ディレクトリ内の同名ファイル削除
　指定ディレクトリ内に同名のファイルが存在する場合、削除する。

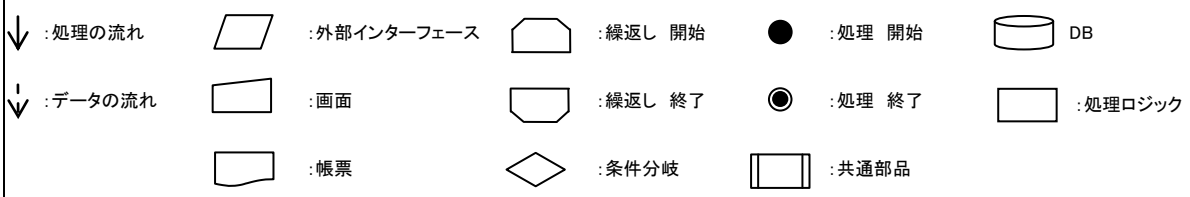
表 3-45 処理定義(R07)

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>処理名</td><td>RDF スキーマファイル登録更新</td></tr>
<tr><td>処理 ID</td><td>R07</td></tr>
<tr><td>関連機能 ID</td><td>F08,F07</td></tr>
<tr><td>開発言語</td><td>PHP</td></tr>
<tr><td>インターフェース</td><td>HTTP</td></tr>
<tr><td>ファイル名</td><td>/var/www/html/lodmanage/schemaupload.php (新規登録)
/var/www/html/lodmanage/schemaupload2.php (更新)</td></tr>
<tr><td>処理内容</td><td>UI からの HTTP のアップロード要求に対応し、アップロードした RDF スキーマファイルを所定の位置に格納する。
なお、データファイルのアップロードと異なり、文字コード変換は行わない。</td></tr>
<tr><td>入力(HTTP Post)</td><td>Multipart による Post
<table>
<tr><th>項番</th><th>パラメタ名</th><th>内容</th><th>値</th></tr>
<tr><td>1</td><td>upfile</td><td>Multipart で送るファイルデータ本体とファイル名</td><td></td></tr>
<tr><td>2</td><td>basename</td><td>更新対象のファイル名のベース</td><td>更新時のみ使用
更新時は省略不可。
登録時は無視される。</td></tr>
</table></td></tr>
<tr><td>出力(HTTP レスポンスコード)</td><td>
<table>
<tr><th>項番</th><th>内容</th><th>レスポンスコード</th><th>備考</th></tr>
<tr><td>1</td><td>正常終了</td><td>200</td><td></td></tr>
<tr><td>2</td><td>アップロードに失敗</td><td>500</td><td></td></tr>
<tr><td>3</td><td>システム内エラー</td><td>500</td><td></td></tr>
</table></td></tr>
<tr><td>出力（ファイル）</td><td>RDF スキーマファイルディレクトリにアップロードされたファイルが出力される。
また、更新の場合はパラメタでファイル名のベース名が指定されたものになる。拡張子はアップロードしたファイルのものが使用される。</td></tr>
</table>

処理フロー



凡例



処理詳細

1. ファイルの種類

　対象外のファイルのアップロードを防止するため、拡張子が ttl,rdf 以外のものがアップロードされた場合はメッセージを示し、処理を終了する。

2. 新規登録/更新

　新規登録と更新でアップロードするファイル名の取り扱いが異なるため処理を分ける。
なお、新規登録と更新の処理を行う PHP ファイル自体は別である。

3. 同名ファイルの存在確認

　新規登録の場合、アップロード先に同じベース名のファイルがある場合、既存のデータの上書きになるため、アップロードを行わずに終了する。
なお、更新の場合は、登録するファイル名は別途パラメタとして取得し、そのパラメタの値でファイル名を決定する。また、アップロード先の同じベース名のファイルの削除を行う。

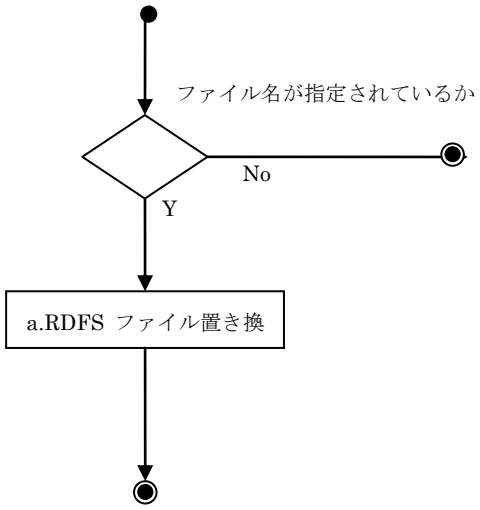
表 3-46 処理定義(R08)

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>処理名</td><td>RDF スキーマファイル削除</td></tr>
<tr><td>処理 ID</td><td>R08</td></tr>
<tr><td>関連機能 ID</td><td>F10</td></tr>
<tr><td>開発言語</td><td>PHP</td></tr>
<tr><td>インターフェース</td><td>HTTP</td></tr>
<tr><td>ファイル名</td><td>/var/www/html/lodmanage/schemadel.php</td></tr>
<tr><td>処理内容</td><td>Web 画面からの操作で、指定された RDF スキーマファイルの削除を行う。
UI 上は、削除するファイルを選択して実行した後に、一度確認画面が入るが、その確認画面中に hidden で本機能のためのパラメタが記載されている。
なお、公開サーバ上のデータの更新のために、中身のない空の RDF スキーマファイルを管理サーバの公開ディレクトリに更新し、実際にファイルの削除は行わない。

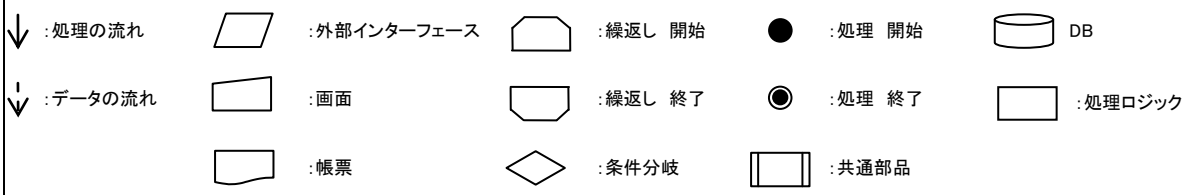
なお、後述する B02:公開サーバデータ更新（表 3-50）では、公開サーバ上の RDF スキーマファイルが空の RDF スキーマファイルで更新された場合、公開サーバ上の RDF スキーマファイルを削除する。

また、CSRF 対策のため、トークンをパラメタに付与し、また確認画面が表示されてから実際に削除を行うまでの時間を 2 分間に制限する。</td></tr>
<tr><td>入力(HTTP Post)</td><td><table><tr><th>項番</th><th>パラメタ名</th><th>内容</th></tr><tr><td>1</td><td>filename</td><td>削除対象ファイル名</td></tr><tr><td>2</td><td>tk</td><td>トークン</td></tr></table></td></tr>
<tr><td>出力(HTTP レスポンスコード)</td><td>削除成功・失敗にかかわらずレスポンスコードは 200 となる。
成功、失敗の内容はレスポンスの HTML の画面上に表示される。</td></tr>
<tr><td>出力（ファイル）</td><td>以下のディレクトリ中の、指定されたファイルのベース名を持つファイルが、空の RDF ファイルで上書きされる。
・生成物ディレクトリ</td></tr>
</table>

処理フロー



凡例



処理詳細

a. RDFS ファイル置き換え

　削除対象の RDFS ファイルを置き換える。

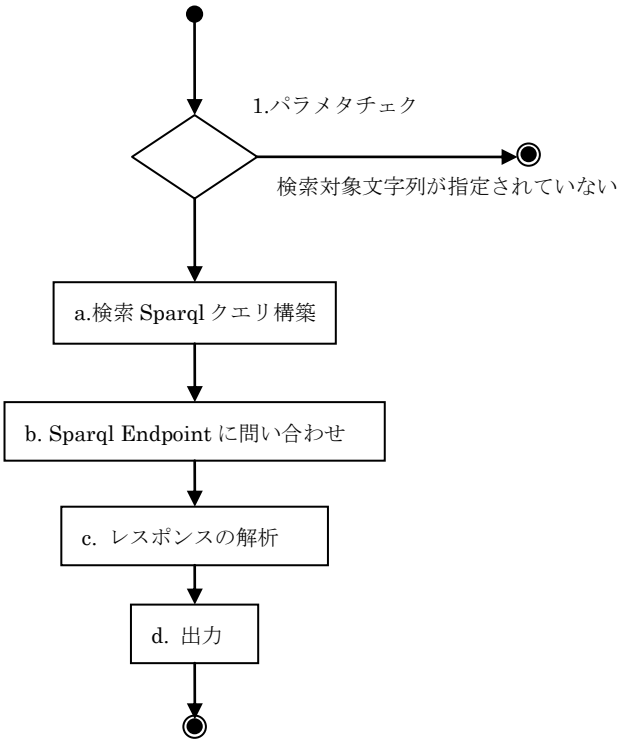
RDFXML 形式の場合中身のない(Triple が 0 件)RDFXML ファイルで置き換える。

TTL 形式の場合、ファイルサイズが 0 のファイルで置き換える。

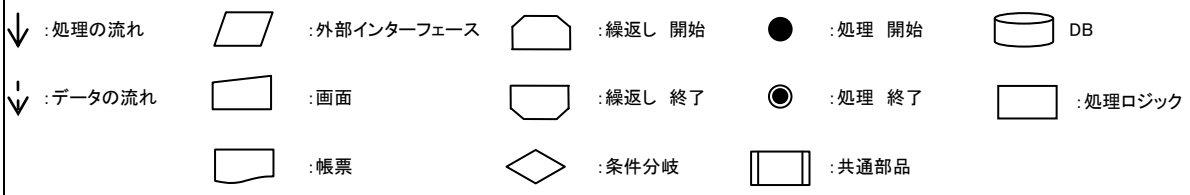
表 3-47 処理定義(R09)

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>処理名</td><td>検索</td></tr>
<tr><td>処理 ID</td><td>R09</td></tr>
<tr><td>関連機能 ID</td><td>F14</td></tr>
<tr><td>開発言語</td><td>PHP</td></tr>
<tr><td>インターフェース</td><td>HTTP</td></tr>
<tr><td>ファイル名</td><td>/var/www/html/lodmanage/searchresult.php</td></tr>
<tr><td>処理内容</td><td>UI からの検索要求に対して、公開サーバの Sparql Endpoint にクエリを投げ、条件に合ったデータが存在するデータファイルの一覧を返す。</td></tr>
<tr><td>入力(HTTP Post)</td><td>
<table>
<tr><th>項番</th><th>パラメタ名</th><th>内容</th><th>値</th></tr>
<tr><td>1</td><td>prop</td><td>検索対象の熟語を指定する。</td><td>検索対象の熟語の URI</td></tr>
<tr><td>2</td><td>value</td><td>検索対象の目的語を指定する。</td><td>検索対象文字列</td></tr>
<tr><td>3</td><td>type</td><td>検索の種類を指定する。</td><td>file: データファイルのプロパティを検索対象とする。
data:すべてのデータを検索対象とする。
val:述語を指定せずに目的語のみで検索する。
省略時/上記以外: data として扱う。</td></tr>
</table>
</td></tr>
<tr><td>出力(HTTP)</td><td>検索結果の画面が返る。詳細は画面定義側に記載する。</td></tr>
</table>

処理フロー



凡例



処理詳細

a. 検索 Sparql クエリ構築

公開サーバの Sparql Endpoint に対して、パラメタで指定された述語、目的語を持つグラフを取得するクエリを作成する。

b. Sparql Endpoint に問い合わせ

a.で作成したクエリを公開サーバの Sparql Endpoint に問い合わせる。
このとき、本機能が稼働する管理サーバからアクセス可能な公開サーバの Sparql Endpoint をスクリプト中に記載する必要がある。インターネット側のドメインや IP 等を指定する場合に、ネットワーク構成によっては管理サーバから公開サーバの公開側にアクセスできない場合がある事には留意する。
なお、目的語の検索対象はリテラルに限定される。（リソースは対象外）

c. レスポンスの解析

d. Sparql Endpoint に問い合わせで帰ってきた検索結果の xml データを読み込み、検索結果の Triple のグラフ名から、元のデータ名を取得し、検索結果とする。

3.3.5. バッチ処理設計

（1）　バッチ処理方式概要

自治体行政情報流通連携基盤システムでは、データサイズ等でオンライン処理が困難な場合や、UI を利用せずに、外部からデータを FTP 等で登録する場合の変換処理を行うための機能として、データのバッチ処理機能を提供する。

また、公開サーバ上で、定期的に管理サーバへアクセスし、管理サーバ上で更新されたファイルを取得し、そのファイルで公開サーバ上のファイル、データベースを更新するためのバッチ処理機能を提供する。

1）設計方針

管理サーバ、公開サーバとも、単一種類のバッチ処理である。

2）多重･同時実行制御

前のバッチが起動中の場合は起動せず、多重・同時実行は原則として行わない。ただし､30 分以上経過しても処理が終了しない場合は､終了を待たずに多重で実行する。

（30 分は実証時の設定値。自治体毎の要件により、スクリプト中の設定（多重起動までの分数）を変更する必要有。）

3）性能設計、負荷分散方式

管理サーバ、公開サーバとも単一種類のバッチ処理のみ実行され、また原則として多重起動は行わないため、性能設計、負荷分散については特に考慮しない。

（2）　バッチ処理一覧

以下に、バッチ処理の一覧を示す。

表 3-48　データ変換･登録ツール バッチ処理一覧

項番	要素	実装言語	処理 ID	関連機能 ID	サーバ
1	バッチ処理スクリプト	bash	B01	F04,F08	管理
2	公開サーバデータ更新	bash	B02	F04,F08,F09,F08,F07,F10,F05,F15	公開
3	変換/バリデート/登録更新	Java	B03	F04,F08,F02,F03,F05	管理

（3）　バッチ処理定義

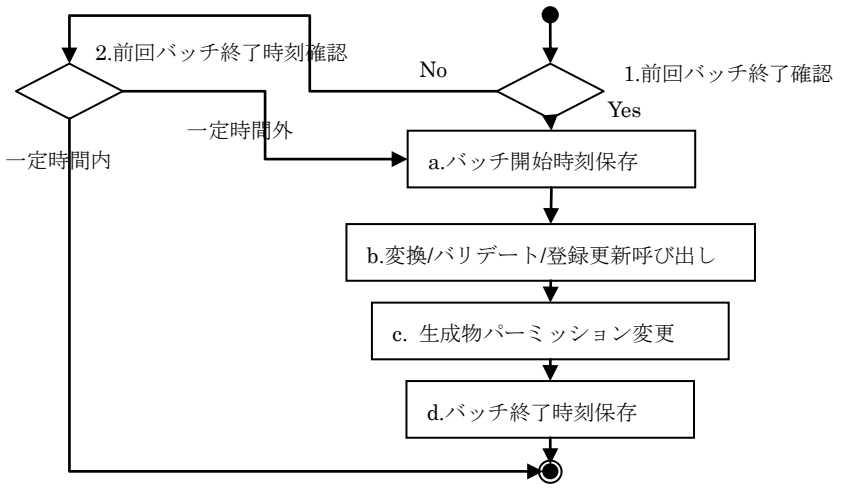
以下にバッチ処理定義を示す。

なお、ファイル名のパスは自治体行政情報流通連携基盤システムの実証時の導入構成である。

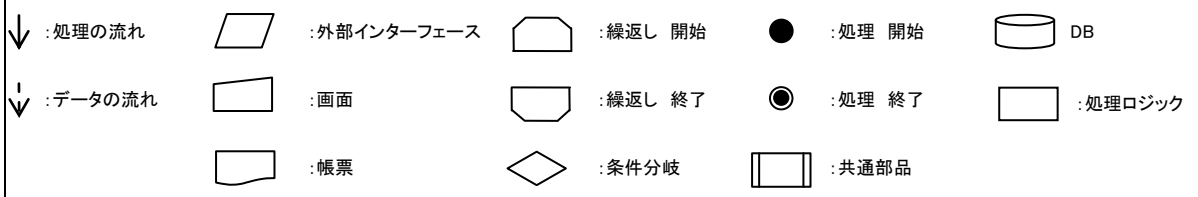
また、B01:バッチ処理スクリプト中で呼ばれる B03:変換/バリデート/登録更新は、R03:変換/バリデート/登録更新 と同一の実装内容である。

表 3-49 処理定義(B01)

項目	内容
処理名	バッチ処理スクリプト
処理 ID	B01
関連機能 ID	F04,F08,F02,F03
処理内容	データサイズ等でオンライン処理が困難なデータや、UI を利用せずにデータを登録する場合の変換処理を行う。
実行方法	cron により呼び出される。
実行タイミング	毎分
実装言語	bash
ファイル名	/home/adminuser/jrrktool/runconvert.sh
排他処理	前回のバッチ処理が終了していない場合は、バッチ処理はキャンセルされる。ただし、前回のバッチ処理終了後一定時間(30 分)経過している場合はバッチ処理を実行する。
エラー処理	変換・バリデートのエラーは変換/バリデート/登録更新側でエラー処理を行い、ログを出力する。 リカバリは行わず、次回以降の実行で再度実行する。
処理フロー	



凡例



処理詳細

1. 前回バッチ終了確認
　a.バッチ開始時刻保存と、d.バッチ終了時刻保存で作成されるファイルのタイムスタンプを比較し、前回のバッチが終了しているかどうかを確認する。

2. 前回バッチ開始時刻確認
　d.バッチ終了時刻保存で作成されるファイルのタイムスタンプを確認し、一定時間以上前であれば異常終了したと見なしてバッチを実行する。一定時間以内であればバッチ処理を終了する。

a. バッチ開始時刻保存
 touch でファイルを作成して、時刻を保存する。

b. 変換/バリデート/登録更新呼び出し
 変換/バリデート/登録更新の機能を呼び出す

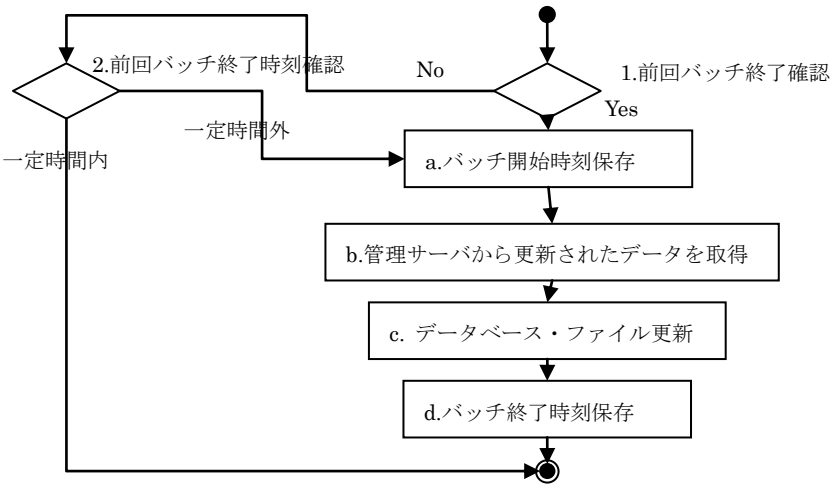
c. 生成物パーミッション変更
　本スクリプトは、一般ユーザ(adminuser)から呼び出されるため、一般ユーザのパーミッションで生成物が生成されるが、オンライン処理変換では Apache の実行ユーザで実行されるため、バリデート前の RDF ファイル、バリデート後の RDF ファイル、エラーファイルのパーミッションを書き込み可に変更し、それぞれの実行時に上書きし、削除時の問題が発生しないようにする。

d. バッチ終了時刻保存
　touch でファイルを作成して、時刻を保存する。

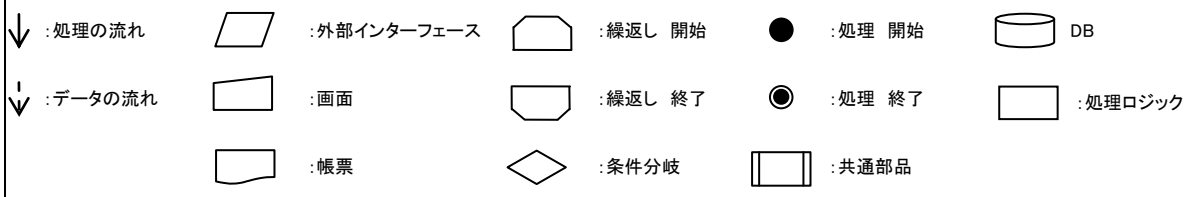
表 3-50 処理定義(B02)

項目	内容
処理名	公開サーバデータ更新
処理 ID	B02
関連機能 ID	F04,F08,F09,F08,F07,F10,F05,F15
処理内容	管理サーバ側の R05:管理サーバ更新データ一覧の URL にアクセスし、前回の B02:公開サーバデータ更新の起動時間より最近に更新されたデータを取得し、公開サーバ側のデータベース・ファイルを更新する。
実行方法	cron により呼び出される。またサーバ起動時に fuseki 起動直後に実行される。
実行タイミング	毎分とサーバ起動時
実装言語	bash
ファイル名	/home/loduser/jrrktool/updatetdb.sh
排他処理	前回のバッチ処理が終了していない場合は、バッチ処理はキャンセルされる。ただし、前回のバッチ処理終了後一定時間(30 分)経過している場合はバッチ処理を実行する。
エラー処理	データベース更新のエラーはログを出力して終了する。 リカバリは行わず、次回以降の実行で再度実行する。
備考	本処理の実装にあたり、Sparql Endpoint を構成するソフトウェアの Apache Jena Fuseki の更新ツールを呼び出している。そのため、Sparql Endpoint を構成するソフトウェアを変更する場合、本処理の実装を一部変更する必要がある。

処理フロー



凡例



処理詳細
1. 前回バッチ終了確認 　a.バッチ開始時刻保存と、d.バッチ終了時刻保存で作成されるファイルのタイムスタンプを比較し、前回のバッチが終了しているかどうかを確認する。 2. 前回バッチ開始時刻確認 　d.バッチ終了時刻保存で作成されるファイルのタイムスタンプを確認し、一定時間以上前であれば異常終了したと見なしてバッチを実行する。一定時間内であればバッチ処理を終了する。 a. バッチ開始時刻保存 　touch でファイルを作成して、時刻を保存する。 b. 管理サーバから更新されたデータを取得 　管理サーバへ HTTP でアクセスし、管理サーバ上の R05 管理サーバ更新データ一覧のページにアクセスする。このとき、前回のバッチ開始時刻を引数として渡すことで、前回のバッチ処理以降に管理サーバ上で更新されたファイルのリストを取得する。 　このリストを元に更新されたファイルを wget で取得する。 c. データベース、ファイル更新 　取得したファイルで、公開サーバ上のデータベースを更新する。また公開サーバ上のファイルも取得したファイルで更新する。 　このとき、管理サーバ上でファイルが削除された場合は、空の RDF ファイルを用いてそのファイルを更新する。(データベースへ空のファイルで更新することで、元のファイルから生成したデータが空のデータにより上書きされることによって削除されることになる。) また公開中のファイルに関しては、空の RDF ファイルを用いて更新した場合、公開しているファイルを削除する。 d. バッチ終了時刻保存 　touch でファイルを作成して、時刻を保存する。

表 3-51 処理定義(B03)

<table>
<tr><th>項目</th><th>内容</th></tr>
<tr><td>処理名</td><td>変換/バリデート/登録更新</td></tr>
<tr><td>処理 ID</td><td>B03</td></tr>
<tr><td>関連機能 ID</td><td>F04,F08,F02,F03,F05</td></tr>
<tr><td>実行タイミング</td><td>B02 バッチ処理スクリプト により呼ばれる</td></tr>
<tr><td>開発言語</td><td>Java</td></tr>
<tr><td>インターフェース</td><td>コマンドライン</td></tr>
<tr><td>ファイル名</td><td>/home/adminuser/jrrktool/jar/convtool.jar</td></tr>
<tr><td>備考</td><td>R03:変換/バリデート/登録更新 と同一実装。
処理内容、処理フローに差異がある。</td></tr>
<tr><td>処理内容</td><td>指定されたディレクトリ、ファイル名、パラメタから、データの RDF へのフォーマット変換を行い、その後、バリデート、登録・更新を行う。

バリデートは、ファイル中の述語の URI で指定された
・指定されたディレクトリ中のダブリンコア等の RDFS
・自治体行政情報流通連携基盤システムで使用するボキャブラリのディレクトリ中の RDFS
で定義されているボキャブラリの URI とのマッチングを行い、存在しない URI を使用している場合はエラーとする。

なお、バッチ処理では値の型チェックは行わないが、値の型付け、リソースへの変換は行う。

バリデート OK のファイルに関しては、R05: 管理サーバ更新データ一覧にて、公開サーバの所定ディレクトリにコピーされる。</td></tr>
<tr><td>入力
(コマンドライン)</td><td>本ツールの Java の実行オプションは以下の通り。

<table>
<tr><th>項番</th><th>内容</th><th>値</th></tr>
<tr><td>1</td><td>クラスパス</td><td>-cp "/home/adminuser/jrrktool/jar/*"

変換ツールの jar のパスを指定する。</td></tr>
<tr><td>2</td><td>ロガー設定</td><td>-Dlog4j.configuration=file:///home/adminuser/jrrktool/nulllog4j.properties

※jena のライブラリは Log4j を使用するが、データ変換・登録ツールでは使用しないため、初期化エラーのログが出力されてしまうので、その出力を抑止するための設定である。</td></tr>
<tr><td>3</td><td>実行クラス</td><td>net.msecnd.vo.jrrk.LodConvert</td></tr>
</table>
</td></tr>
</table>

実行時パラメタは以下の通り

項番	内容	値
1	変換対象ディレクトリ	ディレクトリが指定された場合はそのディレクトリ以下のファイルが変換対象となる。
2	バリデート前 RDF ファイル出力ディレクトリ	/var/www/html/lodmanage/tempout
3	最終出力先ディレクトリ	/var/www/html/rdf/
4	変換・バリデート失敗時のエラーファイル出力先ディレクトリ	/var/www/html/lodmanage/failed/
5	実証時に定義した RDF スキーマファイルの格納ディレクトリ	/var/www/html/schema/
6	ダブリンコア等の既存の RDF スキーマファイルの格納ディレクトリ	/var/www/html/lodmanage/commonschema/
7	バリデート対象外の URI を列挙したファイルを格納するディレクトリ (表 3-66 参照)	/var/www/html/lodmanage/ignorelist/
8	変換するドキュメントの Namespace ※ドキュメントそのもののメタデータを RDF で表現するために指定	http://[自治体等のドメイン]/[パス]
9	バリデータで値の型チェックを行うかの指定	（バッチでは、値の型チェックを行わないため true 以外が入る）
10	値のないタグを RDF に出力するかの指定	true:空タグを出力する。 上記以外：空タグを出力しない。
11	値をリソース化するときの URI の Prefix	http://[自治体等のドメイン]/[パス]

出力
(コマンドライン)

バッチ処理時

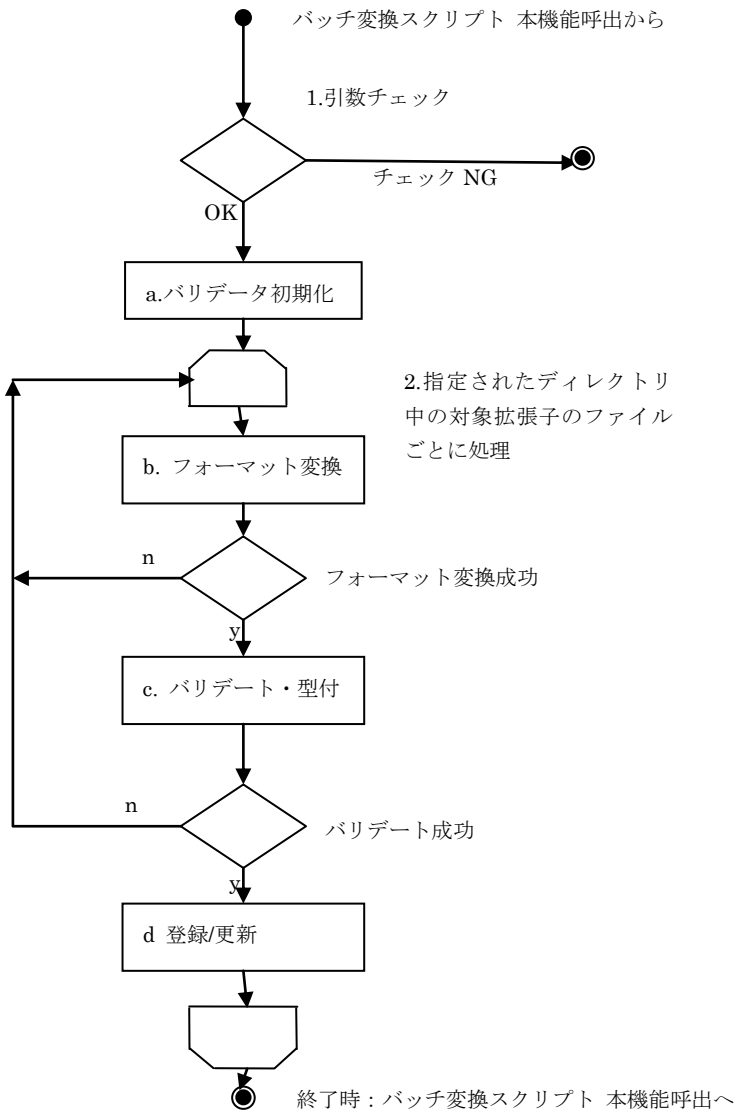
項番	内容	終了コード	備考
1	正常終了	0	
2	システム内エラー	-1	

個別のデータファイルの変換の失敗に関しては終了コードに示さない。

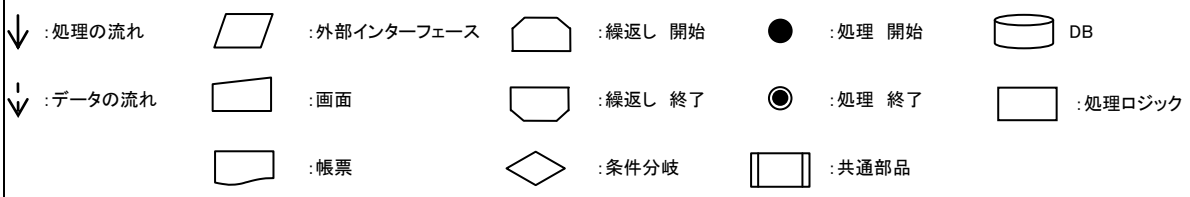
出力
（ファイル）

項番	変換	バリデート	出力
1	成功	成功	最終出力先ディレクトリに変換済みの RDF が出力される。
2	成功	失敗	エラーファイル出力先ディレクトリにエラー内容の xml ファイルが出力される
3	失敗	-	エラーファイル出力先ディレクトリにエラー内容の xml ファイルが出力される

処理フロー



凡例



処理詳細

1. 引数チェック

　引数のディレクトリが存在しているかのチェックと、引数の入力ファイル、ディレクトリが存在しているかのチェックを行う。ファイル、ディレクトリが存在しない場合は、システム内エラーで処理を終了する。

2. 指定されたディレクトリ中の対象拡張子を持つファイルごとに処理

　対象になるデータは、拡張子が txt(タブ区切り CSV),ttl,csv(カンマ区切り CSV), json(SDF ヘッダファイル)である。

a. バリデータ初期化

　変換に先立ち、バリデータの初期化を行う。バリデータの初期化では、指定されたディレクトリ中のダブリンコア等共通語彙の RDFS と、自治体行政情報流通連携基盤システムで使用する語彙のディレクトリ中の RDFS を読み込み、RDFS で定義されている語彙の URI と値の型の組のリストを作成する。またチェック対象外の URI のリストの読み込みを行う。

b. フォーマット変換

　指定されたデータの RDFXML への変換を行う。
変換に際して、元のファイルが CSV、SDF、TTL の場合は RDF に変換する。なお、この際にパラメタのドキュメントの Namespace を用いて、元ファイル中に記載されているファイルそのもののメタデータも RDFXML 内に記述する。

c. バリデート・型付

　「a.バリデータ初期化」で作成した語彙の URI と値の型の組のリストと、バリデート対象外の URI のリストを用いて、対象データ中の述語の URI が、リスト中に存在するか、もしくはチェック対象外の URI であるか調べる。
URI と値の型の組のリスト中に存在しない、かつチェック対象外ではない URI がある場合は、エラーファイルにその旨を出力し、オンライン処理時は画面へも出力する。

なお、値の型チェックはバッチ処理では行わないが、値の型付け、リソース化は行う。

d. 登録/更新

　パラメタで指定されたディレクトリに、生成された RDF ファイルを配置する。

3.3.6. メッセージ設計

ここでは、データ変換・登録ツールが UI 上で出力するメッセージの内容について述べる。

（1）　メッセージ定義一覧

以下に UI 上でのメッセージ一覧を示す。

表 3-52 メッセージ一覧

項番	表示内容	説明	遷移元画面 ID	表示画面 ID
1	同名ファイルが存在するためアップロードできません。	データファイル、スキーマファイルの新規登録時に、ファイルのベース名が同じものが存在する時に表示される	SC_05,SC_11, SC_C_01	SC_R_01 SC_C_02
2	[ファイル名] をアップロードしました。	データファイル、スキーマファイルの新規登録・更新成功時に表示される	SC_05,SC_07, SC_11,SC_14 SC_C_01	SC_R_01 SC_C_02
3	登録処理を行いました。	データファイルの変換/バリデートに成功したときに表示される	SC_05,SC_07, SC_11,SC_14, SC_17, SC_C_01	SC_R_01 SC_C_02
4	CKAN 登録を行いました。	CKAN 連携（自動登録）に成功したときに表示される	SC_C_01	SC_C_01
5	CKAN 登録に失敗しました。	CKAN 連携（自動登録）に失敗したときに表示される	SC_C_01	SC_C_01
6	フォーマット変換に失敗しました。	データファイルの変換に失敗したときに表示される。	SC_05,SC_07, SC_17, SC_C_01	SC_R_01
7	バリデートに失敗しました。	データファイルのバリデートに失敗したときに表示される。	SC_05,SC_07, SC_17, SC_C_01	SC_R_01
8	ファイルをアップロードできません。	サーバ側でアップロード処理に失敗した場合に表示される	SC_05,SC_07, SC_11,SC_14 SC_C_01	SC_R_01
9	ファイルが選択されていません。	データファイル、スキーマファイルのアップロード時に、ファイルを選択していない時に表示される。	SC_05,SC_07, SC_11,SC_14 SC_C_01	SC_R_01
10	サポート外のファイルです。	データファイル、スキーマファイルのアップロード時に、未対応の拡張子を持つファイルをアップロードすると表示される	SC_05,SC_07, SC_11,SC_14 SC_C_01	SC_R_01
11	ファイル[ファイル名]を削除しました。	指定したファイルが削除されたときに表示される	SC_06,SC_13	SC_R_01
12	ファイル[ファイル名]が見つかりません。	指定したファイルが存在しない場合に表示される	SC_06,SC_13	SC_R_01
13	検索結果がありません。	検索で結果が 0 件だったときに表示される	SC_15	SC_16
14	検索文字列を入力してください。	検索で検索文字列を指定しなかったときに表示される。	SC_15	SC_16
15	システムでエラーが発生しました。	定義外のエラーが発生したときに表示される	SC_05,SC_07, SC_11,SC_14, SC_17, SC_C_01	SC_R_01

3.3.7. データ変換・登録ツール単独での利用について

データ変換・登録ツールの各コンポーネントが動作するために必要な資材を以下に示す。

（コンポーネントについては表 3-15 参照）

表 3-53 各コンポーネントと必要な資材

項番	設置サーバ	コンポーネント名称	開発言語	動作に必要な資材
1	管理	UI	PHP	OS-64bit(Linux) HTTP サーバ(Apache) PHP
2	管理	Upload	PHP	OS-64bit(Linux) HTTP サーバ(Apache) PHP
3	管理	変換/バリデート/登録更新	Java	OS-64bit(Linux) Java 別資材（表 3-54、表 3-55 参照）
4	管理	オンライン処理スクリプト	bash	OS-64bit(Linux)
5	管理	バッチ処理スクリプト	bash	OS-64bit(Linux)
6	管理	管理サーバ更新データ一覧	bash	OS-64bit(Linux)
7	公開	公開サーバデータ更新	bash	OS-64bit(Linux) Apache Jena

「公開サーバデータ更新」については、データベース（TDB）への登録・更新・削除を行う処理があり、TDB を保持した Apache Jena が必須となる。

また、「変換/バリデート/登録更新」コンポーネントについては、データ形式変換（CSV から RDF、または SDF から RDF）の部分で、Apache Jena に同梱されている一部の資材、また別途フリーで取得可能な一部の資材が必要となる。

以下に、CSV から RDF に変換するために必要な資材と、SDF から RDF に変換するために必要な資材について述べる。

（1）　CSV から RDF に変換するために必要な資材

CSV から RDF に変換するために必要な資材を以下に示す。

表 3-54 CSV⇒RDF 変換に必要な資材一覧

項番	資材	備考
1	commons-codec-1.6.jar	Apache Jena 同梱
2	httpclient-4.2.3.jar	Apache Jena 同梱
3	httpcore-4.2.2.jar	Apache Jena 同梱
4	jcl-over-slf4j-1.6.4.jar	Apache Jena 同梱
5	jena-arq-2.11.0.jar	Apache Jena 同梱
6	jena-core-2.11.0.jar	Apache Jena 同梱
7	jena-iri-1.0.0.jar	Apache Jena 同梱
8	jena-sdb-1.4.0.jar	Apache Jena 同梱
9	jena-tdb-1.0.0.jar	Apache Jena 同梱
10	log4j-1.2.16.jar	Apache Jena 同梱
11	slf4j-api-1.6.4.jar	Apache Jena 同梱
12	slf4j-log4j12-1.6.4.jar	Apache Jena 同梱
13	xercesImpl-2.11.0.jar	Apache Jena 同梱
14	xml-apis-1.4.01.jar	Apache Jena 同梱

なお、項番 1～14 にある資材（Apache Jena 同梱）は、以下 URL から取得可能である。
https://jena.apache.org/download/index.cgi

（2）　SDF から RDF に変換するために必要な資材

SDF から RDF に変換するには、（1）で述べた資材に加え、以下の資材が必要となる。

表 3-55 SDF から RDF 変換に必要な資材一覧

項番	資材	備考
1	commons-logging-1.1.3.jar	別途サイトからダウンロードが必要な資材
2	fluent-hc-4.3.1.jar	別途サイトからダウンロードが必要な資材
3	gson-2.2.4.jar	別途サイトからダウンロードが必要な資材

なお、項番 1、2 にある資材は、以下 URL から取得可能である。
https://hc.apache.org/

項番 3 にある資材は、以下 URL から取得可能である。
https://code.google.com/p/google-gson/

3.4. その他設計

3.4.1. ファイル設計

本項では、自治体行政情報流通連携基盤システムが入出力として使用するファイルの内容について述べる。

（1）　登録ファイル

自治体行政情報流通連携基盤システムに対し、自治体職員や外部システム(CMS 等)から登録することが可能なファイルは、以下の 3 つである。

1)規定のフォーマットに加工されたファイル（CSV）
2)SDF（Simple Data Format）のファイル（CSV + JSON）
3)RDF スキーマ情報ファイル（RDFXML、Turtle）

以降で、1)～3）について説明する。

1）規定のフォーマットに加工されたファイル（CSV）

表形式のデータに対し、ボキャブラリ定義や、ボキャブラリ情報を持つサイトの URL 等を付与した CSV ファイル。詳細な加工手順やイメージは、『データ加工手順書』を参照のこと。

2）SDF（Simple Data Format）のファイル（CSV + JSON）

表形式のデータについて、データの定義情報（ヘッダ）を JSON ファイルで記載し、データの内容を CSV ファイルで記載する形式のファイル。SDF は、W3C の CSV on the Web Working Group Charter[5]において、RFC 4180 の新しいバージョンの規格が検討されている中の対象に挙がっており、W3C 標準となる動きがある。

[5] http://www.w3.org/2013/05/lcsv-charter/

以下に、SDF の JSON と CSV のイメージを示す。

例)SDF（JSON）のイメージ

```
{
  "name": "sdftest",
  "resources": [
    {
      "path": "sdftest.csv",
      "schema": {
        "fields": [
          {
            "uri": "http://www.example.org/id",
            "id": "place",
            "type": "string"
          },
          {
            "uri": "http://purl.org/dc/terms/title",
            "id": "title",
            "type": "string"
          },
          {
            "uri": "http://www.example.org/pos",
            "id": "pos",
            "type": "geopoint"
          }
        ]
      }
    }
  ]
}
```

データ名

定義対象のファイル

ボキャブラリ定義の URI

定義するボキャブラリ名

データ型

例) SDF（CSV）のイメージ

```
place,title,pos
A-1,"ガルリ　アノネコ","35.935645,136.0602959"
A-2,"DECO!","35.9356667,136.0603013"
A-3,"ボヌールヴィエント","35.935695,136.0603094"
A-4,"POISON APPLE","35.9357145,136.060312"
```

定義した各ボキャブラリ
（place,title,pos）
のデータ値を記載

3）RDF スキーマ情報ファイル（RDFXML、Turtle）

データに対して定義したボキャブラリ情報を持つファイル。自治体行政情報流通基盤システムでは、登録するデータ値のデータ型チェックに利用している。また、定義されたボキャブラリ情報を確認することが可能となる。

以下に、RDF スキーマ情報ファイルのイメージを示す。

例)RDF スキーマ情報ファイル（RDFXML）のイメージ

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:jrrk="http://purl.org/jrrk#"
  xmlns:schema="http://schema.org/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:pc="http://purl.org/procurement/public-contracts#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:qb="http://purl.org/linked-data/cube#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:dcterms="http://purl.org/dc/terms/">
```

定義している
ボキャブラリのサイト

各ボキャブラリの
定義情報を続けて記載

```
<rdf:Description rdf:about="http://purl.org/jrrk#ClimateStatistics">
  <rdf:type rdf:resource="rdfs:Class"/>
  <rdfs:label xml:lang="ja">気象統計</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://purl.org/jrrk#highestSnowfallPerDay">
  <rdf:type rdf:resource="rdfs:Property"/>
  <rdfs:subPropertyOf rdf:resource="qb:MeasureProperty"/>
  <rdfs:range rdf:resource="xsd:float"/>
  <rdfs:label xml:lang="ja">一日の最大降雪量</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://purl.org/jrrk#highestTemp">
  <rdf:type rdf:resource="rdfs:Property"/>
  <rdfs:subPropertyOf rdf:resource="qb:MeasureProperty"/>
  <rdfs:range rdf:resource="xsd:float"/>
  <rdfs:label xml:lang="ja">最高気温</rdfs:label>
</rdf:Description>
```

```
<rdf:Description rdf:about="http://purl.org/jrrk#lowestTemp">
    <rdf:type rdf:resource="rdfs:Property"/>
    <rdfs:subPropertyOf rdf:resource="qb:MeasureProperty"/>
    <rdfs:range rdf:resource="xsd:float"/>
    <rdfs:label xml:lang="ja">最低気温</rdfs:label>
</rdf:Description>
・・・・・・（ボキャブラリ定義情報が続く）
```

例)RDF スキーマ情報ファイル（Turtle）のイメージ

```
@prefix schema: <http://schema.org/> .
@prefix jrrk: <http://purl.org/jrrk#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix pc: <http://purl.org/procurement/public-contracts#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix qb: <http://purl.org/linked-data/cube#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

jrrk:ClimateStatistics a rdfs:Class ;
    rdfs:label "気象統計"@ja .

jrrk:highestSnowfallPerDay a rdfs:Property ;
    rdfs:subPropertyOf qb:MeasureProperty ;
    rdfs:range xsd:float ;
    rdfs:label "一日の最大降雪量"@ja .

jrrk:highestTemp a rdfs:Property ;
    rdfs:subPropertyOf qb:MeasureProperty ;
    rdfs:range xsd:float ;
    rdfs:label "最高気温"@ja .

jrrk:lowestTemp a rdfs:Property ;
    rdfs:subPropertyOf qb:MeasureProperty ;
    rdfs:range xsd:float ;
    rdfs:label "最低気温"@ja .
・・・・・・・・・・・（ボキャブラリ定義情報が続く）
```

定義している
ボキャブラリのサイト

各ボキャブラリの
定義情報を続けて
記載

（2）　ファイル一覧

入出力として使用するファイルの種類を以下に示す。

表 3-56　入出力ファイルの種類

項番	名称	拡張子	ディフォルトファイル名	備考
1	タブ区切りデータファイル(CSV)	txt	-	自治体職員が登録するデータファイル(CSV)
2	カンマ区切りデータファイル(CSV)	csv	-	自治体職員が登録するデータファイル(CSV)
3	データファイル(SDF)ヘッダ	json	-	自治体職員が登録するデータファイル(SDF)データのヘッダ情報
4	データファイル(SDF)データ	csv	-	自治体職員が登録するデータファイル(SDF)データの内容
5	RDF ファイル(RDFXML)	rdf	-	変換後の RDF ファイル
6	RDF ファイル(Turtle)	ttl	-	変換後の RDF ファイル
7	RDF スキーマファイル(RDFXML)	rdf	-	自治体職員が登録する RDF スキーマ
8	RDF スキーマファイル(Turtle)	ttl	-	自治体職員が登録する RDF スキーマ
9	エラー結果ファイル(出力のみ)	xml	[変換前のファイルベース名].xml	登録・更新作業時のエラー内容を出力
10	バリデート非対象 URI リスト	txt	ignorelist.txt	バリデートの非対象としたい URI のリスト

（3）　ファイル定義

前述した各ファイルの定義を以下に示す。

表 3-57 ファイル定義：タブ区切りデータファイル(CSV)

項目	内容
名称	タブ区切りデータファイル(CSV)
拡張子	txt
内容	オープンデータコンソーシアム形式でヘッダ、ボキャブラリが付与されたタブ区切りの CSV ファイル
区切り文字	タブ
文字コード	システム内部では UTF-8 を前提とする。 ※UI 経由のアップロード時に文字コードを指定することで UTF-8 に変換されて格納される。

表 3-58 ファイル定義：カンマ区切りデータファイル(CSV)

項目	内容
名称	カンマ区切りデータファイル(CSV)
拡張子	csv
内容	オープンデータコンソーシアム形式でヘッダ、ボキャブラリが付与されたカンマ区切りの CSV ファイル オープンデータコンソーシアム形式の CSV かどうかの判定は、同名のデータファイル（SDF）ヘッダ（.json）が存在せず、先頭行が'@comment' か'@prefix' で始まるかどうかで判定するものとする。
区切り文字	カンマ。データ内にカンマが入る場合はデータ自体をダブルクオートで囲むこと。
文字コード	システム内部では UTF-8 を前提とする。 ※UI 経由のアップロード時に文字コードを指定することで UTF-8 に変換されて格納される。

表 3-59 ファイル定義：データファイル(SDF)ヘッダ

項目	内容
名称	データファイル(SDF)ヘッダ
拡張子	json
内容	SimpleDataFormat 形式のヘッダファイル。
文字コード	UTF-8

表 3-60 ファイル定義：データファイル(SDF)データ

項目	内容
名称	データファイル(SDF)データ
拡張子	csv
内容	カンマ区切りの CSV ファイル 同名のデータファイル（SDF）ヘッダがあり、 先頭行が'@comment'および@prefix' で始まらない。
区切り文字	カンマ
文字コード	UTF-8

表 3-61 ファイル定義：RDF ファイル(RDFXML)

項目	内容
名称	RDF ファイル(RDFXML)
拡張子	rdf
内容	RDFXML
文字コード	システム内部では UTF-8 を前提とする。 ※UI 経由のアップロード時に文字コードを指定することで UTF-8 に変換されて格納される。

表 3-62 ファイル定義：RDF ファイル(Turtle)

項目	内容
名称	RDF ファイル(Turtle)
拡張子	ttl
内容	Turtle 形式の RDF
文字コード	UTF-8

表 3-63 ファイル定義：RDF スキーマファイル(RDFXML)

項目	内容
名称	RDF スキーマファイル(RDFXML)
拡張子	rdf
内容	RDFXML
文字コード	UTF-8

表 3-64 ファイル定義：RDF スキーマファイル(Turtle)

項目	内容
名称	RDF スキーマファイル(Turtle)
拡張子	ttl
内容	TTL 形式の RDF
文字コード	UTF-8

表 3-65 ファイル定義：エラー結果ファイル

項目	内容
名称	エラー結果ファイル
拡張子	xml
内容	エラー内容が記載された xml ファイル
文字コード	UTF-8

表 3-66 ファイル定義：バリデート非対象 URI リスト

項目	内容
名称	バリデート非対象 URI リスト
拡張子	txt
内容	バリデートを行わない URI を列挙したテキストファイル URI の末尾が*のものは、その URI のホストに対してバリデートを行なわない。 先頭が#で始まる行はコメント行
文字コード	UTF-8
記載例	http://www.example2.org/testdata/data1 http://www.example.org/* http://schema.org/* http://www.w3.org/* #http://www.example2.org/*

（4）　ファイル命名規則

自治体職員が登録するデータファイルの命名規則を以下に示す。

- 漢字ファイル名は使用不可
- ファイル名は半角英数と半角アンダーバー(_)と半角ハイフン(-),かつスペースを含まないものとする。

以下にファイル名とその扱いの例を示す。

city_data1.csv　→　使用可能
city data2.csv　→　使用不可(スペース混じりのため)
都市_data3.csv　→　使用不可(漢字が含まれているため)
CITY_data3.csv　→　使用不可(全角英数が含まれているため)

（5）　データベース

自治体行政情報流通連携基盤システムでは公開サーバ上の RDF の Triple を格納・検索するためのデータベース(RDF Store)として、Apache jena プロジェクトの TDB を使用する。

TDB は以下の特徴を持つ。

- Apache jena fuseki との親和性が高い。
- メモリキャッシュを持ち、単独マシン上で稼働する高速な RDF Store です。
- 複数サーバを束ねたクラスタリンクを必要としないため、クラウド環境でのリソースのスケールアップ、スケールダウンの設計が容易である。

3.4.2. ログ管理設計

（1）　ログ管理方針

自治体行政情報流通連携基盤システムのログ管理方針については、導入自治体の運用ポリシーに従うものとする。

なお、以下については本実証で構築した環境についての説明となる。

また、apache ログ、システムログについては言及しない。

（2）　ログローテーション

導入自治体の運用ポリシーに従うものとするが、データ変換・登録ツール、標準 API とも、日次でログファイルを別ファイルに移してローテートすることを推奨する。

（3）　ログ削除

導入自治体の運用ポリシーに従うものとする。

（4）　ログバックアップ

導入自治体の運用ポリシーに従うものとする。

（5）　ログ管理対象一覧

ログ管理対象を以下に示す。

表 3-67 ログ管理対象一覧

項番	対象サーバ	要素	内容	ログファイル名
1	管理	データ変換・登録ツール	バッチ処理スクリプトログ	/var/log/jrrk/runconvert.[yyyymmdd].log
2	管理	データ変換・登録ツール	オンライン処理スクリプトログ	/var/log/jrrk/runconvertsync.[yyyymmdd].log ※同期更新が行われなかった日のログファイルは存在しない。
3	公開	データ変換・登録ツール	公開サーバ更新ログ	/var/log/jrrk/updatetdb.[yyyymmdd].log
4	公開	標準 API (fuseki)	アプリケーションログ	/var/log/fuseki/fuseki-log.[yyyy-mm-dd] ※当日のログは日付部分なし。

（6）　ログ出力内容

各ログの出力内容を以下に示す。

表 3-68 管理サーバ/データ変換・登録ツール/バッチ更新スクリプトログ (/var/log/jrrk/runconvert.[yyyymmdd].log)

項番	ログ内容	説明
1	[START] YYYY-MM-DD hh:mm:ss	更新処理の開始時刻を示す
2	[END] YYYY-MM-DD hh:mm:ss	更新処理の終了時刻を示す
3	[INFO] エラーファイルの方が新しいため変換を行いません。［ファイル名］	変換対象ファイルにエラーがあり、以前変換処理を行った時点のエラーファイルがあることを示す。
4	[INFO]RDF ファイルの方が新しいため変換を行いません。[ファイル名]	変換対象ファイルがすでに変換処理を行って変換済みのファイルがあることを示す。
5	[convert]Convert proc Done	変換処理が終了したことを示す。
6	[convert]Convert proc fail	変換処理が失敗したことを示す。
7	～	実際に変換を行った場合に変換内容のメッセージが表示される。
8	Cannot run multiple instances.	すでにバッチ更新スクリプトが実行中であることを示す。処理は行なわない。
9	Too old instance found.	指定時間(30 分)より以前に起動されたバッチ更新スクリプトが存在することを示す。処理を続行する。

表 3-69 管理サーバ/データ変換・登録ツール/オンライン更新スクリプトログ (/var/log/jrrk/runconvertsync.[yyyymmdd].log)

項番	ログ内容	説明
1	[START] YYYY-MM-DD hh:mm:ss	変換処理の開始時刻を示す
2	[END] YYYY-MM-DD hh:mm:ss	変換処理の終了時刻を示す
3	[convert]Convert proc Success	変換処理が成功したことを示す。
4	[convert]Convert proc fail	変換処理が失敗したことを示す。
5	[type] check value type: [true/false]	バリデータで値の型チェックを行うかどうかを示す。
6	～	実際に変換を行った場合に変換内容のメッセージが表示される。

表 3-70　公開サーバ/データ変換・登録ツール/公開サーバ更新ログ (/var/log/jrrk/updatetdb.[yyyymmdd].log)

項番	ログ内容	説明
1	[START] YYYY-MM-DD hh:mm:ss	変換処理の開始時刻を示す
2	[END] YYYY-MM-DD hh:mm:ss	変換処理の終了時刻を示す
3	[date] [Unix Epoch]	管理サーバへデータ取得のために使用する時刻の Unix Epoch を示す。
4	[rdf]copy [ファイル名]	管理サーバから取得したデータファイルを公開領域にコピーしたことを示す。
5	[rdf]delete [ファイル名]	削除対象データファイルを公開領域から削除したことを示す。
6	[rdf]TDB update [ファイル名]	[ファイル名]のファイルで TDB を更新したことを示す。
7	[rdf]TDB update fail. Exit. [ファイル名]	[ファイル名]のファイルで TDB を更新に失敗して、処理を終了したことを示す。
8	[schema]copy [ファイル名]	管理サーバから取得したスキーマファイルを公開領域にコピーしたことを示す。
9	[schema]delete [ファイル名]	削除対象スキーマファイルを公開領域から削除したことを示す。
10	Cannot run multiple instances.	すでに公開サーバ更新バッチが実行中であることを示す。処理は行なわない。
11	Too old instance found.	指定時間(30 分)より以前に起動された公開サーバ更新バッチが存在することを示す。処理を続行する。

表 3-71　公開サーバ/標準 API(Fuseki)/アプリケーションログ (/var/log/fuseki/fuseki-log.[yyyymmdd].log)

項番	ログ内容	説明
1	hh:mm:ss INFO Fuseki :: [message]	Fuseki のリクエスト処理の内容を示す。
2	hh:mm:ss INFO Config:: [message]	Fuseki 起動時の設定の内容を示す。
3	hh:mm:ss INFO Server:: [message]	Fuseki 起動時の設定の内容を示す。

3.4.3. ユーザ定義

自治体行政情報流通連携基盤システムで定義しているシステム側のユーザは以下の通りである。

なお自治体行政情報流通連携基盤システムでは、メンテナンスを行う管理者以外がシステムにログインする必要はないため、管理者権限を持たないログイン可能なユーザは存在しないものとする。

表 3-72 システム側ユーザ一覧

項番	対象サーバ	機能	ユーザ名	権限
1	管理	管理者	adminuser	sudo 可
2	公開	管理者	loduser	sudo 可

管理サーバ側の Web 画面へのログインユーザ（ファイルを登録する自治体職員のユーザ）は以下の通りである。以下は、実証時でのユーザとなるため、権限に差異はないが、実運用では権限ごとのユーザが必要と思われる。

表 3-73 管理サーバ 変換ツール ユーザ一覧

項番	対象サーバ	機能	ユーザ名	権限
1	管理	管理者	lodadm	-本実証でのユーザ

3.4.4. パスワードポリシー

自治体行政情報流通連携基盤システムでは固有のパスワードポリシーは持たず、また固有のパスワードポリシーに依存した実装ではない。パスワードポリシーは導入自治体の運用ポリシーに従うものとする。

4.　自治体行政情報流通連携基盤システムの構築･動作確認手順

4.1.　自治体行政情報流通連携基盤システムの構築手順

本項では、自治体行政情報流通連携基盤システムの構築手順を示す。
なお、前提として以下の点留意すること。

- OS、ネットワーク設定、セキュリティ設定については、自治体行政情報流通連携基盤システム固有の箇所以外の説明は省略する。導入自治体のポリシーに従うこと。
- ミドルウェアについては､導入パッケージによりディレクトリ構成やオプション等が異なるため要点のみの記載とする。

4.1.1. 公開サーバの構築

公開サーバの構築の流れの概略を以下に示す。

- OS のインストール（本書での解説は省略する。クラウド環境の場合は OS 展開済みのイメージを使用する場合もある）
- ミドルウェアの導入設定（本書での解説は省略する）
- Jena Fuseki のインストール
- 変換ツール導入
- 個別設定

（1）　サーバ構築

以下の設定でサーバを構築する。

表 4-1 サーバ構成(公開サーバ)

項番	要素	内容	備考
1	OS	64bit Linux	bash,wget が動作する必要がある。 apache で proxy を使用するため、SELinux の設定を変更もしくは無効化する必要がある。
2	開放ポート	80(HTTP)	80 以外のポートは外部に対して閉じるが、必要に応じて作業用の SSH のポートを開放する。 3030 に関しては、B02 公開サーバデータ更新が内部で localhost でのアクセスで使用するため、設定に留意する。
3	apache		mod_proxy,mod_rewrite が必要 自動起動は OFF にする
4	Java	Oracle Java7	
5	Ruby	Ruby 1.8 以上	ruby-devel　irb　も必要
6	ユーザ	loduser を作成	loduser が sudo 可能な設定にする。

公開サーバの構築は、ユーザ：loduser で作業する。

1）Apache Jena Fuseki のインストール

Jena のサイト(http://jena.apache.org/) から、jena-fuseki のバイナリを取得、解凍し、解凍してできるディレクトリを fuseki にリネームして、/usr/lib/ の下にコピーする。

続いて、以下のファイルのパーミッションを変更する。

表 4-2 Jena Fuseki 関連ディレクトリ(公開サーバ)

項番	パス	パーミッション	内容
1	/usr/lib/fuseki/s-*	755	ツール群
2	/usr/lib/fuseki/fuseki	755	本体
3	/usr/lib/fuseki/fuseki-server	755	本体

次に、サーバ起動時にデータの更新、fuseki の起動、apache の起動をするための設定を追加する。

/etc/rc.local の末尾に以下を追加する。

```
/home/loduser/jrrktool/runfuseki.sh
sleep 60
/home/loduser/jrrktool/updatetdb.sh

/etc/init.d/httpd start
```

2）データ変換・登録ツールの導入

以下のディレクトリを作成する。

表 4-3 変換ツール 関連ディレクトリ(公開サーバ)

項番	パス	パーミッション	内容
1	/home/loduser/work	644	作業用
2	/home/loduser/jrrktool/	777	データ変換・登録ツール格納先
3	/home/loduser/jrrktool/updatefile	777	データ変換・登録ツール作業ディレクトリ
4	/var/log/jrrk/	777	データ変換・登録ツールログ出力先
5	/var/log/fuseki	777	標準 API ログ出力先
6	/usr/lib/fuseki/DB2	777	データベースのデータ格納先

次に、B02:公開サーバデータ更新モジュール、設定ファイルを導入する。

表 4-4 スクリプト等（公開サーバ）

項番	パス	パーミッション	内容
1	/home/loduser/jrrktool/updatetdb.sh	755	B02 公開サーバデータ更新モジュール
2	/home/loduser/jrrktool/runfuseki.sh	755	fuseki の起動
3	/etc/httpd/conf.d/fuseki.conf ※パスは環境により異なる場合がある	644	Apache の設定 (fuseki への proxy) ※Sparql Endpoint を構成するソフトウェアを変更する場合は、変更後のソフトウェアの設定に従う
4	/usr/lib/fuseki/log4j.properties	644	fuseki ロガー設定
5	/usr/lib/fuseki/jrrk-tdb.ttl	644	fuseki 設定

次に、バッチ処理を cron に設定する。

表 4-5 cron 設定（公開サーバ）

項番	設定内容	変更内容
1	cron (crontab –e で設定)	MAILTO="" * * * * * /home/loduser/jrrktool/updatetdb.sh

3）個別設定

以下のファイルについて、各環境に合わせて修正する。

表 4-6 環境依存ファイル（公開サーバ）

項番	パス	変更内容
1	/home/loduser/jrrktool/updatetdb.sh	管理サーバの URL(wget の引数の 2 箇所) 公開サーバからアクセス可能な URL を指定する。

また、環境によっては、上記以外に以下のファイルの修正が必要な場合がある。

表 4-7 環境依存設定一覧（公開サーバ）

項番	設定内容	対象ファイル
1	公開サーバホスト名	/etc/httpd/conf/httpd.conf ただし、前段に負荷分散装置等を挟む場合は、その設定に従うこと。
2	管理サーバホスト名	/home/loduser/jrrktool/updatetdb.sh
3	管理サーバへの取得間隔	crontab –e で /home/loduser/jrrktool/updatetdb.sh の項目を編集する。
4	公開するデータの基底 URI	管理サーバ側で指定

上記の設定が完了したら、サーバを再起動する。

4.1.2. 管理サーバの構築

管理サーバの構築の流れの概略を以下に示す。

- OS のインストール（本書では省略。クラウド環境の場合は OS 展開済みのイメージを使用する場合もある）
- ミドルウェアの導入設定（本書では省略）
- 変換ツール導入
- 個別設定

（1）　サーバ構築

以下の設定でサーバを構築する。

表 4-8 サーバ構成（管理サーバ）

項番	要素	内容	備考
1	OS	64bit の Linux	bash が動作する必要がある。 PHP から Shell を呼び出している箇所があるので、SELinux の設定を無効化する必要がある。
2	開放ポート	80(HTTP) 443(HTTPS)	80,443 以外のポートは外部に対して閉じるが、必要に応じて作業用の SSH のポートを開放する。
3	apache		mod_headers, mod_ssl,mod_rewrite が必要 自動起動は ON にする
4	Java	Oracle 版 Java7	
5	PHP		php-mbstring も必要 導入後、以下を php.ini に設定 *expose_php=Off* *post_max_size = 20M* *upload_max_filesize = 20M* *default_charset = "UTF-8"* また、*error_reporting* も、 *error_reporting = E_ERROR* に変更するが、導入直後は障害切り分けのために変更しない。
6	ユーザ	adminuser を作成	adminuser が sudo 出来るようにする。

管理サーバの構築は、ユーザ：adminuser で作業する。
以下 sudo する必要のある箇所があるが、説明は省略する。

1）データ変換・登録ツールの導入

以下のディレクトリを作成する。

表 4-9　変換ツール関連ディレクトリ（管理サーバ）

項番	パス	パーミッション	内容
1	/home/adminuser/work	644	作業用
2	/home/ adminuser/jrrktool/	777	データ変換・登録ツール格納先
3	/var/log/jrrk/	777	データ変換・登録ツールログ出力先
4	/home/ adminuser /jrrktool/jar/	755	データ変換・登録ツールライブラリ

次に、データ変換・登録ツール、設定ファイルを導入する。

表 4-10　変換ツール導入（管理サーバ）

項番	パス	パーミッション	内容
1	/home/adminuser/jrrktool/runconvert.sh	755	データ変換（バッチ）
2	/home/adminuser/jrrktool/ runconvertsync.sh	755	データ変換（オンライン）
3	/home/adminuser/jrrktool/jar/*.jar	644	データ変換・登録ツールライブラリ
4	/etc/httpd/conf.d/manage.conf ※パスは環境により異なる場合がある	644	Apache の設定 (レスポンスヘッダ、ssl, mime 設定) 認証用のパスワードファイル等は別途作成する。
5	/var/www/html/*	ファイル：644 ディレクトリ：777	データ変換・登録ツール UI

次に、バッチ処理を cron に設定する。

表 4-11　cron 設定（管理サーバ）

項番	設定内容	変更内容
1	cron (crontab –e で設定)	MAILTO="" * * * * */home/adminuser/jrrktool/runconvert.sh

2）個別設定

以下のファイルを環境に合わせて修正する。

表 4-12 環境依存ファイル（管理サーバ）

項番	パス	変更内容
1	/var/www/html/lodmanage/searchresult.php	管理サーバからの公開サーバの Sparql Endpoint の URL ($endpoint) 生成する RDF の基底 URI※ ($ownns)
2	/home/adminuser/jrrktool/runconvert.sh	生成する RDF の基底 URI※ （Java を呼んでいる箇所の後半の URL）
3	/home/adminuser/jrrktool/ runconvertsync.sh	生成する RDF の基底 URI※ （Java を呼んでいる箇所の後半の URL）
4	/var/www/html/lodmanage/ckanupload.php	外部からの公開サーバの Sparql Endpoint の URL ($ckanSparqlEndpoint) 外部からの公開サーバの RDF 公開 URL ($openRdfUrlBase) 生成する RDF の基底 URI※ （$ownns）
5	/var/www/html/cmsupload.php	$allowipar (外部システム（CMS 等）サーバの IP)もしくは、 $allowhostar (外部システム（CMS 等）サーバのホスト名)

※ 生成する RDF の基底 URI は、データ設計時に定義するが、通常は導入自治体のドメインで、オープンデータ用に定義された URL になる。

なお、環境によっては、上記以外に以下のファイルの修正が必要な場合がある。

表 4-13　環境依存設定一覧（管理サーバ）

項番	設定内容	対象ファイル
1	公開サーバホスト名	/var/www/html/lodmanage/ckanupload.php（管理サーバから解決出来る URL で指定） /var/www/html/lodmanage/searchresult.php（外部から解決出来る URL で指定）
2	管理サーバホスト名	/etc/httpd/conf/httpd.conf
3	B01 バッチ変換スクリプトの実行間隔	crontab –e で /home/adminuser/jrrktool/runconvert.sh の項目を編集する。
4	公開するデータの基底 URI	/var/www/html/lodmanage/searchresult.php /var/www/html/lodmanage/ckanupload.php /home/adminuser/jrrktool/runconvert.sh /home/adminuser/jrrktool/runconvertsync.sh
5	アップロードの上限サイズ	/etc/php.ini の upload_max_filesize と post_max_size
6	外部システム（CMS 等）のアップロード元	/var/www/html/cmsupload.php
7	公開サーバの追加・削除時の設定	不要（管理サーバは公開サーバの情報を持たない。）

上記の設定終了後、サーバを再起動する。

4.2. 自治体行政情報流通連携基盤システムの動作確認手順

本項では、自治体行政情報流通連携基盤システムを構築した後に、正常に構築され、動作するかを確認するための手順について述べる。

4.2.1. 公開サーバの動作確認手順

1) HTTP サーバ/SPARQL-based Command の動作確認

管理サーバにまだデータファイルをアップロードしない時点で、以下の URL にアクセスする。なお以下の URL は SPARQL Based Command にアクセスして、公開サーバのデータベース上の RDF の Triple を 10 件表示するものである。

http://[公開サーバの公開側のホスト名]/ /data/sparql?
query=SELECT+*+WHERE+%7B+graph+%3Fg%7B+%3Fs+%3Fp+%3Fo+.+%7D%7DLIMIT+10&output=text
(実際には一行)

以下のように表示されれば正常に動作していることとなる。

```
-----------------
| s | p | o | g |
==================
-----------------
```

なお、管理サーバにすでにデータファイルを登録している場合、データファイルの中身が反映され、以下のような表示になる場合があるが、この場合も正常に動作している。（出力の右側と下側を省略）

```
-----------------------------------------------------------------------------------------
| s                                | p                                | o     | g
=========================================================================================
| <http://www.example.org/r3> | <http://www.example.org/c15> | "18" | <http://localh
```

4.2.2. 管理サーバの動作確認手順

公開サーバ動作確認後、

https://[管理サーバホスト名]/loduser/lodmanage/setupdate.php

にブラウザでアクセスし、apache の Basic 認証で設定したユーザ名、パスワードでログインする。

その後、RDF スキーマ登録画面でサンプルの RDFS ファイル(testschema.rdf)を登録する。

さらに、サンプルの CSV ファイル(test.csv)を、データ登録で同期処理にチェックを入れて登録する。

このとき、登録に成功すれば管理サーバ側は正常に動作している。

4.2.3. 公開サーバ･管理サーバの結合確認手順

管理サーバの動作確認手順終了後、以下の URL を開く。

http://[公開サーバの公開側のホスト名]/

data/sparql?query=SELECT++%3Ftitle+%0D%0AWHERE+%7B%0D%0A+GRAPH+%3Fg%0D%0A%7B%0D%0A+%3Chttp%3A%2F%2Fwww.example.org%2Fr2%3E+%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Felements%2F1.1%2Ftitle%3E+%3Ftitle%0D%0A%7D%0D%0A%7D%0D%0A&output=text

※上記は以下のクエリを URL エンコードしたもの。

```
SELECT   ?title
WHERE {
 GRAPH ?g
{
 <http://www.example.org/test> <http://purl.org/dc/elements/1.1/title> ?title
}}
```

以下が表示されれば問題ない。

```
| title |
===========
| "jrrk"   |
```

管理サーバで変換されたデータが、公開サーバ上のデータベースに格納され、そのデータが検索されたので、管理サーバと公開サーバで連係が行われいると確認できる。

また、公開サーバ上で、管理サーバで変換した RDF ファイル（http://[公開サーバの公開側のホスト名]/rdf/test.rdf)、管理サーバに登録した RDFS ファイル(http://[公開サーバの公開側のホスト名]/schema/testschema.rdf)が取得できれば連携が行われていると確認できる。

5. 本書のまとめ

本書は、「情報流通連携基盤の自治体行政情報における実証に係る請負」にて設計・構築を行った「自治体行政情報流通連携基盤システム」の目的、求められる要件、設計内容について説明してきた。本書によって、各自治体が独自に、かつ設計コストを抑えて「自治体行政情報流通連携基盤システム」を構築することが出来ることにより、自治体行政情報流通連携基盤システムが各自治体に普及展開され、自治体のオープンデータ推進に役立つことを願っている。