

地方自治体のオープンデータ公開プロセスを支援する アプリケーションの検討

渡邊 輔[†] 陳 思楠[†] 佐伯 幸郎^{††} 中村 匡秀^{†,†††}

[†] 神戸大学 〒657-8501 神戸市灘区六甲台町 1-1

^{†††} 理化学研究所・革新知能統合研究センター 〒103-0027 東京都中央区日本橋 1-4-1

^{††} 高知工科大学 〒782-8502 高知県香美市土佐山田町宮ノ口 185

E-mail: [†]tasuku@ws.cs.kobe-u.ac.jp, ^{††}chensinan@gold.kobe-u.ac.jp, ^{†††}saiki.sachio@kochi-tech.ac.jp,
^{††††}masa-n@cmds.kobe-u.ac.jp

あらまし 社会の情報化が進む中、様々なデータを活用した課題解決が期待されている。そのため、政府や自治体が保有するデータをオープンデータとして公開する動きが世界的に広がっている。しかし、人的リソースや担当者のスキルの問題から、自治体の規模によってオープンデータの公開状況には大きな差がある。また、公開されるオープンデータの標準化が進んでいないという課題も存在している。そこで本研究ではオープンデータの公開プロセスを3つに分割し、各プロセスに関わる担当者のスキルに合わせたデータ作成を支援するアプリケーションを提案する。自治体の職員が、システムを介して専門知識なしで標準化されたデータ形式を定義し、データを作成することができようシステム機能を検討した。最後に、システムを介したオープンデータの公開プロセスをプロトタイプを用いて検証を行った。

キーワード オープンデータ、地方自治体、データ標準化、データ検証

Study of Applications to Support the Open Data Publication Process of Local Governments

Tasuku WATANABE[†], Sinan CHEN[†], Sachio SAIKI^{††}, and Masahide NAKAMURA^{†,†††}

[†] Kobe University, Rokkodai-cho 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

^{†††} Riken AIP, 1-4-1 Nihon-bashi, Chuo-ku, Tokyo, 103-0027 Japan

^{††} Kochi University of Technology, 185 Tosayamadacho Miyanokuchi, Kami, Kochi, 782-8502 Japan

E-mail: [†]tasuku@ws.cs.kobe-u.ac.jp, ^{††}chensinan@gold.kobe-u.ac.jp, ^{†††}saiki.sachio@kochi-tech.ac.jp,
^{††††}masa-n@cmds.kobe-u.ac.jp

Abstract As the informatization of society progresses, it is expected that a variety of data will be used to solve problems. For this reason, the movement to release data held by governments and municipalities as open data is spreading worldwide. However, due to the problems of human resources and the skills of those in charge, the status of open data release varies greatly depending on the size of the municipality. In addition, there is a lack of standardization of open data to be released to the public. In this study, we propose an application that divides the process of open data release into three parts and supports data creation according to the skills of the personnel involved in each process. We have studied the functionality of the system so that local government officials can define standardized data formats and create data without specialized knowledge through the system. Finally, the process of publishing open data through the system was verified using a prototype.

Key words Open data, Local government, Data standardization, Data validation

1. はじめに

情報化社会が進行するなか、様々な場所に蓄えられたデー

タを活用することで社会を豊かにする方法が模索されている。2021年に施行された官民データ活用推進基本法[1]では、ネットワークを通じて流通する多様な情報を効果的に活用し、急速

な少子高齢化の進展への対応等の様々な課題を解決することを目指し、国や地方自治体に保有する官民データの活用を推進する施策を行うことを求めており、政府や地方自治体は**スマートシティ** [2] を推進する取り組みを行う担当課を設置するなど、データを活用したまちづくりに向けた取り組みが実際に進められている。そのような背景のもと、国や地方自治体が保有するデータを**オープンデータ** [3] として公開する動きが広がっている。オープンデータは営利、非営利目的問わずに二次利用可能、機械判別に適したデータ構造を持つ、無償利用可能などの特性から様々なステークホルダーによる利活用が期待され、実際にオープンデータを活用したサービスが提供されている。

しかし、オープンデータの公開には、公開に関わる担当者のスキルや時間の制約などの問題があり、公開されるデータの質や量に大きな差がある。これは、特に小規模な地方自治体で顕著な問題となっている。また、オープンデータの利活用の利便性を考えると、**データの標準化** [4] が必要である。標準化されていないデータは、データの利活用を阻害するだけでなく、データの品質を保証することが困難となるため、迅速な対応が求められるが、データの公開を担当する職員のデータの標準化に関する知見の不足などから思うように進んでいない。

そこで本研究では、オープンデータの公開プロセスに着目し、標準化されたオープンデータ公開を各担当者のスキルに合わせて支援するアプリケーションを提案する。このアプリケーションは、(S1) データチェック関数実装、(S2) データスキーマ登録、(S3) データセット作成の3つのステップから構成されており、各ステップにおいて、担当者を分割することで、スキルに合わせたデータ作成を支援する。S2、S3では、WebUI等の直感的な操作によるデータスキーマ登録及びデータ作成を提案しており、専門的な知識を有さない地方自治体の職員による、標準化されたオープンデータ作成が可能となる。なお本稿ではオープンデータのデータセットの列名やその値がどのような形式であるかを定義する情報のことをデータスキーマと呼ぶこととする。S1では技術者がデータをチェックする関数を実装し、システムに登録する。S2では、オープンデータ公開担当者が、標準的なデータ項目定義書を参考にデータスキーマをシステムに登録する。S3では、オープンデータ作成担当者が、登録されたデータスキーマをもとにシステム上で形式のチェックを行いながらデータセットを作成する。

また、本システムの3つのステップを介したデータ作成プロセスの実現性検証のため、プロトタイプシステムを実装し、自治体標準オープンデータセット [5] に準拠したデータセットを作成するテストを行った。その結果、提案した3ステップによるオープンデータの形式チェックが可能であることを確認した。

2. 準備

2.1 スマートシティとオープンデータ

AI、IoTをはじめとする各種技術開発の進展とともに、社会のデジタル化が急速に進展しつつある。この流れを受けて、政府及び地方自治体はデジタル技術をまちづくりに取り入れ、都市機能の最適化や市民生活の質の向上を目指すスマートシティ

の取り組みを進めている。その取り組みの一つとしてオープンデータの取り組みがある。政府や自治体が保有する公共データを公開し、データ利用を促進することで都市に新たな価値を生み出すことを目的としており、各団体はオープンデータ基本指針 [6] に従ってデータ公開を行っている。オープンデータ基本指針によると、オープンデータは

- (1) 営利、非営利目的問わずに二次利用可能であること
- (2) 機械判別に適していること
- (3) 無償で利用できること

を満たすものであり、誰でもデータを活用できる。

2.2 オープンデータの活用

オープンデータは官民間問わず様々な分野での活用が期待されている。例えば農業の分野では、農林水産消費安全技術センターが公開している**農薬データベース**を活用し、個人の勘や経験に頼らず、農薬の適正な使用を支援したり、正確な農作業の記録を支援するアプリケーションが開発されている [7] [8]。他にも、観光の分野では、地図や観光施設、公共施設等のオープンデータを活用して、イラストマップや古地図を見ながら街歩きができ、観光・グルメ・バス停などの情報が確認できる地図アプリケーションが開発されている [8] [9]。また、スマートシティ実現のための実証実験として、都市 OS [10] と呼ばれるデータ利活用基盤を用いてセンサーデータ等のリアルタイムと組み合わせ、オープンデータを活用する取り組みも行われている [11]。

2.3 オープンデータ公開の現状と課題

このようにオープンデータは様々な分野で活用されている一方で、オープンデータの公開については、いくつかの課題が存在する。デジタル庁による、地方自治体へのオープンデータの取り組みに関するアンケート結果 [12] によると、2020年の段階で、オープンデータをすでに公開している自治体は56.6%とおおよそ半数にとどまっており、地方自治体の規模ごとにみると、人口30万人以上の大規模都市(県も含む)で96.9%、20万人～30万人の中規模都市で95.2%、5万人～20万人の小規模都市で71.3%、5万人未満の市町村で41.9%と、規模が小さいほど公開率が低くなっている。また、オープンデータに取り組むにあたっての課題として、「オープンデータを担当する人的リソースがない」や「オープンデータの効果・メリット・ニーズが不明確」、「オープンデータにどう取り組んでよいかかわからない」、「オープンデータの利活用が進まない」などが多く挙げられている。これらのアンケート結果から、特に小規模な地方自治体で、オープンデータの公開には人的リソースや、専門知識の不足といった課題や、オープンデータの効果やメリットが不明確で活用が進んでいないという課題があることがわかる。

3. 提案手法

3.1 研究の目的とアプローチ

本研究では、2.3章で述べた、オープンデータの公開における人的リソースや、専門知識の不足といった課題や、オープンデータの活用が進んでいないという課題を解決するために、オープンデータ公開プロセスを支援するアプリケーションを提案する。具体的には、公開するオープンデータがどのような形式で

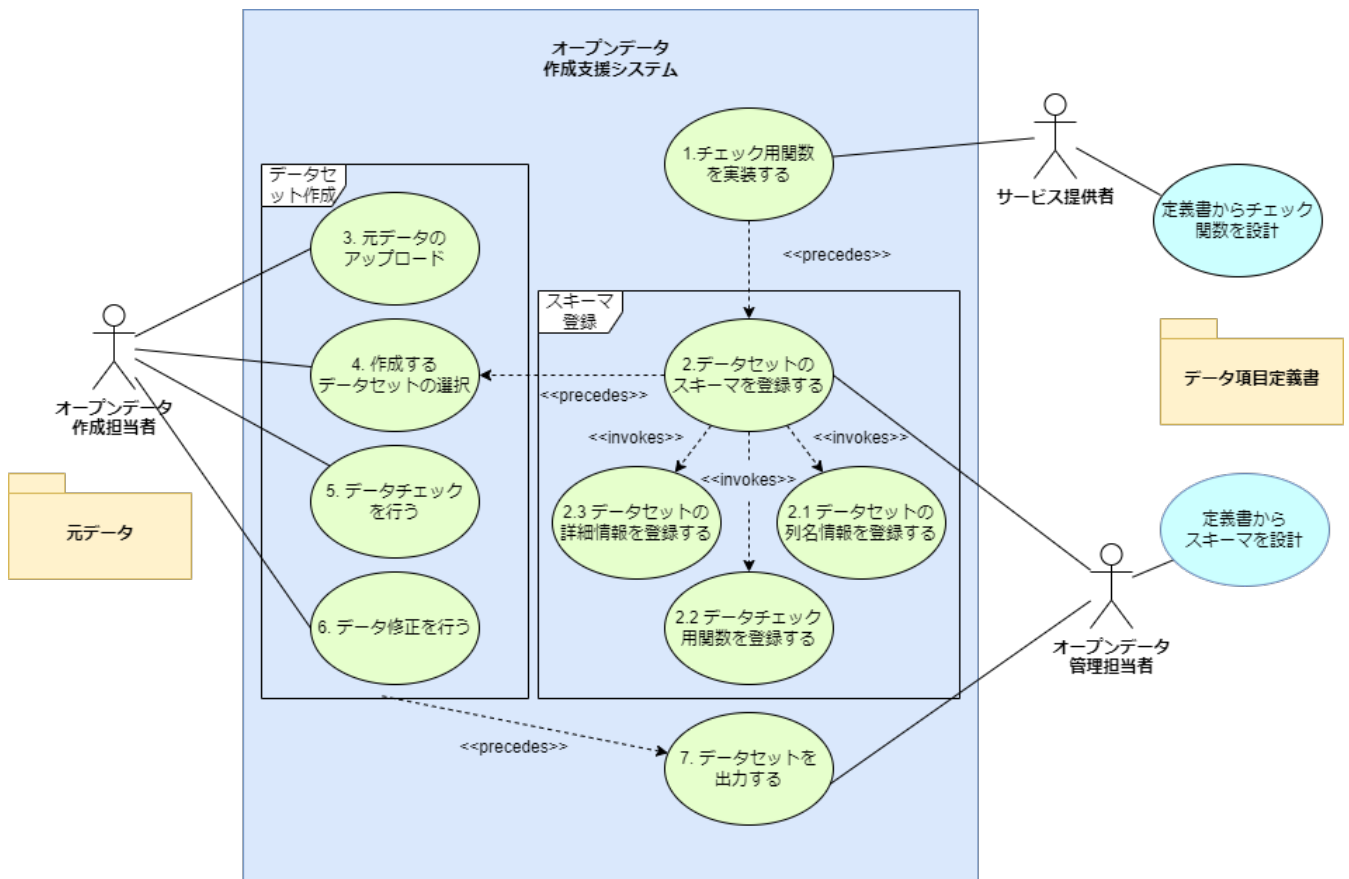


図1 提案システムのユースケース図

あるべきかを示すデータスキーマをシステム内で定義し、これから作成するデータがスキーマに従っているかをチェック・問題点を指摘することで、オープンデータ公開プロセスを支援する。データをチェックする関数を作成する「サービス提供者」、スキーマを定義する「オープンデータ管理担当者」、チェック結果をもとにデータを作成する「オープンデータ作成担当者」を分離することで、オープンデータ公開に関わる各担当者のスキルに合わせたデータ作成を支援することができる。特に、データの作成を行う者に関しては、専門知識なしに作業を行うことができるように設計することで、専門知識を持たない職員でも無理なく作業に参加でき、人的リソースの不足を解消することを目指している。なお、このシステムの利用者のスキルレベルとしては、以下のようなものを想定している。

サービス提供者 Python を用いたシステム開発の知見を持ち、データ項目定義書をプログラムに落とし込める

オープンデータ管理担当者 データ項目定義書を理解できる

オープンデータ作成担当者 Excel 等でデータの編集作業ができる

また、このシステムを利用し推奨される標準的なデータセットの形式(自治体標準オープンデータセット等)に準拠したデータを公開することによって、オープンデータを活用する際に、他のデータセットの連携やアプリケーションでの利用、ローコード・ノーコードツール[13]での利用等が容易になり、オープンデータの活用がより進められることが期待される。

3.2 ユースケース

図1に本システムのユースケース図を示す。各ユースケースの詳細については、以下の通りである。

UC1: チェック用関数を実装する

サービス提供者がデータ項目定義書を参考に、列データの形式が正しいかをチェックする関数を引数や戻り値などの情報を含めてシステムに登録する。

UC2: データセットのスキーマを作成する

オープンデータ管理担当者がデータ項目定義書を参考に、データセットのスキーマをシステムに登録する。登録する内容は、データセットの列名の情報や、データセットの各列に適用するチェック用関数の情報、データセットのメタデータである。

UC3: 元データのアップロード

オープンデータ作成担当者は、フォーマットチェックを行いたいデータをシステムにアップロードする。

UC4: 作成するデータセットの選択

オープンデータ作成担当者は、UC2で作成されたデータセットのスキーマからこれから作成するデータセットを選択する。

UC5: データチェックを行う

システムは、UC3でアップロードされたデータをUC4で選択されたデータセットのスキーマに従ってチェックを行い画面に表示する。

UC6: データ修正を行う

オープンデータ作成担当者は、UC5で表示されたチェック結

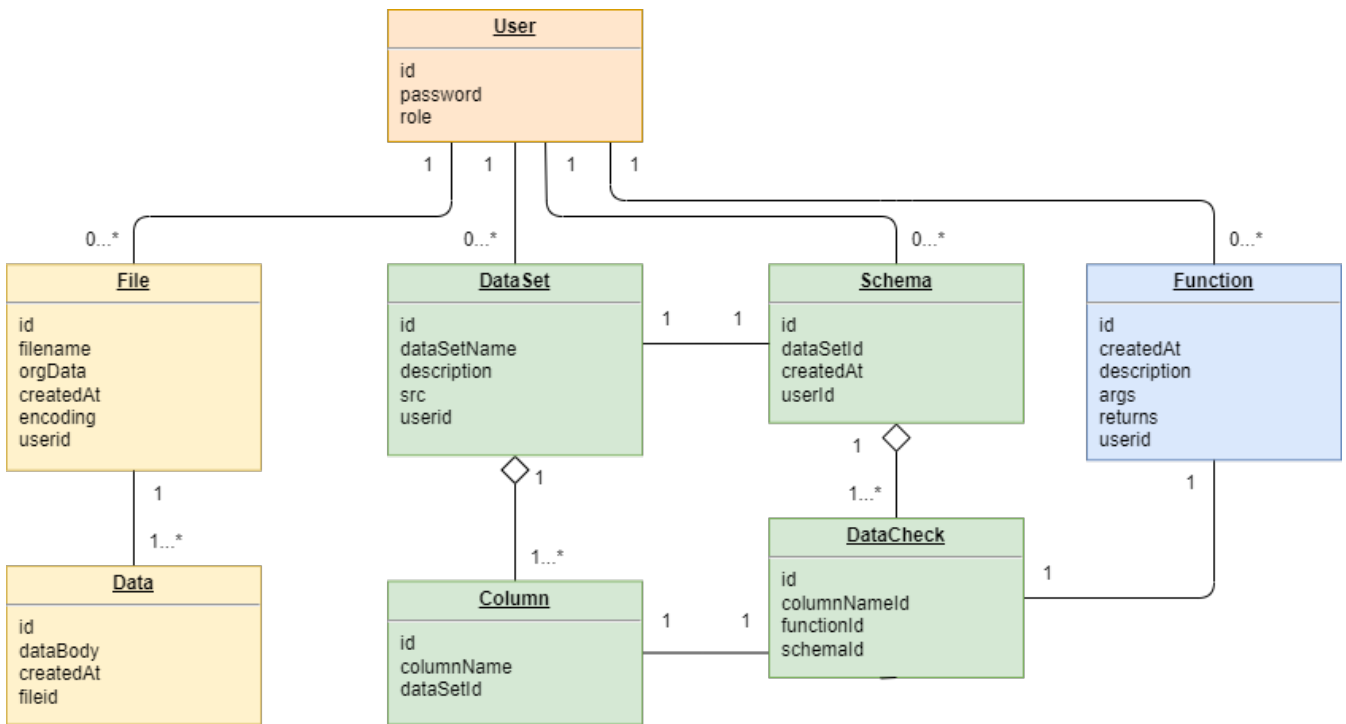


図2 提案システムのドメインモデル図

果をもとに、データを修正し、必要に応じて UC5 に戻る。

UC7: データセットを出力する

オープンデータ管理担当者は、UC5 で表示されたチェック結果が全て正しいことを確認した後、データセットを出力、公開する。

3.3 ドメインモデル

図2に本システムのドメインモデル図を示す。四角はエンティティを表し、エンティティ間の関連端にある数字は多重度を表す。User はシステムを利用するユーザの情報を持つテーブルであり、role によってオープンデータ作成担当者、オープンデータ管理担当者、サービス提供者それぞれの権限を持つ。Function はデータチェック用の関数の情報を持つテーブルである。DataSet はデータセットの情報を持つテーブルであり、複数の列データを持つ。Schema はデータセットのスキーマの情報を持つテーブルであり、一つの DataSet に結びつく。また、データセットのどの列に対してどの関数を適用するかを表すテーブル DataCheck を持つ。File はアップロードされたファイルの情報を持つテーブルであり、1つの File エンティティに対して、編集後の一時保存データエンティティである Data エンティティが複数結びつく。

3.4 全体アーキテクチャ

本システムの全体アーキテクチャを図3に示す。本システムによるオープンデータの作成は3つのステップから構成されている。1つ目では、サービス提供者によるデータチェック関数の実装を行う。2つ目では、オープンデータ管理担当者によるデータセットのスキーマの作成を行う。3つ目では、オープンデータ作成担当者によるデータ作成・データチェックを行う。

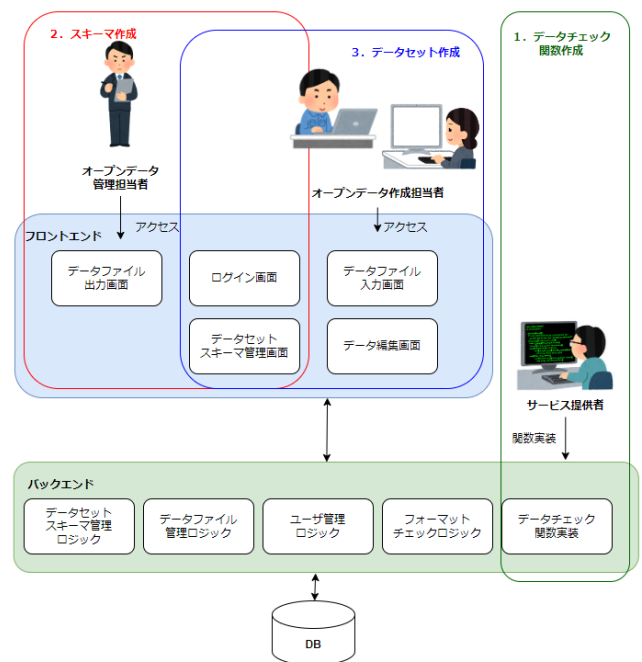


図3 提案システムの全体アーキテクチャ

3.5 機能概要

本節では、システムの3つのステップを実現するために必要な機能の詳細を述べる。

3.5.1 データチェック関数作成

このステップでは図4に示すように、データ項目定義書の説明・形式・制約等からルールを抽出し、データチェック関数を作成する。その後、関数の説明や引数、戻り値の型等の情報とともに関数をシステムに登録する。

データ項目定義書(例)

データ項目(指定緊急避難場所一覧) (注1)				
項目 No.	項目名	区分	説明	形式
2	ID	○	地方公共団体内で指定緊急避難場所(注2)が一層に決まるよう、IDを設定し、記載。	文字列(半角英数字)

① 定義書からルールを読み取る

- ・ ID列は値が半角英数字である
- ・ ID列は値が重複してはならない

② ルールを満たすことを確かめる関数を実装する

```
def function(列データ):
    // 半角英数字であることを確かめる処理
```

③ 関数を詳細情報とともにシステムに登録する

```
Id: func:1
説明: 半角英数字であることを確かめる
引数: pd.Series
戻り値: bool
```

図4 データチェック関数作成の流れ

3.5.2 スキーマ作成

このステップではデータセットのスキーマを作成する。その実現のために、データセットスキーマを登録・編集・削除する機能が必要となる。データセットスキーマの登録時には、データ項目定義書の情報を元に、データセットの名前、説明、データセットに含まれる列の名前や必須項目かの情報を登録する。また、データセットスキーマには、どの列に対してどのデータチェック関数を適用するかを示す情報も登録する。データセットスキーマの登録は、プログラミングの知識を持たないオープンデータ管理担当者でも行えるように、WebAPPのUI上で作成できるようにする必要がある。スキーマ作成画面のイメージを図5に示す。

3.5.3 データセット作成

このステップでは前ステップで作成されたデータセットのスキーマを選択し、そのスキーマに従ってデータセットを作成する。その実現のために、作成するデータセットのスキーマ選択機能、元データのアップロード機能、データチェックの実行機能、データの編集機能、データファイル出力機能といった機能が必要となる。いずれの機能も、プログラミングの知識を持たないオープンデータ作成担当者でも行えるように、WebAPPのUI上で作成できるようにする必要がある。また、データ作成の利便性のため、スキーマの選択時に詳細を表示する機能や、データチェックの結果、データのどの部分を修正すべきかを示す機能も必要となる。データチェック画面のイメージを図6に示す。

4. データチェックの実例

4.1 概要

3章で提案したシステムで行うデータチェックプロセスのうち重要な部分である、データチェック用ライブラリ作成、各データセットごとのスキーマ作成、データチェックの実行について、プロトタイプシステムを実装し、想定するオープンデータ作成プロセスに沿って実際にデータチェックを行う。プロトタイプは、Jupyter Notebook [14] 上で動作する Python スクリプトとして実装する。



図5 データスキーマ作成画面例



図6 データチェック画面例

4.2 想定するデータセット

今回は、デジタル庁が公開している自治体標準オープンデータセットのうち、「03. 指定緊急避難場所一覧」に準拠した指定緊急避難場所一覧データセットを作成することを想定する。データ項目定義書に記述されている各種要件に従うようにデータセットを作成することを目標とし、例えば、入力データが列「ID」にて、「半角英数字」であること、「重複していない」ことを満たしているかを確かめることができるシステムを構築できることをプロトタイプを用いて示す。

4.3 データチェック用ライブラリ作成

まず最初に、サービス提供者がデータチェック用ライブラリを作成する。具体的には、Pythonのライブラリである pandas [15] において列データを表す Series というオブジェクトを引数とし、その列内の値が各種条件を満たしているかを判定する関数を作成する。ここではデータ項目定義書を参考に以下のことを確かめる関数を作成した。

- ・ 値が半角英数であること
- ・ 値が全角カタカナであること
- ・ 値が電話番号フォーマットに従っていること (数字, -, +, (,), /のみで構成されていること)
- ・ 列内で値が重複していないこと
- ・ 配列を引数にとり、その配列内の値が列内の値と一致すること

ソースコード 1 データチェックライブラリのリスト

```

1 function_dict = {
2   "func:0001": {
3     "function": is_alphanumeric_with_row,
4     "args": [pd.Series],
5     "return": (bool, pd.Series),
6     "description": "シリーズの各値が英数字であることを確認します。"
7   },
8   ...
9 }

```

次に、関数をリスト化し、一意の id によって関数を呼び出せるように保存する。提案システムでは、データベースにリストを保存するが、プロトタイプでは、python の dict 型のオブジェクト [16] としてソースコード上に実装した。オブジェクトの例をソースコード 1 に示す。このように、関数をリスト化し、一意の id によって関数を呼び出せるように保存することで、データチェックの際に、データセットのスキーマに従って、必要な関数を呼び出すことができる。

4.4 各データセットごとのスキーマ作成

次に、オープンデータ管理担当者が、データセットごとに、データ項目定義書を参考にして、列名のスキーマ及び、データチェック関数のリストを作成する。列名のスキーマは、データセットの各列が定義書通りの名前になっているか、必須項目となっている列は存在するかを確かめるために用いられ、各列の名前や、一意の id、必須項目であるかを記述する。データチェック関数のリストは、データセットのどの列に対してどのチェック関数を適用するかを紐づけるために用いられ、対象の列 id と、データチェック用ライブラリの id、引数を記述する。実際の提案システムでは、列名のスキーマ及びデータチェック関数のリストは、WebAPP の UI 上で作成でき、専門知識なしに作成できることを想定しているが、プロトタイプでは、python の dict 型のオブジェクトとしてソースコード上に実装した。列名スキーマの例をソースコード 2 に、データチェック関数のリストの例をソースコード 3 に示す。ここでは、列名スキーマの例として、「03. 指定緊急避難場所一覧」の列名スキーマを作成し、また、データチェック関数として、以下の表 1 に示すようなチェックを行うこととする。

表 1 データチェック関数のリスト

対象の列	適用する関数	引数
ID	値が半角英数であることを確かめる	なし
ID	列内で値が重複していないことを確かめる	なし
名称_カナ	値が全角カタカナであることを確かめる	なし
電話番号	値が電話番号フォーマットに従っていることを確かめる	なし
災害種別_洪水	配列を引数にとり、その配列内の値が列内の値と一致することを確かめる	[1,NaN]

4.5 データチェックの実行

最後に、オープンデータ作成担当者が、各自で作成したデータを前手順で作成したスキーマに従ってチェックし、データを修正しながらデータセットを作成する。ここでは、三田市が公

開している三田市指定緊急避難場所一覧データセット [17] を用い、オープンデータ作成担当者が当該データを作成したものと想定して、これまで作成したスキーマに従ったデータチェック、データ修正のプロセスを示す。

まず最初に、データの読み込みを行い、続いて列名スキーマによる列名チェックを行う。列名チェックでは、必須となっている列が存在するか、列名が定義書通りの名前になっているかを確認する。列名スキーマから必須項目となっている列を取り出し、データセットに存在するかを確かめるコード、及び、データセットの列名が列名スキーマ内に存在するかを確かめるコードを用いて列名チェックを行った。その結果、「ID」、「名称_カナ」、「災害種別_洪水」など合計 12 の必須項目となっている列が存在しないことがわかった。また、データセットには「No」、「住所」の 2 つの列名スキーマと異なる列が存在することがわかった。その後、列名チェックの結果をもとに、列名が定義書通りになるように修正を行う。今回は一例として、「No」を「ID」に変更し、「災害種別_洪水」列を新たに追加するとい

ソースコード 2 列名のスキーマ

```

1 column_schema = {
2   "03. 指定緊急避難場所一覧": [
3     {
4       "column_name": "全国地方自治体コード",
5       "cid": 1,
6       "required": False,
7     },
8     {
9       "column_name": "ID",
10      "cid": 2,
11      "required": True,
12    },
13    ...
14  ]
15 }

```

ソースコード 3 データチェック関数のリスト

```

1 check_func_dict = {
2   "03. 指定緊急避難場所一覧": [
3     {
4       "target_id": 2,
5       "func": "func:0001",
6       "args": []
7     },
8     {
9       "target_id": 2,
10      "func": "func:0005",
11      "args": []
12    },
13    {
14      "target_id": 4,
15      "func": "func:0002",
16      "args": []
17    },
18    {
19      "target_id": 17,
20      "func": "func:0003",
21      "args": []
22    },
23    {
24      "target_id": 25,
25      "func": "func:0006",
26      "args": [[1, np.NaN]]
27    }
28  ]
29 }

```

ソースコード 4 データチェックの出力結果 (一部)

```
1 target_id: 17, func: func:0003, args: []
2 対象の列名: 電話番号
3 シリーズの各値が電話番号の形式であることを確認します。
4 結果:
5 (False, 47 NaN
6 48 NaN
7 53 NaN
8 dtype: object)
9 target_id: 25, func: func:0006, args: [[1, nan]]
10 対象の列名: 災害種別_洪水
11 シリーズの各値が指定された文字のみで構成されているかどうかを確認
12 します。
13 結果:
14 (False, 0 0
15 1 0
16 8 0
...)
```

う修正を行った。

次に、データチェック関数のリストに従って、データチェックを行う。データチェック関数のリストから、データセットの列に対して、適用する関数を取り出し、関数を実行する。今回は表 1 に示すようなチェックを行い、ソースコード 4 のような出力を得た。その結果によって、「電話番号」列の 47, 48, 53 行目に許可されていない値「NaN」が存在すること、「災害種別_洪水」列の多くの行に許可されていない値「0」が存在することがわかった。その後、データチェックの結果をもとに、データセットの値を修正する。今回の例では、「電話番号」列の 47, 48, 53 行目に電話番号を入力し、「災害種別_洪水」列の値「0」を「NaN」に変更するという修正を行った。

このようにしてデータチェックと修正を繰り返すことで、定義書に従ったデータセットを作成することができる。

5. まとめ

本稿では、オープンデータの公開における人的リソースや、専門知識の不足といった課題や、オープンデータの活用が進んでいないという課題を解決するために、オープンデータ公開プロセスを支援するアプリケーションを提案した。また、データチェックプロセスの検証のために、プロトタイプシステムを実装し、自治体標準オープンデータセットのうち、「03. 指定緊急避難場所一覧」に準拠した指定緊急避難場所一覧データセットの作成を想定した検証を行った。その結果、提案システムによるデータチェックプロセスの実行によって、定義書に従ったデータセットの作成を支援する機能の実装可能性が示された。

今後の展望として、本稿で提案したアプリケーションの実装を行うことが挙げられる。具体的には、スキーマやデータをシステムで管理する仕組みや、システムに登録されたスキーマに従ってデータをチェックできる仕組みの実装を進めていく。また、本稿で提案したアプリケーションの実装を行った後、実際に自治体標準オープンデータセットの作成を想定した検証を行うことで、本稿で提案したアプリケーションの有用性を検証することが必要であると考えられる。

謝辞 本研究の一部は JSPS 科研費 JP19H01138, JP20H05706, JP20H04014, JP20K11059, JP22H03699, JP19K02973, 若手研究

23K17006 の助成を受けて行われている。

文 献

- [1] “官民データ活用推進基本法 | e-gov 法令検索,” <https://elaws.e-gov.go.jp/document?lawid=428AC1000000103>. (Accessed on 1 August 2023).
- [2] B.N. Silva, M. Khan, and K. Han, “Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities,” *Sustainable Cities and Society*, vol.38, pp.697–713, 2018. <https://www.sciencedirect.com/science/article/pii/S2210670717311125>
- [3] 大向一輝, “日本におけるオープンデータの進展と展望,” *情報管理*, vol.56, no.7, pp.440–447, 2013.
- [4] 一輝大向, “オープンデータ活用: 1. オープンデータと linked open data,” *情報処理*, vol.54, no.12, pp.1204–1210, nov 2013.
- [5] “自治体標準オープンデータセット (正式版) | デジタル庁,” https://www.digital.go.jp/resources/open_data/municipal-standard-data-set-test/. (Accessed on 1 August 2023).
- [6] “オープンデータ基本指針 | 政府 cio ポータル,” <https://cio.go.jp/node/2357>. (Accessed on 1 August 2023).
- [7] “アグリノート | gap にも使える営農支援ツール 農業は記憶から記録へ,” <https://www.agri-note.jp/>. (Accessed on 24 July 2023).
- [8] “オープンデータ 100 | デジタル庁,” https://www.digital.go.jp/resources/data_case_study/. (Accessed on 1 August 2023).
- [9] “さばえぶらり,” <http://atr-c.jp/burari/product/oldmap/sabae.html>. (Accessed on 24 July 2023).
- [10] “Sip サイバー/アーキテクチャ構築及び実証研究の成果公表 7. 都市 os,” https://www8.cao.go.jp/cstp/stmain/a-whitepaper3_200331.pdf. (Accessed on 1 August 2023).
- [11] 萩原隼士, 森本章倫, “都市解析に資する都市 os の試験的な構築とその運用に関する一考察,” *都市計画報告集*, vol.20, no.2, pp.239–243, 2021.
- [12] “地方公共団体へのオープンデータ取組に関するアンケート結果 (令和 2 年度) | デジタル庁,” https://www.digital.go.jp/resources/data_questionnaire/. (Accessed on 29 July 2023).
- [13] F. Sufi, “Algorithms in low-code-no-code for research applications: A practical review,” *Algorithms*, vol.16, no.2, 2023. <https://www.mdpi.com/1999-4893/16/2/108>
- [14] “Jupyter,” <https://jupyter.org/>. (Accessed on 1 August 2023).
- [15] “pandas,” <https://pandas.pydata.org/>. (Accessed on 1 August 2023).
- [16] “辞書オブジェクト,” <https://docs.python.org/ja/3/c-api/dict.html>. (Accessed on 1 August 2023).
- [17] “三田市指定緊急避難場所一覧 | 三田市オープンデータカタログサイト,” https://data.bodik.jp/dataset/282197_kinkyuhinansho. (Accessed on 1 August 2023).