

SHACL (Shapes Constraint Language) による アプリケーションプロファイル記述の試み

An Attempt to Describe an Metadata Application Profile with SHACL (Shapes Constraint Language)

高久雅生^{1*} 江草由佳²
Masao Takaku¹ Yuka Egusa²

¹ 筑波大学

¹ University of Tsukuba

² 国立教育政策研究所

² National Institute for Educational Policy Research

Abstract: Linked Open Data 形式のメタデータ記述のスキーマ説明を行うアプリケーションプロファイル記述を容易に公開するための手法として、SHACL (Shapes Constraint Language) に基づく文書化とデータ検証を一つの形式で両立する手法を提案する。実際のデータセットとして、著者らが提供している教科書 LOD を例にとり、手法の詳細について報告する。

1 はじめに

Linked Open Data (LOD) を始めとするウェブ上での構造化データの活用が進展して久しい。公共団体における公共データ、図書館や博物館による書誌・所蔵情報、学術研究機関の研究データなど、さまざまな領域のデータが構造化され、LOD データとして活用できるようになっている。こうした構造化データの流通と交換がウェブ上で広く行われるようになるにつれ、次の課題はその利活用に移りつつある。特に、提供される構造化データがどのような内容と構造を持つかが分からなければ、具体的なデータの探索やブラウジングが行えず、活用することも難しい。

このような課題を解決するために、メタデータの領域で議論されてきたのがアプリケーションプロファイル (Application Profile) と呼ばれる方法論で、アプリケーションプロファイルとは、特定のデータセットのために使われる語彙や記述規則、構造などをプロファイルとして記述する [1]。例えば、Dublin Core Metadata Initiative (DCMI) が 2008 年に発表したシンガポール・フレームワークはそのようなアプリケーションプロファイル記述のための枠組み提案である [2]。シンガポールフレームワークは、機能要件文書 (Functional require-

ments), ドメインモデル文書 (Domain model), 記述セットプロファイル (Description Set Profile; DSP) に加えて、記述ガイドライン (Usage guidelines) の 4 点の文書から構成されると規定され、アプリケーションプロファイル記述のための構成要素が明示された。

具体的なアプリケーションプロファイル記述としては、欧州の文化財ポータル Europeana における Europeana Data Model (EDM) [3]、国立国会図書館による DC-NDL [4] などがある。

アプリケーションプロファイル記述の課題は、(a) 人間が読んでわかる文書記述能力や可読性の高さの側面、(b) 記述内容を手軽に編集して提供できる記述形式の側面、(c) 機械可読可能性を高めてデータ検証などに役立てる形式記述の側面など、セマンティクスとシンタクスの双方の観点からなる要件を満たしながら、それぞれのトレードオフを図る必要がある点にある。最近では、これらの要件を満たすために、YAML 形式による編集環境の提案 [5]、CSV からの形式記述による方式 [6] など、いくつかの新しい提案と実装が出てきているものの、アプリケーションプロファイルの記述能力とデータ検証可能性の高さと、記述の容易さを両立するための研究は少なく、課題は十分に解決されていない。

そこで本研究では、手軽に記述できる記述形式と人間が読んで分かる読解力という要件を満たすツールとして、基礎的なグラフ構造記述に SHACL (Shapes Con-

*連絡先: 筑波大学 図書館情報メディア系
茨城県つくば市春日 1-2
E-mail: masao@slis.tsukuba.ac.jp

straint Language) [7] を採用したうえで、SHACL への変換ツールと、SHACL データから人が読んで理解できるプロファイル記述への変換ツールの両者を組み合わせることにより、手軽にアプリケーションプロファイル記述が可能な環境を提供することを目指す。

2 提案手法

図 1 に提案手法の概要を示す。本手法では、プロファイル記述を Microsoft Excel におけるスプレッドシート内に行い、その内容記述をスクリプト `xlsx2shapes` により RDF/Turtle 形式の SHACL 表現に変換する。生成された SHACL データを元に、メタデータ記述の検証が行えるだけでなく、ウェブサイト生成スクリプト `ttl2html` が人間が読んでわかる形式のウェブ公開用文書に変換する。

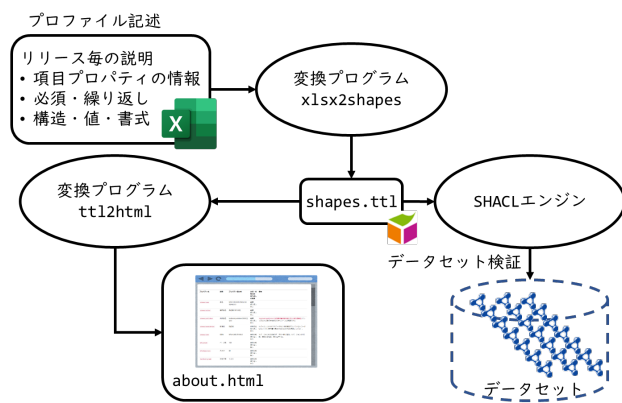


図 1: 提案手法の概要

以下では、著者らが提供している教科書 LOD データセット [8][9] におけるプロファイル記述を例に取りながら提案手法の詳細を述べる。

2.1 プロファイル記述

プロファイル記述は、記述対象クラスの実体インスタンスが持つ構造について、SHACL に基づいて記述することを想定する。ただし、SHACL による制約をそのまま RDF 形式で記述することは容易ではないため、記述を助ける目的で Microsoft Excel 形式での簡易記述を行うこととする（図 2）。

Excel 形式での記述に際しては、記述対象の各リソースが持つプロパティの種類と数、それぞれのプロパティの出現回数、書式、記載例等の情報をプロファイル情報として記述する。Excel ファイルの各シートには通常、対象となる 1 リソース分（ノードシェイプ; node shape）のプロファイル情報を記述する。これは SHACL におけ

る 1 つのノードシェイプについての情報を 1 つの Excel シートに書くこととなる。すなわち、Excel ファイルは、記述すべきリソース（ノードシェイプ）の種類の数に対応するシート数から構成されるファイルとなる。

図 2 に示すように、Excel 形式における記述では、1 行目は特別な行であり、先頭行最左列のセル（A1）には定義したいリソースの Shape クラスの URI を記述し、先頭行の他の列（B1 以降）はヘッダ行として、主にプロパティシェイプ（property shape）を示すヘッダを記載する。このヘッダには通常、プロパティシェイプにおける制約条件に対応するプロパティ名（`sh:name` や `sh:description`, `sh:minCount` など）を記述する。ただし、この際のプロパティ名には純粋な IRI だけでなく、言語別の説明文書の記述を書き分けられるよう、IRI の末尾に言語コードの指定を行えるようにしており、例えば `sh:name@ja` や `sh:description@en` という形で、末尾に「@言語コード」を付けた記述を許容している。

続く 2 行目では該当リソースのノードシェイプの条件を記載する。2 行目冒頭の A2 セルは通常 `sh:targetClass`（ノードシェイプのターゲットクラスを示す）を記載し、続く B2 セルに対応するクラス IRI を記述する。図 2 の例では、A2 セルに `sh:targetClass` とし、B2 セルに `https://w3id.org/jp-textbook/Textbook` と対象クラスの IRI を記述する。

3 行目以降は、当該リソースにおけるプロパティシェイプの具体的な制約条件を記述し、1 行が当該リソースの持つ各プロパティに対応する情報になる。なお、各列は、1 行目のヘッダ行に書いた IRI に対応した値を記述することにしており、例えば、図 2 の例では、C1 ヘッダセルには `sh:name@ja` とあるので、日本語のプロパティ名称を書く列となる。教科書リソースの例では「書名」や「教科」といったラベルを記述することとなる。他の列では、I1 ヘッダセルは `sh:minCount` とプロパティの繰り返しの最小値の条件が指定されているため、教科書リソースの書名プロパティは必須の制約条件であるのでその記述は「1」となる。

2.2 SHACL 生成

前節で述べた Excel 形式のプロファイル記述をその内容にあわせて SHACL 形式のシェイプグラフに変換する。変換プログラム `xlsx2shapes` はコマンドラインツールとして Excel ファイルを引数にとり実行すると RDF/Turtle 形式の SHACL シェイプ記述に変換し出力する（図 3）。

教科書 LOD における SHACL 記述の生成例の抜粋を図 3 に示し、以下、SHACL 記述の表現について説明する。なお、教科書リソースは多くのプロパティを持

	A	B	C	D	E	F	G	H	I	J	K
1	https://w3id.org/jp-textbook/TextbookShape	sh:path	sh:name@ja	sh:name@en	sh:description@ja	sh:description@en	skos:exam	sh:maxCoi	sh:minCoi	sh:class	sh:datatype
2	sh:targetClass	https://w3id.org/jp-textbook/Textbook									
3	sh:property	http://schema.org/name	書名	Title			NEW CRO	1	1		http://www.s
4	sh:property	http://schema.org/editor	編著者名	Editor(s)			森住衛 ほか	1	1		http://www.s
5	sh:property	http://schema.org/publisher	出版者名	Publisher	<code>textbook:pub	Link to the publisher	textbook:publisher/2		1	https://w3id.org/jp-tex	
6	sh:property	http://schema.org/bookEdition	版情報	Edition	※バージョン2.0までのデータ内には版(改訂版)					http://www.s	
7	sh:property	http://schema.org/isbn	ISBN	ISBN	※バージョン6.0以降のデータの一部に	978-4-385-70190-5				http://www.s	
8	sh:property	http://id.loc.gov/ontologies/bibframe/extent	ページ数	The total pages			120	1		http://www.s	
9	sh:property	http://id.loc.gov/ontologies/bibframe/dimensions	大きさ	Dimensions			B4	1		http://www.s	
10	sh:property	https://w3id.org/jp-textbook/grade	対象学年	Target grade			1, 2, 3	6		http://www.s	
11	sh:property	https://w3id.org/jp-textbook/authorizedYear	検定年	Authorized year			2001			http://www.s	
12	sh:property	https://w3id.org/jp-textbook/catalogue	掲載教科書目	Textbook cati	<code>textbook:cat	Link to the catalog	textbook:catalogue/s		1	https://w3id.org/jp-tex	
13	sh:property	https://w3id.org/jp-textbook/usageYear	使用年	Usage year			2005		1	http://www.s	
14	sh:property	https://w3id.org/jp-textbook/usageYearRange	使用年の区間	Range of Usa	"yyyy-yyyy" もしく	Either of "yyyy-yyyy"	2005-2005	1	1	http://www.s	
15	sh:property	https://w3id.org/jp-textbook/textbookSymbol	教科書記号	Textbook symbol			英語	1	1	http://www.s	
16	sh:property	https://w3id.org/jp-textbook/textbookNumber	教科書番号	Textbook number			904	1	1	http://www.s	
17	sh:property	https://w3id.org/jp-textbook/subject	権目(科目)	Subject	<code>textbook:cu	Link to the subject	textbook:c		1	https://w3id.org/jp-tex	
18	sh:property	https://w3id.org/jp-textbook/subjectArea	教科	Subject area	<code>textbook:cu	Link to the subject	textbook:c		1	https://w3id.org/jp-tex	
19	sh:property	https://w3id.org/jp-textbook/curriculum	対応する学習	Curriculum G	<code>textbook:cu	Link to the curricul	textbook:c		1	https://w3id.org/jp-tex	
20	sh:property	https://w3id.org/jp-textbook/school	学校種別	Type of scho	対応する学校種別を	Link to the type res	textbook:s		1	https://w3id.org/jp-tex	
21	sh:property	https://w3id.org/jp-textbook/item	所蔵情報	Holding item	<code>nier:record	Link to the blank node resource(s), which			1	http://id.loc.gov/ont	
22	sh:property	http://www.w3.org/2000/01/rdf-schema#seeAlso	関連リンク	関連リンク	繰り返しあり。<a	Repeatable. Links	http://id.ndl.go.jp/jpno/89034331				
23	sh:property	http://id.loc.gov/ontologies/bibframe/note	注記	Notes			2009年度より出版社を「日本文教出版株式			http://www.s	
24											
25											
26											
27											
28											
29											
30											
31											

図 2: Excel におけるプロファイル記述

ち、例えば、「書名」や「教科」(国語など)、その教科書の「使用年」、教科書を利用する「学校種別」(小学校など)などがある。図 3 内の 3~12 行目は「書名」、13~22 行目は「編著者」、23~33 行目は「出版社名」を示している。

図 3 の 1 行目および 2 行目においては、新たなノードシェイプ定義を示しており、ターゲットクラスは `<https://w3id.org/jp-textbook/Textbook>` というデータセット独自のクラス IRI を指定している。3~12 行目は教科書リソースの書名プロパティを示しており、4 行目で対応するプロパティパスが `<http://schema.org/name>` であることを示し、5~6 行目は、このプロパティの名称ラベルが「書名」(英語であれば Title)であることを示している。7 行目は、SKOS (Simple Knowledge Organization System) 語彙 [13] を借り、プロパティのデータ例として「NEW CROWN ENGLISH SERIES 3」という文字列を示している。8~9 行目は、教科書リソースの「書名」プロパティの出現回数として、最大 1、最小 1 として、書名プロパティは必ず 1 つ必要なので「必須」であるとし、かつ、繰り返しのないプロパティであるという制約を示している。10 行目は、教科書の「書名」は「文字列」リテラルとして表現し、データ型が文字列であることを示している。11 行目は、この教科書リソースのプロパティ群を表示させるときなどに、どのプロパティから順に表示させるかを表現しており、こ

の書名プロパティは 1 番目に表示されることを示している。

なお、8~9 行目の繰り返し回数や 10 行目のデータ型といった制約条件は、RDF データセットのグラフ検証を行う際に使用される。

2.3 HTML 生成

前節の要領で生成された RDF/Turtle 形式のファイルを入力として、人間が読んでわかるような各リソースおよび使用プロパティ語彙の説明として「表」形式の HTML 記述へと変換する。この SHACL 記述を HTML 文書に変換する機能は、Linked Data データセットの公開、提供を目的としたツール `tt12html` [10] の一機能として構築されている。

前節で述べた教科書 LOD を例とした SHACL データは、教科書 LOD のサイトで `about.html` [8] として公開している(図 4)。図 4 におけるプロパティ語彙を説明する表では、各リソースのプロパティ群に対して、各プロパティの名前、説明、プロパティ値の例、必須・省略の別、繰り返しの有無、備考という各項目について説明される。これらの各項目は、SHACL の対応する制約条件を自動で変換して生成する。例えば、「必須・省略の別」では、`sh:minCount` に 1 以上の数値が指定された場合、「必須」として示される。もし

```

1 <https://w3id.org/jp-textbook/TextbookShape> a sh:NodeShape;
2   sh:targetClass <https://w3id.org/jp-textbook/Textbook>;
3   sh:property [
4     sh:path <http://schema.org/name>;
5     sh:name "書名"@ja;
6     sh:name "Title"@en;
7     skos:example "NEW CROWN ENGLISH SERIES 3";
8     sh:maxCount 1;
9     sh:minCount 1;
10    sh:datatype <http://www.w3.org/2001/XMLSchema#string>;
11    sh:order 1
12  ];
13  sh:property [
14    sh:path <http://schema.org/editor>;
15    sh:name "編著者名"@ja;
16    sh:name "Editor(s)"@en;
17    skos:example "森住衛 ほか 29 名";
18    sh:maxCount 1;
19    sh:minCount 1;
20    sh:datatype <http://www.w3.org/2001/XMLSchema#string>;
21    sh:order 2
22  ];
23  sh:property [
24    sh:path <http://schema.org/publisher>;
25    sh:name "出版者名"@ja;
26    sh:name "Publisher"@en;
27    sh:description "<code>textbook:publisher/<var>掲載教科書目録の発行年</var>/<var>発行者略
28    ↳ 称</var></code>として示される発行者をあらわすリソースと関連付ける"@ja;
29    sh:description "Link to the publisher resource, represented as
30    ↳ <code>textbook:publisher/<var>catalogue publication year</var>/<var>publisher's
31    ↳ abbreviated name</var></code>"@en;
32    skos:example "textbook:publisher/2003/三省堂";
33    sh:minCount 1;
34    sh:datatype <https://w3id.org/jp-textbook/Publisher>;
35    sh:order 3
36  ];

```

図 3: プロファイル記述から生成された SHACL のシェイプ記述例 (抜粋)

くは、sh:minCount が指定されなかった場合、または sh:minCount に 0 が指定された場合、「省略可能」として示される。また、sh:maxCount が設定されていない、または 1 より大きな数値が指定された場合、「繰り返し有り」として示され、sh:maxCount が 1 以下の数値が指定された場合には「繰り返し無し」として示される。さらに、備考欄は sh:description プロパティの説明内容がそのまま示されることとなるため、図 3 内の 27～28 行目に見られるように、HTML 生成を意識して、SHACL 記述内でも HTML 記述を行うことができる。

また、前節で述べたように、各説明テキストには言語コードを付けることができるようにしたのは、静的サイト生成の際に、言語切り替えを行って、例えば、日英両言語でのアプリケーションプロファイル提供を行えるようにするためである。

3 考察

前節でも例示したように、提案手法は教科書 LOD の開発を通じて考案し、開発してきたものである。教科書 LOD ではプロファイル記述は 13 種類のノードシェイプに対して表現され、シェイプによる記述は 836 トリプルに渡る。教科書 LOD のデータセット全体は約 29 万トリプルから構成され、前述したアプリケーションプロファイル記述はデータセット全体をカバーしていることから、これを用いて全体に対するデータ検証を行うことができる。

提案手法のプロファイル記述の表現力は、基本的に教科書 LOD のような中小規模のデータセットで必要とされる構造記述や書式記述の大半をカバーするように実装してきており、多岐にわたる SHACL の表現力をそのまま実現する形となっている。構造記述としては、先述の sh:minCount や sh:maxCount を通じた必須・繰り返しの有無に加えて、ブランクノードを通じた構造や sh:or を通じた論理和集合による構造表現

教科書リソースが持つプロパティ

教科書リソースが持つプロパティを下表に示します

プロパティ名	説明	プロパティ値の例	必須・省略の別 繰り返し の有無	備考
<code>schema:name</code>	書名	NEW CROWN ENGLISH SERIES 3	必須 繰り返し 無し	
<code>schema:editor</code>	編著者名	森住衛 (ほか29名)	必須 繰り返し 無し	
<code>schema:publisher</code>	出版者名	textbook:publisher/2003/三省堂	必須 繰り返し 有り	<code>textbook:publisher/(掲載教科書目録の発行年)/(発行者略称)</code> として示される発行者をあらわすリソースと関連付ける
<code>schema:bookEdition</code>	版情報	改訂版	省略可能 繰り返し 有り	※バージョン2.0までのデータ内には版情報が入っているレコードはない (古い教科書の場合のために念のため用意している)。
<code>schema:isbn</code>	ISBN	978-4-385-70190-5	省略可能 繰り返し 有り	※バージョン6.0以降のデータの一部に追加。※バージョン6.1以降、複数ある場合、繰り返される。
<code>bf:extent</code>	ページ数	120	省略可能 繰り返し 無し	
<code>bf:dimensions</code>	大きさ	B4	省略可能 繰り返し 無し	
<code>textbook:grade</code>	対象学年	1, 2, 3	省略可能 繰り返し 有り	
<code>textbook:authorizedYear</code>	検定年	2001	省略可能 繰り返し 有り	
<code>textbook:catalogue</code>	掲載教科	textbook:catalogue/中学	必須	<code>textbook:catalogue/(学校種別)/(教科書目録発行年)</code> として示され

図 4: 教科書 LOD における説明記述 [8]

などに対応している。書式表現の制約条件としては、`sh:languageIn` や `sh:uniqueLang` といった言語タグ付きの制約を付けた文字列を指定できる。他にも、値制約 (`sh:minInclusive`, `sh:maxInclusive`)、文字列長の制約 (`sh:minLength`, `sh:maxLength`)、正規表現による指定 (`sh:pattern`) などが可能である。

本手法では SHACL を経由して出力しているため、SHACL エンジンを用いたデータセット検証が可能である。例えば、教科書 LOD では年度ごとに新しい教科書約 100 ~ 300 冊分のメタデータの追加があるが、それらの追加時にはデータ検証を行うことにより、想定外のデータが混入していないか、欠損が無いかなどを SHACL データ検証機能を通じて確認している。これら検証機能を通じて、エラーが発生した場合には、入力データの誤りだったり、想定外の事情によるデータ欠損などの事例が判明することもあり、長期的なデータの品質保証にもつながっている。なおこの際、SHACL エンジンとしては通常の検証機能を有するものがそのまま使えるため、例えば教科書 LOD では、SHaCLEX [11] や pySHACL [12] といった複数の実装から実行環境として使いやすいものを使っている。

最後に、提案手法の限界にいくつか触れておく。提案

手法はプロファイル記述の方式には SHACL の基本機能をそのまま使うことを想定しているが、文書化する際の機能として、多言語化と例示を提供する機能が SHACL 仕様に見当たらなかったため、独自に対応した。まずプロパティの値の形式などの例を示したいことがあるが、このような値の例示を記述する語彙が SHACL には存在しなかったため、SKOS 語彙から `skos:example` を用いている。また、SHACL には多言語文書化を想定した機能もなかったため、独自に、言語タグを指定することにより対応言語分の文書化を行うこととしている。多言語化対応の方式はプロファイル記述の文書化が 2 言語で完結してメンテナンスされる場合には合理性があると思われるが、一方で、単一言語しかメンテナンスできない場合や言語ごとの保守担当が異なる場合などには保守性が低くなってしまう等の問題も出てくるものと思われる。

また、文書化の出力は HTML 形式を前提としており、SHACL 形式への文書化要素の入力もそれにあわせて HTML を埋め込むような方式としているが、印刷用の PDF 形式や EPUB 形式といった異なる出力ニーズへの対応や、より多様な文書出力が求められる場合には、Markdown などをはじめとする他のマークアップ

方式なども検討する必要がある。

同様に、現状の文書化出力は `ttl2html` という著者が独自に開発した静的サイト構築ツールに埋め込まれて使うこととしているが、静的サイト構築ツールとしてはより汎用的な Jekyll [14] や Hugo [15] といった他のツール群 [16] との連係といった方式も検討する必要があるものと考えられる。加えて、汎用的な静的サイト構築ツールに加えて、Linked Open Data に特化したツール類 (例えば、Jekyll-RDF [17]) との連係も今後検討すべきものと考えられる。

また、プロファイル記述を Excel のような表形式で記述しているが、このような形態での文書表現がわかりやすい形であるかという点についても検討を要するように思われる。小規模であれば、表形式で十分に説明が尽くせると思われるが、説明文書の分量が増えた場合やより詳細な説明が必要となった場合に、どのように拡張すればよいかといった指針があると望ましいと思われる。より柔軟に分かりやすい記述が可能な形式を検討することも今後の課題として必要と思われる。

4 おわりに

本研究では、手軽に記述できる記述形式と人間が読んで分かる読解力という2点を満たすアプリケーションプロファイル記述を行う実践手法として、グラフ構造記述に SHACL を用いる方法論を示した。また、SHACL 形式をそのまま記述するのではなく、表計算ツール Excel を入力ツールとして用いて、Excel から SHACL への変換ツールと、SHACL データから人が読んで理解できるプロファイル記述への変換ツールの両者を開発することにより、手軽にアプリケーションプロファイル記述が可能な環境を提供した。

教科書 LOD を例として、中小規模のデータセットのプロファイル記述に対応する記述力があること、また、人間が読んでわかりやすいプロファイル提供に使えることを示し、さらに、SHACL のデータ検証機能をそのまま使うことにより、データセットの質と保守性の向上に活かすことができることを示した。一方で、今後の課題としては、ツールや実践における汎用性と拡張性、多言語機能などといったいくつかの側面を示した。

本ツールはオープンソースソフトウェアとして Github 上において公開している [10]。

謝辞

本研究は JSPS 科研費 JP20K12543 の助成を受けたものです。

参考文献

- [1] Mariana Curado Malta, Ana Alice Baptista, Paul Walk: Developing metadata application profiles, IGI Global, 2017, 248p.
- [2] Mikael Nilsson, Tom Baker, Pete Johnston: The Singapore Framework for Dublin Core Application Profiles, <https://dublincore.org/specifications/dublin-core/singapore-framework/> (2008)
- [3] Europeana Data Model: Europeana Pro, <https://pro.europeana.eu/page/edm-documentation> (accessed 2021-11-01)
- [4] 国立国会図書館ダブリンコアメタデータ記述 (DC-NDL), <https://www.ndl.go.jp/jp/dlib/standards/meta/> (accessed 2021-11-01)
- [5] Nishad Thalath, Mitsuharu Nagamori, Tetsuo Sakaguchi, Shigeo Sugimoto: Authoring Formats and Their Extensibility for Application Profiles, Proceedings of International Conference on Asian Digital Libraries (ICADL 2019), Lecture Notes in Computer Science, Springer, vol 11853, pp.116-122 (2019)
- [6] DCMI Application Profiles IG: DC Tabular Application Profile, <https://github.com/dcmi/dctap/> (accessed 2021-11-02)
- [7] Holger Knublauch, Dimitris Kontokostas (Eds): Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017, <https://www.w3.org/TR/shacl/> (accessed 2021-11-02)
- [8] 教科書 LOD プロジェクト: プロジェクトについて - 教科書 Linked Open Data (LOD), <https://w3id.org/jp-textbook/about> (accessed 2021-11-03)
- [9] 江草由佳, 高久雅生: 教科書 Linked Open Data (LOD) の構築と公開, 情報の科学と技術, vol.68, no.7, pp.361-367 (2018)
- [10] masao/ttl2html: Static site generator for Linked Data, <https://github.com/masao/ttl2html> (accessed 2021-11-05)
- [11] SHACL EX library, <https://www.weso.es/shacl-ex/> (accessed 2021-11-05)
- [12] pySHACL, <https://doi.org/10.5281/zenodo.4750840> (accessed 2021-11-05)

- [13] Alistair Miles, Sean Bechhofer (Eds.): SKOS Simple Knowledge Organization System Reference, W3C Recommendation 18 August 2009, <https://www.w3.org/TR/skos-reference/> (accessed 2021-11-05)
- [14] Jekyll Simple, blog-aware, static sites — Transform your plain text into static websites and blogs, <https://jekyllrb.com/> (accessed 2021-11-05)
- [15] The world's fastest framework for building websites, Hugo <https://gohugo.io/> (accessed 2021-11-05)
- [16] Static Site Generators - Top Open Source SSGs, Jamstack, <https://jamstack.org/generators/> (accessed 2021-11-05)
- [17] Jekyll RDF, <https://aksw.org/Projects/JekyllRDF> (accessed 2021-11-05)