

デジタルツイン基盤とその3D データ活用

青谷和真¹ 川村地平¹ 塚田学¹ 落合秀也¹ 江崎浩¹

概要：建物内の機器や設備の稼働状況、環境情報などの情報の可視化、シミュレーションを行うデジタルツインへの注目が高まっている。本稿では、デジタルツインアプリケーション作成のために行ってきた活動をまとめた。その手順は、3D データの収集、オントロジーを用いたメタデータの付与、建物情報を提供する API 作成、の3つに分けられる。3D データの収集は、3D スキャニングにより、大学内の建物や工場など、3箇所で行った。属性情報の付与に関しては、構造情報の記述には BOT、部屋の名称やセンサ等の記述には他のオントロジーを適宜使用して RDF を作成し、SPARQL エンドポイントとして公開することで外部から参照することを可能にしている。アプリケーションへ建物情報を渡す API は FastAPI を用いて実装した。この API は形状情報や属性情報を返すだけでなく、建物内の混雑状況の可視化などに使用するため、任意の視点からの部屋の2次元の画像を取得することができる。また、収集した3D データの活用例として、CityGML と組み合わせることで屋内外の情報を統合した3D マップビューワーを作成した。

KAZUMA AOTANI¹ CHIHEI KAWAMURA¹ MANABU TSUKADA¹ HIDEYA OCHIAI¹
HIROSHI ESAKI¹

1. はじめに

近年の IoT 化の進行などにより機器や設備の稼働状況、環境情報などをリアルタイムで収集することが可能となっており、これらの情報の可視化やシミュレーションを行うデジタルツインへの注目が高まっている。デジタルツインは特に製造業における生産性の向上やスマートシティの実現といった分野での活用が期待される [1][2]。

デジタルツインアプリケーションの作成には、対象の建物の3次元形状及びそれに付随する属性情報が用いられる。BIM(Building Information Modeling)と呼ばれる、建物の形状情報と属性情報を併せ持った建築情報モデルの使用が有効であるが、既存の建物には BIM が存在しない場合が多い。そのため、3D スキャンを用いて建物の形状情報を取得し、それに対し属性情報を付加し、BIM 化するというプロセスが必要となる。

ある建物のデータを使用したデジタルツインのユースケースは多数存在し、これらのデータは多様なドメインのデータと相互に参照し合うことでより価値を発揮する。そのため、建物に関するメタデータは一つの固有のシステム

の中に取り組むのではなく、オープンデータとして汎用的に扱える必要がある。そこで、属性情報の付加の際には、オントロジーに従って記述する事で、建物に関する情報から LOD (Linked Open Data) を作成し、このデータをアプリケーション側に提供するための API を用意することが望ましい。

本稿では、まずデジタルツインアプリケーション作成に関連する技術を紹介する。次に、3章で、デジタルツインアプリケーションのためのデータ基盤の作成として実際に行なった、データ収集および3D マップの作成、オントロジーを用いたメタデータの付与、建物情報を提供する API 作成のそれぞれの工程について説明する。

3D マップの作成に関しては、東京大学工学部2号館、エービー白河工場事業所、NTTCOM スマートシティラボにて行なった。撮影には MatterportPro2 を使用し、点群データ及びメッシュファイルを得た。属性情報の付与に関しては、BOT (Building Topology Ontology) ^{*1}を用いて構造情報を記述し、部屋の名称やセンサ等の記述には他のオントロジーを適宜使用した。

アプリケーションへ建物情報を渡す API は FastAPI を

¹ 東京大学 The University of Tokyo

^{*1} <https://w3c-lbd-cg.github.io/bot/#hasBuilding>

用いて実装した。この API は単純に形状情報や属性情報を返すだけでなく、建物内の混雑状況の可視化などに使用することを想定し、任意の視点からの部屋の画像を取得できるようにしている。

また、デジタルツインを街単位で活用する例として、スマートシティへの応用などが挙げられる。街の 3D データをオープンデータとして公開するプロジェクトは国内外で見られるが、国土交通省によって公開された東京都のデータを 3 章で作成したデータと組み合わせた実装について 4 章で説明する。

最後に、5 章で、まとめと今後の課題について説明する。

2. 関連技術・研究

2.1 デジタルツイン

デジタルツインは、狭義では原子レベルから幾何学的なレベルまで、現実上の物体を完全に記述する一連の情報であると定義される [3]。これらは設計、生産、運用、保守に至るまでの製品のライフサイクルをシミュレーションすることに用いられる。一方で広義にはシステムの状態やプロセスのシミュレート、監視、制御などができる統合的システムであると定義されている。このデジタルツインには、物理的空間、デジタル空間、そして 2 つの空間をつなぐ情報処理層の 3 つから構成される。BIM 等から抽出した形状データや実空間から収集した時系列データを用いて、サイバー空間でシミュレーションやデータの可視化などを行うアプリケーションは「デジタルツイン」と呼ばれる。デジタルツインは元々製造業で先行してきた概念であり、シミュレーションによって現実と同様のプロダクトをサイバー空間上に作り、想定されるトラブルの検証などに利用されていた。現在はデジタルツインは製造業に加え、ヘルスケア、スマートシティといった分野での活用が盛んである [4]。

2.2 セマンティクスウェブ技術

2.2.1 オントロジ

オントロジは「概念 (Concept) と関係性 (Relation) の組を持ち、形式化されているためにコンピュータによる推論 (Reasoning) が可能であるもので、特定のドメインに対して有用にデザインされている」と定義されている [5]。オントロジはセマンティックウェブにおいて重要であり、これを記述するために RDF などが用いられる [6]。

2.2.2 RDF

RDF (Resource Description Framework) は Web 上にあるリソースを記述するためのモデルである。RDF では、トリプルと呼ばれる、主語、述語、目的語の 3 つの要素でリソースの関係を表現する。これらのトリプルの集合を RDF のグラフと呼び、リソースの識別には URI (Uniform Resource Identifier) を用いる。RDF のデータを格納する専用データベースをトリプルストア (Triplestore) と呼び、

RDF のクエリ言語 SPARQL を用いてクエリを発行することができる。

2.2.3 LOD

LOD (Linked Open Data) とはオープンライセンスのもとで公開され、無料で再利用が可能な Linked Data のことである。前節の RDF を、外部リンクを含みオープンライセンスのもとで公開することで LOD とすることができる。

オープンデータを公開する際には、次の 4 つの条件が提案されている [7]。

1. モノやコトに対する識別子に URI を使用する
2. 誰もが参照できるよう HTTP の URI を使用する
3. URI を参照した際には、RDF や SPARQL などを用いて有用な情報が提供される
4. 多くの情報が得られるよう外部へのリンクを含める

RDF や LOD は、再利用性の高さ、処理に特定のアプリケーションを必要としないこと、URI を用いてリソースを一意に識別できるという理由からオープンデータとして高く評価されている。また、LOD では外部と参照し合うことでデータの価値を高めることができる。

2.3 3D データ

2.3.1 CityGML

街情報などの 3D データを公開し、利活用する例が国内外で見られる。スマートシティの実現など、3D データをより高度なデータマネジメントに応用するには、ジオメトリ情報だけでなく建物の名称や高さといった多様なメタデータを情報として保有している必要がある。このような要件を踏まえた日本全国の 3D 都市モデルの整備・活用・オープンデータ化プロジェクトとして、国土交通省が主導する PLATEAU と呼ばれるものがある。このプロジェクトでは東京都 23 区をはじめとした複数都市の 3D モデルを公開しており、人流計測や混雑状況の可視化に用いられている。これらの都市モデルは CityGML という形式のファイルで表現されており、ジオメトリの情報に加え、構造物の属性情報が XML 形式で組み込まれている。また、CityGML のもう一つの特徴としてモデルの詳細度に応じてレベルが分かれており、これらを一元管理することができる [8]。図 1 に CityGML ビューワーにより 3D マップを表示させている画面を示す。

現在策定中の CityGML3.0 では、[9] では、CityGML の見直しが行われており、時系列データなど、動的な情報を記述できるようにするなど、新たな LOD の策定などに取り組んでいる。

2.4 BIM

建物の形状情報と属性情報を持ったデータ形式として Building Information Modeling (BIM) が存在する。BIM は「建物の物理的・機能的特性をデジタルで表現したもの



図 1 Cesium により表示した CityGML

であり、ライフサイクルで発生する何らかの意思決定のために、信頼性の高い情報を提供する共有知識資源である」と定義されている [10]。より具体的には、建物の 3 次元形状の情報だけでなく、部屋の名称や面積、材料といった属性情報を付加した建築情報モデルである。BIM は現在主に建設プロセスの効率化や安全衛生面の管理、施設の管理といった分野に利用されている [11][12]。特に近年は IoT との連携によりデジタルツインを実現し、より多くの分野での活用が期待されている [13]。

一般に BIM の作成には Autodesk, GRAPHISOFT などが開発した専用のオーサリングツールを利用する。建築オブジェクトとそのプロパティを読み込む、及び書き込むための国際標準として、Industry Foundation Classes(IFC)が存在し、BIM はこの IFC 形式で出力される。

BIM をデジタルツインアプリケーションに活用するためには建築オブジェクトの形状情報とメタデータを取得することが必要となる。IFC ファイルからこれらを抽出する API の作成が行われている。メタデータについては IFC 形式から BOT に従った記法に変換してグラフ DB に格納、形状情報については gltf や obj 形式に変換し PostGIS に格納することで、それぞれデジタルツインアプリケーションに対して提供しやすいうように保持している。この一連の流れを図 2 に示す。

3. 実験

デジタルツインアプリケーションの構築には、対象となる物体及び、それをデジタル空間上で再現するための 3D データ、メタデータが必要となる。3D データの取得には 2 章で述べたように BIM を用いることが多いが、BIM の作成には専門的な知識や時間、労力などのコストが要求される。新しく建設される建物に関しては建設プロセスの管理のために BIM が用意されていることが多いが、既存の建物の多くは BIM が用意されていないことがほとんどである。そこで、3D スキャンによる 3D データの取得をし、それに対してオントロジーに従った記法によりメタデータの付与をした。さらに、アプリケーションに対して情報を渡すための API を作成した。本章では、この各ステップに

対する詳細を記す。

3.1 データ収集

3D データの取得に関しては、フォトグラメトリックな方法、LIDAR などのセンサを用いた手法などがある。今回は 3D スキャンに MatterPort 社 [14] の提供するサービスを利用した。撮影に用いた機材は MatterPortPro2 で、赤外線による形状情報の取得、4K カメラによるテクスチャ情報の取得が可能になる。形状情報は点群、OBJ ファイルとして取得することができる。

撮影場所は東京大学本郷キャンパスの工学部 2 号館、エービー白河工業事業所、NTTCOM スマートシティラボの 3 箇所である。

3.1.1 東京大学キャンパスマップ

Matterport を用い、東京大学工学部二号館の B1F から 13F までを撮影した。ただし、4F、10F 以外の階では階段からエレベーターホールまでの短い区間のみを撮影した。4F では、それらに加えて講義室や実験室を撮影し、10F では、研究室を撮影した。4F の撮影には約 5 時間 (撮影回数:350 回)、10F の撮影には約 2 時間 (撮影回数:120 回) を要した。撮影した講義室のデータの一部を図 3 に示す。これらの 3D モデルを用いて、3D マップを構築していく。撮影に使用した Matterport Pro2 というカメラでは、一度に 3.5m の範囲を撮影できるが、撮影範囲が広くなると、徐々に誤差が蓄積されていき、構造的に整合性が取れなくなる場合がある。図 3 では本来閉路となるはずの廊下が、ループを構築できないという問題が発生している。

3.1.2 エービー白河事業所マップ

BIM の活用分野の一つにファクトリーオートメーションが挙げられる。本節では、工場内の情報の可視化等に活用するため作成した 3D モデルについて説明する。撮影を行った場所はエービー白河事業所で、ここではプラスチック金型制作などが行われている。工場内には大型の機械が多く、オクルージョンが多く発生したため、撮影には多くの地点を要した。撮影に要した時間は 4 時間半程で、139 地点での撮影情報を用いた。に工場を上から見た画像を示す。また、工場内には環境センサや Lidar センサが複数箇所に設置されており、温度、湿度、照度、騒音や機械の電力といった情報を取得することが出来る。

3.1.3 NTTCOM スマートシティラボのマップ

東大グリーン ICT プロジェクト (GUTP)*2にてデジタルツインアプリケーション開発のためのワークフローの検討および実証実験が進められている。デジタルツインアプリケーション開発の最初の段階として、今回 NTTCOM スマートシティラボにて撮影を行なった。この撮影では、Matterport Pro2 に加え、レーザーによるスキャ

*2 <https://www.gutp.jp/>

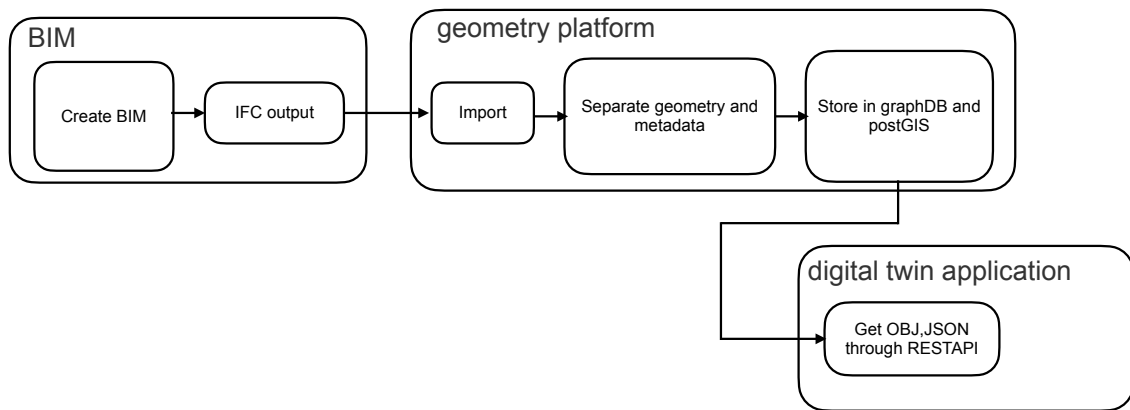


図 2 BIM から形状情報、属性情報を抽出し、アプリケーションへ渡すためのシステム構成概要



図 3 工学部 2 号館の点群データ。閉路の始点と終点の位置がずれている。

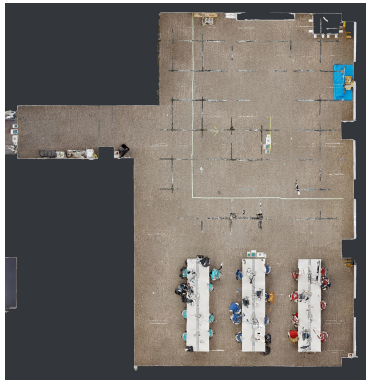


図 4 NTT コミュニケーションのスマートラボの点群データ

ンを行う BLK360 も用いられた。両者で撮影をしえられた点群の比較では、主に密度に差があり、物体の形状などは BLK360 の方が精度が上であった。例えば、床に貼られたテープが、BLK360 では一直線となっていたが、Matterport Pro2 では途中で途切れるなどしていた。この様子を図 5 に示す。一方で、今回撮影を行なった規模では、部屋全体の精度としてはそれほど大きな差は出なかった。この撮影に要した時間は 1 時間ほどで、撮影地点は 48 箇所であった。

3.2 BOT によるメタデータ付与

建物に関する情報として必要なものとして、構造情報が挙げられる。これを記述するために、オントロジーとして BOT[15] を利用している。

BOT では、図 7 に示すように、区域から始まり、階層的に建物、階数、空間を記述する。また、各部屋に対しては、その部屋の名称、固有の ID、設置されている場合はセンサの ID がメタデータとして付与されている。

3.3 API 設計

3.1.1 で収集したデータを用いて、建物の形状情報、属性情報をアプリケーションに対して提供する API を、Python の FastAPI で実装した。API の構成としては、建物に関するメタデータ、3D データ、時系列データを与えるものを用意している。メタデータについては、引数を指定しない場合、その建物の名称、所在地、その建物に含まれる階数や部屋についての情報を JSON 形式で取得することができる。部屋の ID をパラメータとして指定した場合には、その部屋内の情報として、部屋固有の ID、部屋の名称、3D データの格納先のパス、センサなどが設置されている場合はそのセンサの名称と MQTT ブローカーの IP アドレスを取得できる。また、外部アプリケーションとの連携等を考慮し、3D データ上の任意の視点からの画像を生成し提供するというを行なっている。この機能の実装例がなかったため、今回 2 通りの実装を行い、その評価を行なった。

3.3.1 3D マップから 2D 画像への変換

前節で記述したメタデータ及び部屋の画像を取得することのできる API を Docker 内に FastAPI を用いて作成した。3D マップからの画像の取得は python を用いた手法と Unity による手法の 2 種類の手法を用いて実装を行なった。Python を用いた実装では、VM 上に Xvfb による仮想ディスプレイを作成し、pygame を用いて obj ファイルを読み込み、指定された視点からの仮想ディスプレイのキャプチャを行なっている。視点の向き、視点の位置、部屋は POST リ

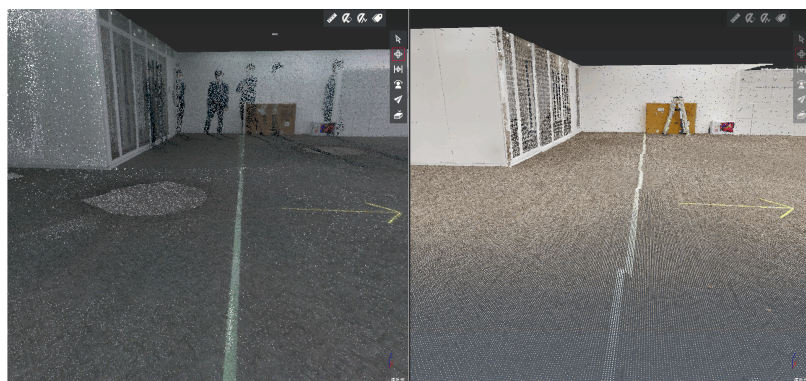


図 5 Matterportpro2 と BLK360 で撮影した点群データの比較. 左側が BLK360 によるもので右側が Matterportpro2 によるもの. 点の密度や精度に差があり, Matterportpro2 によるものは床の上の線に途切れが生じるなどした.

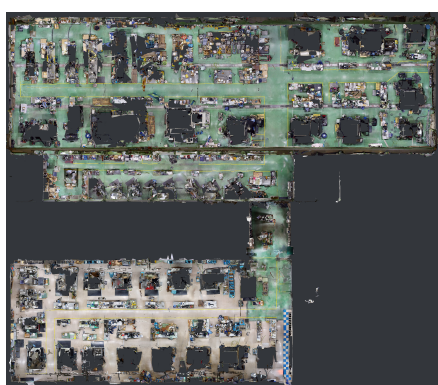


図 6 エービー白河工場の点群データ

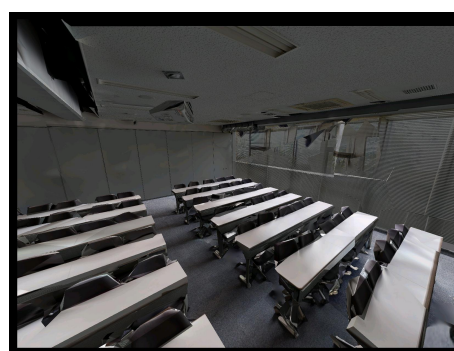


図 8 本節で述べた API から取得した自由視点からの二次元画像

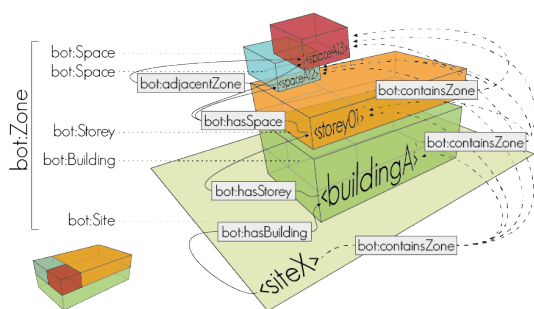


図 7 区域内のクラス間の関係性 [15]

クエストにより指定している。

Unity を用いた実装では, FastAPI を通した POST リクエストから Unity へアクセスする。POST リクエストには, 上部から撮影するか斜部から撮影するかといった情報やどの部屋を撮影するかといった情報を含んでいる。POST リクエストを受け取った後は, Xvfb を用いて仮想ディスプレイを作成し, ビルド済みの Unity アプリケーションデータを用いて 3D モデルを配置し, キャプチャを行い画像ファイルを生成する。最後にその画像ファイルを Base64 形式でユーザに返却するといった構成になっている。図 8 に, この API を通じて取得した部屋の画像を示す。

3.3.2 API 評価

Python で作成した 2 次元画像と Unity で作成した 2 次元画像を図 9 に示す。Python による実装の方が部屋の様子が鮮明に写っているが, Unity の場合は約 4.3 秒, Python の場合は約 13.6 秒と応答時間に大きく差が出た。上記で設計した API のうち, 特に応答時間が長くサーバーに負荷のかかる 2D 画像の生成機能について, 応答時間を評価した。一つの部屋の obj ファイルは 4~16MB 程度であり, これを読み込み画像を生成するまでの時間はおおよそ 3D データのサイズに比例した。図 10 にファイルサイズと応答時間の関係を表したグラフを示す。応答時間は Python により実装した API で計測し, 10 回の平均を取った。また, ネットワーク環境等による影響をなくするためローカルホスト上にサーバーを立ててアクセスを行なった。

真上から見た画像や, 横から見た画像など, 利用頻度が高いと思われる画像について, キャッシュをしておくことで応答時間は 1.5×10^{-2} 秒程度まで削減することができた。また, 同時に 100 件のリクエストを投げるとサーバーが処理落ちするため, RabbitMQ による非同期処理を実装している。

特に Python での実装に関して, 実用的な API とするためにはより応答時間を短縮することが求められる。応答時間は主に 3D データのサイズに比例するため, 例えばポリ

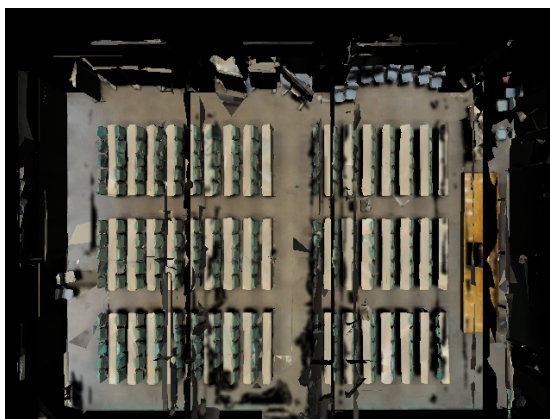
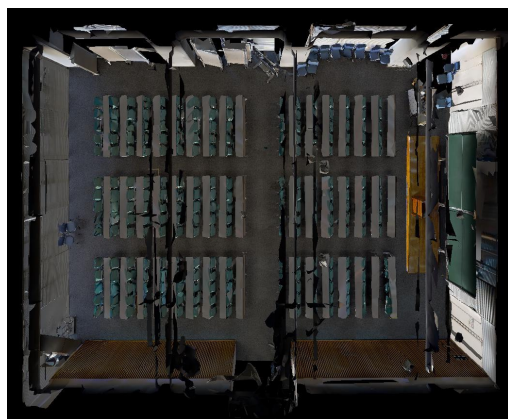


図 9 Python (左), Unity (右) によって生成した部屋の 2 次元画像

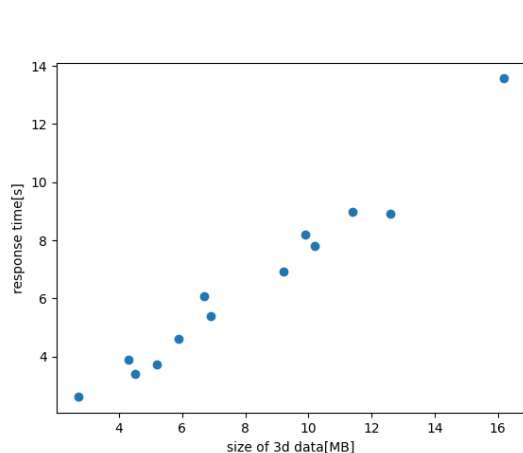


図 10 3D データのファイルサイズと API の応答時間の関係をプロットした散布図

ゴン数を削減することは有効である。一方でポリゴン数を削減しすぎると画像の詳細度の低下に繋がるため、最適なポリゴン数の模索が求められる。また、今回用意した 3D データは 3D スキャンにより取得してきたものであり、頂点の数が BIM などによる 3D モデルよりも多くなってしまっている傾向がある。そこで、点群から自動で BIM を作成する際に用いられる平面検出など [16] を用いることで、3D モデルをより簡単な形に置き換えることでファイルサイズの削減につながる事が予想される。

4. PLATEAU と屋内情報の活用

PLATEAU で公開されているデータは街の外観を捉えた 3D モデル及びそれに付随するメタデータである。メタデータには、住所、高さ、面積などに加え浸水ランクなどのデータが含まれており、XML 形式で記述されている。一方で、前章で作成した 3D マップは建物内部の 3D データとメタデータを含んでいる。これらの屋内外情報を繋ぐことによって、より詳細度の高い 3D マップの作成が可能になる。また、人流データなど、これまで屋外での情報と屋内での情報が分断されていたものをつなぐことで、デー

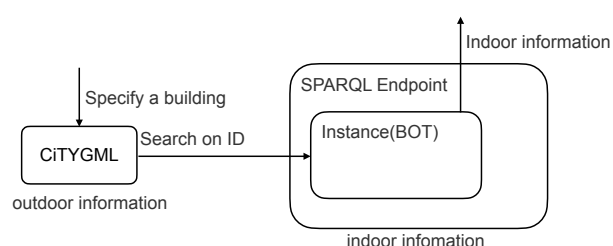


図 11 屋内情報と CiTyGML を組み合わせたビューワーの構成図

タの利活用が可能になる。

4.1 実装

CityGML 形式で与えられた東京都 23 区の 3D マップと前章で用意した工学部 2 号館の 3D データを用いることで屋内外の情報を繋いだビューワーを作成した。ビューワーのシステム構成を図 11 に示す。CityGML の読み込み及び可視化には Cesium という 3D ビューワーエンジンを活用しており、建物情報及びメタデータの取得を行なっている。街全体のビューワーから任意の建物を指定すると、その建物固有の ID から、SPARQL エンドポイントに対して検索を行い、オントロジーで記述されたインスタンスを特定することができる。

4.2 課題点

屋内に関する情報はオントロジーに従った記述により RDF 形式で記述されているが、屋外情報に関しては CityGML で与えられており、それぞれの情報を一括で管理することが現状では出来ていない。また、CityGML の形式で与えられたファイルを読み込むには、Cesium のような専用のツールを用いることが必要とされるため、自由な開発を阻害してしまっている現状にある。そのため、2.4 節で触れたような、メタデータや形状データを取り出し提供する API などが重要になってくると考えられる。

5. まとめと今後の課題

本稿では、デジタルツインアプリケーション作成の枠組みとして、建造物の 3D データ取得からメタデータの付与、建物に関する情報をアプリケーションに提供する API の作成を行なった。また、この屋内の情報と、オープンデータとして提供されている 3D マップを組み合わせることで、屋内外のデータを繋いだより詳細な 3D マップビューワーを作成した。

今後の課題としては、今回作成した API の応答時間をより早くするための 3D データの処理や、この API を利用して現実世界へのフィードバックまでを含めたデジタルツインアプリケーションの作成することで実用性の評価を行うことや、3D スキャンを通じて獲得した 3D データと BIM を同様に扱うためのデータ処理などが挙げられる。また、CityGML のデータをより扱いやすくするための枠組みの導入などが必要となる。

参考文献

- [1] A Fuller, Z Fan, C Day, and C Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, Vol. 8, pp. 108952–108971, 2020.
- [2] Chiara Cimino, Elisa Negri, and Luca Fumagalli. Review of digital twin applications in manufacturing. *Comput. Ind.*, Vol. 113, p. 103130, December 2019.
- [3] Yu Zheng, Sen Yang, and Huanchong Cheng. An application framework of digital twin and its case study. *J. Ambient Intell. Humaniz. Comput.*, Vol. 10, No. 3, pp. 1141–1153, March 2019.
- [4] T Ruohomäki, E Airaksinen, P Huuska, O Kesäniemi, M Martikka, and J Suomisto. Smart city platform enabling digital twin. In *2018 International Conference on Intelligent Systems (IS)*, pp. 155–161. ieeexplore.ieee.org, September 2018.
- [5] Ayesha Ameen, Khaleel Ur Rahman Khan, and B Padmaja Rani. Extracting knowledge from ontology using jena for semantic web. In *International Conference for Convergence for Technology-2014*, pp. 1–5. ieeexplore.ieee.org, April 2014.
- [6] 粕谷 貴司 (Takashi Kasuya). BIM を用いた Software Defined BACS の実現に関する研究 (The design and implementation of Software Defined BACS based on shared BIM repository). PhD thesis, Ph.D Thesis, The University of Tokyo, 2020.
- [7] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205–227. IGI Global, 2011.
- [8] Thomas H Kolbe, Gerhard Gröger, and Lutz Plümer. CityGML: Interoperable access to 3D city models. In Peter van Oosterom, Siyka Zlatanova, and Elfriede M Fendel, editors, *Geo-information for Disaster Management*, pp. 883–899. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [9] Tatjana Kutzner, Kanishk Chaturvedi, and Thomas H Kolbe. CityGML 3.0: New functions open up new applications. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, Vol. 88, No. 1, pp. 43–61, February 2020.
- [10] Markus Koenig Christian Koch Jakob Beetz André Borrmann. *Building Information Modeling :Technology Foundations and Industry Practice*. Springer, 2018.
- [11] Shu Tang, Dennis R Shelden, Charles M Eastman, Pardis Pishdad-Bozorgi, and Xinghua Gao. A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Autom. Constr.*, Vol. 101, pp. 127–139, May 2019.
- [12] Alex Bradley, Haijiang Li, Robert Lark, and Simon Dunn. BIM for infrastructure: An overall review and constructor perspective. *Autom. Constr.*, Vol. 71, pp. 139–152, November 2016.
- [13] Kendrik Yan Hong Lim, Pai Zheng, and Chun-Hsien Chen. A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. *J. Intell. Manuf.*, Vol. 31, No. 6, pp. 1313–1337, August 2020.
- [14] Matterport. <https://matterport.com>.
- [15] Mads Holten Rasmussen, Maxime Lefrançois, Georg Ferdinand Schneider, and Pieter Pauwels. BOT: The building topology ontology of the W3C linked building data group. *Semant. Web*, pp. 1–19, October 2020.
- [16] Qian Wang, Yi Tan, and Zhongya Mei. Computational methods of acquisition and processing of 3D point cloud data for construction applications. *Arch. Comput. Methods Eng.*, Vol. 27, No. 2, pp. 479–499, April 2020.