

Artykuł przeglądowy

W poszukiwaniu optymalnego formatu danych dla trójwymiarowych modeli miast

In the quest for an optimal data format for three-dimensional city
models

Piotr Cichociński

AGH Akademia Górniczo-Hutnicza

Abstract

Despite the continuous development of geographic information systems (GIS) methods and tools, most spatial data sets used in Poland are still two-dimensional. In order to be able to carry out various types of simulations and analyses, 3D models of the cities are needed that comply with commonly accepted standards. Such a widely used, open standard for representing objects and structures in 3D city models is the City Geography Markup Language (CityGML). However, there has been insufficient support for CityGML-formatted data by available software, one reason for which is the eXtensible Markup Language (XML) encoding – verbose, hierarchical, complex and unsuitable for use on the web. Therefore, it was concluded that in order to popularise 3D city models, two problems would have to be solved: simplifying the data exchange format and making it more compatible with GIS software. For this purpose, three formats developed in recent years were analysed: CityJSON, FlatGeobuf, GeoParquet. They implement an approach referred to as “cloud-native”, which refers to the management of datasets (including spatial datasets), optimised for hosting in cloud environments. However, the review has shown that there is currently no format for 3D city models that is immediately applicable and further work is needed to build a fully functioning solution.

Słowa kluczowe: CityGML, CityJSON, cloud native, FlatGeobuf, GeoParquet

Keywords: CityGML, CityJSON, cloud native, FlatGeobuf, GeoParquet

Wstęp

Pomimo ciągłego rozwoju metod i narzędzi systemów informacji przestrzennej (GIS) większość zbiorów danych przestrzennych używanych w Polsce jest wciąż dwuwymiarowych (2D), tak samo jak używane wcześniej tradycyjne papierowe mapy. Takie podejście utrudnia, czy wręcz nawet uniemożliwia prowadzenie złożonych analiz, które wymagają danych trójwymiarowych (3D). Tymczasem trójwymiarowe modele są coraz częściej tworzone dla miast na całym świecie (szczególnie tych największych). Już w roku 2012 było ponad tysiąc takich modeli (Morton i in., 2012). Ich zastosowania wykraczają daleko poza wizualną reprezentację. Badania przeprowadzone przez Biljeckiego i in. (2015) wykazały, że modele 3D miast były wykorzystywane w co najmniej 29 różnych przypadkach, które łącznie reprezentowały ponad 100 różnych zastosowań. Nowsze projekty zostały opisane między innymi w publikacjach (Biljecki i in., 2016), (Eller i in., 2022), (Gholami i in., 2022), (Harter i in., 2023), (Johansson i in., 2016), (Katal i in., 2022), (Rodríguez i in., 2017), (Virtanen i in., 2021), (Waksmundzka i in., 2021) i (Willenborg i in., 2018).

Aby móc przeprowadzać różnego rodzaju symulacje i analizy, modele 3D miast powinny być zgodne ze powszechnie przyjętymi standardami. Takim szeroko stosowanym, otwartym standardem reprezentowania obiektów i struktur w trójwymiarowych modelach miast jest City Geography Markup Language (CityGML) (Gröger i in., 2012). Zapewnia on ramy do organizowania i opisywania najważniejszych elementów miejskich, takich jak budynki, drogi i ukształtowanie terenu, w oparciu o ich właściwości fizyczne (kształt, rozmiar), relacje przestrzenne, przeznaczenie i wygląd zewnętrzny. Wykorzystuje do tego celu hierarchiczne struktury powiązanych obiektów, połączenia między nimi i opisy ich położenia w przestrzeni. CityGML pozwala również na różne poziomy szczegółowości (Level of Detail – LOD). Wykraczając poza proste formaty graficzne, CityGML umożliwia tworzenie szczegółowych modeli 3D miast, które mogą być wykorzystywane do zaawansowanych analiz i wspomagania podejmowania decyzji.

Sformułowanie problemu

Pomimo rosnącej popularności modeli miast opartych na CityGML, Noardo i in. (2020) stwierdzili niewystarczający poziom obsługi danych zapisanych w formacie CityGML przez dostępne oprogramowanie. Można odnieść wrażenie, że zdecydowana większość wysiłków związanych z CityGML została poświęcona na opracowanie modelu danych i wydaje się, że bardzo niewiele uwagi poświęcono opracowaniu użytecznego formatu wymiany. Rzeczywiście, kodowanie eXtensible Markup Language (XML) jest

rozwickłe, hierarchiczne, złożone i nieprzystosowane do użycia w sieci (Bydłosć i in., 2010). Wady te utrudniają wykorzystanie CityGML w praktyce, co można zaobserwować na podstawie niewielkiej liczby pakietów oprogramowania obsługujących pełne możliwości odczytu/zapisu/edycji plików CityGML oraz stosunkowo niedużej liczby zbiorów danych przechowywanych w plikach CityGML. Pliki CityGML są znane z tego, że bardzo trudno jest je analizować, porównywać i wydobywać z nich informacje (Ding i in., 2023), (Nguyen i in., 2017). Ma to głównie związek z faktem, że GML dopuszcza kilka różnych sposobów przechowywania tej samej geometrii (Rouault, 2014), a pliki CityGML mają głęboką hierarchię z wieloma XLinkami (jest to metoda tworzenia wewnętrznych i zewnętrznych odnośników w dokumentach XML). Te problemy komplikują opracowanie oprogramowania, które może skutecznie odczytywać, wizualizować, analizować i przekształcać dane CityGML (Ledoux i in., 2019).

Pewną trudnością w popularyzowaniu modeli 3D miast jest słaba dostępność odpowiednio ustrukturyzowanych, szczegółowych danych wektorowych. Wydaje się jednak, że problem ten został ostatnio rozwiązany za sprawą oprogramowania „geoflow” (Peters i in., 2022), które pobiera chmury punktów i poligony 2D dla kolejnych pojedynczych budynków i generuje na tej podstawie ich modele 3D. Jest to w pełni zautomatyzowany proces, nie wymagający interwencji operatora.

Również w Polsce udało się w relatywnie prosty sposób uzyskać modele budynków 3D, będące trójwymiarowym odwzorowaniem znacznej części budynków z bazy danych obiektów topograficznych BDOT10k (Główny Urząd Geodezji i Kartografii, 2023a). Dane na poziomie szczegółowości LOD2 dostępne są dla 236 powiatów, obejmujących obszar 10 województw, natomiast dane na poziomie szczegółowości LOD1 udostępniane są dla całego kraju. Od roku 2023 zasób ten jest sukcesywnie wzbogacany o modele 3D drzew o wysokości powyżej 4 m (Główny Urząd Geodezji i Kartografii, 2023b). Wszystkie modele są dostępne bezpłatnie i mogą być swobodnie wykorzystywane, ale jak dotąd nie znalazły szerszego zastosowania. Jednym z powodów może być niewygodny i nieefektywny sposób dostępu do tych danych. Chociaż są one prezentowane w Geoportalu 3D, jego funkcjonalność ogranicza się do przeglądania tych danych i podstawowych analiz, takich jak wizualizacja zacielenia. Aby szerzej wykorzystać te dane, należy je pobrać za pośrednictwem portalu mapowego w żmudnym, ręcznym procesie. Korzystając z narzędzia identyfikacji, użytkownicy klikają na mapę w granicach powiatu, dla którego chcą uzyskać dane. Powoduje to wyświetlenie listy dostępnych zbiorów danych, które należy pobierać pojedynczo. Uzyskane w ten sposób pakiety danych są jeszcze podzielone na pliki odpowiadające arkuszom map topograficznych w skali 1:5000. Do przeglądania i przetwarzania danych zaleca się korzystanie z bezpłatnego oprogramowania QGIS (QGIS.org, 2024). Należy jednak pamiętać,

że na poziomie szczegółowości LOD2 te zbiory danych można już sklasyfikować jako Big Data.

Propozycja rozwiązania

Z powyższego wynika, że dla spopularyzowania trójwymiarowych modeli miast należałoby rozwiązać dwa problemy: uproszczenie formatu wymiany danych i zwiększenie jego zgodności (szczególnie) z oprogramowaniem GIS oraz ułatwienie dostępu do danych. Z pomocą przychodzi stale zwiększająca się dostępność taniej, niezawodnej i skalowalnej przestrzeni dyskowej w chmurze, zapewniająca nowe, bardziej wydajne i elastyczne sposoby pracy z danymi geoprzestrzennymi. Jako przykłady można tutaj wymienić Amazon S3, Microsoft Azure, czy Google Cloud. Z tego powodu tradycyjne bazy danych i dostawcy usług danych przestrzennych mogą stać się niepotrzebnym obciążeniem w dostępie do (w szczególności) ogromnych zbiorów danych geoprzestrzennych. W ich miejsce coraz większą uwagę poświęca się podejściu określanemu w języku angielskim jako „cloud-native”, które odnosi się do zarządzania zbiorami danych (w tym przestrzennych), zoptymalizowanymi pod kątem hostingu w środowiskach chmurowych.

W ramach tego podejścia między innymi pojawiły się nowe formaty danych (i obsługujące je biblioteki programistyczne), pozwalające na efektywny dostęp do danych zapisanych na dyskach w chmurze. Zdaniem Opaleychuk (2024) kluczowym hasłem ich autorów jest „pobierz tylko to, czego potrzebujesz”. Ma to szczególne znaczenie w przypadku ogromnych zbiorów danych, które nie mają szansy zmieścić się w całości w pamięci komputera (Barciauskas i in., 2023) i oznacza, że użytkownik powinien mieć możliwość wczytania tylko niezbędnych w danym momencie kafelków lub części ("chunks") danych. Ponieważ wiele standardów „cloud-native” uwzględnia zapytania, częściowe odczyty i inne metody usprawniające pobieranie danych (Green, DiStefano, 2024), dostawcy danych mogą polegać na tych standardach, aby zoptymalizować swoje zbiory danych bez konieczności dodatkowej konfiguracji po stronie serwera.

W pierwszej kolejności zostały opracowane i spopularyzowane formaty dla danych rastrowych i chmur punktów, takie jak Cloud-Optimized GeoTIFF (COG), czy Cloud-Optimized Point Cloud (COPC). Dane wektorowe są powszechnie uważane za bardziej wymagające (trudniejsze) do optymalizacji na potrzeb chmurowych, ale można zaobserwować pewne postępy w kierunku osiągnięcia tego celu i kilka interesujących ścieżek do zbadania (Holmes, 2024).

Omówienie wyników

Formatom, którym być może byłoby najbliżej do spełnienia powyższych kryteriów są rozwiązania bazujące na JavaScript Object Notation (JSON). Jest to popularny wśród programistów format, zatem dostępnych jest wiele narzędzi i bibliotek programistycznych umożliwiających odczytywanie zapisanych w nim danych. W kontekście prowadzonych tutaj rozważań w pierwszej kolejności należy wymienić CityJSON (Ledoux i in., 2019), który powstał jako alternatywa dla kodowania GML stosowanego w CityGML. Ma być łatwy w użyciu, zarówno w sytuacji odczytu zbiorów danych, jak i w przypadku ich tworzenia. Tam, gdzie było to możliwe hierarchia obiektów CityGML została spłaszczona dla uproszczenia zapisu modeli, lecz zdaniem twórców nie prowadzi to utraty informacji.

Jednak JSON nie do końca spełnia wymagania podejścia „cloud-native”. Żeby to osiągnąć, trzeba rozważyć i ewentualnie wprowadzić pewne modyfikacje. Zostały do tej pory podjęte co najmniej dwie takie próby. Cloud-Optimized GeoJSON był eksperymentem mającym na celu utworzenie wektorowego odpowiednika dla Cloud-Optimized GeoTIFF (COG). Polegał on głównie na wprowadzeniu zorientowanego przestrzennie „spisu treści” w celu przyspieszenia dostępu do konkretnych obiektów zapisanych w zbiorze (Ingold i in., 2020). Natomiast Newline-delimited GeoJSON dzięki usunięciu nadrzędnego elementu „FeatureCollection” umożliwia niezależny odczyt każdej linii (reprezentującej jeden obiekt) bez konieczności wczytywania całego (prawdopodobnie dużego) pliku (Bennett, 2019).

Oprócz JSON do rozważenia pozostają jeszcze dwa formaty: FlatGeobuf i GeoParquet. W odróżnieniu od kodowania JSON FlatGeobuf jest formatem binarnym. Wyróżniają go dwie cechy: możliwość pobierania wybranych fragmentów za pośrednictwem żądań zakresu (range requests) protokołu HTTP oraz unikalny sposób dostępu do konkretnych obiektów, łączący indeks R-Tree z określonym ich uporządkowaniem przestrzennym (wzdłuż krzywej Hilberta) (Williams, 2022). Wadą takiego rozwiązania jest duży stopień skomplikowania struktury danych. Ponadto trzecia współrzędna (Z) nie jest traktowana na równi z pozostałymi dwoma (XY) (jest zapisywana w osobnej tablicy), co może powodować problemy w niektórych zastosowaniach.

GeoParquet jest również binarnym formatem bazującym na Apache Parquet, będącym nowoczesną alternatywą dla plików CSV (GeoParquet, 2024). W odróżnieniu od tradycyjnego podejścia – wierszowego, dane przechowywane są pionowo – kolumnami. W przypadku podejścia wierszowego przykładowe dane zaprezentowane w tabeli 1 byłyby zapisane w następujący sposób:

188801,Jan,Maleski,3.0;188802,Antoni,Pietkiewicz,4.0;188832,Ten,Trzeci,5.0

Natomiast w formacie kolumnowym tak:

188801,188802,188832;Jan,Antoni,Ten;Maleski,Pietkiewicz,Trzeci;3.0,4.0,5.0

Powoduje to, że zapisane w poszczególnych kolumnach dane tego samego typu znajdują się obok siebie, co ułatwia wykonywanie operacji na wybranych atrybutach (na przykład obliczenie średniej ocen). Takie struktury spotykane są najczęściej w hurtowniach danych i mają zastosowanie przy przetwarzaniu ogromnych ilości danych. Informacja o położeniu i kształcie (geometrii) obiektów kodowana jest zgodnie ze standardem Well-Known Binary (WKB).

Tabela 1. Przykładowe dane

Nr albumu	Imię	Nazwisko	Ocena
188801	Jan	Maleski	3,0
188802	Antoni	Pietkiewicz	4,0
188832	Ten	Trzeci	5,0

(opracowanie własne)

W obu przypadkach pomysłodawcy zadbali o współpracę z autorami oprogramowania geoinformacyjnego (w tym szczególnie bibliotek programistycznych), już na wczesnym etapie rozwoju zapewniając ich obsługę.

Podsumowanie i wnioski

Z zaprezentowanego przeglądu widać, że nie ma obecnie formatu dla trójwymiarowych modeli miast możliwego do natychmiastowego zastosowania i konieczne są dalsze prace celem zbudowania w pełni funkcjonującego rozwiązania. W przypadku CityJSON byłoby to doprowadzenie do zgodności z wymaganiami podejścia „cloud-native” – w szczególności „pobierz tylko to, czego potrzebujesz”. Natomiast FlatGeobuf i GeoParquet (ten pierwszy pod warunkiem zapewnienia pełnej obsługi danych 3D) można potraktować jedynie jako meta formaty (podobnie jak JSON), na bazie których należałoby zdefiniować struktury danych zgodne ze standardem CityGML.

Podziękowania

Praca finansowana z subwencji Ministerstwa Nauki i Szkolnictwa Wyższego dla AGH w Krakowie.

Literatura (References)

- Barciauskas A., Mandel A., Barron K., Deziel Z. (2023). Cloud-Optimized Geospatial Formats Guide. Cloud-Native Geospatial Foundation. <https://guide.cloudnativegeo.org>
- Bennett S. (2019). Newline-delimited GeoJSON. GitHub. <https://stevage.github.io/ndgeojson>
- Biljecki F., Stoter J., Ledoux H., Zlatanova S., Çöltekin A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4), 2842-2889.
- Biljecki F., Arroyo Ohori K., Ledoux H., Peters R., Stoter J. (2016). Population estimation using a 3D city model: A multi-scale country-wide study in the Netherlands. *PloS one*, 11(6), e0156808. <https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0156808&type=printable>
- Bydłosz J., Cichociński P., Basista I. (2010). The possibilities of geoinformation resources recorded in GML accessing with chosen GIS software. *Geomatics and Environmental Engineering*, 4(1), 33-44.
- Ding L., Xiao G., Pano A., Fan H., Calvanese D., Meng L. (2023). Querying CityGML Data through Virtual Knowledge Graphs. *Abstracts of the ICA*, 6, 53. <https://ica-abs.copernicus.org/articles/6/53/2023/ica-abs-6-53-2023.pdf>
- Eller L., Svoboda P., Rupp M. (2022). A deep learning network planner: Propagation modeling using real-world measurements and a 3D city model. *IEEE Access*, 10, 122182-122196. <https://ieeexplore.ieee.org/iel7/6287639/6514899/09954403.pdf>
- GeoParquet (2024). Geospatial data in Apache Parquet. <https://geoparquet.org>
- Główny Urząd Geodezji i Kartografii (2023a). Modele 3D budynków. <https://www.geoportal.gov.pl/pl/dane/inne-dane/modele-3d-budynkow>
- Główny Urząd Geodezji i Kartografii (2023b). Modele 3D drzew. <https://www.geoportal.gov.pl/pl/dane/inne-dane/modele-3d-drzew>
- Gholami M., Torreggiani D., Tassinari P., Barbaresi A. (2022). Developing a 3D city digital twin: Enhancing walkability through a green pedestrian network (GPN) in the City of Imola, Italy. *Land*, 11(11), 1917. <https://www.mdpi.com/2073-445X/11/11/1917/pdf>
- Green I., DiStefano D. (2024). An Inside Look at Cloud-Native Geospatial Formats. Foursquare. <https://location.foursquare.com/resources/blog/capabilities/an-inside-look-at-cloud-native-geospatial-formats>
- Gröger G., Kolbe T.H., Nagel C., Häfele K.H. (2012). OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0. Open Geospatial Consortium, Doc. No. 12-019.
- Harter H., Willenborg B., Lang W., Kolbe T. H. (2023). Life Cycle Assessment of building energy systems on neighbourhood level based on semantic 3D city models. *Journal of Cleaner*

- Production, 407, 137164.
<https://www.sciencedirect.com/science/article/pii/S0959652623013227>
- Holmes C. (2024). An Exploration of ‘Cloud-Native Vector’. Medium.
<https://cholmes.medium.com/an-overview-of-cloud-native-vector-c223845638e0>
- Ingold T., Walker, Hards B., Little D. (2020). COGJ: Cloud Optimized GeoJSON. GitHub.
<https://github.com/planetfederal/cogj-spec>
- Johansson T., Segerstedt E., Olofsson T., Jakobsson M. (2016). Revealing social values by 3D city visualization in city transformations. *Sustainability*, 8(2), 195. <https://www.mdpi.com/2071-1050/8/2/195/pdf>
- Katal A., Mortezaadeh M., Wang L.L., Yu H. (2022). Urban building energy and microclimate modeling—From 3D city generation to dynamic simulations. *Energy*, 251, 123817.
<https://www.sciencedirect.com/science/article/pii/S0360544222007204>
- Ledoux H., Arroyo Ohori K., Kumar K., Dukai B., Labetski A., Vitalis S. (2019). CityJSON: A compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data, Software and Standards*, 4:4
- Morton P.J., Horne M., Dalton R.C., Thompson E.M. (2012). Virtual city models: Avoidance of obsolescence. In *Education and Research in Computer Aided Architectural Design in Europe—30th eCAADe Conference* (pp. 213-224).
- Nguyen S.H., Yao Z., Kolbe T.H. (2017). Spatio-semantic comparison of large 3D city models in CityGML using a graph database. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 99-106. <https://isprs-annals.copernicus.org/articles/IV-4-W5/99/2017/isprs-annals-IV-4-W5-99-2017.pdf>
- Noardo F., Arroyo Ohori K., Biljecki F., Ellul C., Harrie L., Krijnen T., ... & Stoter J. (2021). Reference study of CityGML software support: The GeoBIM benchmark 2019—Part II. *Transactions in GIS*, 25(2), 842-868.
- Opaleychuk K. (2024). Geospatial Cloud Native Data Overview: Mapping the Future. FME Support Center. <https://support.safe.com/hc/en-us/articles/26440396767501-Geospatial-Cloud-Native-Data-Overview-Mapping-the-Future>
- Peters R., Dukai B., Vitalis S., van Liempt J., Stoter J. (2022). Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogrammetric Engineering & Remote Sensing*, 88(3), 165-170.
- QGIS.org, (2024). QGIS Geographic Information System. QGIS Association. <http://www.qgis.org>
- Rodríguez L.R., Duminil E., Ramos J.S., Eicker U. (2017). Assessment of the photovoltaic potential at urban level based on 3D city models: A case study and new methodological approach. *Solar Energy*, 146, 264–275. <https://core.ac.uk/download/pdf/288003205.pdf>
- Rouault E. (2014). GML madness. Geo tips & tricks. <https://erouault.blogspot.com/2014/04/gml-madness.html>
- Virtanen J.P., Jaalama K., Puustinen T., Julin A., Hyyppä J., Hyyppä H. (2021). Near real-time semantic view analysis of 3D city models in web browser. *ISPRS International Journal of Geo-Information*, 10(3), 138. <https://www.mdpi.com/2220-9964/10/3/138/pdf>.

- Waksmundzka, K., Fijałkowska, A., Chmiel, J. (2021). Wykorzystanie analiz przestrzennych 3D do oceny efektywności instalacji paneli fotowoltaicznych na przystankach komunikacji miejskiej. *Roczniki Geomatyki*, 19(3 (94), 57-64.
- Willenborg B., Sindram M., Kolbe T.H. (2018). Applications of 3D city models for a better understanding of the built environment. In: M. Behnisch, G. Meinel (ed.), *Trends in Spatial Analysis and Modelling – Decision-Support and Planning Strategies*, Springer Verlag. URL <https://mediatum.ub.tum.de/doc/1348882/document.pdf>
- Williams H. (2022). Kicking the Tires: Flatgeobuf. Horace Williams. <https://horace.works/2022/02/23/kicking-the-tires-flatgeobuf>

Streszczenie

Pomimo ciągłego rozwoju metod i narzędzi systemów informacji przestrzennej (GIS) większość zbiorów danych przestrzennych używanych w Polsce jest wciąż dwuwymiarowych. Aby móc przeprowadzać różnego rodzaju symulacje i analizy, potrzebne są modele 3D miast, które będą zgodne ze powszechnie przyjętymi standardami. Takim szeroko stosowanym, otwartym standardem reprezentowania obiektów i struktur w trójwymiarowych modelach miast jest City Geography Markup Language (CityGML). Obserwuje się jednak niewystarczający poziom obsługi danych zapisanych w formacie CityGML przez dostępne oprogramowanie, czego jednym z powodów jest kodowanie eXtensible Markup Language (XML) – rozwlekłe, hierarchiczne, złożone i nieprzystosowane do użycia w sieci. Dlatego stwierdzono, że dla spopularyzowania trójwymiarowych modeli miast należałoby rozwiązać dwa problemy: uproszczenie formatu wymiany danych i zwiększenie jego zgodności z oprogramowaniem GIS. W tym celu dokonano analizy trzech opracowanych w ostatnich latach formatów: CityJSON, FlatGeobuf, GeoParquet. Realizują one podejście określane w języku angielskim jako „cloud-native”, które odnosi się do zarządzania zbiorami danych (w tym przestrzennych), zoptymalizowanymi pod kątem hostingu w środowiskach chmurowych. Dokonany przegląd pokazał jednak, że nie ma obecnie formatu dla trójwymiarowych modeli miast możliwego do natychmiastowego zastosowania i konieczne są dalsze prace celem zbudowania w pełni funkcjonującego rozwiązania.

Dane autorów / Authors details:

dr hab. inż. Piotr Cichociński

ORCID 0000-0002-8633-1235

Piotr.Cichocinski@agh.edu.pl

Przesłano / Received	3.12.2024
Zaakceptowano / Accepted	12.03.2025
Opublikowano / Published	21.10.2025



© Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).