

(10) 3次元プロダクトモデルのグラフDBによる共有

四月朔日 勉¹・矢吹 信喜²

¹正会員 川田テクノシステム株式会社 テクニカルイノベーションセンター 課長
(〒114-0023 東京都北区滝野川6-3-1)
E-mail:t-watanuki@kts.co.jp

²フェロー会員 大阪大学教授 大学院工学研究科 環境・エネルギー工学専攻
(〒565-0871 大阪府吹田市山田丘2-1)
E-mail:yabuki@see.eng.osaka-u.ac.jp

CIM (Construction Information Modeling/Management) で扱われるプロダクトモデルには、土木構造物のライフサイクル全体で発生する膨大なデータが含まれる。そのような巨大なデータを関係者間で共有し、効率良く扱うためには、何らかのデータベースに格納する必要がある。リレーショナルデータベースを利用するのが一般的ではあるが、深い階層構造を有する構造物のプロダクトモデルに対しては、十分な性能を発揮できないと考え、様々な方式のデータベースに格納する方法を検討、考察した。さらに、格納したデータを操作する際の性能を、実験を通して確認し、IFC-Bridgeで記述された橋梁プロダクトモデルを格納するのに適したデータベースの方式を明らかにした。

Key Words : IFC, CIM, BIM, product model, graph database

1. はじめに

現在、日本における土木構造物の設計は、主に2次元の設計図を用いて行なわれているが、欧米で用いられているBIM (Building Information Modeling) に影響を受ける形で、土木構造物を3次元モデルで表現し、ライフサイクル(計画、設計、施行、維持管理)全体で利活用し生産性を上げる試みを、国土交通省が平成24年度からCIM (Construction Information Modeling/Management) と称して開始している。

日本ではライフサイクルの各フェーズは別々の組織が担当する場合がほとんどであるため、多数の関係者間でプロダクトモデルが共有できるだけでなく、複数の組織間で共有できることが求められる。また、CIMで用いられるプロダクトモデルには、土木構造物のライフサイクルの各フェーズで発生する様々なデータが格納されるため、膨大なデータ量を扱える必要もある。

現状、メールやファイル転送サービス等を用いた、ファイルによる共有がなされていると考えられるが、データ量が膨大であるため効率が悪い。よって、将来的にはネットワーク上でプロダクトモデルを共有し、膨大な情報の中から目的の情報だけを抽出することが求められる。

そのためには、プロダクトモデルを何らかのデータベースに格納する必要があるが、一口にデータベースと言

っても、様々な方式が存在する。本研究では、IFC-Bridgeで記述された橋梁プロダクトモデルを多数の関係者間で共有し、そこから必要なデータのみ抽出することを想定した場合に、適したデータベースの方式を明らかにする。

2. プロダクトモデルのデータベースへの格納

IFC-Bridgeプロダクトモデルをあるデータベースに格納する際、格納先データベースが取り扱い可能なデータモデルに変換して格納する必要があり、逆に、データベースに格納されたデータを取得する際はIFC-Bridgeプロダクトモデルへの逆変換が必要になる。データベースの方式の違いは、取り扱いが可能なデータモデルの違いとすることができる。

データベースの方式は多数提案されているが、本研究の最終的な目的は実際にプロダクトモデルを共有することであるから、すでに実用化されていて、十分な実績が存在するものに限定することとし、以下を検討対象とした。

- ・リレーショナル型
- ・キー・バリュ型
- ・ドキュメント指向

・グラフ型

これらのデータベースにIFC-Bridgeで記述した橋梁プロダクトモデルを格納することを考える。

(1) リレーショナル型

関係モデルに基いたデータベースである。IFC-Bridgeの階層構造をそのままモデル化することはできないため、IFC-Bridgeのクラスが関係モデルの表に対応するようにモデル化し、各表に階層構造を表わす関係を定義する。各クラスの属性は、表の列として定義する。

(2) キー・バリュー型

あるキー（通常文字列）に対応する値を格納する非常にシンプルなDBである。性能向上が容易であるといったメリットがある反面、表現力が低く、IFC-Bridgeのような複雑なデータ構造を表現するのは困難である。よって、キー・バリュー型データベースへの格納は行なわない。

(3) ドキュメント指向

自由な形式のドキュメントを格納できる。格納データの記述の自由度が高い。ドキュメントの文法は様々であるが、JSON形式やXML形式である実装が多い。

表現できるデータ構造の自由度が高いため、IFC-Bridgeプロダクトモデルを格納する際も様々なデータ構

造で格納することが可能である。しかし、ドキュメント指向DBにおけるデータ構造の検討は、本研究の範囲を越えるため、単純に1橋梁を1ドキュメントとして格納することにした。

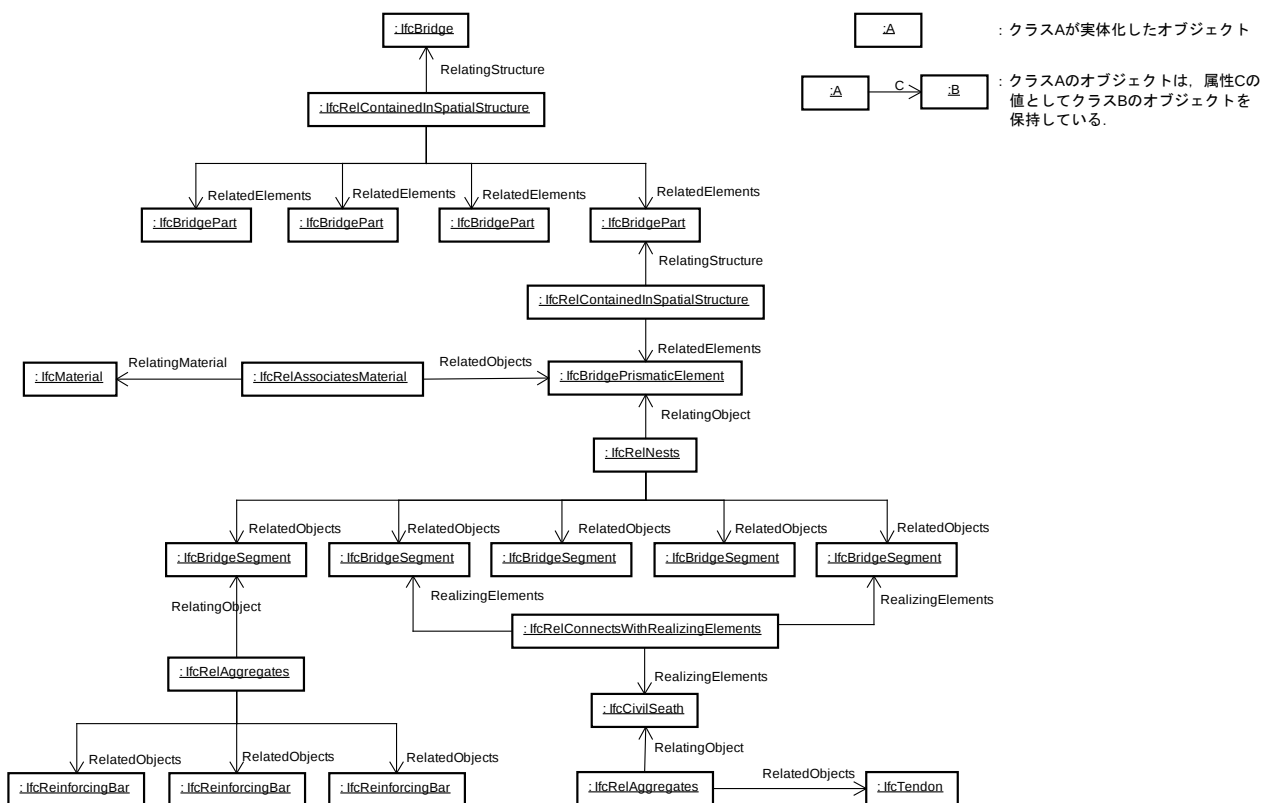
(4) グラフ型

グラフ理論に基づいたデータベースであり、ノードとノード同士を結ぶエッジによりデータを構成する。各ノードとエッジにはプロパティ（キーにより識別される値）を複数保持することができる。このモデルをプロパティグラフモデル^{1,2)}と言う。

以下のルールでプロパティグラフモデルに変換した。

- 1つのIFC-Bridgeエンティティを1つのノードにする。
- エンティティの属性データの内、文字列型、数値型、列挙型である属性はプロパティグラフモデルのプロパティとして格納する。
- エンティティの属性データの内、文字列型、数値型、列挙型以外の属性はプロパティグラフモデルのノードとして表現し、ノード同士をエッジで結ぶ。

IFC-Bridgeで記述された橋梁プロダクトモデルの一部をUML (Unified Modeling Language) のオブジェクト図で表現した（図-1）。IFC-Bridgeのクラスを実体化したオブジェクト同士が関連し、木構造に近い構造を構成してお



り、かつ深い階層構造を有していることが分かる。ドキュメント指向とグラフ型では、これをほぼそのままの形で格納することが可能であるが、リレーショナル型では、多数の表の間の関係として表現しなくてはならない。

3. 実験

IFC-Bridgeで記述した橋梁プロダクトモデルを格納するのに適したDBを明らかにするため、第2章で考案したデータ構造で、以下の製品にプロダクトモデルを格納し、検証を実施した。

- ・SQL Server 2008R2 (リレーショナル型)
- ・MongoDB (ドキュメント指向)
- ・Neo4j (グラフ型)

CIMで扱うプロダクトモデルはデータ量が膨大で、かつ階層構造が深いという特徴があるため、データ量が増加した場合と、扱うデータの階層が深くなった場合の性能を確かめることに着目した。

(1) 実験内容

1. データベースの格納データ量が増加する場合に、橋梁プロダクトモデルの一部分を取得する処理の実行時間の変化を計測する。
2. 取得対象データの階層構造が増加する場合に、処理時間の変化を計測する。

対象データとして、PC箱桁橋をIFC-Bridgeで記述したものを利用した。PC橋梁の場合、通常、桁は施工単位毎にモデリングされるため、橋長が長くなるに従い桁を構成する部材の数（データ量）が増える。そのため、実験1では、操作対象橋梁の桁長を長くすることで操作対象データを増やした。実験2では、取得対象データの階層が変化するように取得クエリを調整した。

(2) 実験結果

実験1の結果を図-2に示す。グラフ型DBでは格納するデータ量によらず、一定時間でデータを取得できることが分かる。これは、グラフ型DBが近接のノードを $O(1)$ の計算量で取得できる^{1),2)}ことを示している。一方、リレーショナルDBではデータ数を n として $O(\log n)$ で処理時間が増加している。これは、グラフ型には劣るものの十分に実用的な性能である。

実験2の結果を図-3に示す。リレーショナル型DBにおいて、取得対象データの階層構造が深くなり、表と表の間の結合が多く発生すると性能が劣化することが分かる。一方、グラフ型DBではこのような劣化は見られない。グラフ型DBで階層をたどる際の計算量は、階層の深さを m とすると $O(m)$ である^{1),2)}が、1階層あたりの計算量が

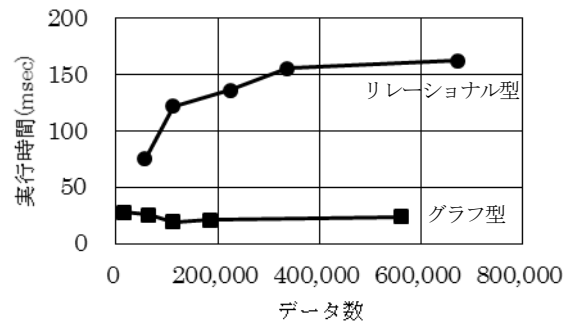


図-2 格納データ数と処理時間の関係

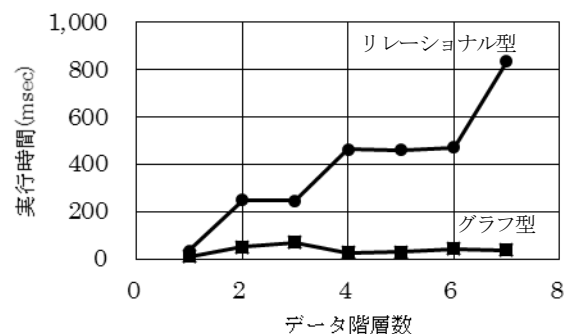


図-3 取得データの階層と処理時間の関係

微小であるため、計測値に現われなかったと考えられる。おそらく100階層を越えるオーダーであれば計測値に現われると思われるが、今回利用したデータはそこまで深い階層は有していないため計測できなかった。

ドキュメント指向DBであるMongoDBでは、格納可能データ（ドキュメント）のサイズに16Mバイトまでという制限があり、大きなサイズのデータを格納できないという問題が発生した。1橋梁分のデータを複数に分割する等、データ構造を工夫することで解決可能であるが、ドキュメント指向DBは格納データの自由度が高いため、データ構造を決定することが難しい。よって、MongoDBを用いた実験は断念した。

4. 結論

本研究の成果、および結論を以下に示す。

IFC-Bridgeで記述したプロダクトモデルを、リレーショナル型、グラフ型のデータベースに格納するためのデータ変換手法を考案した。

上述したデータ変換手法を用いて、IFC-Bridgeプロダクトモデルを具体的なデータベース製品、SQL Server 2008R2、Neo4jに格納し、それぞれのDBからIFC-Bridgeプロダクトモデルを取得することに成功した。

CIMで扱われるプロダクトモデルには、ライフサイクルのあらゆるフェーズの情報が格納されるため膨大で、かつ深い階層構造を有している。よって、3章で示したように、データ量が増加しても性能が劣化せず、階層構造が深いデータの処理が高速なグラフ型データベースが、プロダクトモデルの共有に適していると言える。

本研究の成果により、プロダクトモデルをDBに格納し、関係者間で共有することが可能になる。今後の課題として、共有したプロダクトモデルを活用するための、差分抽出や履歴データ保持を考慮したデータ構造の考案が挙げられる。

参考文献

- 1) Robinson, I., Webber, J. and Eifrem, E.: *Graph Databases 2nd Edition New Opportunities for Connected Data*, O'Reilly, 2015.
- 2) Robinson, I., Webber, J. and Eifrem, E. (佐藤直生監訳, 木下哲也訳) : *グラフデータベース Neo4j によるグラフデータモデルとグラフデータベース入門*, オライリー, 2015.
- 3) Lebegue, E., Fies, B., Gual, J., Arthaud, G., Liebich, T. and Yabuki, N.: *IFC-BRIDGE V3 Data Model - IFC4 Edition R3*, 2013.
- 4) buildingSMART International: *Industry Foundation Classes IFC4 Official Release*, <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/>, (2014).
- 5) 矢吹信喜, 李占涛: 日仏橋梁プロダクトモデルの統合化による新 IFC-BRIDGE の開発と CAD コンバータの改良, 土木情報利用技術論文集, Vol.15, pp.59-66, 2006.
- 6) 矢吹信喜, 志谷倫章: PC 橋梁の 3 次元プロダクトモデルの開発と応用, 土木学会論文集, No.784/VI-66, pp.171-187, 2005.
- 7) 矢吹信喜, 蒔苗耕司, 三浦憲二郎: *工業情報学の基礎*, 理工図書, 2011.