［事例実践論文］

研究データ管理における検索機能の実装とその変遷
Implementation and Evolution of Search Functions in Research Data Management

菊地 伸治†，田辺 浩介†，坂本 浩一†，高田 安裕†，傳法 春樹†，
門平 卓也†，谷藤 幹子‡
Shinji KIKUCHI，Kousuke TANABE，Koichi SAKAMOTO，Yasuhiro TAKADA，Haruki DENPO，
Takuya KAROHIRA and Mikiko TANIFUJI

†国立研究開発法人物質・材料研究機構 材料データプラットフォームセンター
‡国立情報学研究所 オープンサイエンス基盤研究センター
† Data Platform Center, National Institute for Material Science
‡ Research Center for Open Science and Data Platform, National Institute of Informatics

要旨

研究データ管理において保持するデータの再利用を促すためには，そのデータに関するメタデータの記述充実度や検索の容易性が重要になる．研究の専門性が高くなるほど，データ記述で利用される語彙やデータ構造は概念上の深化・特化が進むため，その概念が持つ意味合いを反映した問い合わせ処理の実現が必要になる．さらに，先進性の高い研究データであるほど，高い秘匿性が求められ,アクセス制御についても高度化が求められる．本稿では研究データ管理のケーススタディとして，国立研究開発法人物質・材料研究機構における研究データ管理(RDM)の機能要求の変遷と検索機能の進展に関して概説し，アーキテクチャを更新・拡張させる過程で実装された検索機能の二つの形態に関して多論点から比較評価を行い，技術的妥当性に関する一つの示唆を述べる．

Abstract

In order to promote the reuse of data held in Research Data Management (RDM), the richness of the metadata description and the search validity of the data are important. The higher specialization in a research discipline, the more conceptually specialized the vocabulary and data structures used in data description become, so it is necessary to implement query processing that reflects the connotations of these concepts. Furthermore, the more advanced the research data is, the more confidentiality is required, and the more sophisticated access control is also required. In this paper, as a case study, we give an overview of the evolution of requirements on RDM and the development of retrieval functions at the National Institute for Materials Science, and compare two approaches implemented through updating and extending the architecture from various perspectives including performance and functionality. An evaluation with comparison between them is made and suggestions on technical appropriateness against these requirements is tried to be presented.

1. はじめに

材料科学分野では世界規模でMaterials Informaticsと呼ばれる材料科学とデータ科学を融合した取り組みが進展している．Materials Informaticsとは，蓄積された膨大な実験データ，計算機能力の向上により算出可能となった膨大な計算データを入力として，統計学，パターン認識等のデータ解析技法を用いてプロセスと特性間，あるいは異なる特性間に成り立つ法則性を抽出・発見・予想する，加えて論文からのテキストマイニングで得られる大量データを用いて材料探索を行うなど，機械学習や深層学習などの情報学の知見を用いて新たな材料開発を加速することを含意する[1], [2], [3].「第四の科学手法」の提唱から 10 年以上を経過した今日，類似の動きは材料科学分野に限らず一般化されたパラダイムとして定着・進展し，クラウドコンピューティング・機械学習の成熟化に伴い深化し続けている[4], [5], [6]．その結果，研究開始の創出期から実用に向けた適用応用期までに創出される一連の研究データをシームレス・高品質に管理・提供できる研究データ管理プラットフォームの構築と実現が分野横断に要求され，実装が進んでいる[7], [8]．国立研究開発法人物質・材料研究機構においても, Materials Informatics およびデータ駆動型研究環境の実現に向けて材料データプラットフォーム’DICE’の構築と提供を進めているが，多様で異質な計測・研究システム群を用いて実施される実験・シミュレーション等の研究活動の結果と

［事例実践論文］
2022 年 12 月 28 日受付, 2023 年 5 月 8 日改訂, 2023 年 7 月 20 日受理

して産出される研究上の一次データを組織的に管理する RDM(Research Data Management)サービスの定着化と運用最適化が当初からの課題であった[9], [10].

本稿では, 材料データプラットフォーム’DICE’における研究データ管理(RDM)上の機能要求の変遷と, 顕著に影響が見られた検索機能の進展を概説する. [11]に記されたように科学技術領域でのデータ参照に関する要求は複雑であり, 単なるシステムに対する要求に留まらず, 本質的に多様な要求・課題群を含んでいる. 本稿では, 当該要求・課題群の構造的な分析の上で包括的な解決策を定義するのではなく, むしろ要求の変遷に応じてアーキテクチャを更新・拡張させる過程で実装された検索機能の二つの形態に関して多論点から比較評価を行うことで, 技術的妥当性を評価することを試みる. なお, 本稿の刊行前の段階で材料データプラットフォーム’DICE’が商用クラウドコンピューティング環境上での運用に移行する際に研究データ管理(RDM)の運用・構成が見直された. そこでは利用・運用両面で最適化したエコシステム実現を志向し, 研究データの構造化から解析迄をより効率的に行うよう研究データ管理(RDM)を独立した機能・サービスとしてではなく, 観測・測定作業等で利用される IoT (Internet of Things)デバイスとのデータ共有機能と統合・一体化したサービス構成に更新された. これに伴い, 本稿説明の検索機能の実装も発展的に上記サービスに吸収され, RDM サービス単体の実装は廃止に至っている.

以下, 本稿の構成を述べる. 続く 2.ではシステム全体構成を鳥瞰し, 研究データ管理基盤における検索機能の位置付けとともに, アーキテクチャ上の変遷について説明する. 具体的には, PostgreSQL を用いた QBE(Query by Example)指向の初期実装から全文検索エンジンである Elasticsearch を採用した実装への移行と, その背景や扱うメタデータの構成概要, アクセス制御を記す. 3.では初期段階で開発した PostgreSQL を用いた QBE 指向の実装について概説する. これに対して 4.では Elasticsearch を利用した実装を概説する. 5.では 3, 4.で記した各実装に関する性能特性や Elasticsearch の強み・弱みの点から各種評価を実施する. 6.では関連先行研究を中心に説明する. 最後の 7.では本稿の貢献点を示し, むすびとする.

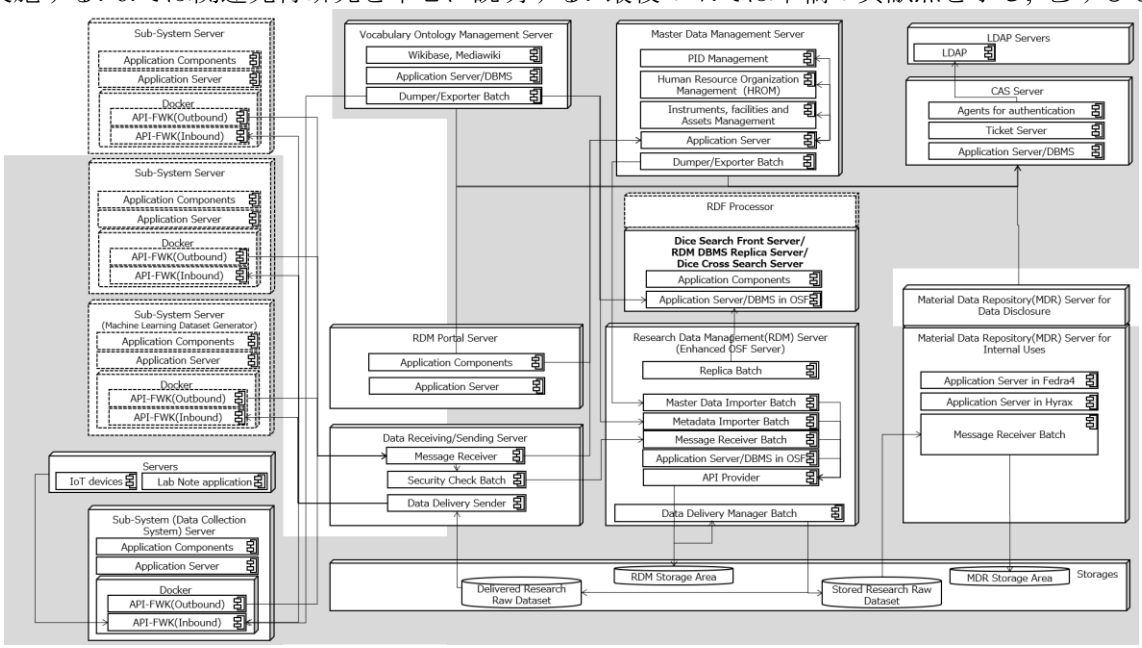


図 1 UML 配置図による全体アーキテクチャ [12], [13]

2. アーキテクチャ概要と変遷

2.1. 検索機能の位置付けと概要

図 1 は, [12], [13]で記された当該プラットフォームの上位構造であるアプリケーション群の構成要素を記した UML（Unified Modeling Language）配置図をもとに, その後の差異部分を含んで更新した配置図である. 一つの直方体で記されるノードは物理マシン, 仮想マシンのいずれかで実装されるサーバ等のサイトを意味し, これらノード間の矢印線は, UML 配置図で本来指定されるべき接続とその種類ではなく, 呼び出し関係（依存関係）として描いている. 図中, 灰色で塗り分けられている部分は, 当該プラットフォームを構成するネットワークのうち, セキュアな内部セグメント領域であり, それ以外は DeMilitarized Zone（DMZ）上のセグメント等に相当する. [12]に基づき機能的に分類すると, 大きく 2 つから構成される. 図中の下半分は業務プロセスを扱う機能群に相当し, 左側から研究データの発生源, その統制機能である Sub-System Server 群, Research Data Management (RDM)に代表される研究データ管

理機能, そして Material Data Repository(MDR)のような研究データの公開機能であり, これらが連携して機能する. これに対して図中上半分は, この業務プロセスを支えるための情報資源管理機能, シングルサインオン等のユーティリティ機能に相当する. 詳細は[12], [13]に基づくが, 表.1 に主要ノード群について [12]に基づき概説した. [12]の時点から新たに発生した差異事項は’RDF Processor’の取りやめ, 代替として新たに’Dice Search Front Server’以下を実装・配置した点である.

研究データ管理において保持するデータの再利用を促すためには, 高度で妥当な検索機能の提供が必要であり, そのためにはデータに関するメタデータの記述充実度や発見の容易性(Discoverability)が重要になる. 発見の容易性を実現するには, データ記述で利用される語彙や概念構造を反映したメタデータを最大限に利用することが要求される. このため, [12], [13]にて報告されたプロジェクト当初段階では, 集積・管理されたメタデータや内部データの来歴管理情報(プロビナンス)を RDF 形式でダンプして取り込み, 多様な関連検索機能を提供する構想の下で検討を進めてきた. しかし, 実際の構築から試験運用を進めると, 汎用目的ゆえに一般性の高い RDF 化に対する利用者の要求は低く, むしろ検索の容易性が求められていた. 具体的には, (i)同一元素を対象とする異分野の複数の計測結果を含んだ検索, (ii)物理量を指定する検索, (iii)同一測定装置による複数の計測結果を含んだ検索, (iv)統制語彙の指定による検索である. さらには多様な項目を条件式の記述をせずに容易, かつ複合的に指定し得る機能である. また, 先進性の高い研究データを扱うほど, 高い秘匿性が求められ, 潜在的にアクセス制御の高度化も求められる. 以上に基づき, 2.2 ではメタデータの構造について, 続く 2.3 では検索機能の初期アーキテクチャ, 2.4 ではアクセス制御について概説する. 最後の 2.5 では要求上の進展と実装に与えた影響を概説する.

表 **1** 主要ノードの定義（[12]をもとに抜粋）

ノード	定義
Research Data Management (RDM) Server	周辺システムに相当する各種サブシステム(Sub-System Server)群から生成される研究データ(メタデータ, 保管対象データ)を集積・管理の上で 流通管理を行う機能である. 実装に当たっては OSS の OSF(Open Science Framework,[18])を採用しており, バックエンドサーバとして機能する.
Data Receiving/Sending Server	周辺の各種サブシステム(Sub-System Server)群から生成される研究データを集配信する機能である.保管対象データのセキュリティ検証, マルウエア対策を実行する機能を含み, 研究データの品質の維持管理を担う基盤としても機能する.
RDM Portal Server	利用者が当該プラットフォームを操作する上で GUI を提供する機能である.
Sub-System Server, API-FWK	研究データを“つくる”,“使う”に相当し, より付加価値の高い研究データを生み出す一連のサブシステム群である. 材料科学の分野毎に多様なレガシーデータベース, 情報システムが存在している. 具体的には各種計測装置から集積した研究データを管理するシステム群等である. これらは各々, その時々にそれら自身への要求事項に従い開発されたもので, 実装環境・言語に標準・統一性を期待することは出来ない異種システム群である. この様なサブシステム群からデータを移送するため, その通信処理を標準化しサブシステム群と連携する専用アダプタが API-FWK である.
Material Data Repository (MDR) Server	Research Data Management (RDM) Server に集積・管理された保管対象データを“公開する”際に利用する機能である.
Storages	巨大で多様なデータ類を保管する領域である.

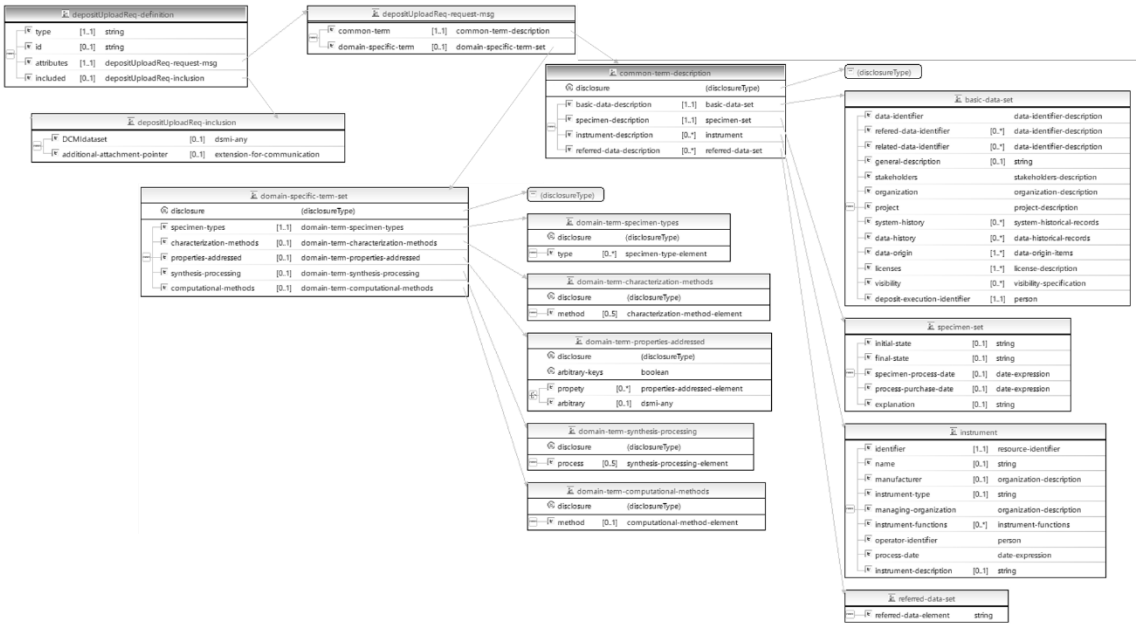


図 2　メタデータの形式定義(抜粋) [12]

2.2. メタデータの構成

図 2 は[12]にて概説された, 当該プラットフォームで流通・交換・保管されるメタデータの主要構造である. これを[12]の記述に基づき説明すると以下のようになる.

“当該プラットフォームのメタデータは, XML Schema で定義の上, Json.schema 形式に変換して利用される. このメタデータは流通・交換・保管の各々で利用されるため多面的な要素を持ち, 単に保管対象データ記述の側面だけではなく, 研究データ登録のための通信メッセージとしても機能する. 保管対象データ記述としては, 研究データそれ自体の書誌情報, 材料科学の論点でその研究データの特徴を記述するための前述分野内容に基づく項目群も含んで構成される. 以上から複雑な構造を内包する. 図 2 の最左上位の要素が depositUploadReq であり, これが記述の最上位になる. この要素とその構成要素である depositUploadReq-request-msg 等は研究データ記述の側面よりも, それを交換する上での通信メッセージの側面が強く, それが故に標準的な JSON:APIv1.0 の記述流儀を継承している[14]. depositUploadReq-request-msg 要素は, 二つの構成要素である common-term-description と domain-specific-term を含む. 前者の common-term-description 要素は研究データに関する書誌情報を記述するためのものであるが, 一部には材料, ならびに装置に関するサマリー・概要情報を含む. 当該 common-term-description 要素の主要部は basic-data-set 要素であり, これは当該プラットフォーム内で長期に渡り, 研究データを識別・一意性を保障するための記述で, PID(Persistent IDentifier)でもある data-identifier 要素や, そのデータの生成母体となったプロジェクト識別子 project 要素, 生成日時等の履歴情報である data-history 要素を含んで構成される. 対して後者の domain-specific-term 要素は, 前述分野内容を具体化した項目群であり, 計測(domain-term-characterization-methods), 物質材料(domain-term-specimen-types), 特性(domain-term-properties-addressed), 合成・プロセス(domain-term-synthesis-processing), 計算(domain-term-computational-methods)の各記述要素を含んで構成される. 前述のように当該メタデータは, 多様な材料科学研究の方法・ユースケースに対して可能な限り共通的に適用されることを目的としているため, 前述 5 要素は選択的に利用される.”

上述の説明に追加で特記すべき点は 2 点ある. 第一は, 図 2 に記載したメタデータの主要構造は記述構造のみを定義するものであり, 特に実際の物質材料(domain-term-specimen-types)の記述では事前に定義した統制語彙を用いる. 統制語彙は当該研究機構で標準として開発した材料分野のオントロジのサブセットを事前に受信し, 保持管理される. 記述の際に語彙を特定すると, メタデータ上ではその語彙に関連付いた URI で記述される. 第二は, 特性(domain-term-properties-addressed)の重要性である. ここでは計測された物理値等が単位系を含んで記述される. 単位系は前述の統制語彙の一つとして扱われる. 物理値はスカラ量, 行列等の多様な型を持つことができ, その表現形式も一定程度考慮されているが, 記述上のコンフォーマンスを確立するまでには至らなかった.

なお, このメタデータの構造については, [15]のように当該研究機構外にも公開・開示されたが, 機能的多面性や材料科学上の特定項目を含むゆえに記述階層が深く複雑であったため, 適用にあたっては困

難な点がある旨の批判も存在した.

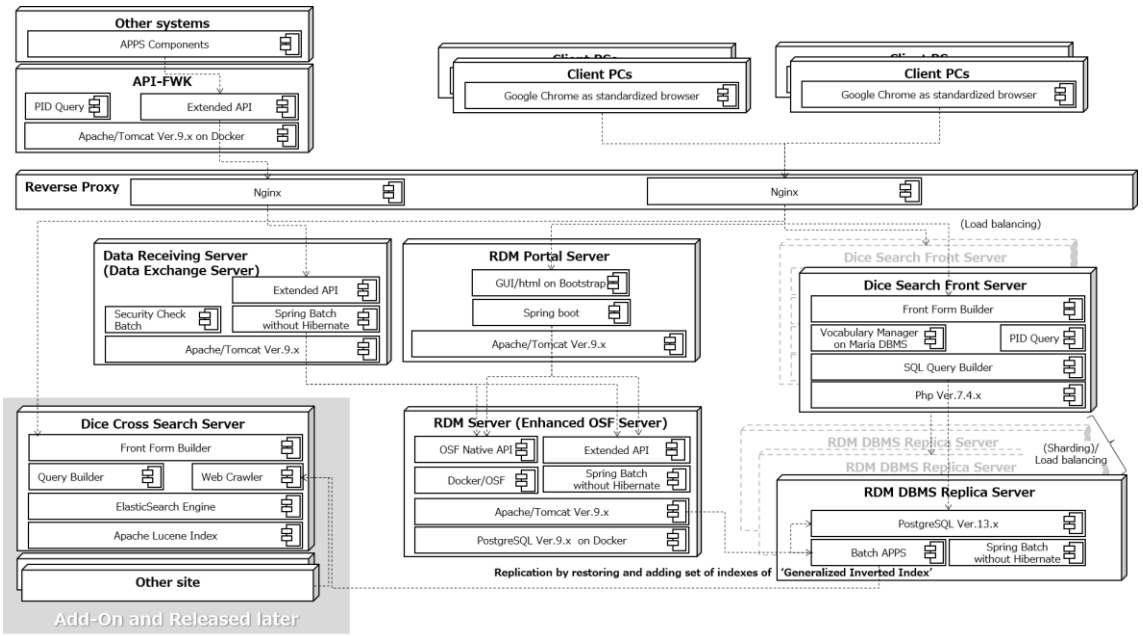


図 3　UML 配置図による検索機能の論理アーキテクチャ

2.3. 検索機能の全体アーキテクチャの概説

図 3 は, 図 1 の UML 配置図から研究データ管理を供する機能要素群を取り出した上で, 検索機能を提供するサーバ群をさらに詳細化した論理アーキテクチャ上の配置図である. 実際の物理的実装はこれに基づくが, 一部は異なる. 図 3 中で検索機能を実現するサーバ群は Dice Search Front Server, RDM DBMS Replica Server, 及び4.での説明に該当する Dice Cross Search Server である. 利用者に対するクライアントは Web ブラウザであり, Google Chrome を標準としている.

検索機能の初期の実装では, 検索容易性を実現することを優先的に目指した. このため Dice Search Front Server, RDM DBMS Replica Server は, 3.にて後述する QBE 指向の機能を提供する. Dice Search Front Server は画面上での操作等のフロント処理, RDM DBMS Replica Server はメタデータ等を管理する機能であり, PostgreSQL 13 を用いてメタデータを JSONB 型 (JSON Binary, JavaScript Object Notation Binary)として管理している. QBE とは, 例示による問い合わせ (Query by Example)の意味であり, 詳しくは[16]で説明されている. これは検索条件を指定する際に利用者の利便性を考慮して可能な限りビジュアルな画面操作で該当問い合わせができるようにしたもので, 主機能は Dice Search Front Server 内で提供される. 詳しくは 3.にて説明する. 画面上の操作の後, フロント処理の一環として等価 SQL に変換し, RDM DBMS Replica Server に問い合わせる. 画面構成については後述するが, 2.2 のメタデータの JSON Schema を参照することで当該メタデータと等価なフォームを自動生成する. RDM DBMS Replica Server は, 全ての研究データ群を管理する RDM Server 上の PostgreSQL 9.6 の完全レプリカを保持しており, Shared Nothing で複数 Server 群を配置することが可能である. 単位時間内の問い合わせ件数が高頻度になり, スケーラビリティに対する要求が強くなった場合は, シャーディング(Shading)を図り一定の応答品質を維持できるようにしている. これはフロント処理を行う Dice Search Front Server も同様であるが, 実際には 1 ペアで運用している.

Dice Search Front Server と RDM DBMS Replica Server は図 1 のアーキテクチャ構造を前提としているため, 先行実装した研究データ群を管理する RDM Server のアーキテクチャ構造に依存して実装されている. そのため, データ記述で利用される語彙や概念構造を反映したメタデータ構造に強い依存性のある問い合わせであっても, 当該メタデータに従う限りにおいて問題は顕在化しない. しかし Materials Informatics が世界規模で進むほど, 新たな課題も顕在化した. その背景を含めて 2.5 で説明する.

2.4. アクセス制御の概説

アクセス制御は, 多層的な制御により実現している. [13]では当該検索機能も利用する CAS(Central Authentication Service, [17])ベースの RDM 認証・認可機構について概説しているが, これは, システムの利用権に関する制御に留まる. 先進性の高い研究データゆえの高い秘匿性を維持するためのアクセス制

御は, CAS で制御する認可層ではなく, これを基層とした上位の研究データ, メタデータ等のコンテンツ管理に対する認可層になる. 3.で説明する実装では, 研究データを管理する RDM Server 内に実装される OSF(Open Science Framework)[18]のプロジェクトごとのアクセス制御にマッピングしている. これに基づき, 各プロジェクトに参加する利用者は厳密に管理され, 然るべきアクセス権が付与される. 前述のように RDM DBMS Replica Server は, RDM Server 上の PostgreSQL 9.6 の完全レプリカとして実装されるので, このアクセス制御機能が継承される. この結果, 検索されたデータに対して, その利用者の所属プロジェクトで定義されたメタデータか否かを評価し, アクセス制御上, 許容できないものは応答の際に除外される.

4.で説明する実装では, そのようなアクセス制御方式を適用することができない. このため, メタデータを拡張してアクセス権項目を付与した JSON メタデータ群を RDM DBMS Replica Server 内のバッチ処理で作成, クローラ(Web Crawler)が取り込む際にアクセス制御情報に関する参照制約として取り込んでいる. 所望のアクセス制御は, 以上のように複合的な連携により実現されている.

2.5. 要求上の進展と実装に与えた影響

Materials Informatics が世界規模で進むほど, 研究データは一組織内だけでの再利用に留まらず, 複数組織と共有することが求められてくる. この結果, 機能サービスとしての研究データ管理についても単独組織のみでの運用が許容されないものになる. 具体的には, 研究分野で[19]のようなデジタルトランスフォーメーション(DX)化が必須になると, クラウドコンピューティング環境下で独立した複数組織での利用運用や, 研究データ管理自身に対する商用サービスの適用が高い優先事項になる. この結果, メタデータの発生源や研究データの管理主体は複数組織にまたがることになる. いわば “連邦制”の適用と具体化が必要になる. この“連邦制”とは, 例えば[20]では以下のように説明される.

> “サービス基盤が連邦制運営されているとは, 独立して運営されているサービス基盤の運営組織が, 一定の合意の下にそれぞれのサービス基盤を接続し, サービスの相互利用を可能にしている状態を指す.”

上記の一定合意には様々な水準があり, その一つにはメタデータやプロトコルの標準適用や, サービスレベルの合意が相当する. しかし 2.2 で示したメタデータの定義があったとしても第三者公的機関による標準化がされていない, もしくは De facto 標準にも至らない未成熟な段階では, 完全な連邦制への移行自体が困難になる. 連邦制による情報検索サービスでは, [21]に代表されるメタデータのハーベスティングが用いられる場合があるが, それでも検索精度を維持するにはメタデータの統一性も優先的に考慮するべき事項である. 特に 2.2 に記すメタデータが複雑ゆえに適用に困難となる事態は, そのまま“連邦制”を運用していく上での障壁になり, その結果, メタデータ構造に強く依存した 3.の実装だけに依存することは, DX 実現に対しては逆に単なる障害になる. 機械可読性を必須要件とするワークフロー自動化や装置のみでの利用を前提とせずに人間系での運用が中心となる場合, DX 化に向けた最低限の連携を目指すに当たっては, 運用可能な複数種類のメタデータ定義の許容や, 例えば, そこでの必須項目の無記入並びにデータ適合性緩和などのメタデータの用法や運用基準をルーズ化させることも妥協点として受容するべきことになる.

上記背景のもとで, 2.2 のメタデータ形式に沿わないデータについてはヒットを保証しない旨の検索精度低下の受容等, 大幅な要求の緩和の下で 4.で説明する全文検索エンジンを利用した実装が, そのソリューションとして求められるに至った. この結果, 図 3 上の Dice Cross Search Server に相当する機能を Elasticsearch によって実装した [22]. この場合, 2.2 のメタデータの適用はベストエフォートとして取り扱われることになる.

以上から検索に対する要求上の進展を総括すると, 初期段階の「曖昧性を持った汎用目的の機能」から第二段階の「データ記述で利用される語彙や概念構造を反映したメタデータによる精緻な問い合わせ処理」, さらには第三段階の「メタデータ発生源や研究データの管理主体の複数組織化, それらの連携を考慮した問い合わせ処理」に変化している. これに応じて, 当初の汎用的な RDF 技術の適用, 第二段階ではより精度を上げるための QBE 指向の適用, 最後の第三段階では検索品質の低下を許容することを引き換えにメタデータ用法やメタデータ運用基準のルーズ化を前提とした全文検索エンジン適用, と進展してきている. ただし当該要求上の進展は, 現実にはシステムを運用する組織の研究戦略上の進展に応じて経時的に発生する要件に基づくものであり, 必ずしも個々要求間の包含関係が明確にされているわけではない. 従って要求を反映した機能群が進化的に発展しているとまでは説明できるものではなく, 現実には局面最適化され, 一部には背反的な実装に留まることも否めない.

3. 初期の検索機能の実装

前述のように初期実装では，材料情報の検索容易性を実現することが最大の狙いであったため，QBE 指向の実現に力点を置いたものになった．図 4 では処理概要のシーケンスを，図 3 上の Dice Search Front Server, RDM DBMS Replica Server, 利用者のクライアントであるブラウザ間で UML 配置図に基づき記述している．最初に CAS Server を利用してシングルサインオンが実行され，これにより検索機能である Dice Search Front Server が提供する QBE ベースの入力フォームと研究データを管理する RDM Server がシームレスにアクセス可能になる．ログイン処理が実施されると，図 5 の(i) に記される QBE ベースの入力フォーム画面が JSON Schema から自動生成され，ブラウザ上に表示される．ここでは QBE の特徴である例示を，種々の方法で実現している．例えば選択項目については，全てラジオボタン選択ができるように展開表示され，文字列項目はプルダウンリストに対応付けている．また後述するような複数の検索ダイアログも利用可能である．これにより利用者は，事前の関連知識が無くとも問合せ内容を指定できるようになる．これらは図 4 上では[Viewing Forms～Inputting a query Conditions]の処理に相当する．

入力フォーム画面上で条件指定後に，図 5 上の[SEARCH]ボタンを押下すると，検索が実行される．具体的には図 4 の[SQL Building & Handling～Throwing a query]を実施し，続けて RDM DBMS Replica Server に対して SQL 記述を送付する．図 6 では，この周辺の中核的処理を記載しており，後に説明する．RDM DBMS Replica Server 内では 2 フェーズで SQL を実行する．Phase.1 では該当条件に基づき RDM DBMS Replica Server 内で管理するメタデータの全 JSON インスタンスに対して該当するものを抽出する．Phase.2 では Phase.1 実施後の検索結果に対して，アクセス制御として利用者の所属プロジェクトで定義されたメタデータか否かを評価し，参照権限のあるもののみを許可するフィルタリングを実施する．当該フィルタリングで残された ResultSubSets からアクセス可能とする URL を Set ごとに作成し，図 4 上の[Viewing a Result Set～Selection by users]を実施するため，ブラウザに一覧形式で戻す．アクセスを可能とする URL を押下すると，RDM DBMS Replica Server ではなく，実際の研究データを管理する RDM Server に直接アクセスし，サムネイル画像，必要情報の表示を行うとともに研究データのダウンロードも可能とする．Phase.1 を高速化するために，RDM DBMS Replica Server へのレプリカ生成をした直後に JSON Schema から定義された GIN(Generalized Inverted Index) インデックス生成スクリプトを実行する[23], [24]. GIN インデックスは JSON 上の全 path に対して定義されている．

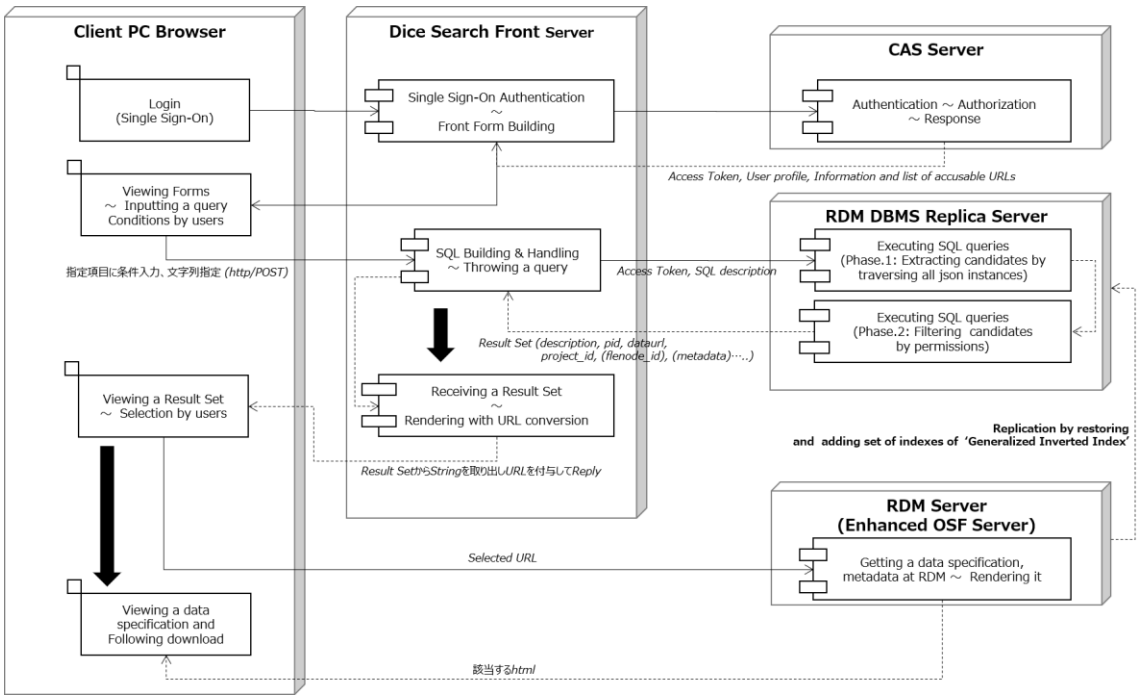


図 4　処理概要のシーケンス

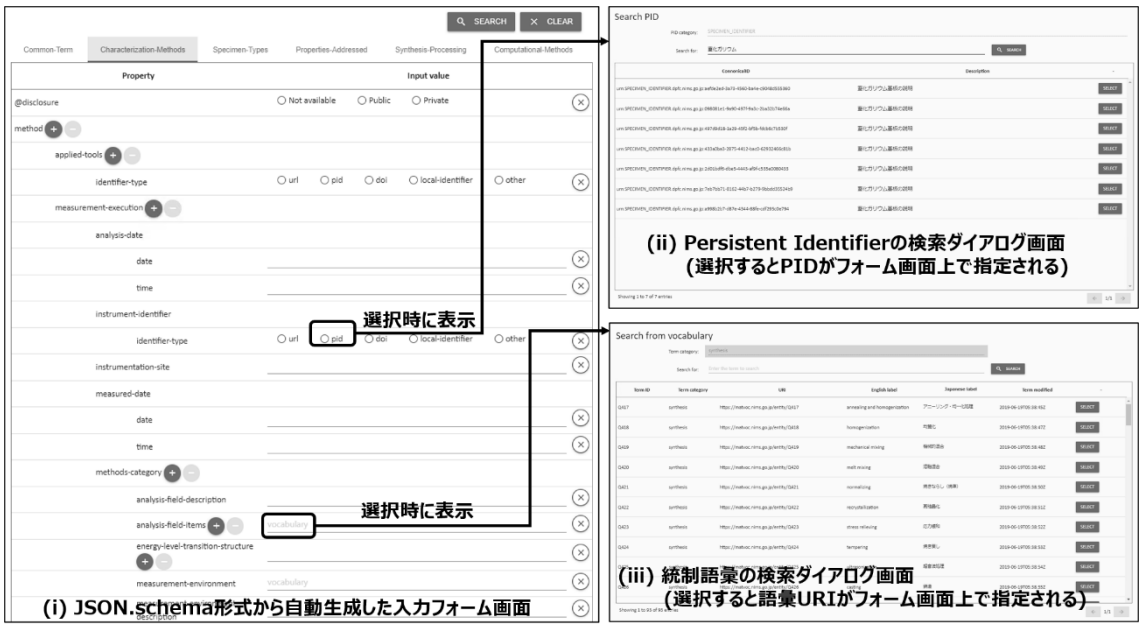


図 5　入力フォームの概要

図 5 で条件入力をする際の利便性を向上するため，複数の入力支援のサブ機能が実装されている．例えば PID を検索対象として指定する場合, (ii)Persistent Identifier 検索ダイアログ画面が表示される．その際, PID を管理するサーバに対して API を介して問い合わせを実施し，該当する検索結果を入手する．その後，利用者が所望のものを選択すると，入力フォーム上に指定表示される．このようなダイアログは標準的な統制語彙に対しても適用している．具体的には’vocabulary’とガイドされる項目を押下すると，(iii)統制語彙の検索ダイアログ画面が表示される．前述のように統制語彙は材料分野のオントロジのサブセットを Dice Search Front Server があらかじめ定期的に受信し，適宜更新している．(iii)統制語彙の検索ダイアログ画面で所望の語彙を選択すると，その語彙に付与された URI が入力フォーム上に指定表示される．2.2 で説明したメタデータは, PID や語彙の URI を標準記述形式として含んでおり，一部 RDF トリプルの側面も含んで記述されている．

JSON Schema 形式は図 6 の記載のとおり, QBE の入力フォームの自動生成だけではなく全てのメタデータインスタンスの検証にも参照される．利用者がフォーム画面上で入力条件を指定すると，図中グレーで囲まれた Dice Search Front Server 内の[SQL Building & Handling～Throwing a query]の相当処理を実行する．ここでは(ii)の画面入力に従い, (iii)の等価中間表現である JSON インスタンスを生成する．その後，図 6 内の Table 内で指定される変換規則に従い, PostgreSQL の JSON 演算子を含む(iv)の SQL のサブ記述を生成する．これらは実行の際に WHERE 句に含まれる．当該実装では基本的に全て文字列型に置換するため，潜在的に要求のあった数値型の範囲指定等の検索はできない．また図 6 内の Table で明らかなように，1 項目内では OR である選言(disjunction)のみを許容するが，複数の項目間では AND である連言(conjunction)も許容する．図 6 の変換処理による SQL のサブ記述のマッピングの例を図 7 と図 8 に記載する．図 7 の最初のものは入力フォーム上でラジオボタンに相当し，これは複数指定されると全て AND である連言に置換される．これに対して同じ項目を配列として複数指定する場合は OR である選言での連結に置換される．図 8 は階層的に組み合わされて指定された場合であり，同配列要素下の異種要素間は AND である連言に置換，配列要素が異なると OR である選言での連結に置換される．これらのマッピング規則は，検索容易性の追求により QBE による例示を高度化するほど，その等価な問合せの SQL のサブ記述を生成するために追加が必要であり，入力フォームでのユーザビリティを高めるほど，等価なサブ記述を生成する規則数はより多く複雑になる．以上の置換処理に基づき，概念構造を反映したメタデータに対する問い合わせ処理が実施可能となる．

4. Elasticsearch を利用した検索機能の実装

2.5 で前述したように検索機能は大幅な見直しを行うことになった．全文検索エンジンである Elasticsearch を利用した実装の論理アーキテクチャは，図 3 の左下部分である Dice Cross Search Server に該当する．ここでは RDM DBMS Replica Server 内で管理され，バッチ処理で定期的に掃き出した JSON インスタンス群をクローラにより Elasticsearch のインデックスに取り込む．このため，RDM Server を介

さない他システムからの JSON ベースの等価ドキュメントも潜在的に取り込めることになる. 2.5 で前述したように要求を緩和し，広域なシステム群内のデータを対象とすることを意図したゆえに Cross Search と命名されている. Elasticsearch のインスタンス配置は試験運用の途上であることも手伝い，特徴的なマルチノードへの配置は実施しておらず，単一インスタンスのみである. GUI については可能な限り 3.の初期実装である QBE の入力フォームを踏襲しており，表面上の操作性に大きな差異はないようにしている.

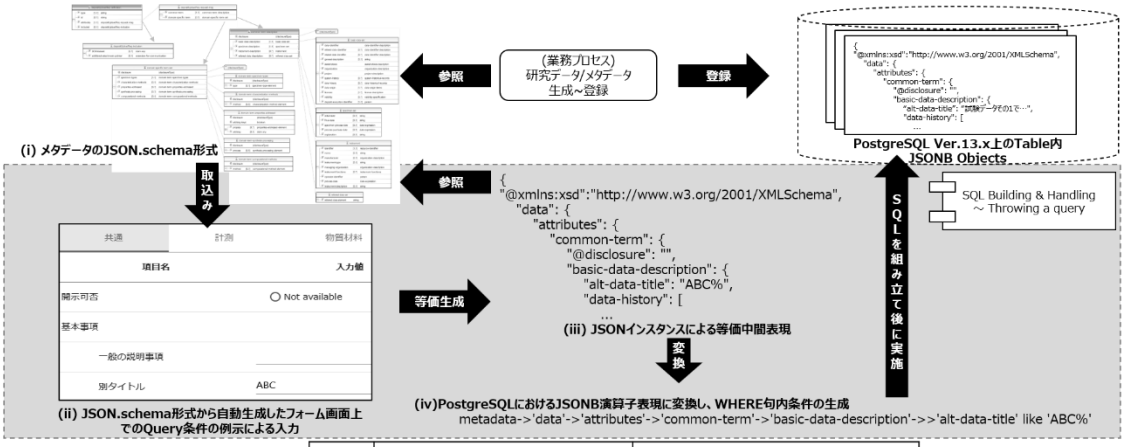
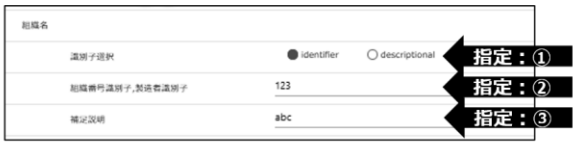


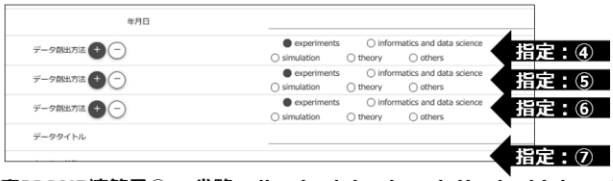
Table. フォーム画面の入力表現のSQL上の変換結果

Cases	Specified terms on the Forms	Expressions in SQL Where Clause
#1.	'ABC'	like 'ABC'
#2.	'%BC'	like '%BC'
#3.	'AB '	like 'AB '
#4.	' '	like ''
#5.	'' ※No spaces	Not Specified
#6.	'ABC'|'xyz'	'ABC' OR 'xyz'
#7.	'AB ' | 'xyz'	'AB ' OR 'xyz'
#8.	' ' | 'A' | 'xyz'	' ' OR 'A' OR 'xyz'

図 6　メタデータを基にした変換処理の概要

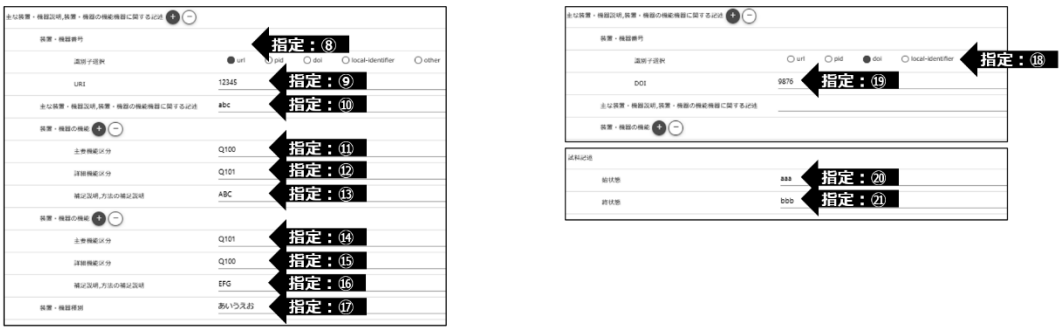


要素JSONB演算子①： 省略->'basic-data-description'->'organization'->>'identifier-type' like 'identifier'
要素JSONB演算子②： 省略->'basic-data-description'->'organization'->>'organization-identifier' like '123'
要素JSONB演算子③： 省略->'basic-data-description'->'organization'->>'description' like 'abc'
連結演算： WHERE (① AND ② AND ③)



要素JSONB演算子④： 省略->'basic-data-description'->'data-origin'->>0 like 'experiments'
要素JSONB演算子⑤： 省略->'basic-data-description'->'data-origin'->>1 like 'experiments'
要素JSONB演算子⑥： 省略->'basic-data-description'->'data-origin'->>2 like 'experiments'
要素JSONB演算子⑦： 他項目の条件
連結演算： WHERE ((④ OR ⑤ OR ⑥) AND ⑦)

図 7　項目と SQL の記述のマッピング(その 1)




```
要素JSONB演算子⑧： 省略->'instrument-description'->0->'identifier'->>'identifier-type' like 'url'
要素JSONB演算子⑨： 省略->'instrument-description'->0->'identifier'->>'uri' like  '12345'
要素JSONB演算子⑩： 省略->'instrument-description'->0->>'instrument-description' like 'abc'
要素JSONB演算子⑪： 省略->'instrument-description'->0->'instrument-functions'->0->>'maincategory-code' like 'Q100'
要素JSONB演算子⑫： 省略->'instrument-description'->0->'instrument-functions'->0->>'subcategory-code' like 'Q101'
要素JSONB演算子⑬： 省略->'instrument-description'->0->'instrument-functions'->0->>'additional-explanation' like  'ABC'
要素JSONB演算子⑭： 省略->'instrument-description'->0->'instrument-functions'->1->>'maincategory-code' like  'Q101'
要素JSONB演算子⑮： 省略->'instrument-description'->0->'instrument-functions'->1->>'subcategory-code' like 'Q100'
要素JSONB演算子⑯： 省略->'instrument-description'->0->'instrument-functions'->1->>'additional-explanation' like  'EFG'
要素JSONB演算子⑰： 省略->'instrument-description'->0->>'instrument-type' like 'あいうえお'
要素JSONB演算子⑱： 省略->'instrument-description'->1->'identifier'->>'identifier-type' like 'doi'
要素JSONB演算子⑲： 省略->'instrument-description'->1->'identifier'->>'doi' like '9876'
要素JSONB演算子⑳： 省略->'specimen-description'->>'initial-state' like 'aaa'
要素JSONB演算子㉑： 省略->'specimen-description'->>'final-state' like 'bbb'
連結演算：WHERE  (((⑧ AND ⑨ AND ⑩ AND ((⑪ AND ⑫ AND ⑬) OR (⑭ AND ⑮ AND ⑯)) AND ⑰) OR (⑱ AND ⑲)) AND ⑳ AND ㉑)
```

図 8　項目と SQL の記述のマッピング(その 2)

Elasticsearch を用いた研究データ管理システム・サービスの実装は, 既に複数存在する. 特に[25], [26]では Elasticsearch の特徴的な事項について, SQL と比較して概説されている. これらの検索性能の比較評価は 5.2 で後述するが, Cross Search Server では, 図 4 の[SQL Building & Handling～Throwing a query]部分を実質, Elasticsearch の API 呼出しに変更して実装している. この結果, 図 6 上で(iv)の JSONB 演算子を含む SQL のサブ記述の代替として, インデックスへのマッピングが施された上で, 図 9 のように問い合わせ用 JSON 形式により検索を実施する. インデックスへのマッピングは[22]にて紹介される Dynamic Mapping で自動的に行われる[27].

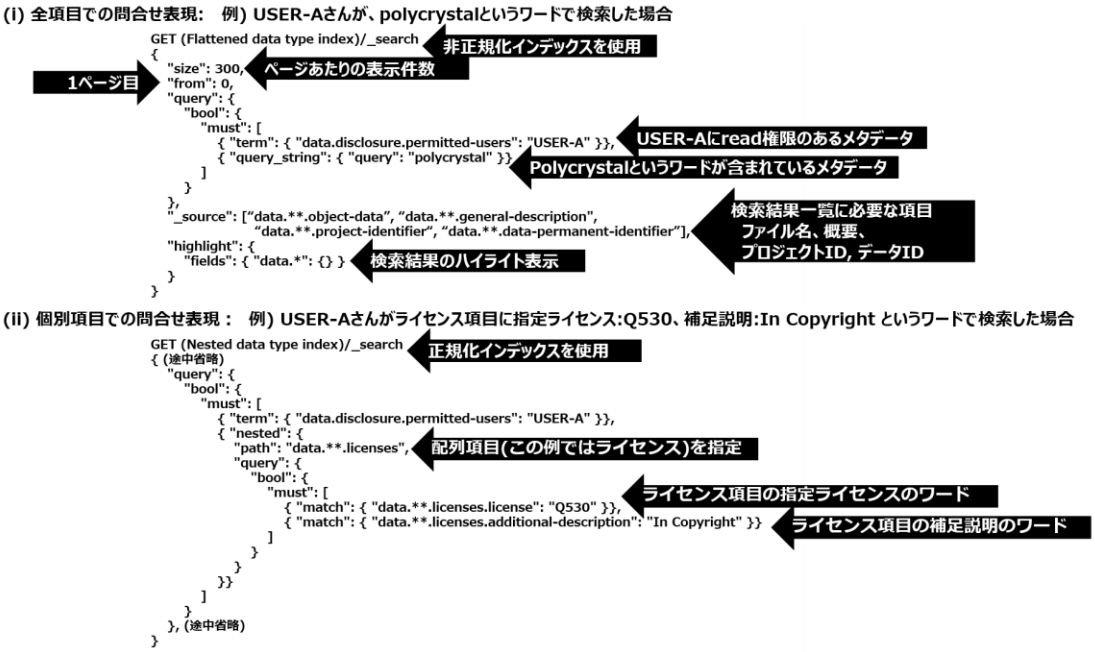


図 9　問い合わせ用 JSON 形式へのマッピング

一般には全文検索エンジンを利用した場合, 検索インデックスの構成によっては同表記で意味の異なる語彙の排除や, 概念やセマンティクスに依存した問い合わせが困難となることも見込まれる. しかし前述の GIN インデックスで見られるように, メタデータ上で表現される概念群をその path 表現を用いて識別するようにインデックスを生成することで, 然るべき概念を指定して検索することも可能となる. 図 10 は Elasticsearch でのインデックス概念のモデル図である. Elasticsearch の内部では, Lucene と呼ぶ検索エンジンが実装され, JSON インスタンス群を Elasticsearch に取り込むと語彙に対する転置インデック

スが作成される．この転置インデックスは，JSON インスタンス群上の全ての語彙を対象として，そこに出現する語彙(term)辞書として作成され，各 term に対して下記項目が保持される．
(i) 各 term が含まれる JSON インスタンス上の path 表現の identifier
(ii) 各 term が各 JSON インスタンスで出現する位置
(iii) 各 term を含むすべてのインスタンス数
また，JSON インスタンス群上の全ての語彙を対象とするため，JSON インスタンス群が参照する JSON Schema 上の語彙も扱う．このため中間表現として path 表現についても扱い得る．検索において”additional-description”という単語に関する条件が指定される場合は，Lucene は転置インデックスから該当する JSON インスタンス上の path を特定の上でさらに該当する JSON インスタンスで該当するものを抽出できる．ただし，その JSON インスタンスは 2.2 で説明した JSON Schema への適合が前提であり，それ以外のものについては保証の限りではない．さらにどこまで両者の検索結果の等価性を実現できるのか，その厳密な検証までには至っておらず，現時点では 3.の初期実装の完全な代替置換である，と言明することはできない．この機能は図 3 の論理アーキテクチャ上では独立した機能要素として描かれているが，試験運用であるため実際の配置では RDM DBMS Replica Server と同じ VM(Virtual Machine)上に配置されている．

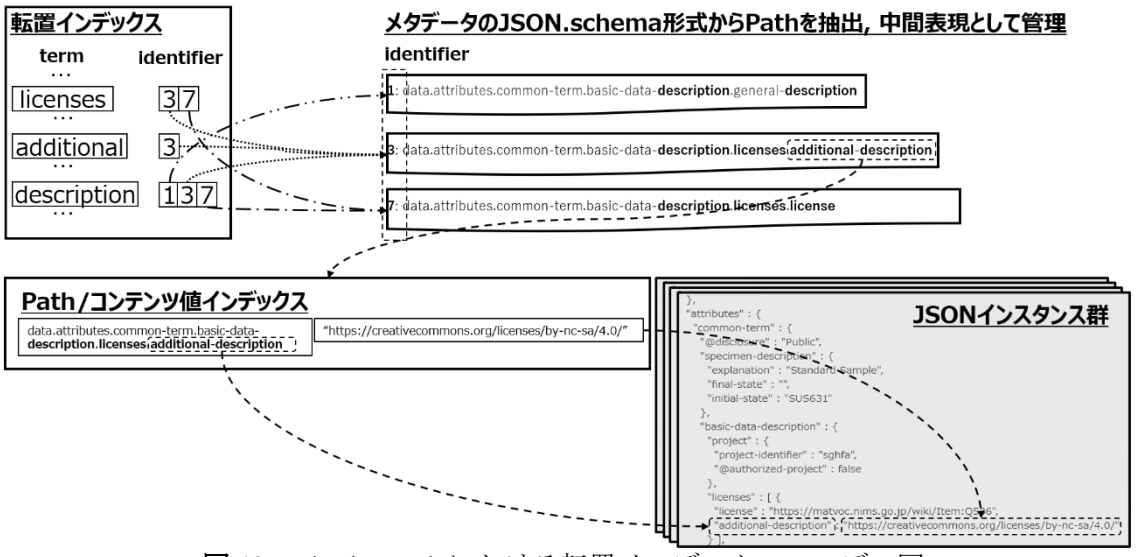


図 10　Elasticsearch における転置インデックスのモデル図

5. 評価

5.1. 評価に向けての論点説明

理想的には本稿にて評価するべき点は，“現実で発生した要求変遷に応じてアーキテクチャを更新・拡張させる過程で採択された技術方式の選択の妥当性そのもの”となる．技術的妥当性としては性能的特性，検索精度等を含んだ機能的な実現度が該当する．しかし 2.5 での説明のとおり，機能要求上の進展は，システムを運用する組織の研究戦略上の進展に応じて経時的に変化しており，必ずしも個々要求間の包含関係が明確にされているわけではない．さらに RDM の運用構成が見直された結果，3,4.で説明した各実装も発展的に他サービスに吸収され，廃止に至っていることから，実装時点で真に妥当性ある方式を採択したか，その良し悪しを評価することは既に難しい．特に 2.5 で言及した “メタデータ適用のベストエフォート化”に基づく検索精度については廃止以前に該当事例が発生していないため，評価はできない．このため，評価は部分的で代替的な論点のみから実施するに留める．具体的には各実装に関する性能特性の比較に基づく潜在能力の評価や Elasticsearch の強み・弱みの点からの定性評価と演繹的評価に限定する．それでも一つの示唆を与えることは可能と考える．

5.2. 性能特性の比較に基づく潜在能力の評価

ここでは 3,4.の実装の性能特性を比較することで潜在能力を示し，類似案件での適用に向けての判断材料を提供することを目的とする．評価では，実際の運用環境でクエリとして与える指定条件が応答性能にどのような影響を与えるかを示すことで，3,4.の何れの実装方式が潜在的に優位かを示す．クエリとして与える指定条件は，2.1 で明記した要件である，(i)同一元素を対象とする異分野の複数の計測結果を

含んだ検索, (ii)物理量を指定する検索, (iii)同一測定装置による複数の計測結果を含んだ検索, (iv)統制語彙の指定による検索, (v)多様な項目を条件式の記述をせずに複合的に指定, の 5 点から各々代表させることが望ましいが, 実際には(v)のみを実施している. 応答性能は RDM DBMS Replica Server と Elasticsearch を実装する Dice Cross Search Server に対する Query をクライアント上のブラウザから送信して結果を受け取るまでの経過時間 (Elapsed Time)で測定する. 経過時間での測定ゆえにネットワークレイテンシが誤差要因となり得るが, ここでは定常状態として扱う. 更に4.で言及したように RDM DBMS Replica Server と Dice Cross Search Server は同じ VM 上に配置されているため, システム上の物理諸元は測定～評価の上では重要要因にはならない. 参考までに表.2 に記す.

表 **2** 性能特性比較で利用したシステム上の物理諸元

項目	指定値
OS	CentOS 7.9
CPU 情報	クロック： 3.0GHz　　コア数： 8 コア
メモリ容量	128GB
ネットワーク等のその他の条件	計測対象のシステムと計測用クライアントは同一 LAN に所属, 1Gbps の回線で接続

データ件数に相当する JSON インスタンス総数はサービス初期段階から成長せず, 3,815 件と小さく, いずれの実装でも同一 JSON インスタンス群の利用を測定上の基本的仕様としている. 従って検索精度の点で評価はできないが, 性能特性評価では外乱となる要因は, ほぼ除外されている. この件数ゆえに, 概ねが VM のメモリ上にキャッシュインされている状態と推定される.

図 11 は, 3.で説明した実装で図 5 に記された入力フォーム上で入力項目数を増加させた場合の応答性能である. 図中の実線は RDM DBMS Replica Server の処理時間を含めて利用者のクライアント上のブラウザに対する経過時間, 破線は RDM DBMS Replica Server 内の処理時間のみの測定値である. ここでは複数入力項目を指定するので, AND である連言のみで処理される. 図 12 は入力フォーム上での任意項目内で OR を指定して選言数を増加させた場合の応答性能等であり, 実線・破線は図 11 の定義と同じである. ここで示される特性では図 6, 図 7, 図 8 でのマッピング規則に応じて GIN インデックスがどのように適用されるか, が暗示されることになる. 全ての項目で GIN インデックスが付与されていることから複数入力項目を AND で連結するほど, 選別される検索結果は少なくなるため, 図 11 上では応答性能が向上する. これに対して図 12 のように任意項目内で OR を指定して選言数を増加させた場合, 複数の支配要因が存在する. 0 件の場合, GIN インデックスの利用はなく, 全ての JSON インスタンスが検索対象になり, そのうち先頭の 1 ページ相当分が戻される. これに対して 1 件のみの場合は, 該当する GIN インデックスのみが参照され, UNION 演算がなされないので最速の応答性能を示す. 選言数を増加させるに従い, GIN インデックスが利用されても UNION 演算のコストが増加するため, 応答性能が悪化することが観察される. このため性能改善に当たっては, 選言部分に対する並列化等の新たな最適化対応が望まれる.

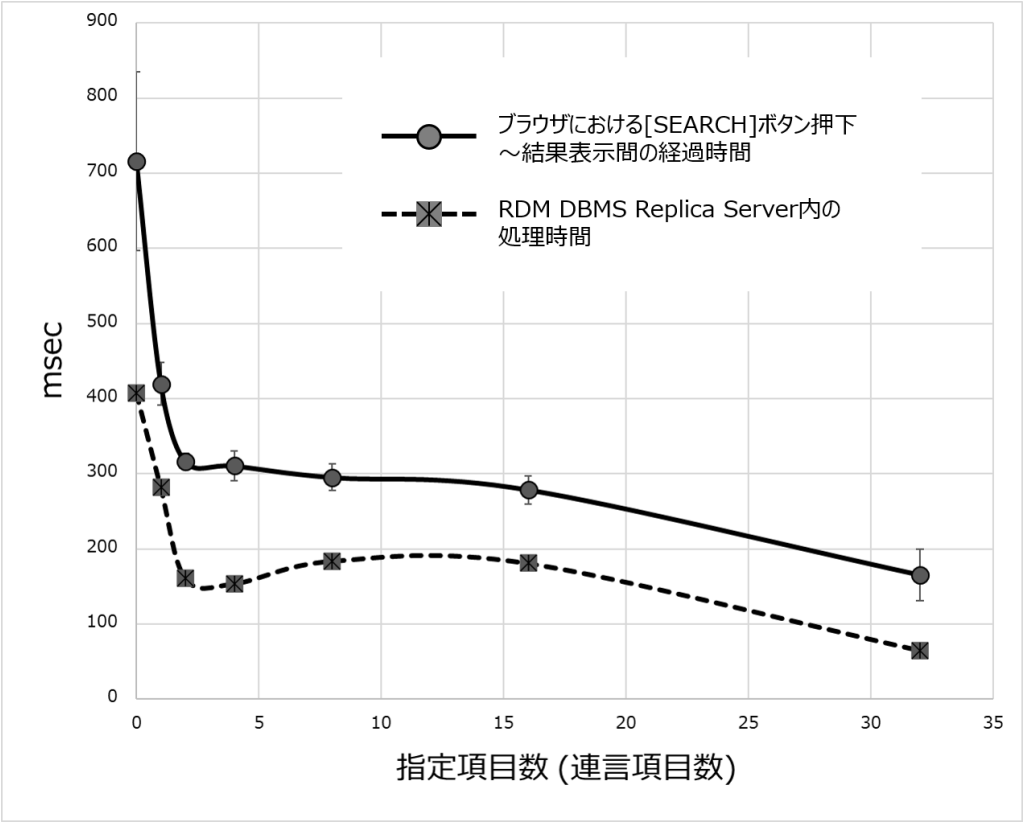


図 11 入力項目数増加に対する応答性能

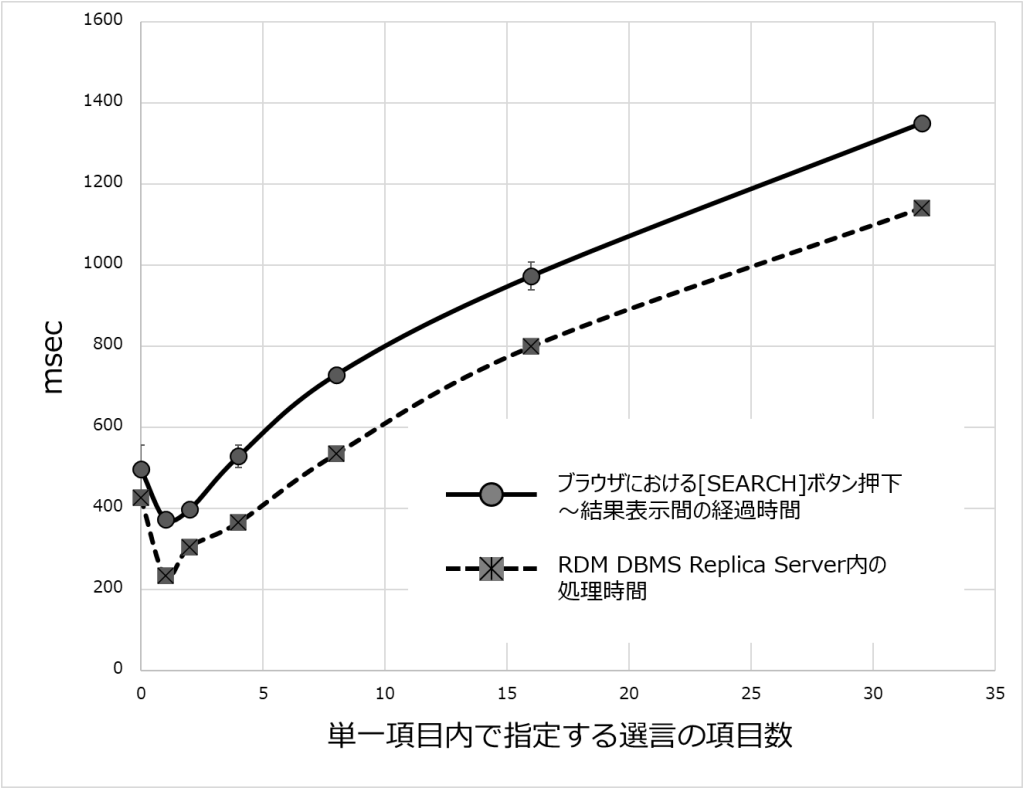


図 12 任意項目での選言数増加に対する応答性能

これに対して図 13 はほぼ同数の格納データ件数のもと, 4.で説明した実装で図 11 と同様に入力フォーム上で入力項目数を増加させた場合の応答性能である. また図 14 は 4.で説明した実装で図 12 と同様に入力フォーム上での任意項目内で OR を指定して選言数を増加させた場合の応答性能である. 図 13 と図 14 では図 11 と図 12 と同一の測定条件, 測定方法による比較とするため, Elasticsearch 内の処理時間を含めて利用者のクライアント上のブラウザに対する経過時間のみを表示する. 経過時間ゆえにネットワー

クレイテンシが誤差要因に成り得るが, 前述のとおり両者とも同一ネットワーク,VM 環境で測定していることから定常と見なし得るため, 無視することができる.

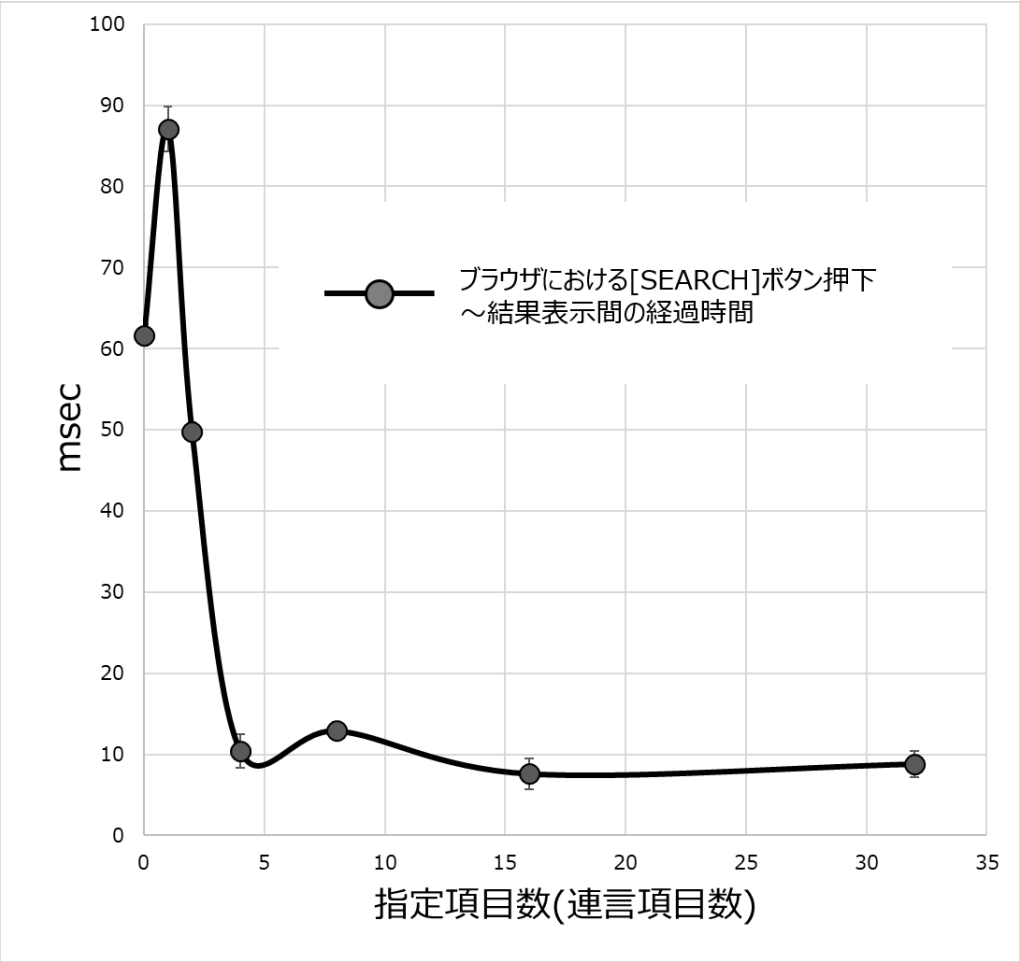


図 13　入力項目数増加に対する応答性能

図 11 に対する図 13 の比較では, Elasticsearch を利用した実装が圧倒的に高速な検索性能を示している. これは図 12 に対する図 14 の比較でも観察される. 詳細な要因分析は本稿の範囲外となるが, 想定される要因の一つとしては, GIN インデックスを利用する 3.の実装では SQL を介することから JDBC ドライバを利用しており, かつ検索時にテーブルの JOIN を多用することから, HTTP/JSON ベースのクエリで内部処理を限定する Elasticsearch よりも大きな内部オーバヘッドが存在し得ることが考えられる.

図 13 と図 14 の挙動については図 11 と図 12 のそれと似た傾向, 異なる傾向のいずれもが観察される. その要因分析に関しては Elasticsearch の内部実装に依存するため, 本稿の範囲外である. しかし図 14 で示されるように Elasticsearch の実装の場合, 入力フォーム上での任意項目内で OR を指定して選言数を増加させた際の応答性能悪化は図 12 と比較して限定的になることは興味深い. 4.にて一部言及したように, 両者間の問い合わせ等価性と差異についてはさらなる検証評価が必要であるとは言え, 発見の容易性を向上させようとするほど, 選言数は増加する傾向にあることが推定される. そのような場合でも一定性能を確保し得ることは, 技術上有利に働くと考えられる.

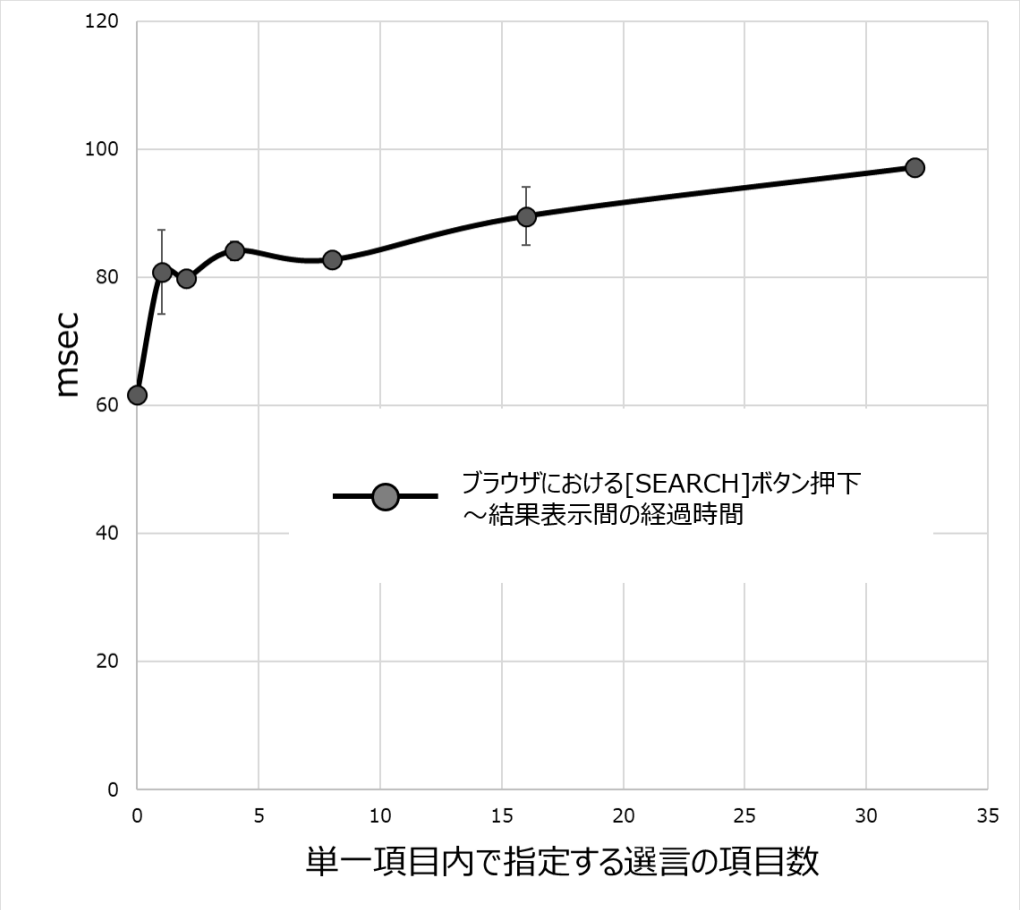


図 14　任意項目での選言数増加に対する応答性能

5.3. 機能特性, 要求変遷から見た評価

本節では Elasticsearch に関する特徴的な事項を確認し, 本稿の実装との比較評価を行う. さらに要求の論点からの整理も行う. [25]では Elasticsearch の特徴的なアーキテクチャを説明の上で, SQL との比較について言及している. ここでは Elasticsearch の持つ強みとして(i) Scalability, (ii) Agility, (iii) Performance を, 弱みとして(iv)セキュリティ・アクセス制御に関すること, ならびに(v)習熟の点を挙げている. これらの強み・弱みをもとに 4.で記した実装を評価すると以下のようになる.

(i) Scalability に関しては, 本稿の実装では検索対象のデータの件数が少なく, Elasticsearch の性能を引き出し得るユースケースとは言えない. Scalability を要求されるユースケースは IoT(Internet of Things)デバイスでストリーミングデータを扱う場合が考えられるが, 今回の研究データ管理領域ではそこまでは至らない. このため著者らの実装では, Scalability の技術的適合性・妥当性を主張することはできない.

(ii) Agility に関しては, あくまでも SQL との比較の上である. 著者らの実装では, SQL の利用とはいえ, その実態は JSON として管理されているデータであり, この点での技術的妥当性・優位性の主張も難しい.しかし Elasticsearch の持つ Dynamic Mapping 機能によって, 自動的に検索インデックスのデータ型を設定することができるため, これに基づくアプリケーション開発・データ統合の生産性の向上は期待できる. この点で Agility についての優位性の主張は可能である.

(iii) Performance に関しては, SQL と比較して優位性が主張されている. これに関しては本稿でも検証された. 特に 選言数を増加させた場合の潜在的な優位性については前述のとおりである.

(iv) セキュリティ・アクセス制御については, 2.4 で記載したとおりであり, メタデータを拡張してアクセス権項目を付与した JSON メタデータ群を RDM DBMS Replica Server 内のバッチ処理で作成し, クローラが取り込む際にアクセス制御情報として取り込む形で対応している.

2.5, 5.1 では, 検索に対する要求変化が現実に経時的に発生する事態に基づくものであり,その中で二つの形態の実装がなされ, かつ局面最適化され, 一部には背反的な実装となり得る制約があることを説明した. 本来, 要求と言う点では以下の(i), (ii)の両立が望ましい.

(i) データ記述で利用される語彙や概念構造を反映した精密な問い合わせの実現

(ii) 研究データに関する異なる管理主体間の連携を満足する実装であること

本稿の実装では，種々の技術的工夫を盛り込むことで．問題点を緩和している．第一は,二つの実装では可能な限り操作性の継承をするように実装し,QBE ベースの入力フォームについても一定程度維持されている．第二は JSON 形式のメタデータを共通的な表現基盤とし，そこで表現される統制語彙の共通性を保つ限りにおいては，両者の一定程度の連続性を持った操作感は維持できる．しかし 2.5 で言及したようにメタデータの適用がベストエフォートである以上，例え操作感の維持をしたとしても，本来要求される検索精度については限界があり得ることは容易に推測される．つまり 4.の実装では，運用上の制約を設けることで操作性や運用の連続性が初めて実現できる，と言うことも可能である．逆に現時点では, DX 化に向けた最低限の連携を目指したとしても，研究データに関する異なる管理主体間の連携という点で十分なソリューションとして寄与し得るか，という問いについては結論までには至らない．

6. 関連研究

関連研究は多岐に渡るが，ここでは関連の強い領域に特化して説明する．一つは科学データにおける Information Retrieval 領域である．もう一つは研究データ管理領域での Elasticsearch の適用である．

Information Retrieval 関連領域での研究は古く，60 年以上の歴史がある．[11]では科学データにおける Information Retrieval 領域で利用者がどのように検索を試み，かつデータを参照するかを複数分野で横断的に評価している．この評価では材料研究分野は扱っていないが，検索に関する潜在的要求を知る上では複数の重要な論点を提供する．下記にいくつかを列挙する;

(i) 分野に限らず利用者は，計測データそのものが持つ多様性に向き合う必要がある．そして妥当に多様性を統合することが必須になるが，これは課題でもある．

(ii) 分野に限らず，ドキュメント化された後工程と曖昧性を伴う前工程が存在する．

(iii) 利用者の一般的な見方(View)は共通である．

(iv) 分野に限らずリポジトリやジャーナル，さらには個人的な人脈からもデータは渉猟される．

このようなデータは非常に価値を持ち得るが，既存リポジトリの検索機能では応えきれていない面も強く，特に研究者は自らの所属学域外に当たることもある． 検索機能の潜在的要件を検討する上では，これらの点は本質的に重要であるが，本稿の検討ではこれらの点を前提として要求を明確化してきたわけではない． 今後，さらなる機能上の進化をさせる上で研究データ管理と合わせて取り込んでいくべき事項と言える．

[28]では製薬領域に特化して，科学データの参照サービスにおける効果的な方法について提案している．[28] の事例として取り上げられている製薬領域のデータベースは主に伝統的な RDB での実装であり，固定的なデータ構造，標準化された記述，大量のデータという特徴がある．その上で業界での標準語彙による検索参照が大きな検索性能の劣化を招いている．そこで[28]では２段階のアプローチを利用している．第一段階では効果的な語彙セットを構築する．ここでは自然言語処理によって語彙集を生成後，意味をなさない語彙を排除して構築する．第二段階は管理であり，Hash Index Tree でキーワードの Hash から DBMS のインデックス，表のインデックス，データ要素のインデックスの階層構造により検索能力を高める工夫をしている．著者らの方法との相違点は複数存在する．著者らの方法では，第一に統制語彙は標準化されていることが前提であり，これは外部から供給される．また多様なデータベースでの管理箇所へのインデックスを提供する[28]の方法は，4.で記した Elasticsearch を利用した実装とほぼ等価であるのに対して，3.で記した初期の実装とは本質的に異なる．標準的なメタデータスキーマの役割に依存しつつ，統制の際に利用する方式については，著者らのアプローチが一つのモデルを提示していると考える．

[29]では科学データ管理向けの連邦制システムのアーキテクチャについて説明しており，比較的新しい実装である．ここではデータ統合で伝統的に見られる方法を採用している．科学技術領域を扱ってはいるが，メタデータの複雑さや概念構造をどのように問い合わせで反映するのかの言及がない．著者らの方法は研究データ管理である RDM Server にて一度，集積した上でサービスを提供することから，異種データの統合を事前に済ませており，[29]のようなデータの統合を行った上での問い合わせをする方式は採用していない．

研究データ管理領域での Elasticsearch の適用については[25], [26], [30]など複数存在する. [25]については前節で説明しているので割愛する．また[26]は NoSQL に関する二つのデータベースの比較評価であり，研究データ管理領域に限定しないため，これも割愛する. [30]では HPC(High Performance Computing)を用いた科学分野で Elasticsearch を適用した例である．ここでは研究データの発見容易性・改善のために Elasticsearch を適用しており，連続的にインデックスを再構築する手順概要，ファイル特にファイル所属のメタデータを取り込んでインデックスを作成することを紹介している．ただし，本研究との関連で

特記するべき事項はない.

最後に上記以外の関連研究に言及する. [31]は, 全文検索に対してオントロジを用いたセマンティクス指定と統合した検索機能向けのインデックスを提案している. 問い合わせは対話式の QBE 指向であり, SPARQL に近い形式を前提とする. 本稿の QBE 指向の機能とは関連性が高いが, 著者らは新たなインデックス作成技術の開発を目的としているのではなく, 実運用上での要求に対する技術的妥当性について述べており, 狙いは異なる.

7. むすび

本稿では. 材料データプラットフォーム’DICE’における研究データ管理上の要求の変遷と検索機能の進展に関して概説し, アーキテクチャを更新・拡張させる過程で実装された二つの形態に対して性能・機能を含んだ多論点からの比較評価を行うことで, 各々の技術的妥当性を示した. 前述したように研究データ管理(RDM)は, 研究データ構造化から解析迄の利用とその運用にて, より効率的・最適化実現を目的として, 商用クラウドコンピューティング環境上へ移行する際に IoT (Internet of Things)デバイスとデータ共有・構造化する機能・サービスと統合・一体化した結果, 本稿説明の検索機能も発展的に上記サービスに吸収され, 実装そのものは既に役目を終えている. 本稿の貢献点を下記に列挙する.

(1) 科学技術領域における研究データ管理領域では新たな要求を受けて活発な開発が進展しており, この中で材料科学分野でのケーススタディを提示した. 特に既存の研究で述べられているインデックス技術の新たな方式提案やサーベイとは異なり, 特定運用環境における現実の要求に対してどのような技術方式を選択し, 統合してきたかを明示した.

(2) 研究データ管理プラットフォームでの検索機能の二つの実装について, 比較の上で技術的妥当性を評価するための材料を提供した. 特に Elasticsearch を利用した実装について, その優位性と限界について示した. ここでは性能, データ統合の生産性, ならびに発見容易性を向上させる上で潜在的に有利となる点, 限界としてはデータ記述で利用される語彙や概念構造を反映した精密な問い合わせをする上では一定程度の犠牲を強いる点である. これに基づくと, 研究データに関する異なる管理主体間の連携という点で十分なソリューションとして機能するかは結論には至らないが, 研究データ管理上の検索機能・検索サービスを実現する上では一つの示唆は提供すると考える.

参考文献

[1] 知京豊裕, “マテリアルズインフォーマティクスの現状と課題～海外の動向と日本の挑戦,” 情報知識学会誌, 2017 Vol.27, No.4, pp.297-304, 2017, https://doi.org/10.2964/jsik_2017_032 , 2023.7.22 参照.

[2] 上島伸文, 及川勝成, “技術解説～計算材料科学・工学の最新動向,” 電気製鋼, 87 巻, 1 号, pp.21-26, 2016.

[3] Y.Liu, T.Zhao, W.Ju and S.Shi. “Materials discovery and design using machine learning,” Journal of Materiomics, Vol.3, No. 3, pp.159-177, 2017, https://doi.org/10.1016/j.jmat.2017.08.002, 2023.7.22 参照.

[4] T.Hey, S.Tansley and K.Tolle. “The Fourth Paradigm: Data-Intensive Scientific Discovery,” Microsoft Research, October 2009, ISBN:978-0-9825442-0-4, https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/, 2023.7.22 参照.

[5] C.S.Liew, et al. “Scientific Workflows: Moving Across Paradigms,” ACM Computing Surveys,Vol.49,No.4, Article 66, 2016, https://doi.org/10.1145/3012429 , 2023.7.22 参照.

[6] 山田直史, 高島洋典, 山田直史, “超スマート社会(Society5.0)実現に向けて,” 情報管理, 2017.8. Vol.60, No.5, 2017.

[7] 大学 ICT 推進協議会, “学術機関における研究データ管理に関する提言,” https://axies.jp/report//publications/proposal/, 2019 , 2023.7.22 参照.

[8] R.C.Amorim, J.A.Castro, J.R. da Silva and C.Ribeiro. “A comparison of research data management platforms: architecture, flexible metadata and interoperability,” Universal Access in the Information Society, Vol.16, pp.851-862, 2017, https://doi.org/10.1007/s10209-016-0475-y, 2023.7.22 参照.

[9] NIMS Now, 2019.1 月号, https://www.nims.go.jp/publicity/nimsnow/vol19/hdfqf100000aoslh-att/hdfqf100000aosp0.pdf, 2023.7.22 参照.

[10]谷藤幹子, “材料データプラットフォームシステムの設計と構築,” 月刊機能材料, Vol.40, No.10, 2020 年 10 月号, 2020.

[11]K. Gregory, P. Groth, H. Cousijn, A. Scharnhorst and S. Wyatt, “Searching Data: A Review of Observational Data Retrieval Practices in Selected Disciplines,” Journal of the Association for Information Science and Technology, Vol.70, No,5, pp.419-432, 2019, https://doi.org/10.1002/asi.24165, 2023.7.22 参照.

[12]菊地伸治, 門平卓也, 鈴木峰晴, 内藤裕幸, “高付加価値科学データ創出を指向した研究データ管理プ

ラットフォームのアーキテクチャ," 信学技報, Vol.119, No.66, SC2019-2, pp.7-17, 2019.
[13]菊地伸治, 内藤裕幸, 門平卓也, 谷藤幹子, "CAS ベースの RDM 認証・認可機構の漸増開発とアセスメント評価," 情報処理学会論文誌デジタルプラクティス（DP）, Vol.2 No.2(Apr. 2021), pp.64-79, 2021.
[14]https://jsonapi.org/format/#status, 2023.7.22 参照.
[15]https://doi.org/10.48505/nims.3240, 2023.7.22 参照.
[16]M. M. Zloof, "Query-by-example: a data base language," IBM Systems Journal, Volume.16, Issue.4, pp.324-343, December.1977, https://doi.org/10.1147/sj.164.0324, 2023.7.22 参照.
[17]https://www.apereo.org/projects/cas, 2023.7.22 参照.
[18]https://osf.io/, 2023.7.22 参照.
[19]青山幹雄, "解説, DX とデータ駆動がもたらす情報通信技術の新たな研究パラダイム," 電子情報通信学会誌, Vol.104, No.6, pp.596-602, 2021.6.
[20]中口孝雄, 村上陽平, 林冬惠, 石田亨, "サービス基盤の連邦制運営のための情報共有・実行制御アーキテクチャ," 電子情報通信学会論文誌, D Vol.J101-D, No.1, pp.193-201, 2018.
[21]F.Simeoni, M.Yakici, S.Neely and F.Crestani, "Metadata Harvesting for Content-Based Distributed Information Retrieval," JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY, 59(1), pp.12-24, 2008, https://doi.org/10.1002/asi.20694, 2023.7.22 参照.
[22]https://www.elastic.co/jp/elasticsearch/, 2023.7.22 参照.
[23]https://www.postgresql.org/docs/current/gin-intro.html, 2023.7.22 参照.
[24]M. Patil, S.V. Thankachan, R. Shah, W.Kai.Hon, J..S.Vitter and S. Chandrasekaran, "Inverted Indexes for Phrases and Strings," Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 2011, pp.555-564, https://doi.org/10.1145/2009916.2009992, 2023.7.22 参照.
[25]O. Kononenko, O. Baysal, R. Holmes and M.W. Godfrey, "Mining modern repositories with elasticsearch," Proceedings of the 11th Working Conference on Mining Software Repositories, May 2014, pp.328-331, https://doi.org/10.1145/2597073.2597091, 2023.7.22 参照.
[26]S. Gupta, R. Rani, "A Comparative Study of Elasticsearch and CouchDB Document Oriented Databases," 2016 International Conference on Inventive Computation Technologies (ICICT), August 2016, https://doi.org/10.1109/INVENTIVE.2016.7823252, 2023.7.22 参照.
[27]https://www.elastic.co/guide/en/elasticsearch/reference/current/dynamic-field-mapping.html, 2023.7.22 参照.
[28]L. Du, M. Li and J. Xu, "An Efficient Method for Scientific Data Retrieval Service," Proceedings of the 2020 3rd International Conference on Big Data Technologies, September 2020, pp.6-10, https://doi.org/10.1145/3422713.3422731, 2023.7.22 参照.
[29]S. Kim and B. Moon, "Federated database system for scientific data," Proceedings of the 30th International Conference on Scientific and Statistical Database Management, July 2018, Article No.33, pp.1-4, https://doi.org/10.1145/3221269.3222332, 2023.7.22 参照.
[30]J. Rosenberg, J.B. Coronel, J. Meiring, S. Gray and T. Brown, "Leveraging Elasticsearch to Improve Data Discoverability in Science Gateways," Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning), July 2019, Article No.:19, pp.1-5, https://doi.org/10.1145/3332186.3332230, 2023.7.22 参照.
[31]H. Bast and B. Buchhold, "An index for efficient semantic full-text search," Proceedings of the 22nd ACM international conference on Information & Knowledge Management, October 2013, pp.369-378, https://doi.org/10.1145/2505515.2505689, 2023.7.22 参照.

著者略歴

菊地 伸治（きくち しんじ）

論文投稿時は国立研究開発法人物質・材料研究機構勤務, 現在, 国立研究開発法人理化学研究所情報統合本部勤務, 1987 年東北大学大学院工学研究科修了, 2013 年会津大学大学院コンピュータ理工学研究科修了, 博士（コンピュータ理工学）. 1987 年～2014 年日本電気株式会社所属, 2014 年～2018 年会津大学特任教授, 2018 年～2023 年国立研究開発法人物質・材料研究機構 NIMS エンジニア, 電子情報通信学会/IEEE Computer Society/ACM/情報システム学会各会員.

田辺 浩介（たなべ こうすけ）

国立研究開発法人物質・材料研究機構技術開発・共用部門データ基盤ユニット主幹エンジニア. 2014 年慶應義塾大学大学院政策・メディア研究科博士課程単位取得退学, 2016 年博士（学術）. 東京工科大学大学院バイオ・メディア研究科助手, 慶應義塾大学メディア・コミュニケーション研究所非常勤講師などを経て,

2012 年より独立行政法人物質・材料研究機構に所属, 2023 年より現職, 情報処理学会/情報通信学会/情報知識学会各会員.

坂本 浩一（さかもと こういち）
国立研究開発法人物質・材料研究機構勤務, 1981 年東京大学理学部情報科学科卒業, 1981 年～2013 年株式会社日立ソリューションズ所属, 2013 年～2019 年株式会社みらい知的財産技術研究所所属, 2019 年から現職, 人工知能学会会員.

高田 安裕（たかだ やすひろ）
国立研究開発法人物質・材料研究機構 技術開発・共用部門データ基盤ユニット勤務.

傳法 春樹（でんぽう はるき）
国立研究開発法人物質・材料研究機構 技術開発・共用部門データ基盤ユニット勤務.

門平 卓也（かどひら たくや）
国立研究開発法人物質・材料研究機構技術開発・共用部門データ基盤ユニット長, 2001 年早稲田大学大学院理工学研究科資源及び材料工学専門分野博士後期課程退学, 2004 年博士（工学）, JST-CREST 研究員等を経て 2007 年国立研究開発法人物質・材料研究機構に入所, 調査分析業務等を経て 2014 年から材料工学分野におけるデータ活用型研究のための基盤構築業務に従事.

谷藤 幹子（たにふじ みきこ）
国立情報学研究所オープンサイエンス基盤研究センター副センター長. 前職の国立研究開発法人物質・材料研究機構にて, 統合型材料開発・情報基盤部門材料データプラットフォームセンター長として, 材料データプラットフォーム DICE の構築に 2017 年～2022 年に携わる. 応用物理学会会員. 内閣府オープンサイエンスの推進に関する検討会委員.