MarkLogic

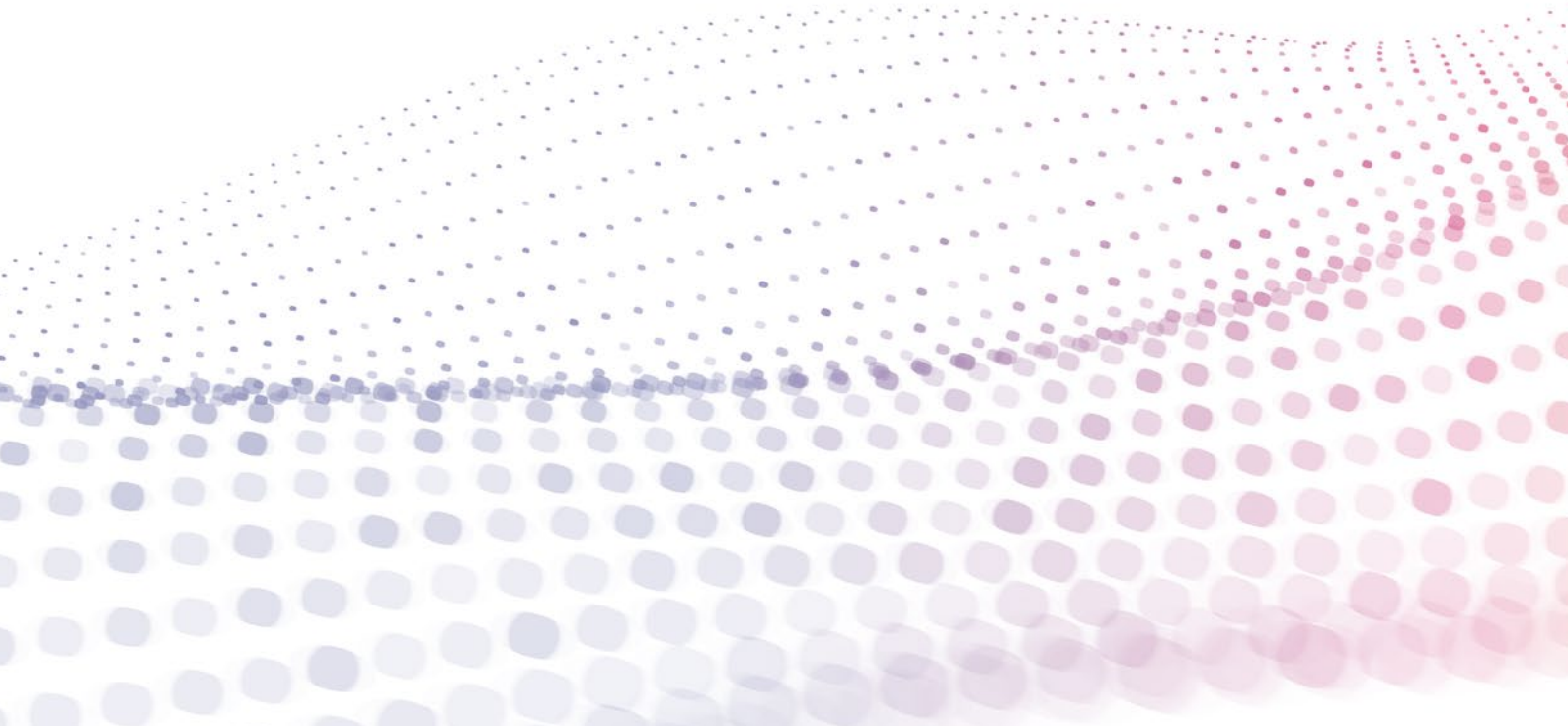
アジャイルなエンタープライズメタデータ管理

MarkLogicで価値あるメタデータをアクションに変える

MARKLOGIC ・ ヘルスケア&ライフサイエンス

経営層やユーザーは、どこから得られたのかわからないようなデータは信用しません。エンタープライズメタデータとは「データに関するデータ」であり、これによってデータが信用できるものになります。残念ながら、ヘルスケアならびにライフサイエンス企業の多くでは、従来のリレーショナルならびにカラムファミリー技術のツールを使ってメタデータの収集や管理を行っていますが、あまりうまくいっていません。

MarkLogic のアーキテクチャはマルチモデルで、メタデータの管理が従来より楽です。またエンタープライズデータの品質やリネージ（経緯）を確固たるものにできます。MarkLogic のスマートなメタデータ管理機能を利用しているヘルスケアならびにライフサイエンス企業は、検索やディスカバリーの改善、規制順守のシンプル化、レポートの正確性と信頼性の改善、より優れたカスタマーサービスなどを実現しています。このホワイトペーパーでは、MarkLogic によるアプローチの特徴とメリットを説明してきます。



本ドキュメントについて

このドキュメントでは、MarkLogic によるエンタープライズメタデータ管理のメリットを紹介します。

- エンタープライズメタデータを定義し、その遍在性、多様性、重要性を確認する
- エンタープライズメタデータ管理の多様な要件を紹介する
- MarkLogic のマルチモデル（＝ドキュメントならびにグラフ）アーキテクチャが、極めて多様なエンタープライズメタデータの扱いに理想的である理由を説明する
- MarkLogic のメタデータ管理機能と、他のあまり柔軟ではないデータアーキテクチャを比較する
- MarkLogic のユニバーサルインデックス、検索、セマンティック機能により、メタデータ管理がさらにシンプルになることを紹介する

また最後に、MarkLogic によるメタデータ管理のユースケースをいくつか紹介し、MarkLogic によってエンタープライズのアジャイル性が劇的に向上するという結論を提示します。

想定される読者：このドキュメントは、エンタープライズメタデータ、またそれを管理するアーキテクチャの重要性について理解したい人を対象に書かれています。データベースのさまざまな概念（リレーショナル、キー / バリュー、グラフ、ドキュメントストアなど）について、最低限の知識があることを想定しています。

エンタープライズメタデータとは何か？

メタデータ管理を効率的に行おうとする際の課題の 1 つとして、メタデータが明快に定義できないことがあります。「他のデータを説明するデータ」という一般的な定義が出発点になりますが、これでは漠然とし過ぎていてあまり役に立ちません。

社内のいろいろな部署の人に、「メタデータとは何ですか」と聞いてみるといいでしょう。これまで私たちがクライアントに聞いてみた際には、業務部門や職種によってその理解や要件はかなり違っていました。私たちが調べたところ、メタデータに関する以下のニーズのさまざまな組み合わせが多いようです。

- ビジネス用語集：各プロジェクトで使われるすべての略語や用語を含む
- タクソノミー（オントロジー）：コレクション内のアイテムを分類するためのもの
- データモデル：さまざまなデータベース用。表、列、データ要素、ドキュメント構造、定義を含む
- リファレンスデータ（コード）：データ分類や数値コードを人間が読めるように変換する際に使用されるもの
- ビジネスルール：データの検証、変換、分類、関係付け、各レコードへのデータクオリティスコアの割り当てに使用するもの
- ワークフローと承認プロセス
- レポート、ダッシュボード、KPI、API 呼び出しのリスト
- 「オペレーショナル」メタデータ：さまざまなレポート作成に通常かかる時間、CPU/ メモリ / ディスク使用量、これらの業務に関連するチャージバックと料金請求などを含む
- レコード間の関係性：同一顧客の異なるシステムにおける異なる表現方法など。このような関係性管理手法はマスターデータ管理（MDM）と呼ばれる
- これらの由来（いつ、どこで、誰が）に関する完全な知識、これらの間の関係性、それらの保持・検証方法

このような調査結果を見るたび、業務部門や分野によって要件が大きく異なることに驚かされます。しかし、すぐにパターンがあることがわかります。例えば、一般的に次のようなことがあります。

- 組織によっては、メタデータは指定されたデータスチュワードによって注意深く管理される必要がある。これが重要なのは、メタデータ構造を少し変更しただけで本番システムに大きなインパクトを与える可能性があるため。

- 組織によっては、メタデータのバージョン管理と有効期間の設定が必要。メタデータの複数の本番バージョンを扱い、サービスを停止することなく本番システムへの変更を取り消すことができる必要がある。

- 組織によっては、メタデータシステム（リファレンスコードのルックアップ、選択肢を含むインターフェイスなど）には、高可用性ならびにリアルタイムのルックアップと検索が要求される。

- 組織によっては、メタデータを利用する際に、重複レコードを素早く特定して削除する必要がある。これは意味が同じものの表現は 1 つにしておきたいため。

- 組織によっては、メタデータに構造があるものとないもの（構造化 / 非構造化）がある。例えば、従来からのスプレッドシートのデータがある一方、複雑なグラフや関係性を表すデータがある。

- 多くの場合、組織はデータの意味ならびに個々のデータ要素やコードのセマンティックを明確に扱う必要がある。

- 各業務部門ごとにメタデータの扱い方をカスタマイズする必要がある一方、組織全体（業務部門間において）は一貫性を持ってデータを扱う必要がある。

MarkLogicによるメタデータ管理で使用される主なエンタープライズ機能

- 柔軟なメタデータ格納
- 検索
- リアルタイムのデータサービス
- エンリッチと注釈
- セマンティック（RDF トリプル）
- データの出自とガバナンス
- きめ細かなセキュリティ
- レポート / 分析
- 拡張性

これまでに「こういった要件すべてを 1 つのデータ技術やアプリケーションで容易に管理することは不可能」という結論に達したことがあったとしても、不思議ではありません。私たちは、多様なタイプのデータやメタデータのさまざまな利用方法に対応するためには、最高に柔軟、アジャイル、安全で信頼できるソフトウェア環境が必要だと考えています。

現在、あなたが考える「メタデータ」と他の利用者が考える「メタデータ」が全く違うということがあり得ます。これは自然なことですが、フラストレーションがたまることでもあります。しかしながら「このような情報を安全に、また組織全体で一貫性のある方法で管理する」という目的を忘れないようにしましょう。というのも、これによってメタデータのアセットを発見して再利用でき、この結果、データアセットのビジネスバリューを最大限にできるからです。組織内で、ビジネスに重要な情報の管理にスプレッドシートを使っているとしたら、こういったデータセットの格納や利用にもう少しましな方法がないのものかという疑問も湧くでしょう。このような疑問は素晴らしいものです。

次に、メタデータの多様性について、またこのような情報を MarkLogic がどう管理するのかについて、明確に定義していきましょう。

極めて多様なメタデータの課題

まずシンプルな問題を考えてみましょう。これは誰がレコードを作成・更新したのかということです。またデータのバリエーションがどのように発生するのかについて確認します。これにより、このような情報の管理において、従来のリレーショナルシステムでは柔軟性が足りないということが明らかになるでしょう。

データ行の末尾に、以下のような「履歴」に関する 4 つの列があるのを見たことはないでしょうか。

CREATED_BY	CREATED_DT	UPDATED_BY	UPDATED_DT
sperterson	2016-02-25	fbrown	2016-03-12
rbatchu	2015-01-15	rbatchu	2016-03-12

これはとてもシンプルで、表内の各行にこの 4 列が必要なだけです。リレーショナルデータベース内のテーブル 1 つ 1 つにこれらの列を追加し、これらの列を更新するコードを追加すれば、それでいいはずです。

しかしちょっと待ってください。この場合、最初と最後の変更だけでなく、その間の変更すべてをトラッキングする必要はないでしょうか。リレーショナルモデルでは、履歴テーブルを別に作ってこれをジョインできます。この場合、クエリが遅くなりますが若干柔軟性が増します。

また、このレコードの古いバージョンも保持したいとします。変更理由のテキストを検索したい場合はどうでしょうか。レコードにキーワードを付けて分類したり、行にデータクオリティのスコアを割り当てたい場合はどうでしょうか。各変更を承認した人は誰か、いつ承認されたのかについて知りたい場合はどうでしょうか。こういったことは、テーブル 1 つ 1 つに列をいくつか足していくというやり方ではできません。データ追加の必要があるのは、数百万行のうちわずか数行だけといった場合はどうでしょうか。各行にプレースホルダーを残しておくべきでしょうか。複数の行にわたって分散しているビジネスオブジェクトを、1 つの追跡可能なエンティティとして扱わなければならない場合はどうでしょうか。

ここにおいて、テーブル形式（リレーショナル構造）で複雑かつ極めて多様なメタデータを格納する際の問題がいくつか明らかになってきます。メタデータを追加し

> 表やリレーショナルシステムには、極めて多様なメタデータを付与できる能力が欠けています。またこの際にこの情報に関係ないレコードに影響を与えるべきではありません。」

たいアイテムが 1 つだけだったとしても、テーブル内のすべての行にこのアイテムを「絶対」追加しなくてはなりません。表やリレーショナルシステムには、極めて多様なメタデータを付与できる能力が欠けています。またこの際にこの情報に関係ないレコードに影響を与えるべきではありません。つまり、より柔軟なアプローチが必要なのです。

次に、単純なテーブル形式のデータ格納以外の選択肢について確認していきます。

メタデータ管理アーキテクチャの他の選択肢

利用できるデータアーキテクチャとして、主に以下の 6 つがあります。

1. リレーショナル
2. 分析（OLAP キューブ）
3. キー / バリューストア
4. カラムファミリーストア
5. グラフストア
6. ドキュメントストア

これら 6 つのデータベースアーキテクチャはかなり長いこと存在しています（メインフレームに使われているものもあります）。しかし新しい種類のものが、データの管理方法を変えてきているということはあまり理解されていません。

ここで、メタデータ管理に分析システムが使えないことは明らかです。分析システムは、主に集約をして過去に

関するレポートを作成するためのものです。またそのスタースキーマパターンは、ほとんど変化がない一元化されたファクトテーブルに基づいています。

またキー / バリューストアやカラムファミリーストアも使えないでしょう。どちらも任意のメタデータのインデックス付け、クエリ、検索に使える柔軟なクエリ言語がないからです。キー / バリューシステムの多くは拡張性を目的として作られており、セマンティックやアジャイル性を目的としていません。

こうなると残された選択肢は、リレーショナル、グラフ、ドキュメントストアの 3 つです。しかしすでに、リレーショナルシステムならびにテーブル形式のモデル（列と行が固定されている）では極めて多様なものにうまく対応できないことは紹介しました。ということで、ここから MarkLogic のマルチモデル（ドキュメントとグラフ）アーキテクチャが検索可能なエンタープライズメタデータの格納に理想的だということを詳しく紹介していきます。

ドキュメント（ならびにグラフ）ストアによる多様なメタデータの管理方法

JSON や XML ファイルに馴染みがある人も多いでしょう。これはツリー構造だと考えられます。つまりデータのペイロード（正味のデータ部分）は各「枝」の「葉」部分にあります。ドキュメントやグラフにはすべてリカーシブ（再帰的）な特徴があります。これはドキュメント内のあらゆる枝はさらに枝（サブブランチ）を持つことができ、このサブブランチもさらにサブブランチを持つことができるということです。グラフにも似たような特徴があります。グラフは、他のメタデータを含む別のグラフをポイントできます。しかしながら取りあえずは、まずドキュメントを取り上げ、先ほどの行の履歴がどのように表現されるのかについて確認しましょう。

これが、リレーショナルアーキテクチャとドキュメントアーキテクチャの主な違いです。リレーショナルモデルでは、表に 1 列追加する場合、この列を表内のすべての行に追加しなくてはなりません。100 万行ある場合には、1 列足すことにより表内に新しいセルが 100 万個追加されます。ここに柔軟性はありません。

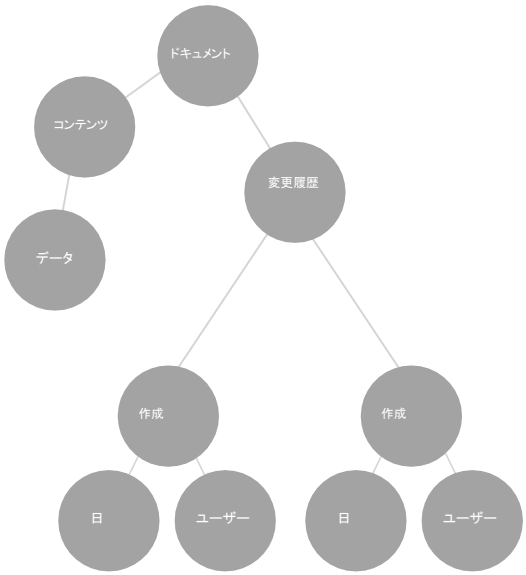


図2：ドキュメントに追加された変更履歴の「枝」

一方、ドキュメントストアはどうでしょうか。ドキュメントストアでは、任意のドキュメント内の任意の枝に、新しい枝をいつでも追加できます。ドキュメントは極めて柔軟に拡張できます。100 万件のドキュメントのうち 1 つだけを対象にできるのです。結論として、ドキュメント（ならびにグラフ）ストアは、極めて多様なメタデータの格納には理想的だということになります。これが採用されるのは、他に比べて柔軟だからです。

ドキュメントストアへのメタデータの格納を一度経験してしまうと、メタデータの格納に表を使うのは拘束衣を着せられるようなものだと感じられるでしょう。ドキュメントストアを利用することで、アジャイルなメタデータ管理ができ、各業務部門の多様なニーズに応えられるメタデータソリューションを短期間でカスタマイズできます。

しかしここで疑問も湧くでしょう。もし各業務部門がデータベース内のすべてのアイテムに自分たちでメタデータを付け始めたら大混乱に陥ることにならないでしょうか。レポートがちゃんとできなくならないのでしょうか。こういった問いへの答えは「そうなることもある」です。

強力なデータガバナンスやコントロールがない場合、ドキュメントやグラフストアの管理は困難になります。しかしながら、MarkLogic にはデータガバナンスとセキュリティを管理するための強力なツールがあります。

MarkLogicにメタデータを格納するさまざまな方法

MarkLogic ではデータをドキュメント（JSON、XML）ならびにグラフ（RDF）として格納できるため、メタデータ格納の柔軟な方法は複数存在します。以下はそのうちよく使われるものの例です。

1. 「ラッパー」あるいは「エンベロープ」を使って、ドキュメントデータとメタデータを同一ドキュメント内に一緒に持つ
2. MarkLogic の「プロパティ」API を使って、最終変更日時などのメタデータを格納する
3. ドキュメント内に組み込まれた RDF トリプルを使って、メタデータを格納する
4. 外部の RDF を使って、ドキュメントの URI をそのメタデータとリンクする
5. 別のドキュメントを使って、ドキュメントのメタデータを格納する。これはセキュリティロールがメタデータとソースドキュメントで異なる場合に便利
6. キー / バリューのペアのメタデータ（マップ）を使って、メタデータを格納する（MarkLogic 9）

ここから、元のデータならびにメタデータの両方におけるバリエーションを検知し、これを修正するための MarkLogic のテクニックをいくつか紹介していきます。まず最初にユニバーサルインデックスを取り上げます。これはどのように機能するのでしょうか。

「ドキュメントストアを利用することで、メタデータ管理をアジャイルにでき、各業務部門が求める多様なニーズに応えられるメタデータソリューションを短期間でカスタマイズできます。」

極めて多様なメタデータの管理に、MarkLogicのユニバーサルインデックスがどう役立つのか

ここまで、ドキュメントストアが極めて多様なデータを格納できるということを説明してきました。ここで重要なのは、MarkLogic データベースのプラットフォームでは、追加（読み込み / ロード）されるとすぐに、枝（コンテンツ「ならびに」構造）の一つ一つにインデックスが付けられるということです。このインデックス（「ユニバーサルインデックス」と呼ばれます）を使うと、データのバリエーションすべてに関するレポートを素早く作成できます。

先ほど取り上げた履歴の例は、これについて考えるのに適しています。ここでは、ほとんどのレコードにおいて必要なメタデータは、当該レコードを作成ならびに更新したユーザーと、これらの変更の日付についてだけだと想定します。また、レコードの中には、変更するにはマネージャの承認が必要な特別なものもあるとします。MarkLogic では、こういった特別なレコードはマネージャの承認を受けた場合にのみ変更が反映されるクエリを作成することで、このワークフローに対応できます。

また MarkLogic の強力なスキーマ検証ツールを使って、これらの規則が守られているかどうかをチェックできます。 MarkLogic には、ドキュメントが一連のデータクオリティ規則を守っているかどうかをチェックするための強力なツールが備わっています。これらの規則は「XML スキーマ」ドキュメントに定義されています。各ドキュメントは、この XML スキーマならびに各ドキュメントに割り当てられたデータクオリティスコア（例えば 1 ～ 100 の値）に対して検証されます。データクオリティのスコアを使うと、質の悪いデータを除外したり、自動処理あるいは人間によってクリーンアップが必要なドキュメントのリストを作成できます。

また MarkLogic を使うと、データベース内の各ドキュメントに対してクオリティスコアを関連付けることもできます。通常、このスコアは 1 から 100 の間の整数です。この検証プロセスでクオリティスコアを変更することもでき

ます。こうすることでドキュメントがレポートに含まれないようにしたり、検索結果のドキュメントのランキングを変えることもできます。

これにより、あらゆるエンタープライズデータにおいて、多様性と一貫性の両方を実現できます。これと同等の柔軟性やパワーを持つリレーショナルモデルは存在しません。

また、MarkLogic で使えるインデックスの数は 1 つだけではありません。ユニバーサルインデックス以外にも、クエリの最適化に利用できるインデックスが多数あります。例えば、順番を持つ情報（日付など）に対してレンジインテックスを任意のタイミングで追加できます。幸いなことに、最初に MarkLogic のメタデータ管理システムをセットアップする際に、これらのインデックスすべてを熟知している必要はありません。MarkLogic にはデフォルトの設定があり、通常ほとんどのプロジェクトではこれだけで十分です。

さてここで、メタデータ管理において最も困難な問題の 1 つを詳しく見ています。これは、メタデータエントリの重複を除外できる信頼性の高いシステムを作る、ということです。多くの組織において、メタデータ管理技術がユーザーに信用されず、使われていないという事実があります。さらに悪いことに、全社的なソリューションが信用できないために、あるいはカスタマイズできないために、業務部門ごとにソリューションを開発してしまうということがあり得ます。

メタデータレポジトリ VS.メタデータレジストリ

多くの組織では、一生懸命になって各業務部門のあらゆるシステムに含まれるさまざまなメタデータ要素をすべて収集しようとしています。この際、各データベース内の特定アイテム（テーブル名、ビュー、列名、関係性など）を定期的にスキャンするツールが使われています。こういったシステムは、レポート作成に必要な適切なフィールドを探す際に役立ちます。これは「メタデータレポジトリ」あるいは「レポジトリカタログ」と呼ばれます。これらは収集と検索にフォーカスしていますが、メタデータを一貫性を持って利用しようとするものではありません。

他のタスクとしては、エンタープライズのデータ標準を満たす用語やデータ要素を提示するシステムの作成があります。これは「メタデータレジストリ」と呼ばれます。これらのシステムは、エンタープライズの標準をキュレーション（整理）しパブリッシュ（提出）します。訓練を受けた「データスチュワード」の役割は、これらのシステム内でデータ要素を管理することです。レジストリは、検索、重複の除外にフォーカスし、承認済みのデータガバナンスプロセスに基づく承認プロセスを注意深く適用します。

信頼できるエンタープライズメタデータ管理システムを構築する

まずシンプルな問題を取り上げてみましょう。あなたは、システム内において誕生日をたった 1 つの方法で表現したいと考えました。このためまず、PersonBirthDate というデータ要素を追加しました。そして他のみんなに、列名、XML 要素名、JSON 名としてこれを使うように依頼します。その後、レジストリを作ってから休暇を数週間取りました。休暇から戻ってみると、あるユーザーは IndividualDOB で検索していて、あなたが以前作った要素を発見することはできませんでした（彼らは自分たちのものを追加していました）。また別の人たちはどちらも気に入らなかったので、こちらも自分たちで CustomerBirthDate を追加しました。こうなると人々は誕生日に関してそれぞれが好きなものを使った表を作り始めます。どのバージョンを使ったらよいのかわからなくなっているからです。

データ定義の重複は、メタデータ管理システムが信用を失ってしまう主な原因の一つです。例えば、誕生日の格納方法の定義が 2 つあった場合、どちらを使ったらよいのでしょうか。自分たちで新しくデータ要素を作ってしまった方が良いのでしょうか。

これは、メタデータ管理システムの構築において検索と承認のプロセスがどれだけ重要かということを表しています。検索ツールが「人」と「個人」という語が関連していると把握している場合、またユーザーが勝手にレジストリ要素を作成できないようになっている場合、ようやく定義の重複を避けることができたと言えます。

MarkLogicのロールベースのアクセス制御によるメタデータの管理

メタデータレジストリは、特に管理アイテムに対するアクセス制御の概念に対応するものです。管理アイテム（administered item）というのは、制御されなかった場合に他のシステムに影響を及ぼすデータのことです。管理アイテムを表示・更新できるロールは、定義しておく必要があります。MarkLogic のロールベースのアクセス制御によって、メタデータレポジトリを注意深くコントロールならびに監査できます。

MarkLogic では、メタデータレジストリ内のドキュメント 1 つ 1 つに対して、柔軟に複数のロールを関連付けられます。これは「ロールベースのアクセス制御」（RBAC：Role-Based Access Control）と呼ばれます。システムに新規ユーザーが追加されると、この人に 1 つあるいは複数のロールが割り当てられます。その後、クエリは更新の前にどのロールの人に対してドキュメントの更新が許可されているかをチェックします。

例えば、ヘルスケア企業においてあらゆる保険金請求に関するデータ標準の定義を担う部署があるとします。メタデータレジストリは、「claims-metadata-manager」というロールを作成できます。すべてのタームとデータ要素は、このロールに関連付けることができます。ユーザーは、システム変更のインパクトを理解するトレーニングを修了した場合にのみ、このロールを持つことができます。このロールがない人はこれらのドキュメントを見ることはできますが、変更することはできません。

MarkLogic のロールベースのアクセス制御によって、メタデータレポジトリを注意深くコントロールならびに監査できます。

MarkLogicの検証・データクオリティツールによるメタデータの管理

データスチュワードには、自分の分野におけるデータ標準を高い品質に保つという責任もあるかもしれません。例えば、保険金請求メタデータ担当のデータスチュワードには、さまざまな形式のドキュメントを検証し、請求ドキュメントの 1 つ 1 つに対してデータクオリティのスコア（1 から 100 などの）を付けるための規則を定義するという任務があるとします。請求のうち、不正確なもの、不完全なもの、確認作業中のものには、低いスコアが付けられます。レポーティングシステムには、スコアが 70 点以上の請求のみを含むようにできます。

表形式のシステムでデータクオリティを管理する際の問題の 1 つとして、複数のシステム全体におけるデータクオリティスコアに関する一元化されたコンセプトが存在しないということがあります。データクオリティはアドホックなプロセスになることが多く、個々の業務部門がデータの検証とスコアリングに別々の方法を使用することが考えられます。また一つのリファレンステーブルを変更した結果、数百万件のレコードのデータクオリティスコアに影響が及ぶことも良くあります。

MarkLogic では、すべてのドキュメント内にデータクオリティ情報を一つのメタデータプロパティとして格納することによってこの問題に容易に対処します。MarkLogic ではあらゆる機能において、このプロパティでデータクオリティを格納しています。XML スキーマでドキュメントを検証したり、クオリティ指標を格納するための特別なテーブルを作るための特別なツールは不要です。MarkLogic では、標準機能としてデータクオリティを管理できます。

「MarkLogicのロールベースのアクセス制御によって、メタデータレポジトリを注意深くコントールならびに監査できます。」

「表形式のシステムでデータクオリティを管理する際の問題の1つとして、複数のシステム全体におけるデータクオリティスコアに関する一元化されたコンセプトが存在しないということがあります。」

ここまで、エンタープライズメタデータの管理にMarkLogicの主要機能がどのように使用されるのかについて見てきました。ここからは、ヘルスケアならびにライフサイエンス企業において、これらの機能を使ってパフォーマンス、品質、サービス、安全性、コスト削減をどのように行っているかについて、具体的な例をいくつか取り上げます。まず1つめの例は、シンプルなビジネス用語集（グロッサリー）です。

ユースケース#1：ビジネス用語集管理と検索

多くのメタデータ管理システムにおいて、一番最初のタスクはシンプルです。つまり各業務部門が個々のプロジェクトで使用している用語を一元的に確認できるようにするというものです。これらの用語はプロジェクトに新しく参加する人に対するオリエンテーションとして重要なだけではなく、ビジネス要件が詳細に説明され、プロジェクトを通じてかつ組織全体において一貫性があるようにするためにも使えます。

ビジネス用語集は、まず列が2つ（ビジネス用語とその定義）あるスプレッドシートのかたちになります。MarkLogicではこういったスプレッドシートを素早くドキュメントストアにロードできます。この際に、各用語が1つのミニドキュメントとなります。mlcp（MarkLogic Content Pump）は、あらゆる区切り文字ドキュメント（CSVなど）を自動的に分割して、これらのドキュメントに入れることができます。

データをロードした後の最初の目標は、簡単な検索アプリケーションを作ることです。これはMarkLogicの検索機能で実現できます。最初に、クエリをサービスに渡すためのwebフォームを作成します。これはクエリにマッチした用語に順番を付けて返すものです。この機能により、ランキングされた（順位に並んだ）マッチが自動的に提供され、検索結果のテキスト内のキーワードが直接強調表示されます（コンテキストにおいてキーワードを確認できます）。

またMarkLogicの検索ランキングシステムでは、「用語」列でマッチしたキーワードの方を「定義」列でマッチしたものよりも上位にランキングすることもできます。このような重み付けは「ワードクエリ重み付け設定」（word query weight setting）と呼ばれます。

マッチしたコンテンツ自体も、ランキングに影響します。例えば、「NoSQL」というキーワードが書籍のタイトルにある場合、これはNoSQLに「ついて」の本だという可能性が高いです。一方、「NoSQL」が書籍内の脚注部分にある場合、そうは考えられません。

この結果、ユーザーが探しているものが検索結果の一番上に来るようになります。これは、自社のビジネスユーザーのニーズに基づいてカスタマイズされた高品質な検索エンジンのようになります。

ここでこのアプリケーションを実装したところ、ユーザーたちも満足だったとしましょう。しかしここでユーザーたちが自分でも用語を追加したいということになりました。また誰が用語を追加したのか、それを承認したのは誰かということもわかるようにしたい、と言っています。

次のステップとして、各用語に簡単なワークフロー（新規追加された用語、レビュー中の用語、プロジェクト担当者によって承認された用語）を追加します。これを行うには、MarkLogicのCPF（Content Publishing Framework）を利用できます。CPFを使うと、各用語の状態を管理するワークフローを作成できます。例えば、ある用語はまず「New Proposed Term」（新規追加候補）として扱われ、これが受け付けられると「Draft」となります。その後、この用語は「Under Review」（レビュー中）、「Approved」（承認済み）、最後に「Published」（提供済み）となります。それぞれの段階においては、承

認されたユーザーのみがこのビジネス用語を次の段階に移動させることができます。提供許可前の用語をユーザーに見せないようにしたり、承認プロセスにある新規用語を確認できるようにできます。

この前の「履歴」メタデータの例と同様に、用語がチームによって承認されるまでにかかった時間をトラッキングし、レポートを作成できます。

ユーザーがどんどん用語を追加していくにつれて、分類やタイプによって用語をフィルタリングしたくなるでしょう。ここで「定義」のソースを分類してみましょう。いくつかの用語定義は政府によるものかもしれませんし、あるものは業界標準からのもの、またあるものは個々のチームやプロジェクト固有のものかもしれません。どのような分類システムであれ、MarkLogic を使うと検索に対して簡単にフィルタを追加できます。この機能は、ファセット検索と呼ばれます。ファセットは、検索結果に追加された複数の検索フィルタのようなものです。通常は検索結果の脇の方に、分類の短いリストとして表示されます。ファセットならびにカテゴリ別の結果の件数は、MarkLogic の検索ツールで管理されています。

次に、他の用語と関係している用語がたくさんあることに気付いたとします。ある用語は製品カテゴリ、他の用語はこのカテゴリに含まれる製品である場合が考えられます。簡単に言うと、ユーザーはフラットな用語の世界から、タクソノミー管理の領域に入ったということになります。これらのサブカテゴリに「broader-term」（広範な用語）というデータ要素を追加することで、ツリー構造のレポートを作成できます。SQL と違って MarkLogic はそもそもリカーシブなので、これを簡単に行えます。ここではまずルート要素を表示し、次にそのサブ要素を表示するような関数を作成します。個々のサブ要素に対して、再度関数を呼び出すことができます。数行からなる関数を使うことで、複雑な深さを持つ詳細なレポートを作成できます。

ビジネス用語集の用語は、他の用語と素早く関連付けることができ、用語に同義語や「もっと詳しく」リンクを表示させることができます。

また MarkLogic のカスタマーの多くは、ビジネス用語の表示に web 標準を使っています。MarkLogic やパートナーは、SKOS（Structured Knowledge Organization System）という標準を使っています。SKOS を使うと、素早いインポートや他の用語とリンク・結合ができる標準語彙を利用できます。

キーワードを入力したけれども、探しているのと完全に同じ用語がない場合、MarkLogic では任意のドキュメントに基づいた別のクエリも利用できます。ある用語に似ている用語も探すことができます。

例えば特定の用語グループを扱っているとします。また新しい用語が追加されたり、特定タイプの用語が「Under Review」（レビュー中）から「Published」（提供済み）に変わった際にメールやメッセージで通知することもできます。

時間の経過に伴い必要に応じて新しいデータ要素が追加され、核となるビジネス用語集は少しずつ拡大していきます。要件が変わるにつれて、アプリケーションも成長し進化します。要素が増えている間も、既存のプログラムは使われ続けます。

自分たちでビジネス用語集を一から作りたい場合、MarkLogic に繋いで利用できるビジネス用語集管理ツールがサードパーティから提供されています。これらのツールの多くでは、SKOS ファイルを直接出し入れ（インポート / エクスポート）できます。これらのサードパーティツールは MarkLogic の能力を活用し、素晴らしいユーザーエクスペリエンスを提供します。

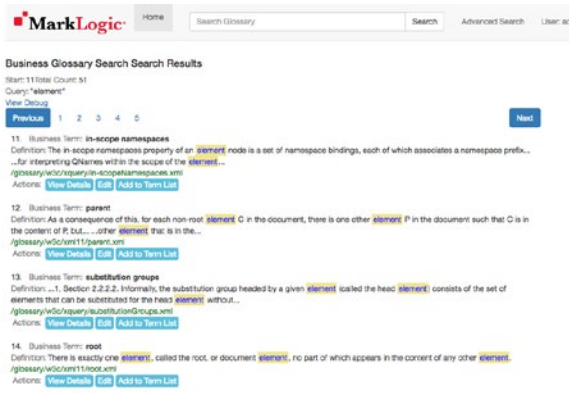


図3：ビジネス用語集の検索結果の例

次のユースケースは、メタデータの最も困難な局面の 1 つを扱っています。ここでは、リファレンスデータを扱い、高可用性のリアルタイムサービスが求められています。

ユースケース#2：リアルタイムのリファレンスデータルックアップ

このケーススタディでは、MarkLogic を使って複雑なリファレンスデータをより意味のあるまた検索可能なものとする方法について紹介します。MarkLogic のスケールアウトアーキテクチャによって、リアルタイムかつ拡張可能なデータエンリッチメントサービスが提供されることを確認します。ここでは数値コードに対して意味のあるラベルを追加するエンリッチを行います。

レガシーの COBOL/ メインフレームシステムでは、利用できるメモリの制約がよく問題になっていました。この制約のために、開発者は普通の名前の変数の代わりに短い数値コードをよく使用していました。このためある人の性別を記録したい場合、文字列「女性」の代わりに、数値コード（2 など）を格納していました。その後、それぞれのアプリケーションはリファレンスデータ（参照データ）を使って、各数値コードを人間が理解できるものに変換して画面やレポートに表示していました。

メモリの価格が下がった現在でも、人間が理解できる文字列ではなく、数値コードで格納することを好む組織はたくさんあります。例えば、米国の郵便番号では、「CA」はカリフォルニアのことです。またヘルスケアの請求で、「つわり」が 11.2 というコードとして入力されます。このようなデータは、リファレンスデータと呼ばれています。というのも、この短縮形のコードは共通の標準をリファレンス（参照）するからです。一般的に、大規模なヘルスケア組織においては、システム内で数千ものリファレンスコードが使用されている可能性があります 。

エンタープライズのリファレンスデータを管理すること自体、複雑なプロセスとなります。さまざまなシステムやレコードにおいて州 / 国コード、郵便番号、通貨コードなどが異なっている可能性があるからです。単一のシステム内においても、時間の経過とともにコードやその意味は変わっていきます。コード（ヘルスケア課金コードなど）によっては、数万の値が含まれ、それぞれのコードに異なる複雑なルールが付いているものもあります。

人間が理解できる文字列の代わりにリファレンスコードを使用することは、RAM やディスクが高価だった時代にはいくつかの利点がありました。しかしながら、現代的なドキュメント指向のシステム（ユーザーが検索エンジンに文字列を入力するもの）が数値コードがわからないと利用できないならば、かなり使いにくいです。MarkLogic のベストプラクティスでは、数値コードならびに普通の言葉の検索に利用できるラベルを、検索しやすい形式で格納します。この形式は、「ハーモナイズされた」あるいは「カノニカルな」形式と呼ばれます。カノニカルな形式は、トランザクション、検索、分析という 3 つの別個の操作に対応できます。これらのシステムでは、各数値をラベルとのペアとして格納します。XML のペアは以下のようになります。

```
<PersonGenderValue>2</PersonGenderValue>
<PersonGenderLabel>Female</PersonGenderValue>
```

数値コードのリファレンスデータを、値 / ラベルのペアの形式に変換するためには、何らかの方法でラベル要素を追加して挿入する必要があります。これを行うために、ここではシンプルなリファレンスデータ管理フレームワークを使用します。

ラベルを挿入するために、このコードではルックアップ関数を使用します。ここで使用されているリファレンスデータのコード（PersonGenderCode）の名前ならびに現在の値（2）が渡されます。これにより、このコードはラベルを返します。この関数呼び出しは次のようになります。

```
ref:label("PersonGenderCode", "2")
```

これはラベル文字列「Female」を返します。

メタデータ管理システムは、他のアーキテクチャでは必要なオーバーヘッドなしでこれらのラベルのルックアップ関数を素早く実行できる関数を提供する必要があります。これをどのように MarkLogic に実装するのか見ていきましょう。

リファレンスデータを格納する場合、通常これをシンプルな JSON または XML ファイル内に格納します。こういった JSON ファイルは以下のようになります。

```
{
   "CodeName":"PersonGenderCode",
   "Items" :
      {"F": "Female",
       "M": "Male",
       "U": "Unknown"
      }
}
```

図4：リファレンスデータを表現したJSONファイル

このファイルを読み取り、短縮形のコードをルックアップし、完全な文字列を返すような簡単な関数を記述できます。しかしながら、このルックアップは高速である必要があります。どうしてでしょうか。例として、毎日100万件の新規ドキュメントが読み込まれ、その各ドキュメントには変換が必要なコードが10個ずつある場合を考えてみましょう。このサービスが遅い場合、かなりのコストが発生します。

MarkLogicでは、こういったルックアップを高速化する機能が2つあります。1つめはサーバーフィールドの利用です。サーバーフィールドを使うと、あらゆるデータコードを直接RAMに置くことができ、時間がかかるディスクへのアクセスを回避できます。2つめの機能は「マップ」と呼ばれる高速なインメモリのキー/バリューストアです。マップはキー/バリューストアで、キーは短い値、返されるアイテムはラベルです。サーバーフィールドとマップを使用することで、これらのルックアップ関数が最初に利用された際に、これをRAMに「固定」できます。これは不要になるまでそのままです。

たまに2、3のドキュメントを変換するだけであれば、サーバーフィールドやマップは不要です。これはシンプルなJavaScriptあるいはXQuery関数で行えます。これらのクエリではMarkLogicのユニバーサルインデックスを使用し、時間がかかるリファレンスデータ全体へのスキャンを行ってコードラベルをルックアップする必要はありません。しかしながら、常にミリ秒レベルの反応時間が求められる場合、MarkLogicではこのようなリクエストを効率的に扱えるということを覚えておいてください。

このように、MarkLogicは極めて信頼性が高くまた拡張性に富んだエンタープライズ仕様のドキュメントエンリッチサービスを提供できます。リファレンスデータのルックアップは、こういったサービスのほんの一例です。

ユースケース#3：セマンティック検索によるサービスやエクスペリエンスの改善とコスト削減

ビジネス用語集のケーススタディでは、MarkLogicがタクソノミーデータをきれいに管理できることを紹介しました。ここからは、こういったツリー構造により時間とお金が節約でき、組織自体が柔軟になることを紹介していきます。

XMLならびにJSONのリカーシブ（再帰的）な構造は、MarkLogic内で容易に管理できます。MarkLogicの高度なクエリ言語（XQuery、JavaScript、SPARQL）を使うと、複雑なネットワーク状やツリー経路アルゴリズムを容易に扱えます。これらは最適なものと言えます。

タクソノミーが本当の形式のメタデータであると言うと、異論もあるかもしれません。これをデータを組織化するための構造だと考える人もいます。しかしながらタクソノミーに関しては、こういった構造を利用することで探しているドキュメントをすぐに発見できるという点でかなり便利だということを理解する必要があります（入力したキーワード自体がドキュメント内になかったとしても結果が得られるのです）。タクソノミーは一連の「サブジェクト（主題）ツリー」として考えることができます。ここでは、ツリーの一番上に抽象的な概念があり、特化したサブジェクトはツリーの下の方にあります。これで、サブジェクトに基づくタクソノミーがどのようなものになるのか、だいたいわかると思います。

次に、任意のドキュメントの分類を、コンピュータがどのように自動化できるのかについて考えてみましょう。ドキュメント内の語を見て、これをそれぞれの主題と比較します。ドキュメント内の語が主題とマッチした場合、このドキュメントはこの主題に関するものだと考えることができます。この主題に関連する用語や同義語がある場合、これらの語もツリー構造内の主題のそばに格納されます。

タクソノミーは、ドキュメントの自動分類用の階層型規則を格納するのに適した場所だと考えることができます。ドキュメントを分類するために、ドキュメント内のすべての語を分析するサービスにドキュメントを送ります。この分析に基づいて、ドキュメントはツリー内の場所への「距離」に基づいて分類されます。

従来は、ドキュメントの自動分類システムは遅く、お金がかかり、実装も困難でした。しかし MarkLogic によって、ドキュメント分類システムの統合が楽になりました。それでは検索の改善に関する 2 つの局面について見ていきましょう。1 つめは SPARQL による「キーワード拡張」、2 つめはドキュメントメタデータのドキュメント内への格納です。

キーワード拡張は、1 つあるいは複数のキーワードを、これと関連する用語を含むように「拡張」するプロセスです。例えば、「激しい動悸」で検索した際に、検索システムが自動的に最寄りの心臓外科を探すというようなことが考えられます。

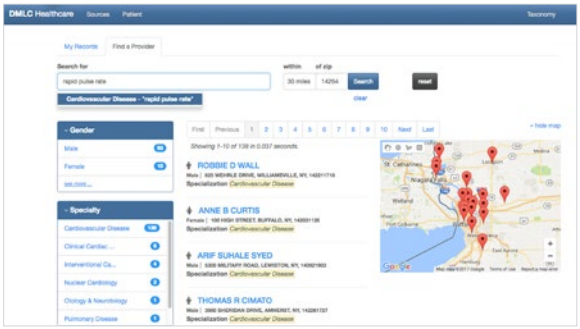


図5：症状（激しい動悸）と専門領域（心臓病）を関連付ける検索アプリケーション

MarkLogic では、関連する語を RDF グラフとしてネイティブに格納できます。ますグラフ内で最初のキーワードを探し、このキーワードの周りで関連する語を探すだけでいいのです。これらの語は、代替ラベル、同義語、より広範な語などに関連付けられています。関連する語の範囲は、クエリ内のセマンティックな距離で調整できます。

もう 1 つのテクニックは、MarkLogic による各ドキュメントに対する分類メタデータの追加です。このメタデータは、特定ドキュメントにデータを挿入するために作成したクエリによって得られたものだったり、あるいは Smartlogic* などのドキュメント分類エンジンによって得られたものだったりします。新しいメタデータがドキュメントに挿入されるとすぐにインデックスが付けられ、すべての語が MarkLogic のワードインデックスに追加されます。

ドキュメント分類タグの追加方法を問わず、これらのドキュメント内に追加されたメタデータによって豊かな検索体験がもたらされます。MarkLogic では拡張キーワードと追加されたメタデータを組み合わせることで、正確な検索ランキングが得られます。これにより探しているドキュメントは通常、検索結果の最初のページに表示されます。

すべての組織において、一貫性があり正確な検索結果がすぐに得られるようになっている訳ではありません。ユーザーが適切なドキュメントを探したり、見つからない場合はドキュメントの作り直しに何時間も費やしていることもあるでしょう。適切な情報を適切な人に届けようとしている組織は、キーワードだけに頼った検索では、検索結果画面の最初のページに適切なドキュメントを表示させることができないことに気づいています。ここにおいて、MarkLogic によるエンタープライズレベルのメタデータ管理システムが登場してくるのです。

さらに詳しく

柔軟なドキュメント / グラフのメタデータアーキテクチャを使うと、従来の表形式しか扱えないシステムは石器時代のもののように見えてくるでしょう。高度なメタデータ管理システムを短期間で本番稼働できるセキュリティや拡張性があるのは MarkLogic だけです。

リレーショナルを超えて

このホワイトペーパーを読んで、リレーショナルデータベースは今日のデータ問題の解決には適していないことを確認してください。
https://www.marklogic.com/resources/beyond-relational-jp/

MarkLogic の概要

MarkLogicは分断されたデータの統合に世界で最も適したデータベースです。この概要データシートで、MarkLogic の差別化要因やカスタマー事例についてご確認ください。
http://jp.marklogic.com/resources/overview-japanese

MARKLOGIC K.K.
150-0001 東京都渋谷区神宮前1-5-8 神宮前タワービルディング 13F
+81 03 4540 0337 | jp.marklogic.com | MarkLogic-JP@marklogic.com

150-0001 東京都渋谷区神宮前1-5-8 神宮前タワービルディング 13F
+81 03 4540 0337
jp.marklogic.com | MarkLogic-JP@marklogic.com