MarkLogic®

データモデリングの新しい考え方

ホワイトペーパー • 2016年3月

これまでのデータモデリングでは不十分な時代が来ています。今日、データの迅速な統合とスマートなアプリケーションの開発において、リレーショナル技術が足枷となっているため、データモデリングに関してより優れたアプローチが求められています。このため、NoSQLとセマンティックによる「マルチモデル」アプローチが選択されています。

現状の確認

データの扱いに関して、改善の必要があるでしょうか？　あるいは今のままで大丈夫でしょうか？　以下の質問に答えていただくと、現状がわかります。「YES」が多いほど、今のデータベースはニーズに応えられていないということになります。

		YES	NO
ビジネスに関する質問	1. 重要なデータのうち、まだデータベースにまだ入っていないものがある。	✓	✗
	2. 関連するデータを複数のデータベースで扱っている。このためデータの全体像は把握できない。	✓	✗
	3. コアシステムから派生した大量のデータセットが存在しており、一元的に管理することができない。	✓	✗
	4. 大規模なITプロジェクトのうち、データ統合ができないことが原因で、予算超過やスケジュールの遅れが発生しているものが存在する。	✓	✗
	5. データベースのスキーマがあまりに複雑で、扱うことが困難。	✓	✗
技術的な質問	6. データモデリングのせいで、アプリケーション開発が遅れたり滞ったりしたことがある。	✓	✗
	7. 「取りあえず動かす」ために、リレーショナルテーブル内の列名を変更したり、列名の解釈を変えたりしたことがある。	✓	✗
	8. データベースのスキーマが頻繁に変更される（月1回など）。またその変更がうまくいかないことがある。	✓	✗
	9. 重要なメタデータやリファレンスデータが、Excelなどの形式でデータベース以外の場所に置かれている。	✓	✗
	10. ミドルティアが複雑なために、パフォーマンスの問題やバグが発生している。	✓	✗

目次

はじめに

データモデリングは、どんな組織においても極めて重要です。データモデルとは、情報の格納方法、現実世界のヒト/場所/モノの記録方法、またそれら相互の関係を詳細に定義するものです。例えば、企業には顧客がいて、顧客は何かを購入します。これらのエンティティ（実体）ならびに関係性をモデリングすることで、データの利用・共有のための基盤ができ、またアプリケーション開発の方向性が決まります。簡単に言うと、データモデルとは、業務が行われる世界を組織がどのように認識しているのかを表現したものです。

しかし残念なことに、データモデリングの従来のアプローチは十分ではありません。データモデリングのプロセスの一部として、概念モデルの作成があります。この概念モデルは、対象分野におけるエンティティと関係性を扱います。これを論理モデルに翻訳し、さらにこれを物理モデルに変換することでデータベースに実装できるようになります。このアプローチは「ERモデリング（実体関連モデリング）」と呼ばれ、1976年の登場以来、標準となっています。

しかし実際には、データベース設計者は、概念モデルを無視しています。ある調査によると、フォーチュン100企業において、概念ERモデリングは1つも存在しなかったことがわかっています。[1] なぜそうなっているのでしょうか？　ここで問題となるのは、現実の世界は複雑すぎてリレーショナルデータベースの行や列に当てはめられない、ということです。リレーショナルデータベースのER図を見ても、ビジネスを理解することはほとんどできません。また、そこに記述されている論理の全体像を把握することもできません。そこにはある種の分断があり、また物理モデルは対象となる世界を複雑かつ貧弱にしか描写できません。

複数のリレーショナルデータベースを完璧に統合・保守するために、企業は多大な努力を払っています。しかし遅かれ早かれ、何らかの変更は避けられません。新しいデータソースが登場したり、これまでとは異なる問いに答えなくてはならなかったり、またデータを新規システムに統合しなくてはならなくなったりするからです。リレーショナルデータベースでは、このような状況にうまく対応できません。しかし今やこういったことはごく当たり前です。

このため、データの迅速な統合とスマートなアプリケーションの開発のために、これまでのアプローチに代わるものとして、NoSQLとセマンティックによる「マルチモデル」が選択されています。NoSQLとセマンティックにより、モデルが柔軟かつわかりやすく、便利になります。MarkLogicユーザーの一人は、NoSQLとセマンティックは「リレーショナル技術の制約をなくす」ものだと言っています。[2]

現在、市場には新しいデータベースが多数存在していますが、MarkLogic® だけがエンタープライズ仕様のマルチモデルデータベースであり、NoSQLドキュメントデータベースとセマンティックのあらゆる長所が1つのプラットフォーム上にまとめられています。この理由により、BBC、NBCユニバーサル、ブロードリッジ、アムジェンといった業界のリーダー企業が、MarkLogicのマルチモデルに基づいて、データモデリングを新しく考え直しています。

より優れたアプローチの必要性

今日の組織にとって、データ統合は最も緊急性のある課題です。例えば、銀行では監督が厳しくなっているため規制報告を改善する必要があります。企業ではM&Aが進んでいます。また政府は国防を改善する必要があります。

国防に関しては、「9/11コミッションレポート」はデータ統合の重要性について「あらゆるソースからの情報を統合して、敵の全体像を把握するのが『スマート』な政府だ」と述べています。[3] 残念ながら、従来のデータベース設計では、データ統合の実現に必要な情報を捕捉できません。それどころかデータ統合に役立つ情報の捕捉すら完全にはできません。データ統合をより速く容易に行うためには、これまでとは異なるデータベースが必要です。

1　M. L. Brodie and J. T. Liu. "The power and limits of relational technology in the age of information ecosystems." Keynote at On The Move Federated Conferences, 2010.

2　ブロードリッジ・ファイナンシャル・ソリューションズのパオロ・ペリッツォーリ氏（SVP & Global Head of Architecture）のインタビューを参照のこと。https://www. youtube.com/watch?v=TB1tLrM_z1k.

3　National Commission on Terrorist Attacks on the United States, The 9/11 Commission Report: Final Report of the National Commission on Terrorist Attacks upon the United States: Official Government Edition (Washington, DC: U.S. G.P.O. 2004) p.401,416. <https://www.gpo.gov/fdsys/pkg/GPO-911REPORT/pdf/GPO-911REPORT.pdf>

これまでのデータモデリングでは不十分

できると言われていたこと

有名なコンピュータサイエンスの論文(1976)において、ピーター・チェンは現実世界に関する情報を「エンティティ」と「関係性」として把握するというアイディアを発表しました。[1] この新しいアプローチは「ERモデリング」と呼ばれ、複数のストレージならびにトランザクションのモデルを統一し、現実世界の表現方法を改善しようというものです。これはその後すぐに、データモデリングの標準となりました。

ERモデリングでは、データベース設計者は最も重要なエンティティに注目します。例えば、物理的なオブジェクト（従業員、車、家など）や概念的なオブジェクト（会社や職業など）があります。次に、属性を絞り込みます（名前、年齢、住所、年収など）。この情報は、その後、物理データベースの実装に役立ちます。このプロセスには、3つの図式モデルがあります。以下のようになります。

概念データモデル

概念データモデルは、一般的なエンティティと関係性の概要を特定するものです。ここでは、経営陣も理解できるように専門用語は使用しません。これは今後、技術的な仕様を決定する際に参照されます。

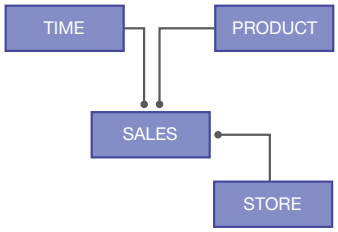

図1：単純な概念データモデルの例（出典：http://www.1keydata.com/）

論理データモデル

論理データモデルは、概念データモデルよりも詳細です。これはヒト/場所/モノとその関係性（それらの間のルールやイベント）を標準化します。この時点では、実装の技術を指定するところまでは行われません。しかし今後の物理モデルを想定したものとなっています。ここでは、エンティティ（表）、属性（列/フィールド）、関係性（キー）を正規化して表現しています。

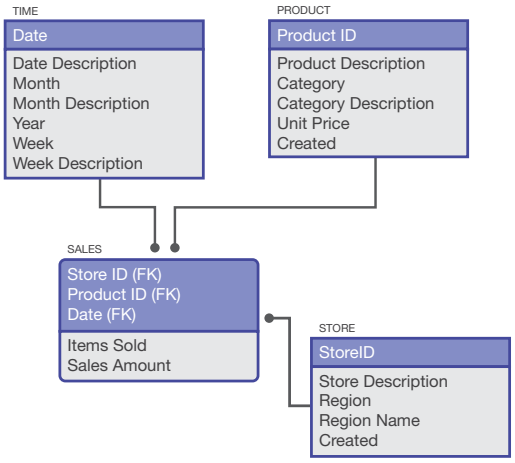

図2：単純な論理データモデルの例（出典：http://www.1keydata.com/）

物理データモデル

物理モデルは、データベース内での実際のデータの格納方法に最も近い表現方法です。これは技術的に詳細に記述され、技術（あるいはベンダー）に依存しています。実際の表名と列名が含まれ、パフォーマンス目標、インデックス、制約定義、データの分散、トリガー、ストアドプロシージャなども考慮します。

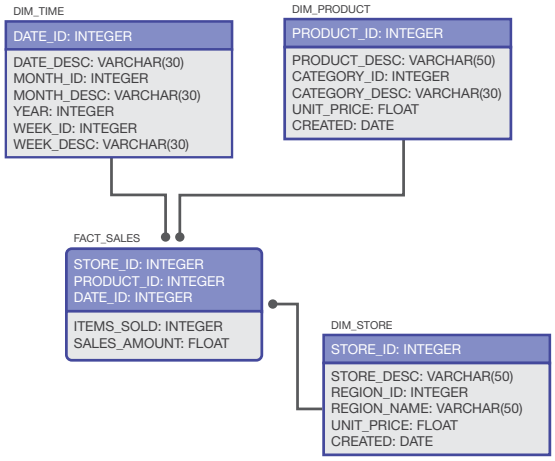

図3：単純な物理データモデルの例（出典：http://www.1keydata.com/）

1 P. Chen. The entity-relationship model - toward a unified view of data. ACM Transactions on Database Systems, 1(1):9–36, 1976. <http://www.inf.unibz.it/~nutt/IDBs1011/IDBPapers/chen-ER-TODS-76.pdf>

「一番の問題は、世界はあまりに複雑だということと、ビジネスの展開が速すぎてITがついていけなくなっていることです」

実際に実現されたこと

残念ながら実際には、エンティティや関係性を記述するはずの概念的ER図は、すぐに物理モデル（リレーショナルデータベースの厳しい制約に基づいて設計されている）に押しやられてしまいます。概念モデルが作成されないこともよくあります。データベース設計者たちは、開発指針となる基礎を作らずに、最初からすぐに物理モデルに取り掛かります。

ある調査によると、フォーチュン100企業10社においては概念的ERモデリングが全く存在しませんでした。また、一般的に言ってデータモデリングに大きな問題があることも判明しました。この調査で明らかになったのは「典型的なフォーチュン100企業においては、約1万個の情報システムが存在し、リレーショナルデータベースには100以上のテーブルが含まれ、それぞれに属性が50から200程度含まれています。決められた概念モデルや正規化手法のいずれも使用されていません。物理モデルの作成は往々にしてパフォーマンス問題が発生するまで先送りになることが多いです」ということです。[2] 他のデータベース専門家も「アプリケーションの50%において（データベース設計は）根本的に不備がある」と言っています。[3]

データベース設計担当者が適切なデータモデリングができない理由

一番の問題は、世界はあまりに複雑だということと、ビジネスの展開が速すぎてITがついていけなくなっていることです。現実世界を描写する概念モデルは、物理モデル（DBAやアーキテクトがリレーショナルデータベース構築の際に利用するもの）にうまく翻訳できません。データベース設計者は、物理モデルにかなりの時間を費やしています。これはリレーショナルデータベースで扱えるように一貫性のあるデータを設計するためです。この従来のアプローチでは、主キー、外部キー、結合によって表を結合する固定的なシステムが生み出されます。残念ながら、これは対象となる世界をうまく表現できません。アプリケーション開発において、これは望ましくありません。というのもインピーダンスミスマッチの問題があるからです。また、変更が必要な場合（新規データソースの追加、サイロ化されたデータソースの統合）にも問題となります。こういった変更は、今日では極めて一般的であるにも関わらず、これにうまく対応できないのです。

その代償

ほとんどの組織では、ビジネスのスピードにIT部門がついていけません。このデータベースアーキテクチャでは、データベース内の表や列がどんどん増えていきます。その結果、データモデルと実際のビジネスエンティティとの関係や、モデルを一か所変更した場合の他の部分への影響などを誰も理解できなくなります。データベースは極めて複雑になり、理解するのが困難で、変更は実質的に不可能です。そして誰もこれに関わりたくなくなってしまいます。

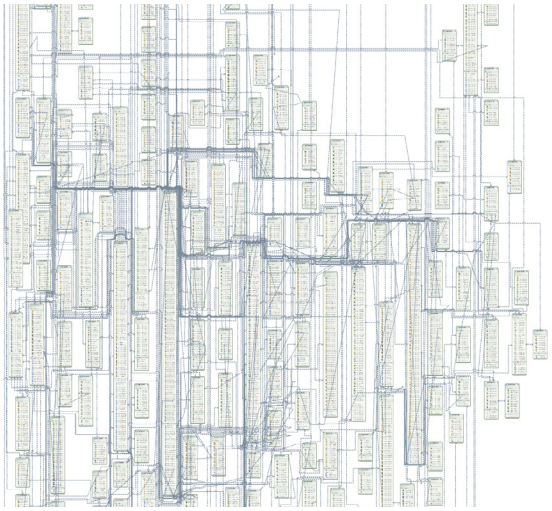

図4：複雑なリレーショナルモデルの一例
（出典：http://www.plandora.org/docs/mer.png）

また、ほとんどの組織においては、スキーマが1つしかないということはあり得ません。通常は、さまざまなサイロに複数のスキーマが存在しています。企業合併/買収、新規データソースの統合などがあると、新しくスキーマが追加されます。こういったスキーマは、時間の経過とともにどんどん増えていき、ますます複雑になっていきます。このため、アプリケーションやレポートで必要な一部のデータを、新しく作ったデータマートに出すことが多くなります。もちろんこれによって、そこここに浮遊するデータのコピーの数が増え、セキュリティ、データガバナンス、データ統合ならびにアプリケーション構築の問題も増えます。

このように分断されたサイロのデータを統合するには、類似する属性を把握する必要がありますが、これはかなり大変です（データに関連付けられたメタデータもないのが普通です）。例えば、スウェーデンのある銀行は、行内の多数のデータベースにおいて「nominal amount（額面価値）」の定義が31種類もあることを発見しました。

2 M. L. Brodie and J. T. Liu, 2010.

3 Roberto V. Zicari, "How good is UML for Database Design? Interview with Michael Blaha." ODBMS.org, July 25, 2011. <http://www.odbms.org/blog/2011/07/how-good-is-uml-for-database-design-interview-with-michael-blaha/>

「ITシステム関連コストの40%は、データ統合問題に起因するものです」

また、それらのデータが実際に何を表しているのかを説明する追加情報は何もありませんでした。このように重要な用語1つの意味が曖昧なことによって、コストやリスクが次々と発生します。しかしこのような問題のある用語が、数十万個も存在しているとしたらどうでしょうか。

これは、データベース管理者だけの問題ではなく、組織全体で重大なリスクが発生することになります。またコストが増加し、プロジェクト期間が延びます。あるリサーチレポートによると、「ITシステム関連コストの40%は、データ統合問題に起因」します。[4] また2015年だけでも、データ統合ソフトウェアに5000億円程度が費やされています。[5]

これまでの30年間、リレーショナルデータベースでこの問題を解決しようとしてきましたが、従来のデータモデリングではうまくいかないのです。今日の組織が対処しなくてはならない多様化していくデータと、リレーショナルデータベースとの間の根本的なミスマッチは、何も対処しなかったならば悪化するばかりです。フォレスター・リサーチは、この問題について「データの構造が複雑になり、量が増えるにつれ、従来のリレーショナルデータベース（事前定義されたスキーマが必要）では不十分になる」と述べています。[6]

リレーショナルモデリングの個々の問題

この項では、今日のデータモデリングに関してリレーショナルデータベースが不得意とする分野をいくつか説明します。リレーショナルデータベースが今日のデータに適さない主な理由をすべて確認したい場合は、ホワイトペーパー『リレーショナルを超えて』をお読みください。アプリケーション開発や拡張性に関する議論なども取り上げています。

エンティティと関係性のモデリングの難しさ

リレーショナルデータベースのER図を見ても、現実世界のビジネスモデルや、そこに記述されている論理全体を把握することは不可能です。このデータモデルは、今日のダイナミックかつ複雑な関係性をうまく表現できません

リレーショナルモデリングでは、通常以下のステップを行います。

1. 各属性を1つのフィールドとする
2. 属性を表にまとめる
3. 各テーブルに主キーを割り当てる
4. 重複する属性がないようにする

ここで問題になるのは、エンティティ間の関係性は、本質的に表内の行と列を使って定義するしかないということです。あるいは、外部キーと制約に基づいて表同士の関係性を表すポインタを利用します。いくつかの関係性（アソシエーションのような継承された関係性など）は暗黙的であり、データベースの外部でドキュメント化され管理されています。場合によっては、単に無視されます。情報が複雑になるにつれて、列、行、ポインタが増えます。これによりモデリングがかなり困難になり、クエリはさらに困難になります。

またビジネスの変化に応じて、スキーマのアップデートも必要です。残念なことに、リレーショナルスキーマの変更には時間とお金がかかるのが普通です。あるMarkLogicユーザーによると「テーブルの列を足したり、置き換えたりするだけでも1億円規模の作業となる」とのことです。[7]変化に対応するには、リレーショナルデータベースでは、通常、多対多の関係性を扱うために結合テーブルを別途作成する必要があり、また下流での問題を回避するために防御的プログラミングを行う必要があります。このような手作業では問題が発生しやすくなります。また変化に対する長期的な対処策とはなり得ません。

またリレーショナルデータベースでは、ヒト/場所/モノのモデリングならびにクエリに関して全員が合意する標準がありません。つまり、リレーショナルデータベース内の各属性は独立しており、それぞれのデータベース管理者が別々にモデリングしています。これに加えて、データが多様なために大量のNULLセルを含む表が多々あります（「疎データ」）。あるいは列名が一般的なVARCHARデータ型で記述され、その非常にわかりにくい名前の意味が

4 M. L. Brodie and J. T. Liu, 2010.

5 Gartner. Forecast: Enterprise Software Markets, Worldwide, 2011-2018, 4Q14. 2014. <https://www.gartner.com/doc/2944023/forecast-enterprise-software-markets-worldwide>; Includes: Data Integration Tools, Data Quality Tools, and Other Data Integration Software.

6 The Forrester Wave™: NoSQL Document Databases, Q3 2014; An Evaluation Of Four Enterprise-Class NoSQL Document Databases Vendors. Noel Yuhanna with Leslie Owens, Emily Jedinak, and Abigail Komlenic.

7 According to one customer at a leading Fortune 100 technology company, the task of adding a column could take them up to a year and cost over a million dollars. For other more complex data modeling projects involving master data management, even lengthier timelines of over five years have been reported.

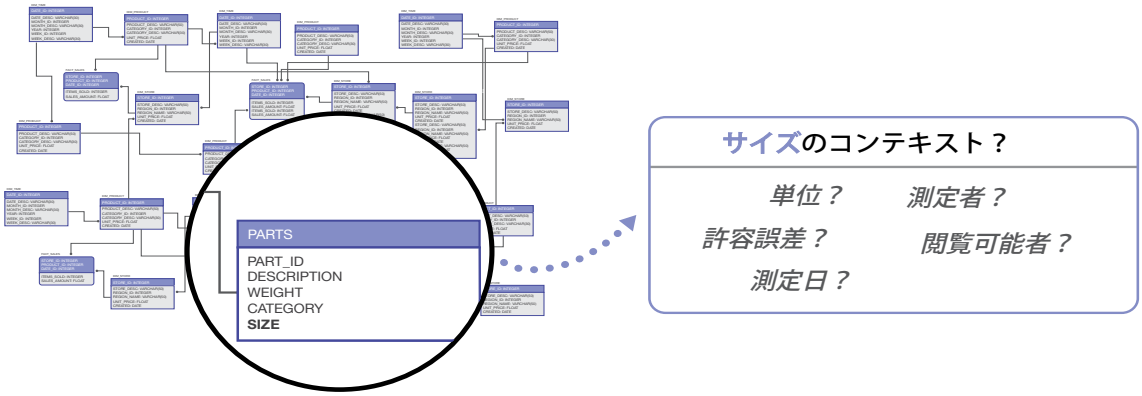

図5：リレーショナルデータベースは、コンテキストを格納できない

わかる人はほとんどいないという場合もあります。こういった問題によって一貫性がなくなり、正規化ならびにクエリの問題が発生します。そして究極的には、データベース設計者はリレーショナル技術の欠点のせいでバラバラになった物事の意味を把握し、論理的な全体として統合（結合）するための方法を別途提供する必要に迫られます。

コンテキスト保持の困難さ

リレーショナルデータベースは、データを数学的に一貫性のある方法で表現します。しかし、データのコンテキスト（データの「セマンティック」）は失われてしまいます。

これはERモデリングに関するピーター・チェンの論文における主要な論点の1つです。彼は「リレーショナルモデルはリレーショナル理論に基づき、データの高い独立性を実現できます。しかし、現実世界に関する重要なセマンティック的情報が失われる可能性があります」と述べています。[8]

例えば、データベース内に部品に関する表があり、その中に「Size」列があったとします。ある行ではこの列の値が「42」となっています。しかし、これに関するコンテキスト情報はどこにあるのでしょうか？「42」の単位は何ですか？　測定時の許容誤差はどのくらいですか？　測定したのは誰ですか？　これは推定値ですか？　測定したのはいつですか？　この列の情報にアクセスできるのは誰ですか？

定性的なデータや、構造があまりないデータ、またクラスやプロパティの関係性を扱わなければならない場合、問題はさらに大きくなります。例えば、表に「Customer」列があるとします。このラベルは何を意味しているのでしょうか？　この「Customer」は、他の重要な情報とどのような関係があるのでしょうか？　この「Customer」グループは、より大きな顧客グループの一部でしょうか？　この「Customer」データを生成しているのは、どのシステムですか？　この「Customer」は、アプリケーションにどのように提供されるでしょうか？　把握している「Customer」は何人いるでしょうか？　ビジネスルールに従っていない「Customer」は誰ですか？

残念ながら、こういったデータのコンテキストはこのデータベース内にはありません。存在したとしても、SharePoint*、Microsoft Excel*スプレッドシートに含まれていたり、あるいは数か月前にプリントアウトしてDBAオフィスの壁に貼ってあるER図内にあるかもしれません。いずれの場合も、データが格納されているデータベース内にはありません。同一データベース内に格納されているデータでさえ、意味のあるものとして活用することは困難です。しかしデータサイロに分断されている場合は、おそらく不可能でしょう。この場合、メタデータやリファレンスデータを入手したり、一緒に扱えるように調整したりすることは困難で、またお金がかかります。

リーダー組織は新しいアプローチを採用

リレーショナルデータベースは確固たる技術であり、データベースとしてこれが最も利用されている理由はたくさんあります。しかし、リレーショナルデータベースをその本来の用途以外にも利用できると考えるのは間違いです。今日の組織は、変化の必要に迫られてアプローチを根本的に検討し直しています。MarkLogicはNoSQLならびにセマンティックにより、従来のデータモデリング手法に取って代わるものとして、数多くの困難な問題を解決します。この項では、MarkLogicに基づく新しいアプローチを採用したことによる成功例を、いくつか簡単にご紹介します。

8　P. Chen, 1976.

「MarkLogicを使ったiPlayerの新規システムでは、初年度に30億件の番組リクエストを処理しました。パフォーマンスが改善され、SQLクエリでは20秒かかったものが、MarkLogicでは20ミリ秒しかかかりません」

BBCのIPLAYERストリーミングサービスの番組メタデータの管理

BBCは、NoSQLならびにセマンティックを基幹アプリケーションに大規模に導入した、最初のグローバル企業です。2012年のロンドンオリンピックの際、BBCは、リレーショナルデータベースからNoSQLとセマンティックを利用する新しいアーキテクチャに移行しました。これは、コンテンツの集約、パブリッシング、再活用を自動化するためです。

以来、BBCはNoSQLならびにセマンティックの利用範囲を拡大しています。一例として、「iPlayer」というテレビストリーミングサービスがあります。こ のサービスは、BBC番組のメタデータを格納・提供する主要コンポーネントとしてMarkLogicを使用して、2014年に再開されました。MySQLとMemcachedに大きく依存していた従来のシステムは パフォーマンスと信頼性に問題があったため、このような方針転換が行われました。またiPlayer用の新規コンテンツ作成プロセス全体も極めて遅く、何日もかかることがよくありました。

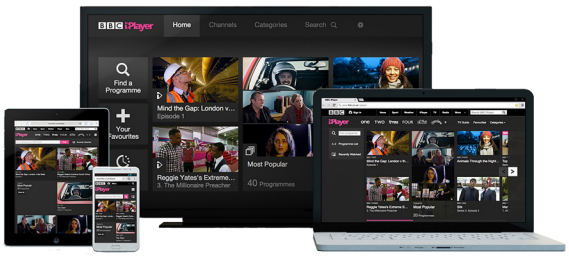

図6：BBCのiPlayerテレビストリーミングサービス

ユーザー増加に対応し、またシンプルかつ信頼性の高いシステムを実現するために、BBCでは新しいやり方が必要となっていました。当初BBCは、この問題をリレーショナル技術で解決しようとし、主要データベースベンダーたちとプロトタイプ作成や検証などを長期間行いましたが、結果的にMarkLogicを選択しました。MarkLogicによるiPlayerの新規システムでは、初年度に30億件の番組リクエストを処理しました。パフォーマンスが改善され、SQLクエリでは20秒だったものが、MarkLogicでは20ミリ秒しかかかりません。またコンテンツ提供にかかる時間は、数日だったものが数分にまで短縮されました。BBCのリードテクニカルアーキテクトは、「MarkLogicは物事を極めてシンプルにしてしまうので、アーキテクトのほうが考え方を大幅に変える必要があります」と言っています。

一流エンターテイメント企業におけるセマンティックメタデータハブ

ある大規模エンターテイメント企業は、MarkLogicを使って数十万という商品のタイトル、キャラクター、配給権、技術情報を管理しています。これらの情報は極めて大切です。以前は、データは情報サイロに分断されており、その結果、低いデータ品質、一貫性のないガバナンス、データの効率的な再利用ができないといった問題が発生していました。このため物理的なアセットを他のオフィスまで届ける配達担当を雇う必要があったほどです。

この問題の解決策が、MarkLogicを使った一元的セマンティックメタデータハブでした。このハブを使うことで、組織内のアセットに関するすべてのメタデータを一か所に格納してアクセスできるようになりました。これにより、競争が激しい業界におけるビジネス上の新しい課題や変化に、素早く対応できるようになりました 。

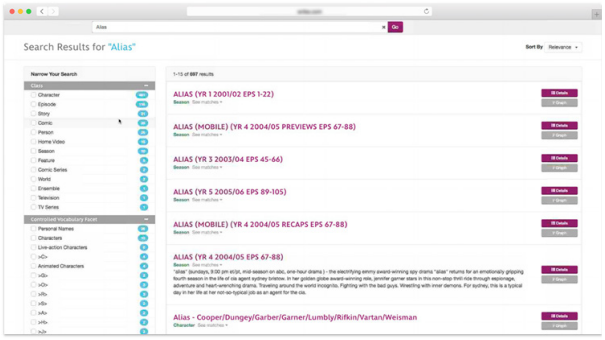

図7：大規模エンターテイメント企業における、セマンティッ クを活用した検索アプリケーション

このハブでは、MarkLogicを使ってデータの格納と検索を行うほか、MarkLogicのパートナーであるSmartlogic*を使って、分類、パブリッシング、タクソノミー/オントロジー管理、セマンティックによるエンリッチメントを行っています。このハブは、複数のサイロからのデータを読み込み、自然言語検索画面からデータのクエリができます。また下流のシステムにデータを提供します。

このデータモデルはNoSQLとセマンティックに基づいているため、データ間の関係性を活用できます。例えば、ある映画に特定の女優が出演しているかどうか、また彼

「他のやり方では、同等の機能を実現するために、異なる3つの技術を統合する必要があるでしょう」

女が特定の役を演じているかどうか、場所、アニメ版かどうかなどを簡単に確認できます。これとは対照的に、リレーショナルデータベースでは、これらの関係性すべてを管理するのは困難で、またコストが莫大になります。MarkLogicに備わっているデータベース、検索、アプリケーションサービスを組み合わせることで、シンプルな情報アーキテクチャを構築できました。他のやり方では、同等の機能を実現するために、異なる3つの技術を統合する必要があるでしょう。

APAにおける学術出版のインテリジェントアナリティクス

APA（アメリカ心理学会）の収益の70%以上は、パブリッシング（出版/情報提供）から得られています。ここでは「情報の第一提供者」であることが、極めて重要です。しかし、データベースが巨大であることが問題となっていました。ここには、記事、書籍、雑誌、論文、ビデオ、テスト、測定データなど5700万件の情報が含まれ、15万人以上のユーザーがアクセスします。APAの旧来のシステムは、OracleとLucene/Solr（リレーショナルデータベースと検索エンジン）に基づくものでしたが、検索結果に一貫性がない、ユーザーエクスペリエンスが良くない、コンテンツ提供が遅いなどの問題がありました。また数百万行もある巨大な表を管理するのも面倒でした。これに加えて、データ自体は極めて構造化されていたのですが、その構造はリレーショナルデータベースに適したものではありませんでした。また、サーバーが大量にあり、大勢の開発者がレガシーシステムの保守に注力しなくてはならなかったため、不要なコストがかかっていました。

図8：セマンティックによるAPAデータの視覚化

APAは、こういったデータを管理するためにMarkLogicに移行しました。MarkLogicでは、データベースに検索機能がビルトインされているため、アーキテクチャがシンプルになります。また次のステップとして、セマンティックを使ったデータのエンリッチメントを行いました。これにより、データの探索と、作者/主題/ジャーナル/スポンサー間の関係性の分析ができるようになりました。

これに加えてMarkLogicを使うことで、APAのシステムで扱えるコンテンツの量が増えました。また、より高い品質、一貫性、アジャイル性、ビジネス拡大に応じた拡張を実現できました。この新しいシステムでは、DBAは保守するインフラが減ったことを喜び、開発者は開発サイクルが速くなったことで新機能に割ける時間が増えたことを喜んでいます。

NOSQLドキュメントデータベースのメリット

前項で紹介した複数の組織がNoSQLとセマンティックを選択した理由の1つとして、データの内容が理解しやすく、柔軟かつ便利なモデルが必要だったということがあります。ここで必要とされるモデルは、データの統合に優れ、よりスマートなアプリケーションを従来よりも短期間で実現するものです。このホワイトペーパーの残りの部分では、このような新しいアプローチの個々のメリットについて説明していきます。

まず最初に、NoSQLドキュメントデータベースのメリットを説明します。NoSQLデータベースには、4つのタイプがあります。ドキュメント、グラフ、カラム、キー/バリューです。NoSQLデータベースのこれらのタイプのうちで、ドキュメントデータベースの人気は他よりも圧倒的に高いです。この項では、ドキュメントデータベースのメリットについてのみ取り上げます。これはMarkLogicではデータをドキュメントとして格納するからです。NoSQLに関する一般的な内容についてもっと知りたい方は、電子書籍『エンタープライズ NoSQL for Dummies』をぜひご覧ください。

ドキュメントモデルの柔軟性

ドキュメントデータベースの主な長所の一つは、柔軟でスキーマに依存しないデータモデルを使用していることです。これにより、データのモデリングとアプリケーション

「JSONならびにXMLドキュメントによって、今日の組織が扱っている多様で複雑なデータをより自然にモデリングできます」

の開発においてアジャイル性と豊かな機能が提供されます。ドキュメントモデルでは、複数の表に含まれるデータを正規化する代わりに、すべてのデータをJSONやXMLドキュメントして保持します。JSONならびにXMLドキュメントによって、今日の組織が扱っている多様で複雑なデータをより自然にモデリングできます。また、データの概念モデル/ビジネスモデル（現実世界のエンティティ）へのマッピングが改善されます。例えば金融取引、患者記録、手術手順などをモデリングする場合、必要な情報すべてを1つのドキュメント内に含めることができます。

```
{
  "hospital": "Johns Hopkins",
  "operationType": "Heart Transplant",
  "surgeon": "Dorothy Oz",
  "operationNumber": 13,
  "drugsAdministered": [
    { "drugName": "Minicillan",
      "drugManufacturer": "Drugs R Us",
      "doseSize": 200, "doseUOM": "mg" },
    { "drugName": "Maxicillan",
      "drugManufacturer": "Canada4Less",
      "doseSize": 400, "doseUOM": "mg" },
    { "drugName": "Minicillan",
      "drugManufacturer": "Drugs USA",
      "doseSize": 150, "doseUOM": "mg" }
  ]
}
```

図9：JSONドキュメントの例。病院における外科手術の手順を表している

リレーショナルデータベースの表内の1行は、1つのドキュメントにほぼ該当します。しかしドキュメントはリレーショナルほど固定的ではなく、また構造化データと非構造化データの両方で使用できます。それぞれのドキュメントごとに、構造のスキーマや属性は異なっています。スキーマは簡単に変更でき、他のドキュメントのことを考慮せずに独立したものとして扱えます。スキーマが異なる新しいデータフィードが突然登場した場合でも、このデータを読み込んで格納できます。この際、構造を事前定義したり、既存のドキュメントを調整する必要はありません。

ドキュメントデータベースには、アプリケーションにおけるデータの利用に関してもメリットがあります。つまり、インピーダンスミスマッチ問題が発生しません。リレーショナルデータベースの場合、インピーダンスミスマッチが、データやアプリケーションプログラミングで使用されているコードを内包するオブジェクトと、データベース内の正規化された行や列の間で発生します。このようなアーキテクチャでは、徐々にパフォーマンスが劣化し、またバグを含むコードが増えていきます。NoSQLの場合、データをJSONやXMLとして扱うことができます。あるいはテクノロジースタックを使えばリッチオブジェクトでさえ扱うことができます。その際に、リレーショナルモデルからの複雑な変換は不要です。これはより単純明快なアプローチで、間違ったアプリケーション開発のリスクを回避できます。というのも反復的な開発アプローチなので、システムは将来的な変更に柔軟に対応できるためです。

より優れたアプローチのインパクト

現在では、サイロに分断されたデータを管理することが多くなっており、データ統合に適したデータベースが求められています。データ統合においては、ドキュメントモデルの柔軟さが大きなアドバンテージになります。データの特定やプロファイリングに無駄な時間をかけたり、複雑なETL処理を管理する代わりに、ドキュメントデータベースでは、データを「アズイズ（そのまま）」読み込むことができます。また必要な変換は、読み込み処理の時点で、あるいはデータベースに読み込んだ後で行うことができます。往々にして、変換はデータを特定のスキーマに合わせるために行われます。あるいは、ドキュメントをエンリッチするために特定のメタデータを追加する必要がある場合もあります。これらの処理では、元のソースデータの完全性を損なわないようにする必要があります。ドキュメントモデルはリレーショナルデータベースに比べて、これらのタスクをより速くかつ単純に行えます。

現在MarkLogicを使っているあるヘルスケアの大企業では、140以上の人事データソースの統合プロジェクトがありました。対象のほとんどが複雑な構造化データで、給与データ、人事評価、昇進、福利厚生データなどがあります。これらのデータを統合・変換した後、下流の50個のシステムにリアルタイムで提供する必要がありました。このプロジェクトは、従来のリレーショナル技術を使った場合、開発に4万時間以上、導入には数年かかると推定されました。MarkLogicを採用したところ、このプロジェクトは1年未満で完了し新しいシステムが稼働し始めましたが、これには会社側も驚いていました。この新規システムでは、複雑なデータ読み込みならびに複雑なデータ出力を処理できるほか、将来におけるデータの変化への対応力も増しました。この会社は今や、人事データすべてを扱うデータレイヤーとしてMarkLogicに依存しています。また、複数のデータサイロが存在した前のシステムに比べて、コストも削減されました。

「ドキュメントデータベースでは、データを『アズイズ（そのまま）』読み込むことができます。また必要な変換は、読み込み処理の時点で、あるいはデータベースに読み込んだ後で行うことができます」

MarkLogicと他のドキュメントデータベースの比較

他の著名なドキュメントデータベースと比較した場合、MarkLogicは以下の点で優れています

- **複数のデータ形式：**MarkLogicでは、JSON、XML、RDF（これは次に説明します）形式でネイティブにデータを格納できます。これに加えてMarkLogicではラージバイナリ（PDF、画像、ビデオなど）も格納できます。ほとんどのドキュメントデータベースでは、JSONしか格納できません。
- **ビルトインの検索機能：**MarkLogicにはユニバーサル（何でもクエリ可能な）インデックスがあります。これは語、フレーズ、構造、値、セキュリティなどに付けられます。またこれに加えてさらに他のインデックスを付けることもできます（レンジ、**地理情報**、トリプルインデックス）。他社のデータベースではインデックス機能は限られています。また全文検索を行う場合は、他のソリューションを統合する必要があります。
- **エンタープライズ機能：**MarkLogicには完全なACIDトランザクション（データの一貫性）が備わっています。ほとんどのNoSQLデータベースには、ACIDトランザクションがありません。つまりデータの消失や破損の可能性があります。これに加えて、MarkLogicには認証済みセキュリティと実績あるHA（高可用性）とDR（災害対策機能）があります。

セマンティックによるさらなるメリット

ドキュメントデータベースは、優れた汎用データベースです。しかしデータモデリングに関して得意でない分野があります。この項では、ファクトと関係性を格納するのにセマンティックが特に適している理由と、データモデリングにおけるこの機能の意味について説明します。セマンティックの概要についてより深く知りたい場合、また実際のセマンティック使用例を知りたい場合、電子書籍『セマンティック for Dummies』をぜひお読みください。

シンプルかつ強力なデータモデル

近年、グラフデータベースの人気が急激に高まっています。またトリプルストア（セマンティックデータを格納するもの）もグラフデータベースの一種として認識されています。データがグラフ構造を取り始め、エンティティ（ヒト/場所/モノ）自体とエンティティ間の関係が重要な場合、セマンティックを使ったほうが良いでしょう。

セマンティックのファクトと関係性は、トリプルと呼ばれます。例えば、トリプルは「John lives in London（ジョンはロンドンに住んでいる）」のようになります。こういったトリプルはSPARQLという標準言語でクエリされます。この言語はかなりSQLに似ています。

トリプルは、ファクトや関係性の両方をシンプルかつ強力に表現します。ある専門家は「モデルが組織全体に関わるものとなり、極めて多様になった場合、SQLでこれらをうまく扱うのはかなり面倒になります。データモデル自体が複雑になり、関係性が増え、多様になった場合、RDF（ならびにSPARQL）の重要性はさらに高まります。これは、情報の多様性が他の要素（量や更新頻度）よりも重要になるためです」と言っています。[9]

セマンティックには、以下のようなユニークな長所があります。

- トリプルは一般に理解されやすく、検索や共有が容易
- トリプルを結び付けてグラフを作成できる。これは機械で読み取ることが可能。また新しいファクトを推論できる

9 Cagle, Kurt. “Why SPARQL Is Poised To Set the World on Fire.” June 4, 2016. <https://www.linkedin.com/pulse/why-sparql-poised-set-world-fire-kurt-cagle>

- RDFトリプルならびにそのクエリ言語SPARQLは、W3Cによって標準化されている
- トリプルストアは、数千億個のファクトや関係性の規模まで拡張できる
- トリプルストアはオントロジーを活用して、データの整理と分類が可能（オントロジーはタクソノミーのようなものだが、より内容が豊かで便利）

より優れたアプローチのインパクト

データの変化を管理

トリプルならびにトリプルストアのユニークな機能は、通常2つの方法で利用されます。いずれの場合もデータのコンテキストを提供することが目的です。1つめのアプローチは、トリプルを使って、組織、専門分野（ドメイン）、あるいは世界全体に関するファクトや関係性を記述することです。その際、記述されるファクトや関係性の個数に制約はありません。これはその後、検索を強化するのに使用することもできます。

ファクトや関係性の定義方法を標準化することは、多様なデータを管理する際に役立ちます。エンティティの定義方法は、各組織ごとにバラバラです（同じものに別の列名が付いている、列名は同じだがそれが意味するものは違うなど）。また自然言語にも多様性があります（英語の「sub」は潜水艦のことかもしれませんし、あるいはSubwayのサンドウィッチのことかもしれません）。

リレーショナルデータベースにおける比較的抽象的で複雑な関係性の定義方法とは違って、トリプルはデータベース内のファクトや関係性を明示的に記述します。これはデータ統合において極めて便利です。というのも結合や複雑な変換をするよりも、トリプルでデータを結び付けるほうが楽だからです。例えばトリプルを使うと、エンティティ「cust123」は、「cus_id_456 」と同じである、あるいは、「cus_id_456」と何らかの関係がある、ということができます。クラスやプロパティの関係性といった関係性も扱えますが、これは従来の物理的なER図 では扱えなかったものです。これらの機能により、セマンティックは検索を拡張する優れたデータモデルを提供します。例えば、「cardiac catheter」（心臓カテーテル）という語で検索する際に、「implantable devices」（埋め込み型機器）に関連する結果を含めるように拡張できます（セマンティックのタクソノミーやオントロジーを利用します）。このような検索拡張の例として、**以下**の囲み記事を参照してください。

セマンティックの実際の使用例

BSI（英国規格協会）は、標準規格の発行ならびに規格関連サービスを全世界で提供しています。BSIは、規格検索用オンラインアプリケーションである「BSOL（British Standards Online）」を開発しました。

残念ながら、従来のキーワード検索ではうまくいきませんでした。例えば、「cardiac catheter」（心臓カテーテル）で検索しても結果はゼロでした。Googleもあまり役に立ちませんでした。というのも「cardiac catheter manufacturing standards」（心臓カテーテル製造規格）で検索しても、このキーワードに基づく一般的なリンクだけが返され、規格自体の情報は得られ**なかったからです**。

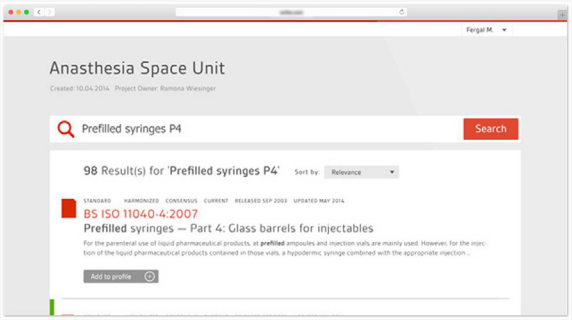

図10：BSIアプリケーションのスクリーンショット

MarkLogicを使うことで、BSIはセマンティックを活用した新しいアプリケーションを構築しました。これにより規格の検索がより質が高く、速くなりました。セマンティックでクエリを拡張することで、「cardiac catheters」で検索しても、それに概念的に関係する結果も返されるようになりました。つまり医療規格に関するドキュメント内で直接「cardiac catheters」に言及していなくても、「implantable devices」（埋め込み型機器）に言及していればこれはセマンティック的に心臓カテーテルに関係しているものだとみなされます。

「リレーショナルデータベースにおける比較的抽象的で複雑な関係性の定義方法とは違って、トリプルはデータベース内のファクトや関係性を明示的に記述します」

メタデータとリファレンスデータの管理

トリプルは、データリネージ（出自、その後の履歴・系譜）、データ保持、時系列、セキュリティ、関連度、その他のさまざまなファクトを説明するためのメタデータ/リファレンスデータとして使用できます。こうすることで、データのコンテキストをどこか他の場所に置いて参照しにいくのではなく、データベース内でデータに関連付けておくことができます。前述のさまざまなサイズの部品に関するデータベースの例で言えば、トリプルを使って、単位（cm）、許容範囲（h17、0-1.20mm）、ジョンが測定したこと、2015年12月1日に測定したこと、製造部門の従業員だけがこの情報にアクセスできることなどを定義できます。

これは、従来のリレーショナルのアプローチ（通常メタデータは存在しないか、あるいは管理が難しい）よりも優れています。またドキュメント指向だけのアプローチよりも優れています。ドキュメントモデルでは、メタデータやリファレンスデータは、JSONやXMLドキュメントとして管理できますが、グラフ構造やマシン読み取り可能なトリプルは活用できません。こういったメリット以外にも、セマンティック推論やクラス/プロパティの複雑な関係の記述ができるので、メタデータの格納に関してはセマンティックが最適なのです。データとメタデータが同一データベース内に一緒に格納されていれば、別々のサイロに含まれている場合よりも、データ統合がよりスムーズになり、エラーが減ります。

NOSQLとセマンティックを合体させたマルチモデルのアプローチ

何でもできるデータモデルというものは存在しません。このため、データや作業の種類に応じて複数のモデルを使用する必要があります。複数のモデルを組み合わせて使う手法は、「ポリグロット・パーシステンス」と呼ばれます。

ポリグロット・パーシステンスを実現するには、異なる種類のデータをさまざまな場所で格納するより、一か所に格納しておくほうが良いです。このアプローチでは、さまざまなテクノロジーをまとめて使用する必要があります。他のデータベースベンダーも同じ結論に至っています。あるアナリストは「マルチモデルアプローチがNoSQLの未来だ」と言っています。[10]

マルチモデルデータベースにより、各モデルのメリットを単独のプラットフォームで実現できるだけでなく、両方のモデルを同一データベース上で利用することでしか得られないユニークなメリットを提供できます。現時点において、MarkLogicだけがエンタープライズ仕様のマルチモデルデータベース（ドキュメントデータベースとセマンティックを統合）を提供しています。このセクションでは、MarkLogicのアプローチのメリットをより詳細に紹介していきます。しかしその前に簡単にまとめておきます。

- ドキュメントデータベース（JSON、XML）
  - 柔軟性とアジャイル性
  - 巨大な規模
- トリプルストア（RDF））
  - エンティティと関係性用に設計
  - コンテキストの提供用に設計
- ドキュメント＆トリプル（JSON、XML、RDF）
  - 上記それぞれのメリットに加えて…
  - 単一のプラットフォーム
  - さらに柔軟なデータモデリング
  - 改善されたクエリ機能

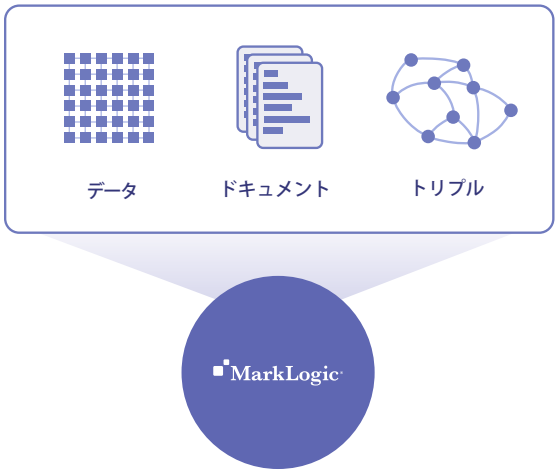

図11：MarkLogicはエンタープライズ仕様のマルチモデルデータベース。データ/ドキュメント/トリプルを1つの統合プラットフォームで格納可能

10 Aslett, Matt. Toward a converged data platform, part one: SQL, NoSQL Databases and data grid/cache. 451 Research. Dec. 3, 2015.

「ポリグロット・パーシステンスを実現するには、異なる種類のデータをさまざまな場所で格納するより、一か所に格納しておくほうが良いです」

統合されたデータベースのシンプルさ

NoSQLドキュメントデータベースとトリプルストアを統合したMarkLogicを使うことで、別々にシステムを準備しなくても両方のモデルのメリットを享受できます。これだけでも十分大きなメリットがあります。つまりこれによって、バックアップ/リカバリー/開発/テスト/検索に関する作業が大幅に削減されます。また管理しなければならないセキュリティモデルは1つだけで、またハードウェアのフットプリントも削減されます。このように、レガシーのデータサイロの管理に大部分の時間とリソースを費やす代わりに、具体的に価値を生み出す活動に注力できます。

さらに柔軟に、さらに強力に

MarkLogicでは、データをドキュメントあるいはトリプルとして格納できます。これはユースケースに応じて選べます。例えば、一般的な使用パターンとしては、前述したようにセマンティックのタクソノミーやオントロジーによってドキュメント検索を拡張する、というものがあります。

使用するデータ型を決める際には、JSONはオブジェクト（顧客、株取引など）に、XMLはテキスト（ブログ投稿、ニュース記事など）、RDFトリプルはファクトや関係性を扱うのに最も適していると考えるといいでしょう。（例：J.J.エイブラムスはスター・ウォーズを監督。原作はジョージ・ルーカス）。下の表1で、各データモデルを詳細に説明しています。このようなオプションがあることで、各作業に適したツールを選択できます。増加する多次元データに対してリレーショナルデータベースを適用することによる面倒な制約に対処する必要はありません。

改善されたクエリ機能

データモデルが優れていれば、データに対してより難しいクエリを実行することができます（通常はコーディングの量も減ります）。MarkLogicは、SQLではできなかった新しいタイプのクエリを実行できます。またSQLよりもシンプルに記述できるクエリもあります。

	JSON	XML	RDF	JSON/XML + RDF
利用法	オブジェクトとして格納されている構造化データに最適	構造化/非構造化データやテキストに最適	ファクトや関係性に最適	データ、テキスト、関係性のシステムに最適
説明	• スキーマ非依存 • JavaScriptでクエリ • コンパクト、パーシングが速い • 値6種：object、array、float、string、boolean、null • 名前空間、コメント、属性なし • web用の一般的なデータ形式	• スキーマ非依存 • XQueryでクエリ • オブジェクト、セット、多くのデータ型（日付、期間、整数など）を格納可能 • 名前空間（埋め込まれたオブジェクトタイプ用）、コメント、属性（メタデータの追加用）を使用 • データモデルとしては、JSONよりも成熟	• エンティティと関係性を定義 • アトミックな構造（これ以上細分化不能） • データとクエリに関して一般的な標準を使用（RDFとSPARQL） • リファレンスデータ、メタデータ、出自用に使用	• ドキュメントにトリプルを含めることができる • トリプルでドキュメントに注釈付け • トリプルのグラフにドキュメントを含めることができる • 強化されたクエリ： - グラフを使ってドキュメント検索を拡張 - ドキュメントをリンクすることでグラフ検索を強化 - トリプルメタデータを使ってドキュメント検索を制約

表1：MarkLogicにおける異なるデータモデルの利用法と説明

「マルチモデルのアプローチでは、各作業に適したツールを選択できます。増加する多次元データに対してリレーショナルデータベースを適用することによる面倒な制約に対処する必要はありません」

MarkLogicのドキュメントモデルには、ドキュメント内の自由文の検索や、重要度（関連性）に基づくデータのランキングといったユニークなクエリ機能があります。MarkLogicのトリプルストアを使うと、関係性のグラフをナビゲートしたり、セマンティック推論ができます。推論では、新しいトリプルをグラフへ追加し、これを既存トリプルの定義や関係性に基づいて活用できます。

MarkLogicではこれら2つのモデルを統合することで、より多くの種類のクエリを活用できます。MarkLogicには、SPARQLクエリと従来のMarkLogicドキュメント検索クエリを混在できるAPIがあります。これにより、グラフによるドキュメント検索の拡張や、ドキュメントとリンクすることによるグラフ検索の強化、トリプルメタデータを使ったドキュメント検索の制約などができます。データへのクエリ方法が多様になることで、そこから得られる答えもより優れたものとなります。

より優れたデータモデルの今後の活用

このホワイトペーパーでご紹介したのは、NoSQLとセマンティックを使うことによるメリットや、より優れたデータモデルによって何が可能なのかのごく一部にすぎません。旧来のデータモデリングからの新しいデータモデリングへの移行は、一見かなり面倒に思えるかもしれません。しかしマルチモデルを導入し、そのメリットを享受しているリーダー企業は、すでに数多く存在しています。この実現に着手できるよう、詳細な情報のためのリンクをいくつかご紹介しておきます。

さらに詳しく

エンタープライズNoSQL for Dummies
http://info.marklogic.com/nosql-for-dummies-jp.html
NoSQLデータベースにもさまざまなものがあります。これらの概要と評価基準についてご紹介しています。

セマンティック for Dummies
http://info.marklogic.com/semantics-dummies-jp.html
セマンティックの世界をより深く理解してください。さまざまなユースケースやセマンティック導入時の考慮事項10個などについてご紹介しています。

カスタマープレゼンテーション
po.st/multimodeldataintegration
MarkLogicユーザーがどのようにデータモデルを分析したのか、どのようにNoSQLとセマンティックを導入したのかについてご紹介しています。

Modern Approach to Data Modeling
po.st/modeling
MarkLogicエンジニアによるプレゼンテーション。マルチモデルデータ統合の現実について詳細にご紹介しています。

MARKLOGIC K.K.
150-0043 東京都渋谷区道玄坂 1-12-1渋谷マークシティウェスト 22 階
+81 3 4360 5354 | jp.marklogic.com | MarkLogic-JP@marklogic.com