

ECAルール・メタプログラミング機構を有するアクティブデータベースシステムの実現

An Active Database System with Meta-Programming Mechanisms for ECA Rules

倉林 修一[△] 清木 康[△]

Shuichi KURABAYASHI Yasushi KIYOKI

本論文では、異種のデータベースがネットワーク上に存在する環境における広域的なデータ処理の自動化機能の提供を目的として、データベース横断的なアクティブルール管理・適用を実現するアクティブデータベース・システムを提案する。本方式は、ルール自身の振る舞いを動的に変更可能なメタプログラミング機構を有するルール言語の導入により、アクティブルールの適用範囲の拡大と応用システム開発の効率の向上を実現する方式である。また、評価実験により提案方式の有効性を示す。

In this paper, we propose a system architecture of active database that provides autonomous and reactive data processing mechanisms over distributed and heterogeneous data sources. The key feature of our approach is to introduce a new active rule language equipped with meta-programming mechanisms that can modify behaviour of active rules dynamically. Application system developers can implement reactive applications rapidly by assembling modular and extensible active rules. We report several experimental results to show the effectiveness of our approach.

1. はじめに

ネットワーク上に存在する情報資源群を対象としたデータの追加や更新などの変化が起きた際に、情報システムが自動的・能動的な反応処理を実行することが出来れば、それらの情報資源の活用範囲を飛躍的に拡大することが可能である。しかしながら、現行のインターネット上の情報資源の多くは、ユーザからの問い合わせ要求に応じて駆動する、受動的なシステム上に蓄積されているため、これらの情報資源群の変化によって発生する新たな情報発信の機会を、ユーザに能動的に提供することが出来ない。既存の情報資源のみならず、将来にわたって増加し続ける情報資源を有効に利用していくためには、サービスが自動的・能動的にユーザの情報獲得を支援する機構の実現が重要である[1]。

アクティブデータベース機構(以下、アクティブDBと呼ぶ)は、事象の発生(Event)、状況判定(Condition)、状況成立時の振る舞い(Action)の三要素を一組にしたECAルールにより、データベースの検索・更新などの操作に反応して、別の検

索・更新のリアクションを自動起動する、イベント駆動型のデータベース・システムである[2][3]。既存のアクティブDB機構は、中央集権型の単一データベース環境での反応処理を目的としており[4]、ECAルール定義に先立ちスキーマの設計を必要とするスキーマ主導のアプローチをとる。そのため、ECAルール中に特定の情報資源のデータ構造に関する知識が明示的に記述(ハード・コード)されるのが前提となっているために、異種のデータベースを横断的に扱うようなECAルールの記述および処理の実現は困難であった。特に、マルチデータベース環境[5]を対象としたアクティブDBの実現において、大局的スキーマ[5]のような粒度の大きいスキーマを用いてルールを記述すれば、ローカルDBの追加やスキーマ構造の変化に対して柔軟に対応することが困難である。一方、マルチデータベース言語システム[5]では、スキーマ操作およびデータ操作を統合的にクエリ内に記述することでデータベース統合を実現するため、ローカルDBに高い自律性を持たせることが出来るが、このような小粒度のスキーマ操作はローカルDBに関する詳細な知識をルール中に記述することになるため、異種のデータベースを横断的に扱うようなルールの記述は困難である。

そこで、本論文では、アクティブルール自身の振る舞いを動的に変更可能なメタプログラミング機構を有するルール言語により、異種データベース群を対象とした横断的なアクティブルール管理・実行を可能とするアクティブDBシステムを提案する。特に、ルール言語のためのメタプログラミング機構として、応用目的に応じてスキーマを動的に再構成するメタオブジェクトモデルと、既存のルールを編集統合するルール・アセンブル機能を示す。この方式は、次の2項目を特徴とした広域アクティブDB機構を実現する。

- 情報資源の自律性：既存の情報資源群を変更することなく広域的なアクティブデータベース・システム環境へ連結可能である。
 - アクティブルールの可搬性・再利用性：既存のルール群をライブラリとして蓄積しておき、それらを編集統合することで新たなルールを定義することが出来る。
- 本方式は、データベース・スキーマを各アプリケーションから共通して利用される共有資源として位置づける従来のアクティブDBシステムと異なり、再利用可能な部分ルールと部分スキーマをひとまとめの要素としてカプセル化し、それらをメタプログラミング機構により連携させるルールを記述することで、ルールの適用可能性を保持したままスキーマの再定義を可能にする方式である。表1に本方式の特徴を示す。

表1 研究の位置付け

Table.1 Differences between our approach and conventional approaches

	パラダイム	ルールとデータの結合	ルールを記述する主体
従来方式	スキーマ主導	密結合	DB管理者がDB管理作業の自動化などのためにルールを記述
本方式	アプリケーション主導	疎結合	開発者やエンドユーザが、自動分類や情報配信などのアプリケーションを実現するために記述

関連研究としては、異種分散データベース環境におけるグローバル・イベント管理システム[6]や、データベース横断的な一貫性制約をチェックする分散メディアータ[3]、P2Pネットワークに接続された異種のデータベース間における協

[△] 学生会員 慶應義塾大学大学院政策・メディア研究科

shuichi.kurabayashi@acm.org

[△] 正会員 慶應義塾大学環境情報学部

kiyoki@sfc.keio.ac.jp

調動作を行うためのECAルール言語[7]などが挙げられる。また、ECAルールを用いてe-learning教材のシーケンシングを行う研究[8]や、既存の能動的な情報資源を統合するECAルールの研究[9]が挙げられる。XMLを対象としたECAルールにより、ドキュメントに能動的な振る舞いを付与する研究[10][11]も行われているが、これらは、単一のXMLリポジトリを対象にしたルール処理を目的としている。

2. 広域アクティブルール言語の設計

本方式は、メディアオブジェクトの特徴をモデル化するスキーマを再利用可能な単位に分割し、そのスキーマを対象とした部分ルールをメタプログラミング機構により連携させるルールを記述することで、ルールの適用可能性を保持したままスキーマの再定義を可能にする方式である。このメタプログラミング機構を有するECAルール言語をaspect-ECAルール言語と呼ぶ。aspect-ECAルール言語は、イベント指定部(E)、条件指定部(C)、アクション指定部(A)に加え、ルールが依拠するデータ構造を記述するメタオブジェクトモデル宣言部の導入により、ルールが個別のデータから独立することになり、ルールの情報資源横断的な適用が可能となる。さらに、メタオブジェクトモデル宣言の統合関数[13]により、副作用を伴わずに複数のメタオブジェクトモデル宣言を統合可能であるため、異なるルールの条件指定部を容易に組み合わせることが可能となる。

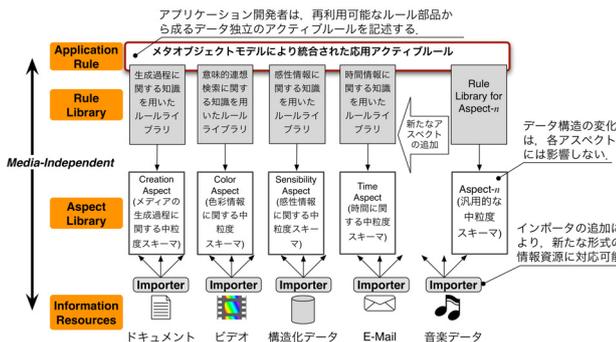


図1 アスペクト指向ルールモデルのコンセプト

Fig.1 A Concept of Aspect-Oriented Rule Model

メタオブジェクトモデルとして用いるアスペクト指向データモデル[12][13]は、全てのデータ・オブジェクトを包括的に記述できるデータ構造を目的としたものではなく、異種のメディアオブジェクトが共通して持つ横断的な特徴(Crosscutting Features)のみを再利用可能な上位スキーマ構造として定義する。この横断的な特徴のための中粒度のスキーマ構造を「アスペクト」と呼ぶ(図1)。アスペクト・クラスは、データ構造、および、属性間の相関量を計量するメソッドを定義する。また、個別のメディアデータのアスペクト情報を、アスペクト・インスタンスと呼ぶ。具体的には、アスペクトは、「メディアデータの生成過程」や「作成者情報」、「日付」のような属性をモデル化するものである。

2.1 アクティブルール言語におけるメタオブジェクトモデル

aspect-ECAルール R は、下記の構文で記述される。

```

R: on <event>,
    where <condition constraints>
    viewpoints of <aspect variables>
    do <action script>
    
```

本方式の特徴は、viewpoints of句におけるメタオブジェク

トモデル宣言にある。viewpoints of句によるスキーマ構造の再構成は、オブジェクト変数宣言句と、アスペクト・アサインメント句から構成される。オブジェクト変数宣言句では、制約が対象とするデータ・オブジェクトの集合を示す変数、オブジェクト変数を宣言する。アスペクト・アサインメント句では、宣言されたオブジェクト変数に対して、制約条件を記述する際に用いるアスペクトを関連づけ、アスペクト変数を宣言する。アスペクト変数は、オブジェクト変数が示すデータ集合における、特定のアスペクトの特徴量の集合を示す。オブジェクト変数 $objvar[0..n]$ を宣言するオブジェクト宣言句は、下記の構文により記述される。

$viewpoints\ of\ objvar_0, \dots, objvar_n;$

また、アスペクト・アサインメント句である $new\dots as$ の列により、オブジェクト変数 $objvar_n$ に対して、 $AspectClass_i$ を $insvar_i$ という名前のアスペクト変数にアサインする。

$new\ AspectClass_i\ as\ objvar_n[insvar_i]$

条件指定句 where では、ルールが発火する条件を、アスペクトに属するメソッドを起動することで、制約を記述する。ここで特筆すべきは、where句で指定される全ての制約は、viewpoints of句で構築されたオブジェクト変数とアスペクト変数のみを用いて記述されるため、制約条件の記述に実データに関する知識が明示的に含まれることが無く、容易にデータ独立性が実現できる点である。

ユーザの好みの楽曲リストと音楽データリポジトリとのマッチングを行うことで、ユーザの嗜好に応じた音楽情報配信を行う aspect-ECAルールの例を図2に示す。この例では、異種の音楽データリポジトリに対する制約を、Artist, Site, Dateの3アスペクトを用いて記述している。

```

define rule MusicRecommendation on IPChanged or FeedUpdated
where myMusic[site].under("file:///home/kurabaya/playlist");
sources[artist].equals(myMusic[artist])
[date].beforeNDays(Date.Today, 7);
viewpoints of sources, myMusic:
new Artist as sources[artist], new Artist as myMusic[artist]
new Site as myMusic[site],
new Date as sources[date];
do RuleChain.connectTo("DetectingSoundEnv", sources);
    
```

図2 aspect-ECAルールの例

Fig.2 An Example of aspect-ECA Rule

2.2 ECAルール・アセンブル方式

ユーザや応用システム開発者がプリミティブなルール統合機能を直接的に用いることは、応用システム開発の生産性の観点から、効率的ではない。そこで、ルールの共有と統合を促進するために、ECAルールの統合プロセスを記述したルール・フレームワークを用いる。ルール・フレームワークとは、条件分岐と繰り返しの制御構造のみを有するシンプルなスクリプト言語であり、パラメータとして与えられた複数のECAルールのメタオブジェクトモデル宣言部を合成し、統合された新たなECAルールを出力するものである。以下に示す4基本ルール・フレームワークを用いて、アプリケーションの目的に応じた応用ルールを生成する(図3)。

- (A) Filtering Framework: 情報配信において、特定条件を満たすデータのみを配信するように、制約条件を記述したルールを連鎖させるフレームワーク。
- (B) New Items Classification Framework: カテゴリの特徴を記述した制約条件ルールを複数用意し、新着情報をそれぞれのカテゴリに自動分類するフレームワーク。
- (C) Old Items Classification Framework: 新着情報をカテゴリ分類基準として、既存の情報をカテゴリに再分類す

るフレームワーク。

(D) Complementing Framework: (C)のOld Items Classification Frameworkで得られたカテゴリ分類を、再帰的に適用することで、カテゴリに分類されるデータを徐々に増加させ、情報補完を行うフレームワーク。

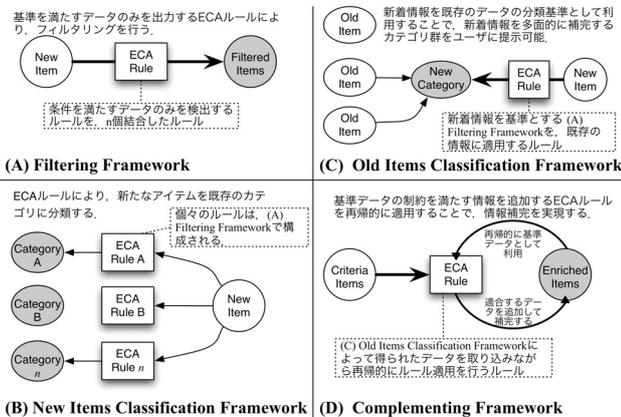


図3 基本ルール・フレームワーク
Fig. 3 Fundamental Rule Framework

2.3 ECA ルールの共有

提案方式におけるECAルール共有の概念を図4に示す。本システムの基本的なコンセプトは、エンドユーザ、高度なルールを記述するスキルを有するエキスパートユーザ、そして、情報資源を抱えるコンテンツ・プロバイダの三者が、ネットワーク上に共有財産として蓄積されたECAルールを利用することで、それぞれの目的に応じた能動的なアプリケーションを容易に実現可能とするというものである。このシステムを提供するサービス管理者に要求されることは、アスペクトの設計・実装のみである。特定のアプリケーションやメディアデータを対象とするための知識は、応用ルール記述のレベルで実現されるように設計することで、ルールをコミュニティの知識として蓄積・共有する。

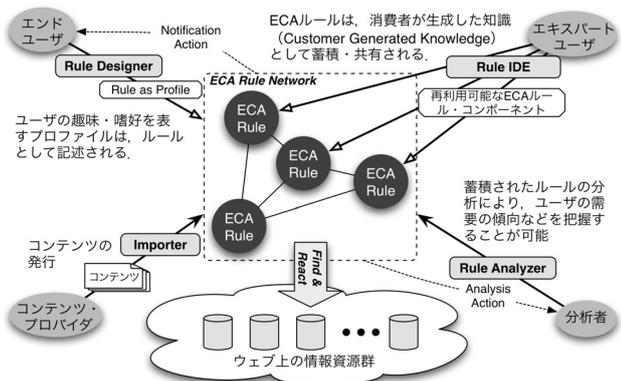


図4 ECAルール共有の概念図
Fig.4 A Basic Concept of ECA Rule Sharing

3. システムの実装

本章では、2章で述べたaspect-ECAルール言語を実装したアクティブDBシステムについて説明する。なお、実装にはJava2SE 5.0を用い、ルールベース部、および、UI部を共にマルチプラットフォーム・アプリケーションとして実装した。図5に実装システムのスクリーンショットを示す。画面左のリストが蓄積されたルール群を示し、画面中央上のリスト群が、それぞれのアスペクトに対して利用可能なルール・ライ

ブラリを示している。また、画面中央下は、既存のデータ集合を対象にルールを適用した場合、制約を満たすデータ集合を示している。また、本システムは、前述のルール・フレームワークを用いて、ユーザが指定した条件に基づきECAルールを半自動生成する機構を有する。具体的には、使用するフレームワークの指定と、フレームワークに対するパラメタとなるルールをライブラリから選択することで、生成されたルールが画面上に展開され、その実行結果が表示される。このように、ユーザはルール・ライブラリの諸機能を選択し、その結果を確認しながらルールを構築可能である。



図5 提案システムのスクリーンショット

Fig. 5 A Screenshot of the Proposed System

本システムは、図6に示すように、中心に位置するルールベース部と、動的なモジュールのロードやネットワーク透過的なリソース・アクセスなどの、ルールベースを実働させるために必要な機能を提供するAppKernelから構成される。アスペクト・クラスやメディアオブジェクトをインポートするためのフィルタの開発者は、ルールベース部が公開するAPIを用いて各モジュールを実装する。

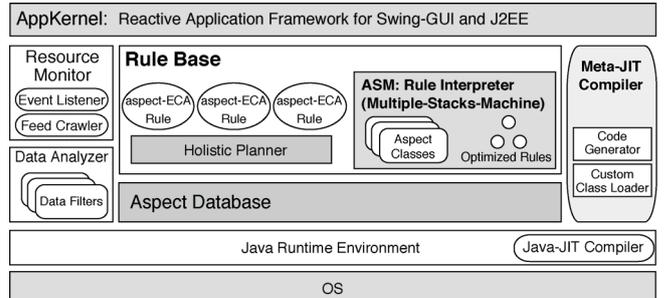


図6 システム実装の概要

Fig. 6 Overview of System Implementation

また、Java言語の実行環境が有するJITコンパイラ等の動的コンパイル技術を用いることで、ECAルールを最適化して実行するMeta-JITコンパイラを実装した(図6右)。ECAルールは、受動的なデータベースにおける問い合わせと異なり、長期間にわたり繰り返し評価される。そのため、ルール評価のコストを低減することは、ルールエンジンのスループット・パフォーマンスを向上させる上で極めて重要である。図6の右部分に示すように、本システムは、定義されたルールをJavaバイトコードへ動的に変換し、システムが実行されているJava仮想マシン上へロードすることで、ルールをJavaアプリケーションとしてハードコードした場合とほとんど同等のパフォーマンスで実行する。

4. 実験

二つの評価実験を行った。実験1は、本システムがルールを評価する際のボトルネックを明らかにすることが目的である。実験2では、ルール最適化機構により、複数のルールが評価された際のスループット・パフォーマンスが向上可能であることを示す。Java2SE5.0で実装した実験システムを、1.25GhzのPowerPCを搭載したマシンを用いて実行した。

実験1では、本システムがルールを評価する際のコスト評価を、1)データ・アクセス時間、2)データ分析時間、3)条件判定時間の三項目において、ルールが発火してから経過時間を計量することで行った。また、データ・アクセス時間を効率化する手法として、転送するデータをgzip圧縮する方法(Opt-1)と、条件判定時間を効率化する手法としてメイン・メモリーに全てのデータを写像後に演算を行う方法(Opt-2)を適用した。評価に用いたデータは、平均ファイルサイズ47Kバイトの33万画素のJPEG画像ファイル300枚である。全画像から生成されたマンセル表色系ヒストグラムを用い、サンプル画像1枚との相関量演算を行うルールを実行した。結果を図7に示す。結果から、本システムのコストは、上記の三項目にほぼ均等に分布しており、特定の箇所にボトルネックが存在するわけではないことがわかる。そのため、個別の効率化手法を適用するのみでは、システムのスループット・パフォーマンスの大幅な向上は望めないことがわかった。

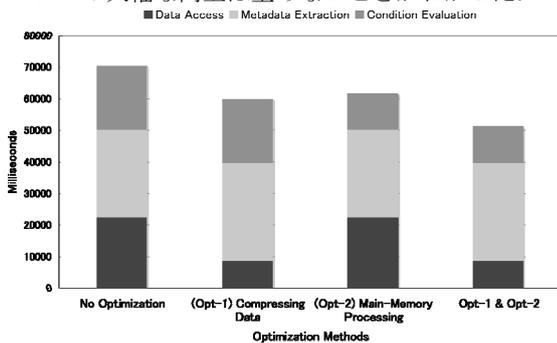


図7 実験1: コスト評価

Fig.7 Experiment-1: Cost Analysis

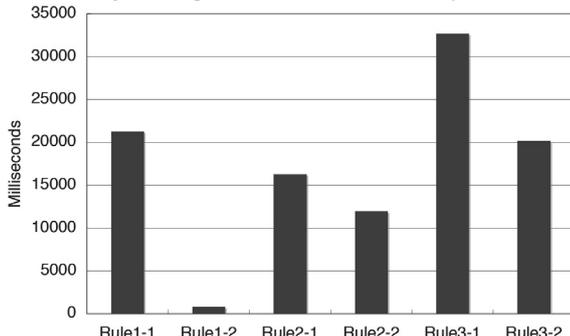


図8 実験2: 複数ルールの評価時間

Fig.8 Experiment-2: Evaluation Time of Rules

実験2では、3章で示したMeta-JITコンパイラにより、複数のルールを評価する際、タスク後半のルールほど高速に処理可能であることを検証する。約3000ページのブログ記事を対象に、時系列演算と文字列照合、および、ページに埋め込まれた画像に対して実験1と同様の相関量演算を行う3種類のルールを設定し、それぞれ異なる2つの制約パラメータを与えて実行した。結果を図8に示す。ルールは、グラフ左のRule1-1からグラフ右のRule3-2に向かって順に評価される。後に評価されるルールほど、Meta-JITコンパイラの効果が現れるた

め、高速化されていることがわかる。

5. おわりに

本論文では、メタプログラミング機構を有するルール言語の導入により、データベース横断的なアクティブルールの管理・適用を実現する広域アクティブDBシステムが実現可能であることを示した。また、評価実験により、本システムが複数のルールを効率的に処理可能であることを示した。今後は、P2Pネットワークのような動的な環境において、ルール・ライブラリを共有する手法の開発、および、ルール処理を複数のマシンで並列的に実行する方式の実現を予定している。

【文献】

- [1] S.Abiteboul, "Issues in Monitoring Web Data", *Database and Expert Systems Applications: In Proceedings of DEXA'02*, LNCS 2453, 2002, pp.1-8.
- [2] J.Widom and S.Ceri. "Active Database Systems: Triggers and Rules for Advanced Database Processing", *Morgan Kaufmann*, 1996.
- [3] N.W.Paton, "Active Rules in Database Systems", *Springer-Verlag*, 1999.
- [4] A.Koschel and P.C.Lockemann. "Distributed Events in Active Database Systems: Letting the Genie out of the Bottle", *Data and Knowledge Engineering*, 1-2(25), 1998, pp.11-28.
- [5] M.W.Bright, A.R.Hurson, and S.H.Pakzad, "A Taxonomy and Current Issues in Multidatabase Systems", *Computer*, 1992, Vol. 25, pp.50-60.
- [6] S.Chakravarthy, "Sentinel: an object-oriented DBMS with event-based rules", *In Proceedings of SIGMOD'97*, 1997, pp.572-575.
- [7] V.Kantere and A.Tsois. "Using ECA rules to implement mobile query agents for fast-evolving pure P2P database systems". *In Proceedings of MEM '05*, 2005, pp.164-172.
- [8] 延原, 庄司, 劉, 横田, "ECAルールを活用したeラーニングシステムにおけるシーケンシング制御の改善", *日本データベース学会 Letters Vol.4, No.2*, 2005, pp.81-84.
- [9] 梶野, 北川, 石川, "配信型情報源に対する統合要求記述とECAルール生成", *電子情報通信学会論文誌D Vol.J85-D1 No.1*, 2002 pp.40-52.
- [10] A.Bonifati, S.Ceri, and S.Paraboschi. "Active rules for XML: A new paradigm for E-services". *The VLDB Journal* 10, 2001, pp.39-47.
- [11] S.Abiteboul, O.Benjelloun, and T.Milo. "Positive active XML". *In Proceedings of PODS'04*, 2004, pp.35-45.
- [12] S.Kurabayashi and Y.Kiyoki. "Aspect-ARM: An Aspect-Oriented Active Rule System for Heterogeneous Multimedia Information", *In Proceedings of WISE'04*, Springer, 2004, pp.659-667.
- [13] S.Kurabayashi and Y.Kiyoki. "An Adaptive Active Rule System for Automatic Service Discovery and Cooperation", *Proceedings of the 24th IASTED DBA2006*.

倉林 修一 Shuichi KURABAYASHI

慶應義塾大学大学院政策・メディア研究科博士課程在学中。2002 慶應義塾大学大学院政策・メディア研究科修士課程修了。アクティブ・データベースシステムの研究・開発に従事。ACM, 情報処理学会, 日本データベース学会, 各学生会員。

清木 康 Yasushi KIYOKI

慶應義塾大学環境情報学部教授。1983 慶應義塾大学工学研究科博士課程修了, 工学博士。1984~1996 筑波大学電子・情報工学系講師, 助教授を経て, 1996 慶應義塾大学環境情報学部助教授, 1998 同学部教授。データベースシステム, 意味的連想検索, マルチメディアシステムの研究に従事。ACM, IEEE-CS, 電子情報通信学会, 情報処理学会各会員。