

アクセス履歴と利用者の位置に基づくコンテンツの再構成

清光 英成[†] 竹内 淳記[‡] 田中 克己[‡]

[†] 神戸大学大学院経済学研究科

kiyomitu@econ.kobe-u.ac.jp

[‡] 神戸大学大学院自然科学研究科

{atsunori, tanaka}@db.cs.kobe-u.ac.jp

急速に普及する携帯電話からのインターネットアクセスサービスにより、利用者の位置情報を考慮することが不可欠となってきた。本論文では利用者のアクセス履歴とアクセスする場所を考慮した Web コンテンツの動的な再構成を議論する。そのため、位置情報を考慮した利用者のページの閲覧履歴やアクセスしている位置情報に基づいてページの閲覧を制御したり、ページコンテンツを動的に変化させることを目的として、コンテンツの動的再構成ルールとページ活性化式を拡張して統合している。更にアクティブデータベースの ECA ルールを参考にしながら議論し、時間的な概念の導入も検討する。

Declarative Reconfiguration Rules for An Active Web based on User's Access History and Location

Hidenari KIYOMITSU[†] Atsunori TAKEUCHI[‡] Katsumi TANAKA[‡]

[†] Graduate School of Economics, Kobe University

[‡] Graduate School of Science and Technology, Kobe University

In this paper, we propose an idea for reconfiguring web pages based on user's histories and their locations. The major objective of the proposed mechanism is the reflection of a web page author's intention about his web data. A reconfiguration rule for providing an appropriate web content including hyperlinks is described in a declarative language, which the author can define easily.

1 はじめに

HTML 言語で Web ページを作成するためのソフトウェアが数多く販売され、ページ作者は JavaScript や DynamicHTML の自動生成機能を用いて変化のあるページを特別な知識なしで作成できるようになってきた。商用のページでは、CGI や ASP* を用いて、ページ作者の意図をページコンテンツに反映し、インタラクティブなページを提供している。しかしながら、CGI や ASP を用いたページの作成にはプログラミングの知識を必要とし、初心者がこれらを導入してページを作成することは困難な場合が多い。

著者らはこのような問題に対して、利用者のリンク巡行履歴に基づいて巡行可能になるリンクを動的に制御するリンク活性化式を導入して、ページ作者の意図をリンク機構に反映する枠組を提案した [1, 2]。また、利用者のページ閲覧履歴に基づいてページコンテンツを動的に変化させるためにコンテンツの動的再構成ルールを宣言的に記述してプログラムを自動生成するアイデアを提案してきた [3]。

利用者が必要とする情報は利用者の地理的な位置によって変化する特性がある。例えば、目的地へのアクセスマップを地理的に近くから見ている利用者と、遠くから見ている利用者とは適当なアクセスマップは異なる。近くから見ている利用者には徒歩でのアクセスマップを表示し、遠くから見ている利用者には最寄り駅までの交通案内を表示したい。しかしながら、従来の Web システムでは利用者が何処からアクセスしようとも同様な情報を表示する静的な側面をもつ。

本論文は、利用者のページ閲覧履歴に基づくアクセス管理に加えて、利用者がアクセスしている位置情報に基づいてページへのアクセスを制御したり、ページコンテンツを動的に変化させることを目的とする。そのため、利用者が実際に存在する位置と、ある場所に利用者がいると仮定した仮想的な位置を記述できるようにページ活性化式とコンテンツの動的再構成ルールを拡張する。さらに、これらをアクティブデータベースシステムで用いられるアクティブルールに準えて議論する。

* Microsoft Active Server Page

2 ページ閲覧履歴に基づくアクセス管理

ページ活性化式は、利用者の閲覧済ページ集合・未閲覧ページ集合と巡覧が可能になるページとの依存関係を記号 \Rightarrow を用いて記述したものである。 $P_{pos} = \{p_{pos_1}, p_{pos_2}, \dots, p_{pos_n}\}$ を閲覧済ページ集合、 $P_{neg} = \{p_{neg_1}, p_{neg_2}, \dots, p_{neg_m}\}$ を未巡行ページ集合、 p を閲覧が可能になるページとする。ページ活性化式 r の構文を、

$$r : P_{pos}, P_{neg} \Rightarrow p$$

$$\text{ただし, } \left\{ \begin{array}{l} P_{pos} \subseteq \mathcal{P}, P_{neg} \subseteq \mathcal{P}, p \notin P_{pos}, \\ P_{pos} \cap P_{neg} = \phi, p \in \mathcal{P}(a) \end{array} \right.$$

とする。ここで、

- WWW 全体に作成されている全てのページの集合を \mathcal{P}
- ページ作者 a の制作した全てのページ集合を $\mathcal{P}(a)$

としている。ページ活性化式 $r : P_{pos}, P_{neg} \Rightarrow p$ の意味は、ページ p を閲覧できるような利用者 u の過去の可能な閲覧済ページ集合 $P(p \notin P)$ 全てからなる集合で規定される。すなわち、

$$\{P \mid p \notin P, P_{pos} \subseteq P \text{ かつ } P_{neg} \cap P = \phi\}$$

であり、ページ p を閲覧できるような全ての可能な閲覧済ページ集合を $SAT(r)$ と表わす。つまり、

1. P_{pos} 中の全てのページを利用者 u がこの時点で閲覧している、
2. P_{neg} 中の全てのページを利用者 u がこの時点で閲覧していない

ならば、ページ p を利用者 u が閲覧できることを表す。

ページ活性化式の構文のただし書きは、 P_{pos}, P_{neg}, p の範囲を以下のように規定している。

- $P_{pos} \subseteq \mathcal{P}$:
ページ活性化式中の閲覧済ページ集合 P_{pos} は全て WWW 上に存在しなければならない。し

かし P_{pos} は、ページ活性化式を定義するページ作者 a の設定した全てのページ集合 $\mathcal{L}(a)$ に限定されない。

- $P_{neg} \subseteq \mathcal{P}$:
ページ活性化式中の未閲覧ページ集合 P_{neg} は全て WWW 上に存在しなければならない。しかし、ページ活性化式を定義するページ作者 a の設定した全てのページ集合 $\mathcal{P}(a)$ に含まれていなくても構わない。
- $p \notin P_{pos}$
ページ活性化式中の閲覧済みページ集合 P_{pos} が p を含んでいてはならない。ページ p を閲覧するためには過去にページ p を巡回していなければならないことになり、このようなページ閲覧はあり得ないからである。
- $P_{pos} \cap P_{neg} = \phi$:
ページ活性化式中の閲覧済みページ集合 P_{pos} と未閲覧ページ集合 P_{neg} の共通部分を存在させてはならない。例えば、ページ $p_i (p_i \in \mathcal{P})$ を閲覧し、かつページ p_i を閲覧していないような利用者のページ閲覧はあり得ないからである。
- $p \in \mathcal{P}(a)$: ページ活性化式が閲覧可能にするページ p はページ活性化式を定義するページ作者 a が作成した全てのページの集合 $\mathcal{P}(a)$ に含まれていなければならない。ページ活性化式はページ閲覧の権利を利用者に与えるかどうかを決定するための記述であるため、ページ作者 a がページを作成できる範囲に限定している。

ページ活性化式はページ作者が定義するが、ページ活性化式の p として記述されなかったページは無条件に閲覧可能である。それはページ活性化式を

$$\{\}, \{\} \Rightarrow p$$

と定義したのと等価であるため、本論文では記述しないことにする。

複数のページ活性化式が同一のページ p を閲覧可能にするように記述された場合、ページ活性化式 r_1, r_2 を

$$r_1 : P_{pos1}, P_{neg1} \Rightarrow p$$

$$r_2 : P_{pos2}, P_{neg2} \Rightarrow p$$

とすると、利用者 u のある時点までの閲覧済みページ集合 $P_{browse}(u)$ が

$$P_{pos1} \subseteq P_{navi}(u), P_{neg1} \cap P_{navi}(u) = \phi$$

または、

$$P_{pos2} \subseteq P_{navi}(u), P_{neg2} \cap P_{navi}(u) = \phi$$

であれば、ページ p が閲覧可能になる。

3 コンテンツの動的再構成ルール

ページ作者が利用者のページ閲覧履歴やサーバが全利用者に提供したコンテンツの履歴に基づいて、ページの内容を変化させるルール定義するために、コンテンツの動的再構成ルールを導入する。本論文では、利用者個別の閲覧履歴を利用者履歴、利用者全体にサーバが提供したコンテンツの履歴をサーバ履歴と呼ぶ。

3.1 利用者履歴

利用者のページ閲覧履歴に依存したコンテンツ(履歴依存コンテンツ)を生成するためのコンテンツの動的再構成ルールは、ページ作者が各利用者に個別の履歴情報を用いて、ページに付加するコンテンツを変化させるために記述する。そのため、属性として

- *times* : ページのアクセス回数
利用者がページに訪れた回数。
- *valid.time* : 同一 URL のページの最終閲覧時刻と比較した有効閲覧期間
最近のアクセスから 3 日以内や 1 週間以上といったコンテンツを表示するのに有効な期間。
- *old.info* : 同一 URL の最後の閲覧のときに付加したコンテンツの ID
元々のページには存在していないコンテンツ、つまり履歴依存コンテンツにページ作者があらかじめ付けた固有の ID。

- *referer* : アクセスした参照元
ページ利用者がどのページからリンクを辿って来たか。

を用意し、表示すべきコンテンツが決定されるような依存関係を記述する。例えば、ページ *index.html* にアクセスした時、過去に *index.html* を *referer.html* からのリンクを巡行して 3 回以上閲覧したことは、

index.html : (*times* > 3, *referer* = *referer.html*)

と記述する。

3.2 サーバ履歴の定義

ページに付加したコンテンツを監視することで、ページ作者は利用者全体に対して今まで付加した履歴依存コンテンツとその回数を把握することができる。ここでは、利用者全体に対して表示したコンテンツの履歴であるサーバ履歴に基づいて動的再構成ルールを定義する。サーバ履歴に対する動的再構成ルールの属性は以下の三つを用意した。

- *content* : 表示した履歴依存コンテンツ
- *combination* : 表示した履歴依存コンテンツの組合せ
- *times* : 履歴依存コンテンツを表示した回数

例えば、コンテンツ ID が *biwa* であるような履歴依存コンテンツをページ *index.html* 中に 10 回表示したことは

index.html : (*biwa*, 10)

と記述する。コンテンツ ID が *ocha* と *biwa* であるような履歴依存コンテンツ両方をページ *index.html* 中に表示したような利用者のアクセスが過去に 5 回あったことは

index.html : ((*ocha*, *biwa*), 5)

と記述する。

3.3 動的再構成ルールの記法

利用者履歴とサーバ履歴を用いてページの見せ方を変えるために、コンテンツの動的再構成ルールの

記法を例を用いて示す。利用者がページ *index.html* にアクセスした時、過去にコンテンツ ID が *ocha* であるような履歴依存コンテンツをページ *index.html* 中に 10 回表示していたら、スタイルシート *front.css* を *index.html* に適用する場合、

index.html : (), (*ocha*, 10) ⇒ *add(front.css)*

と記述する。利用者がページ *index.html* にアクセスした時、過去にコンテンツ ID が *ocha* であるような履歴依存コンテンツをページ *index.html* 中に 5 回表示していたら、履歴依存コンテンツ *ocha* の代わりにコンテンツ ID が *biwa* であるような履歴依存コンテンツを *index.html* 埋め込む場合、

index.html : (), (*ocha*, 5) ⇒ *replace(ocha, biwa)*

と記述する。ここで、履歴依存コンテンツの動的生成ルールを宣言的に記述するための言語仕様を以下に示す。

```

rule ::= URL: (user's history)
      (server history)
      ⇒ function
URL ::= string
user's history ::= [times] [valid_time]
                [old_info] [referer]
server history ::= content ID, times
                | combination, times
function ::= func_name(arguments)
times ::= integer
valid_time ::= span
old_info ::= content ID
referer ::= URL
combination ::= (content ID, content ID)
              | (content ID, combination)
content ID ::= string
span ::= integer time_unit
time_unit ::= minutes | seconds
           | hours | days
           | weeks | months
func_name ::= add | replace | hide
arguments ::= argument
           | argument, arguments

```

試作システムではページ作者の記述したルールは、ルールプロセッサで CGI プログラムに変換される。利用者個別のページアクセスは Cookie として非永続的に記憶されるとともに、利用者に提供したコンテンツ情報をサーバ履歴として蓄積している。試作システムでは *function* にコンテンツの追加・隠蔽・置換といった基本的な関数を用意している。そのため、基本的な関数のみを利用してコンテンツを再構成する場合、ページ作者はプログラミングの知識は必要なく、コンテンツの動的再構成ルールを宣言的に定義するだけでよい。

4 位置情報に基づくアクセス管理

利用者が必要とする情報は利用者の位置によって変化する性質をもつ。例えば、お茶の水女子大学のアクセスマップを茗荷谷駅から見ている利用者と、神戸大学から見ている利用者とは適当なアクセスマップは異なる。茗荷谷駅から見ている利用者には、茗荷谷駅から徒歩でお茶の水女子大学に至るアクセスマップを表示し、神戸大学から見ている利用者にはお茶の水女子大学の最寄り駅までの交通案内を表示したい。そこで本論文では、利用者のページ閲覧履歴とページ巡行履歴に加えて利用者の位置情報を利用したアクセス管理を議論する。

4.1 利用者の位置情報

利用者がアクセスしている位置や過去にアクセスした位置によってページを巡行可能にしたり、表示するコンテンツを変えるために、位置情報をイベントや条件として記述できるようにする。利用者の位置情報には、実際に利用者が存在する実位置と、利用者がある位置に存在すると仮定した仮想位置を考える。

- 実位置
利用者が実際に存在する位置
- 仮想位置
利用者がある位置に存在すると仮定する仮想的な位置

仮想位置は利用者の住所や勤務先が考えられるが、スケジュールを先取りした仮想位置の設定も考慮する。例えば、神戸からお茶の水に向かう時、新大阪から新幹線に乗って移動中の実位置よりも東京という仮想位置の方が重要である。

4.2 位置情報に基づくコンテンツの動的再構成

コンテンツの動的再構成ルールに利用者の位置情報を利用者履歴の要素として記述する。ある利用者が神戸大学からページ *index.html* を 3 回以上閲覧したことは

(index.html, times > 3 : 神戸大学)

と書くことにする。また、利用者の位置情報とページアクセスの組をイベントとして記述する。神戸大学からページ *index.html* にアクセスした時、過去に阪急六甲駅から 3 回以上閲覧していたことは

index.html, 神戸大学 : (times > 3, 阪急六甲駅)

と書くことにする。

ここでは、利用者の位置情報を条件としてだけでなくイベント要素としても扱っている。例として、ある利用者が神戸大学からページ *index.html* にアクセスした時、過去にページ *index.html* を 3 回以上閲覧していれば、スタイルを変化させる場合を考える。

index.html, 神戸大学 : (times = 3), ()
⇒ *add(front.css)*

4.3 位置情報に基づくページ活性化

利用者がページ p_i を位置 A から閲覧したことを $p_i : A$ と書くことにする。 $P_{pos} = \{p_{pos1} : l_{pos1}, p_{pos} : l_{pos2}, \dots, p_{posn} : l_{posn}\}$ を閲覧済ページ集合、 $P_{neg} = \{p_{neg1} : l_{neg1}, p_{neg2} : l_{neg2}, \dots, p_{negm} : l_{negm}\}$ を未閲覧ページ集合として位置情報の記述を加えたページ活性化式 r' を

$$r' : P_{pos}, P_{neg} \Rightarrow p : l$$

と書くことにする。例えば、

$$\{p_1 : A\}, \{p_2 : B\} \Rightarrow p_3 : C$$

と書くと、利用者が位置 A からページ p_1 を過去に閲覧したことがあり、位置 B からページ p_2 を閲覧したことがなければ、ページ p_3 が位置 C から閲覧可能になることを意味する。

ここで、コンテンツの動的活性化ルールとページ活性化式を統合して、アクティブデータベースの ECA ルールを参考に

イベント：条件 \Rightarrow アクション

のような記述にする。

```
read(p), location :  
    user_history, server_history  
     $\Rightarrow$  activate(p), function
```

本論文では Web 等のハイパーテキストシステムを想定しているので、イベントはページへのアクセス要求だけに限定して

$p, location : P_{pos}, P_{neg}, S_{his} \Rightarrow p, function$

と記述することにする。 $p, location$: は ページ p に位置 $location$ から利用者がアクセスするというイベントであり、 P_{pos}, P_{neg} は利用者履歴、 S_{his} はサーバ履歴である。記号 \Rightarrow の右辺 $p, function$ はページ p に $function$ を適用してアクセス権を発行するという意味である。

ページ index.html にアクセスマップというアンカー文字列で最寄りの駅までの交通案内を表現するページ map.html へのリンクを設定しているとしよう。ページ index.html の作者が、茗荷谷駅からアクセスされた時にリンク先を near-map.html に変更したいなら、

```
index.html, 茗荷谷駅:  
     $\Rightarrow$  replace(map.html, near-map.html)
```

と記述すると、利用者が茗荷谷駅からアクセスした時だけ、アクセスマップというアンカー文字列をクリックすることで near-map.html へのリンクを巡回することができる。ここでは、利用者履歴とサーバ履歴を指定していないが、何も書かなければ無条件にアクションが起動されることを示している。

茗荷谷駅からページ near.html にアクセスする利用者が、茗荷谷駅からページ index.html を過去に

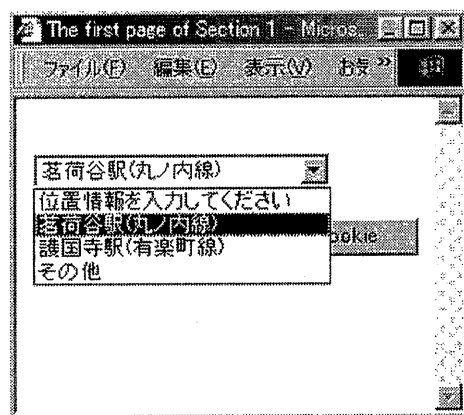


図 1: 利用者の位置情報の入力

閲覧していれば、ページ near-map.html の閲覧できるようにしたいなら、

```
near-map.html, 茗荷谷駅:  
    {index.html:茗荷谷駅}, {}  
     $\Rightarrow$  near-map.html
```

と記述すればよい。

現行のブラウザでは、利用者の位置情報をサーバに送ることができないので、利用者に実位置か仮想位置の位置情報を入力してもらう必要がある。図 1 は、試作システムでの位置入力のイメージを示している。これにより、大学ホームページからアクセスマップへのページが図 2 の最寄りの駅までの交通案内から図 3 へのページに書き換えられて利用者に提供される。

ここで、コンテンツの動的再構成ルールとページ活性化式を用いたアクティブ Web システムとアクティブデータベースシステム [4] を比較する。アクティブデータベースのアクティブルールはイベント・条件・アクションの組である。我々のアクティブ Web システムもアクティブルールに準じた分類で表 1 に整理した。アクティブルールではイベントに構造操作・トランザクション・例外等の組合せを記述できる。例えば、テーブルにタプルが挿入されるトランザクションが中断した事象を条件部分の評価を開始する引金 (trigger) として記述する。一方、我々のアクティブ Web システムは、利用者の

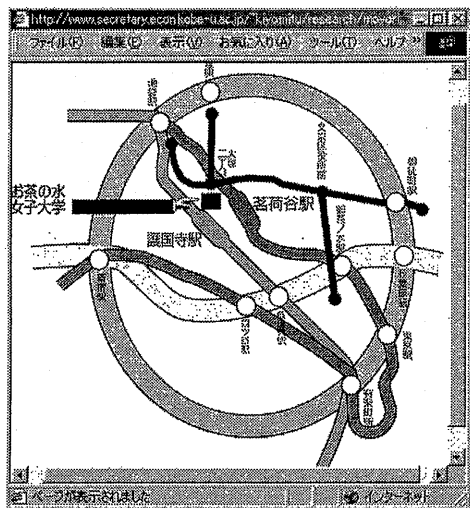


図 2: 最寄り駅までの交通案内



図 3: 最寄り駅からの地図

	Active DB	Active Web
Event	構造操作 トランザクション 例外	ページアクセス 利用者の位置 時刻
Conditon	データベースの 状態	利用者履歴 サーバ履歴
Action	操作	コンテンツの 再構成 ページ活性化

表 1: アクティブデータベースとの比較

位置とアクセスされるページの組を記述し、指定されたページにアクセスされた時に条件部分の評価してアクションを適用する。条件部分は、アクティブルールでは指定したトランザクションが開始されたかや、イベントが評価されているか等の状態を記述する。アクション部分はともに操作を記述している。特に、関数 *replace()* はアクティブデータベースの *do instead* とよく似た機能である [5]。

5 まとめ

急速に普及している携帯電話のインターネット接続サービスを利用する場合、利用者の実位置や目的地等の仮想位置の利用が有効である。そのため、本論文ではコンテンツの動的再構成ルールとページ活性化式を位置情報を記述できるように拡張して統合している。クライアントが移動しない場合は DNS 等によって推定される位置を利用者の実位置と捉えることもできるが、ダイヤルアップ接続の利用者が遠隔地のアクセスポイントに長距離電話で接続することも考えられるので、寺動的に位置情報を獲得する仕掛けが必要である。携帯電話のような通常移動することが想定される接続方法の場合は、移動速度や方向を実位置データとして取得できる仕組みも必要である。携帯電話ではアクセスポイントから位置情報を得ることができるが、GPS サービスを利用して携帯情報端末の位置情報を得たり、ナビゲーションシステムから目的地や経由地などの仮想位置

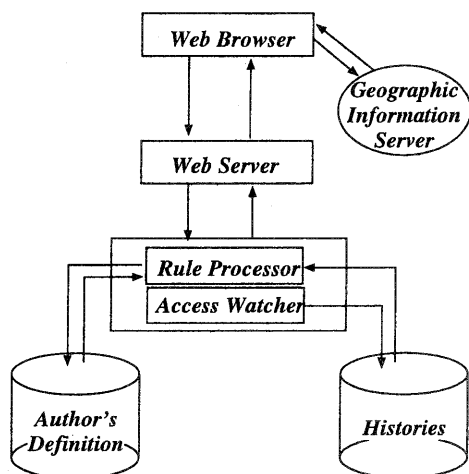


図 4: 位置情報利用のためのシステム概要

を得ることも可能になると考えられる。図 4 に位置情報を利用するためのシステム概要を示した。試作システムでは利用者から位置情報を入力してもらい Cookie として情報を得たが、図 4 の楕円で囲んだ地理情報サーバから利用者の実位置を獲得するようなシステムを想定している。これは、サーバがクライアントに対して Cookie データを発行するだけでなく、クライアントからの地理情報システムへの問合せ操作を行うプラグインを必要とするため、一般的な Cookie と区別するために Geo-Cookie と呼ぶことにする。また、位置情報のような空間的な概念だけでなく、時間の概念をイベント・条件・アクションに導入する。それは、交通機関を利用して目的地に向かって移動している時の実時刻や実位置よりも、目的地や経由地の仮想位置と仮想時刻の方が利用者にとって重要であることが多いからである。例えば、終電の時間より前に経由地に到着するか、後に到着するかで利用者の必要とする情報は変化し、それに従って提供すべきコンテンツも更新されることが期待されている。本研究は、Web のアクセス管理を時間と空間に拡張してアクセス管理を行い、スケジュールの先取り等のアプリケーションと Geo-Cookie を連携させたシステムを今後の課題として検討する。

<謝辞>

本研究をすすめるにあたり、多くの御意見と助言を頂いた本学工学部ならびに自然科学研究科中研究室の皆様へ深く感謝致します。また、本研究の一部は、日本学術振興会未来開拓学術推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(プロジェクト番号 JSPS-RFTF97P00501) による。ここに記して謝意を表します。

参考文献

- [1] 清光英成, 田中克己, "Web ページの巡回に基づく動的なリンク活性化とアクセス管理", 情報処理学会アドバンスデータベースシステムシンポジウム論文集, Vol. 99, No. 19, pp. 115-122, 1999.
- [2] Hidenari Kiyomitsu and Katsumi Tanaka, "Navigation-Dependent Web-Views: Defining and Controlling Semantic Units of Web Pages", *Internet Applications*, IEEE LNCS-1749, pp. 421-426, Springer, 1999
- [3] 清光英成, 竹内淳記, 田中克己, 閲覧履歴に依存した Web コンテンツの動的生成方式, 第 11 回データ工学ワークショップ論文集, 2000 年 3 月 2 日.
- [4] Norman W. Patom, "Active Database System", *ACM Computing Surveys*, Vol. 31, No. 1, pp. 63-103, 1999.
- [5] Klaus R. Dittrich, Stella Gatzju, Andreas Geppert, "The Active Database Management System Manifesto: A Rulebase of ADBMS Feature", *Rules in Database System*, IEEE LNCS-985, pp. 1-17, Springer, 1995.