2005－DBS－137（Ⅱ）（67）
2005／7／14

創発的 XML の提案

石川　博[†‡]　片山　薫[†‡]　高橋俊介[‡]　長屋未来[‡]　布施貴義[‡]　横山昌平[‡]

首都大学東京システムデザイン学部[†]
東京都立大学工学部・工学研究科[‡]

概要 インターネットやユビキタスネットワーク上で XML データを応用する研究がおこなわれている．推薦のような，さまざまな分野で広く使われるような応用は，できるだけロジックは少ない量で，かわりにデータだけで記述できることが望ましい．そこでコンテクスト，エンバイロメント，入力，出力，フィルタ，活性化と安定化という概念を導入して，コンテクスト内の XML データがその識別子に基づいて自己組織化することで応用が実現できるようなデータ集約型応用のための枠組み（創発的 XML）を提案する．

On Emergent XML

Hiroshi ISHIKAWA[†‡] Kaoru KATAYAMA[†‡] Shunsuke TAKAHASHI[‡]
Hideki NAGAYA[‡] Takayoshi FUSE[‡] Shohei YOKOYAMA[‡]

Tokyo Metropolitan University, System Design[†]
Tokyo Metropolitan University, Engineering[‡]

Abstract Applications emergent in the Internet and ubiquitous networks such as recommendation use XML data intensively. It is desirable to describe such applications in less logic. We introduce basic notions such as contexts, environments, input, output, filters, activation, and order. We propose a framework for describing XML data-intensive applications based on the notions. XML data in the context are self-organized by matching their identifiers.

1．はじめに

1．1　背景

インターネットビジネス［石川 02］やユビキタスコンピューティングにおいて XML データを利用する応用［木俵ら 03］がますます増えている．ここでは，データの挿入は起きるが，それ以外の変更は少なくて，検索が大部分であるようなデータ集約型応用を考える．例えば，応用として幅広く利用されている情報の推薦［Ishikawa ら 02］に注目したい．

これらの応用はさまざまな領域にあまねく存在するので，プログラミングをできるだけ記述しないで済むような計算の枠組みが必要であり，それを提案する．そのためにまずコンテクスト（文脈）とエンバイロメント（環境）という概念を導入する．コンテクストは今注目している局所的なデータ空間（いわゆるデータベースに相当，ただし２次記憶上である必要はない）である．エンバイロメントはグローバルなデータ空間である．それらの間でデータのコピーがおこなわれる．またそれら以外のデータ空間（複数あってよい）からの入力（センサ）とそれらへの出力（表示）も考えられる．ただし提案する枠組みは実現のために特定のデータベースシステム（データモデル）を仮定しない．

想定する応用ではデータに対して陽に操作を指定するのではなく，データがコンテクストに挿入されるとさらに別の操作が引き起こされる．その意味で一種のアクティブデータベース（能動データベース）［石川 94］の考えを取り入れている．本稿ではその主導的な原理（法則）を考えたい．しかしながらデータの記述そのものが応用の記述に対応するようにしたい．言い換えると問い合わせ言語やアクティブデータベースにおける ECA（イベント－条件－操作）記述言語を提案するのが目的ではない．提案する枠組みでは，実現できる応用に制限はあるが，一般性を失はない範囲でさまざまな応用が記述できるようにしたい．

無線 IC タグ［Taniguchi 05］［RFID 05］応用やモバイル機器では，複数の理由でデータ本体でなくその識別子（ID）のみを格納することが想定される．ID の指す実体に対応するデータはサーバに格納されている．そこで ID に基づく計算の枠組みが求められている．

提案する枠組みでは，事前にロジックをはっきりとは決めず，データが自己組織化［都甲ら 99］されていく．その意味で創発的［ジョンソン 04］であるといい，XML をとりあえず前提とするので創発的 XML（Emergent XML）という名前を使った．しかしながら特別な場合として RDB［石川

05］や OODB［石川 05］を含んでいると考えられ，本来はより一般的な枠組みであると考えている．
本稿では創発的 XML の導入とその可能性を論じる．詳細な実現や評価にはまだ着手しておらず，その意味で構想の段階であるが，今後の研究方向の一つを指し示すことができればよいと考えてあえて報告する．この節の残りでは関連する研究について述べる．さらに続く各節で，提案システムの概要，応用の記述，実現に関する考察を述べる．

１．２　関連研究

XML の能動的な問い合わせ言語の研究には［Bailey ら 02］や［Ishikawa ら 01］がある．これらは ECA に基づき，問い合わせ言語で XML 応用を記述することを目指すが，ここで提案する方法はロジックよりはむしろデータで応用を記述することを目指す．モバイル環境でのアクティブデータベースとしては寺田らの研究［寺田ら 00］がある．その研究は ECA ルールを XML の形式に埋め込んだ ML を導入している．やはり言語を目指している点でわれわれの研究と方向が異なる．
Abiteboul ら[Abiteboul ら 04]は，XML に動的要素を追加したものを分散環境でいかに効率的に処理するかに研究の主眼が置かれており，データを中心とした応用記述の枠組みの提案ではない．
コンテクストを意識した応用技術としてコンテクストアウエアネスの研究（[Moran ら 01]など）があるが，それらでは現状では地理的情報の利用や UI の研究に主眼が置かれている．
ユビキタスネットワーク環境でのマイニングに関する提案［久保ら 05］があるが，グリッドとセンサ利用によるメタデータの付与とその利用に依存している．そこでは応用を記述する枠組みを規定しているわけではない．ID に注目する点では RDF［RDF 05］とその周辺技術が関連する．
アクティブデータベースにおけるイベント・アクションの連鎖をコントロールする必要があるが，そのための方式として寺田らの研究［寺田ら 02］のようにトリガグラフの適用が考えられる．

２．提案システムの概要

２．１　概念の定義

まず基本概念を定義しておこう．データ形式は XML であることを仮定する．本当はかならずしも XML でなくてもよいのだが，すくなくとも識別子（ID）を持つことは必要とする．
ID は部分構造を持つ．すなわちカテゴリ（スキーマ）ID（IDs）と，インスタンス ID（IDi），およびバージョン ID（IDv）からなる．通常 IDs は顧客，商品などのカテゴリに対応する．IDi は個別の顧客（名）や商品（名）に対応する． そして IDv は商品個体の状態遷移に対応する．そのほか製造番号などの応用に依存する固体識別子も記述したい場合があり，それは ID のデータ抽象化として対応できるようにしたい．
データ空間はいわゆるデータベースに対応する概念である．まずコンテクストとエンバイロメントがある．そのほかに ID がそこから入力されるデータ空間とデータがそこへ出力されるデータ空間がある．
データ空間そのものの生成・消滅がある．データ空間も ID を持つ．データ空間はネットワーク（インターネット，アドホックネットワークなど）上に分散することを妨げない．
コンテクストには特別な性質がある．コンテクストに ID だけが入力されるとそれが指示するデータをエンバイロメントからコンテクストへコピーする．これを活性化という．活性化されると ID の指示する本体の ID は ID+と記述する．特にこれを正の活性化状態という．
一方，本体に含まれる他の ID（外部 ID）は ID－と記述する．それを負の活性化状態という．
ID+と ID－は結合することで秩序化の状態となる．この秩序化の状態を本稿では安定であるということにする．
言い換えると IDsIDiIDv が指定されていれば，それにすべて一致するものが選ばれる． IDsIDi だけが指定されるとれだけで一致をチェックし，その際もしあれば IDv が補われる．IDs のみの場合はそれだけで一致をチェックし，その際もしあれば IDiIDv が補われる．ただし IDv を無視して一致をチェックし，一致しても IDv を補わないオプションがあり，それをデフォールトにする．
ID の入力では，まずコンテクストからサーチする．もしあれば，その ID のバージョンを増やす．さもなければ，エンバイロメントから入力する．
コンテクスト内で活性化された ID の状態は正負にかかわらず安定ではない．その場合はエンバイロメントから対になる（一致する）ID を持つデータが検索されてコンテクストへコピーされる．その結果 ID は安定化し，コンテクスト全体も安定となる．ただし安定化の概念は，XInclude［XInclude 05］のように他の XML 断片を埋め込んで結果として一つの XML になることとは異なり，あくまでも対応する（一致する）ID を持つ XML データはそのまま並存する．また XInclude がベースとする XPointer［XPointer 05］が単一方向であるのに対して，ID の安定化は双方向リンクの概念を含んだものになる．
入力からは能動的にコンテクストへデータがコピーされる．

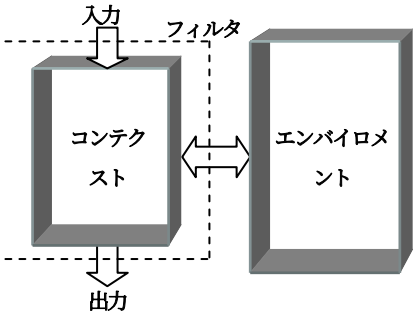


図1　データ空間の構成図

コンテクストが安定となるとコンテクストからデータが出力される.

データ空間同士でのデータのコピーでは，フィルタと呼ばれる条件を必要に応じて記述できる．指定されたフィルタ条件を満たすもののみがデータ空間にコピーされる．詳細な条件は既存の形式（例えば XPath［XPath 05］）に準拠した形式で指定することが可能である.

コンテクストが全体として安定かどうかをチェックする最大回数を指定できる．最大回数が指定されない場合は，コンテクストの極大集合が得られるまで安定化の操作は継続される.

われわれの提案を ECA に当てはめて解釈すれば，データの入力イベントにより安定化という条件がチェックされ，その条件が満たされれば，指定されたフィルタにより必要なデータに関するさらなる入力イベントが引き起こされるとなる.

このような安定化の作用が自己組織化（厳密には自己集合化）[ナノエレクトロニクス 05]の原理である．理解の助けとなるアナロジーとしては，非平衡系における物理化学的な現象［田中 02］［都甲ら 99］などがあるであろうか.

以上述べたデータ空間の典型的な構成を図１に示す．なおデータ空間の実装としては磁気ディスクベースの通常のデータベースのほかに主記憶データベースも許すことはいうまでもない.

２．２　操作の定義

ここでは概要で述べた操作のより詳細な定義を述べたい．操作はデータ構造と結びついており，抽象的データ構造として定義される.

まずデータ空間 Dataspace に対する操作を定義しよう．それは以下のようになる.

Dataspace.create()：データ空間を作りだす.

Dataspace.destroy()：データ空間を消滅させる.

Dataspace.insert({ID})：データ空間に指定された ID 集合を挿入する.

Dataspace.delete({ID})：データ空間から指定された ID 集合を削除する.

Dataspace.get({ID})：データ空間から指定された ID 集合の本体を検索し，値として返す.

Dataspace.getID ()：データ空間からすべての ID 集合を検索し，値として返す.

Dataspace.isUnstable()：データ空間が不安定であることをチェックし，真偽値を返す.

ただしデータ空間が不安定であるとは，以下の条件がともに成り立つ場合である.

- それに含まれる ID のどれかが不安定である.
- データ空間が変化（増殖）する．すなわち活性化が引き続き行われる.

つまり ID がすべて安定にならなくてもデータ空間が変化しなければデータ空間は安定になるし，ID がすべて安定になっても，データ空間が変化すればデータ空間は不安定となる.

Dataspace.setFilter()：データ空間にフィルタを定義する．デフォールトは素通りを許す.

Dataspace.filter()：データ空間に定義されたフィルタを実行する.

次に ID に関する重要な操作を定義しよう.

ID.isUnStable()：ID が不安定であることをチェックし，真偽値を返す.

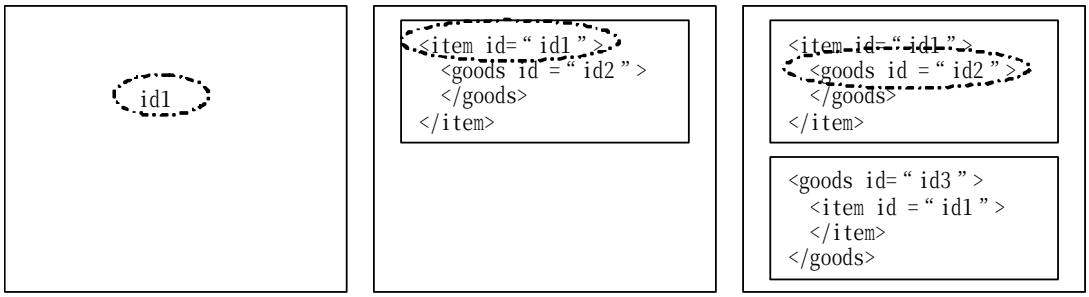


図2　不安定状態の例

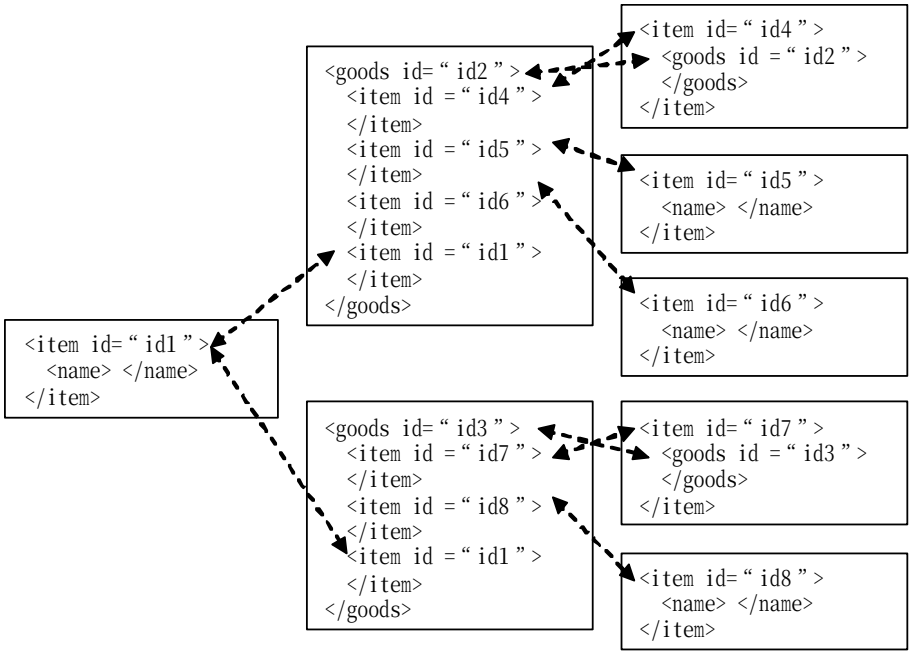


図3　安定状態の例

ただし ID がコンテクスト内で不安定であるとは以下の場合である.

- ID しかなく，ID 本体はない.
- ID 本体があるが，それを参照する別の ID 本体はない. すなわち正の活性化状態（ID+）である.
- ID 本体に含まれる別の ID はあるがそれが参照する ID 本体はない. すなわち負の活性化状態（ID−）である.

これらを模式的に表すと図2のようになる.

ID.setVersion()：IDｖを ID の一致のチェックに使うように設定する.

ID.equals (ID)：ID 同士の一致をチェックする.

次にデータ空間の安定化のチェック回数 Times に関する操作を定義する.

Times.initialize()：回数を初期化する.

Times.increment()：回数をカウントアップする.

Times.setLimit()：最大回数を指定する.

Times.isLimit()：最大回数かチェックし，真偽値を返す.

2．3　システムの概要

ここではシステムの動作原理（すなわち自己組織化の原理）を説明する. ID の入力というイベントによって引き起こされるシステムのメイン処理を擬似コードで記述すると以下のようになる.

```
// コンテクストへ入力する.
Context.insert(Context.filter(Input.getID()))
// コンテクストが安定するまでエンバイロメントからコンテクストへデータをコピーする.
Times.initialize()
do
  for i ∈ Context.getID()
    if i.isUnStable()
      Context.insert(Context.filter(Environment.Get(i)))
  Times.increment()
while NOT Times.isLimit()  AND Context.isUnStable()
// コンテクストの内容を出力する.
Output.insert(Output.filter(Context.getID()))
```

図3に安定状態を模式的に表す.

3．応用の記述

応用では，すでにある応用としてインターネットビジネス応用を取り上げ，提案する枠組みで簡単に記述することが出来ることを示す. 次に新しい応用として IC タグ応用をとりあげ，提案する枠組みがどのような可能性があるか説明したい. さらにバージョンを利用した応用を説明する.

応用の記述において特徴的な点は以下である. ユーザは応用（ロジック）を記述するためにいわゆるプログラミングをしない. 代わりにデータそのものを定義する. ただし必要に応じて最大回数，ID チェックにおけるバージョン利用の有無を指定し，各種フィルタの定義を行う. これらがロジックの必要最小限の部分に対応する.

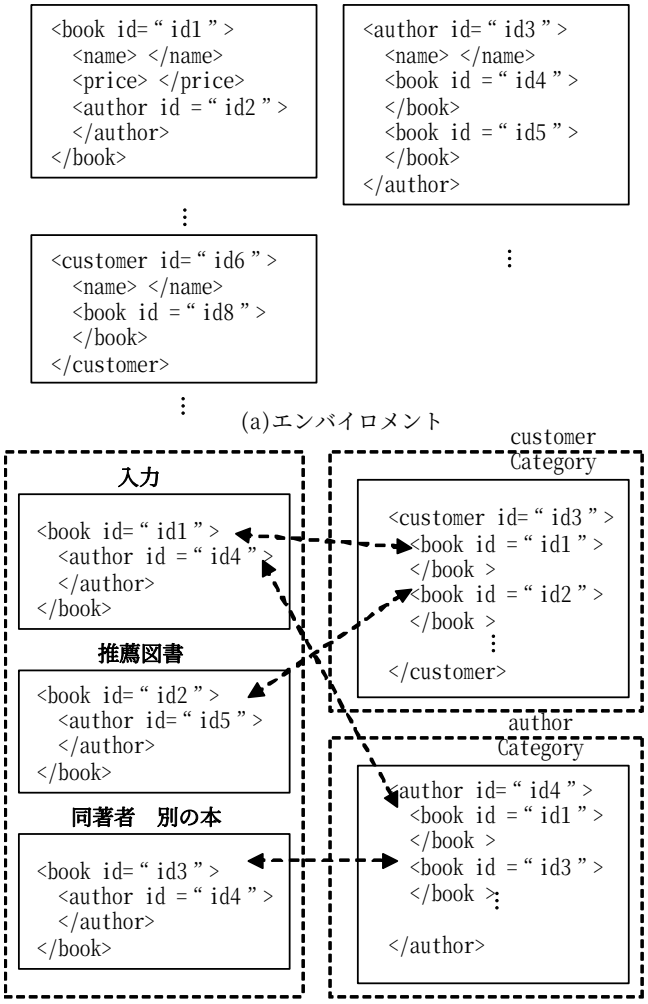


図4　（b）出力

3．1　インターネットビジネス

Amazon［Amazon 05］に代表されるようなインターネットビジネスでは，しばしば商品の推薦を販売促進のために行う．ここでは以下のような推薦を提案の枠組みで実現することを考えよう．

- 【推薦】この本を買った人は，こんな本も買っています．
- 【推薦】この本を書いた人は，こんな本も書いています．

ここでは例えば図 4(a)のような XML データによって応用が記述されていると仮定する．これらはインターネットショッピングサイトにあるデータベースサーバが管理するエンバイロメントに存在する．コンテクストはクライアントのインターネットブラウザが管理するデータ空間である．入力はインターネットブラウザを通して利用者が選択した商品（書籍）の ID である．出力はインターネットブラウザへの表示になる．

- 最大回数は 2 である．

各種フィルタとしては，それぞれ以下のように定義する．

- 入力フィルタは，無設定である．入力は this とする．
- コンテクストフィルタは，コンテクストがブラウザ（ショッピング）であることから，商品 X（書籍など），消費者（顧客）Y，製造者（著者など）Y´のみを入力の対象とする．
- 出力フィルタは以下のようになる．

Y（Y´）の個数は与えられた閾値以上であるような<Y（Y´）が X に this を含む場合 X をすべて求める>．

出力フィルタとして定義するものはこれひとつで十分であるが，括弧内を個別に表すと以下のようになる．

- この本<this>を買った人<Y>は，こんな本<X>も買って<Y>います．
- この本<this>を書いた人<Y´>は，こんな本<X>も書いて<Y´>います．

ID チェックの際にバーションは無視する．

出力される結果は例えば図 4(b)のようになる．実際にブラウザで出力するには最後に XSLT【XSLT 05】等で XHTML に変換する．

3．2　無線 IC タグ応用

スーパーマーケットなどで商品をショッピングバスケットに入れた場合に，複数の商品に関してあらたな推薦や注意（一種の“逆”推薦）を行うという応用が考えられる．例えば以下のような例である．

- 【推薦】この食材でこんな料理ができます．
- 【注意】この薬の組み合わせは危険です．

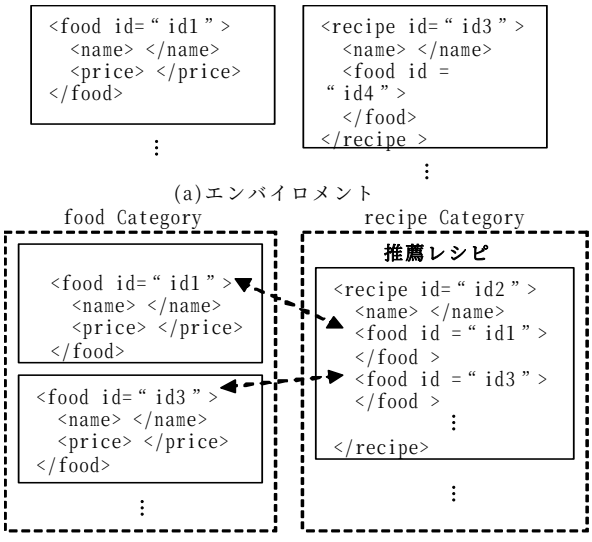
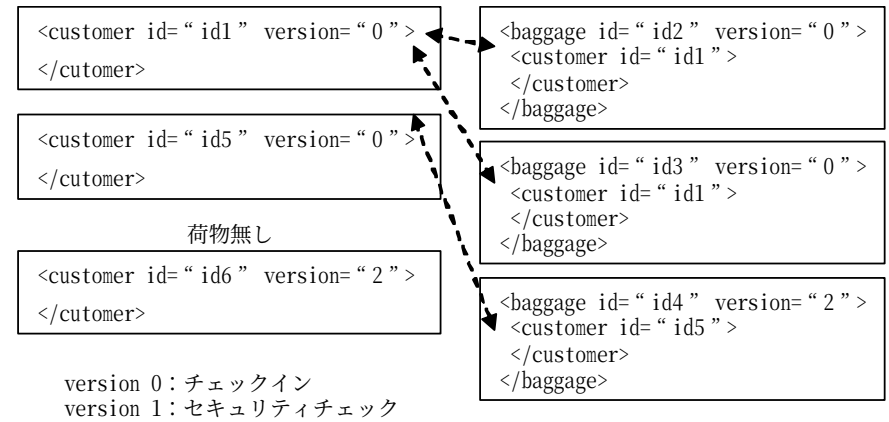
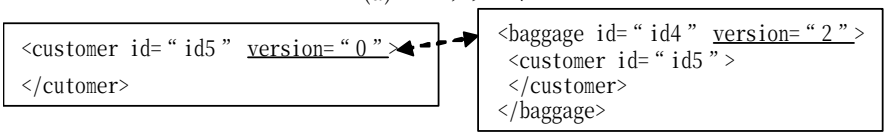


図5(b) 出力



(a) コンテクスト



顧客と荷物のバージョンが合っていないので，異常として出力される

図6（b） 異常出力

ここでは例えば図 5(a)のような XML データによって応用が記述されていると仮定する.

これらはスーパーマーケットなどのデータベースサーバが管理するエンバイロメントに存在する.

コンテクストはショッピング（バスケット）に対応し，通常はサーバで管理するデータ空間である．入力はショッピングバスケットに入れられた商品につけられた無線 IC タグが，無線 IC タグリーダを通してコンテクストへコピーされる．出力はスーパーマーケット内（ショッピングバスケット，売り場，チェックアウトなど）での表示に対応する.

・ 最大回数は1である.

各種フィルタとして，それぞれ以下のように定義する.

・ 入力フィルタは無設定である．入力は this とする.
・ コンテクストフィルタは，コンテクストがショッピングであるから，商品 X と関連情報（推薦あるいは注意）Y のみを入力の対象とする.
・ 出力フィルタは以下のようになる.

<Y が X に this を含む場合の Y を求める>.

出力フィルタとして定義するものはこれひとつで十分であるが，これを個別に表すと以下のようになる.

・ この食材<this>でこんな料理<Y>ができます.
・ この薬の組み合わせ<this>は危険<Y>です.

ID チェックの際にバーションは無視する.

例えば出力される結果は図5(b)のようになる.

3．3 バージョン利用

バージョンの変化は，ID の状態遷移に対応すると考えられる．バージョンに基づく応用として，飛行場におけるチェックインを考える．エンバイロメントは航空会社のサーバにある．コンテクストはチェックイン（顧客とその荷物）である．入力は顧客の携帯する IC カードや荷物のバーコード（もしくは IC タグ）からである．顧客がチェックイン時に同時に荷物をあずけると，荷物から顧客が関連付けられると同時にそれらのバージョンが初期化される（バージョン 0）．その後，顧客と荷物は別々の経路でセキュリティチェックを受け，それぞれのバージョンが増える（バージョン 1）．最終的にその顧客が搭乗ゲートを通過するとそのバージョンが増える（バージョン 2）．一方荷物が飛行機に積み込まれるとそのバージョンも増える（バージョン 2）．ここでの重要な注意事項は，荷物がチェックインされ，積載されたのに顧客が飛行機に乗らない（すなわち搭乗ゲートを通過しない）と飛行機はセキュリティ上の理由で離陸できないことである．すなわち，この例では顧客とその荷物のバージョンがともに 2 でなくてはならない．ここでの出力は搭乗ゲートなどの PC である.

・ 最大回数は1である.

各種フィルタとして，それぞれ以下のように定義する.

・ 入力フィルタは同一便である.
・ コンテクストフィルタは，コンテクストがチェックインであるから，顧客 X と荷物 Y のみを入力の対象とする.
・ 出力フィルタは，以下のようになる.

<X.IDv ≠Y.IDv である X と Y>
これにより最終的に図 6(a)のようなコンテクストから図 6(b)のような結果が出力される.

4．実現に関する考察

効率的な実現はこれからの研究課題のひとつである．データ表現が ID ベースであることはデータの格納管理にハッシュの適用を示唆する．また P2P ネットワークでの情報検索［石川 05］と同様に，ハッシュは分散環境においてデータを配置するべきノードの選択とも関係する.
ただし ID として何を用いるか，どういう構造を持たせるかは応用に依存する．例えば URI（Uniform Resource Identifier）[URI 05]のように汎用指向で考えるか，EPC（Electronic Product Code）[EPC 05]のような応用指向で考えるか選択肢が分かれる．さらにその組み合わせも考えれる．したがってシステムのエンジンとしては，ID は抽象的データ型として定義され，その詳細は隠蔽されていると仮定する．そのかわり ID のチェック機構などの詳細は，その抽象的データ型に付随するアクセスメソッドとして実現するようにする.
データ空間が増殖しないことは，極大集合が求められた場合と，なんらかの条件で枝狩りを行う場合が対応すると考えられる．現状では後者のために単純にデータ空間を安定化する操作の繰り返し回数の最大値を指定できるようにしている.
後者のより進んだ方式として Web コミュニティの発見の分野で用いられている２部グラフや最大フロー最小カット［石川 05］などのグラフ理論の応用が考えられる.

5．おわりに

インターネットやユビキタスネットワークにおいて XML データを集約的に利用した応用を最小限のロジックで記述する枠組みとして創発的 XML を提案した．簡単な応用例に関してそれが容易に記述できることを示した．まだ本研究はその緒についたばかりであり，残された課題は多い．今後の課題をまとめると以下のようになる.

- 概念の洗練化
- 記述できる応用範囲を明確化する.
- 実験を行い実現性や妥当性を評価する.
- 効率的な実現方式を詳細化する.

謝辞

この研究の一部は科学研究費補助金（特定領域研究，研究課題番号：16016273）による.

参考文献

[Abiteboul ら 04] Serge Abiteboul, Omar Benjelloun, Bogdan Cautis, Ioana Manolescu, Tova Milo, Nicoleta Preda: Lazy Query Evaluation for Active XML, Proceedings of SIGMOD 2004, pp.227 - 238 (2004).

[Amazon 05] http://www.amazon.com Accessed 2005.

[Bailey ら 02] James Bailey, Alexandra Poulovassilis, Peter Wood: An Event-Condition-Action Language for XML,
Proceedings of the 11th Int. Conf. on the World Wide Web, pp. 486 - 495 (2002).

[EPC 05] http://www.epcglobalinc.org/
Accessed 2005.

[石川 94] 石川博: アクティブデータベース，情報処理, Vol. 35, No. 2, pp.120–129 (1994).

[Ishikawa ら 01] Hiroshi Ishikawa, Manabu Ohta： An Active Web-based Distributed Database System for E-Commerce (2001).

[石川 02] 石川博： e‐ビジネス技術入門教科書―ビジネスモデルと情報技術(IT) IT TEXT, CQ 出版 (2002).

[Ishikawa ら 02] Hiroshi Ishikawa, Manabu Ohta, Shohei Yokoyama, Junya Nakayama, and Kaoru Katayama ： Web Usage Mining Approaches to Page Recommendation and Restructuring, International Journal of Intelligent Systems in Accounting, Finance, and Management, vol.11, pp.137-148 (2002).

[石川 05] 石川博： 次世代データベースとデータマイニング―DB&DM の基礎と Web・XML・P2P への適用 IT text, CQ 出版 (2005).

[ジョンソン 04]スティーブン・ジョンソン（著）山形浩生（訳）：創発 蟻・脳・都市・ソフトウェアの自己組織化ネットワーク，ソフトバンクパブリッシング (2004).

[木俵ら 03] 木俵豊,是津耕司,勝本道哲:ユビキタス環境におけるデバイス連携コンテンツの適合化手法，情報処理，研究報告 2003-DBS131 (2003).

[久保ら 05] 久保類，真鍋義文、盛合敏：グリッドコンピューティングによる実世界情報マイニングの提案，情報処理，研究報告 2005-DBS-136(2005).

[Moran ら 01] Thomas P. Moran, Paul Dourish: Introduction to This Special Issue on Context-Aware Computing, Human-Computer Interaction, 16 (2) p. 87-95 (2001).

[ナノエレクトロニクス 05]
http://www.nanoelectronics.jp/kaitai/selfassemble/index.htm Accessed 2005.

[RDF 05] RDF Model and Syntax Specification. http://www.w3.org/TR/REC-rdf-syntax/ Accessed 2005.
[RFID 05] http://itpro.nikkeibp.co.jp/rfid/ Accessed 2005.
[田中02] 田中 博：生命と複雑系，倍風館(2002).
[Taniguchi 05] Yoji Taniguchi : IC Tag Based Traceability: System and Solutions, Keynote talk, IEEE ICDE 2005.
[寺田ら 00] 寺田 努，塚本昌彦，西尾章治郎： アクティブデータベースを用いた地理情報システム，情報処理学会論文誌，Vol. 41, No. 11, pp.3103-3113 (2000).
[寺田ら 02] 寺田 努，塚本昌彦，西尾章治郎：移動体計算環境におけるアクティブデータベースの動的トリガグラフ構築機構の設計と実装，情報処理学会論文誌：データベース，Vol.43, No. SIG12(TOD16), pp. 52--63 (2002).
[都甲ら 99] 都甲 潔，江崎 秀，林 健司：自己組織化とは何か，講談社 (1999).
[URI 05] http://www.ietf.org/rfc/rfc3986.txt Accessed 2005.
[XInclude 05] http://www.w3.org/TR/2004/REC-xinclude-20041220/ Accessed 2005.
[XPath 05] XPath: XML Path Language (XPath) Version 1.0. http://www.w3.org/TR/xpath Accessed 2005.
[XPointer 05] http://www.w3.org/TR/2002/WD-xptr-xpointer-20021219/ Accessed 2005.
[XSLT 05] XSLT: XSL Transformations (XSLT) Version 1.0. http://www.w3.org/TR/xslt/ Accessed 2005.