

# ネイティブXMLサーバー 「Tamino」のご紹介 ～電子カルテの効果的な管理のために～

2004年5月27日

株式会社ビーコンIT  
アライアンス事業部 技術部  
安原 琢也  
E-mail: yasuhara@beacon-it.co.jp

©2004 Beacon Information Technology Inc. Japan

## 会社概要

- **社名: 株式会社 ビーコンIT**  
(登録社名 株式会社 ビーコン インフォメーション テクノロジー)
- **設立: 昭和51年8月18日**
- **資本金: 8億925万円**
- **社員数: 241名 (平成15年10月1日現在)**
- **拠点**
  - ◆ 本社(新宿)、東京事業所(恵比寿)、東京ソフトウェア・ラボラトリ
  - ◆ 札幌テクノセンター、横浜事業所、大阪事業所、名古屋事業所
- **主要取扱製品**
  - ◆ Tamino XML Server、EntireX、ADABAS、Natural
  - ◆ TimeCube、Waha!Transformer、EXTES、EXTES Xuras
  - ◆ Hyperion Essbase、Redbrick Warehouse

(1)XMLデータ保存管理向けデータベース製品紹介

tAMINO

主旨

■現状:

- ◆IT化の必要性
- ◆標準化の促進
- ◆XMLデータ(電子カルテ)の保存が本格化

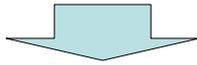
■課題:

- ◆効果的なXMLデータの保存・活用
- ◆パフォーマンス/スケーラビリティ

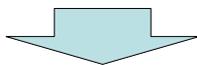
## これからの電子カルテシステムとは

### ■ 医療関係者への情報提供

- ◆ 施設内、施設を越えた情報共有・交換がなされていない



### ■ 情報交換規約の標準化をベースとし、「交換」と「管理」の双方の機能を兼ねたデータベースを基盤として構築されるべき



ネイティブXMLデータベース(NXD)  
「Tamino」

## ネイティブXMLデータベースとは

- ① 変換・マッピング・操作をすることなく格納できる。
- ② 論理モデルを定義、格納、検索できる。
- ③ XML文書の集合(コレクション<sup>\*1</sup>)を扱える。
- ④ XML標準技術を使用する。
- ⑤ 物理的な格納モデルは階層型である。

XML:DB Initiative 並びにZapThinkの定義、さらにBeaconITの考えを追加したものです。  
\*1:XML:DB Initiativeでは「XML文書は基本単位として格納され、RDBの行と同等である」という表現をしている。

## Taminoとは

- XMLドキュメントの効果的な管理・運用・活用のためのハイパフォーマンスなネイティブXMLデータベース
- 企業内のあらゆる型のコンテンツの検索や管理をサポート
- 企業間(B2B)でもプラットフォームに依存せず、既存資産を活用した仕組みを容易に構築可能



## Taminoによるデータ管理例

The screenshot shows the "Software AG Tamino X-Picker" interface. On the left is a "Navigation Tree" showing a hierarchy of collections and files. A red box highlights a "collection" containing "image" and "XVL" folders. Red arrows point from these folders to a browser window showing an image of a red camera and another window showing XML code. A yellow callout box on the right states: "ひとつのコレクション内に ①XML文書 ②画像ファイル ③XVLファイル を一元管理可能" (One collection can manage ① XML documents, ② image files, and ③ XVL files in a unified manner). The XML code in the bottom right window includes tags like <article>, <part>, <chapter>, and <img>. At the bottom, it says "©2004 Beacon Information Technology Inc. Japan".



- ## 機能と特長
- 
- スケーラビリティ
    - ◆ マルチプラットフォーム (Windows、Linux、Unix) 対応
  - ネイティブXML
    - ◆ XMLによる情報の最適な保存形式の提供
    - ◆ 非XMLデータ(画像・音声等)の管理
  - 使いやすい管理・操作インターフェース
    - ◆ ブラウザ/JavaアプリケーションによるGUI管理・操作ツールの提供
  - 時間とコストの節減
    - ◆ 効率的なアクセスと高速・柔軟な検索を実現(日本語形態素解析機能)
  - 外部との連携機能
    - ◆ RDB連携機能による他社RDBMSのXMLデータ変換
  - セキュリティと信頼性の提供
    - ◆ キャッシュ・サーバ、アクセス制御、認証メカニズム
  - Taminoはオープンである
    - ◆ スキーマ(XML Schema)、クエリ(XQuery)、シグネチャー
- ©2004 Beacon Information Technology Inc. Japan

## Tamino活用のメリット

### 診療文書に求められる特性

- ◆ 保存性(Persistence)
- ◆ 管理責任 (Stewardship)
- ◆ 真正性(Potential for authentication)
- ◆ 完全性(Wholeness)
- ◆ 見読性(Human readability)

出典: 村上 英博 (東京住電医療情報システムズ株式会社)  
「診療文書標準HL7 CDA(Clinical Document Architecture)」より



- 最適な管理プラットフォームの提供によるコスト低減
- 容易な管理機能・効果的な検索機能の提供
- 認証、電子署名による機密性の確保

## Taminoの評価ポイント

- スキーマ定義 = データ管理
- テキスト検索に強く、かつインデックスの実装が容易
- パラメータの数が少なく、構築・運用が容易
- 管理ツールの使い勝手がよい
- 大量データの格納・検索も対応可能
- 安定性が高い

**tamino** Number one in XML management

## Taminoスキーマ定義

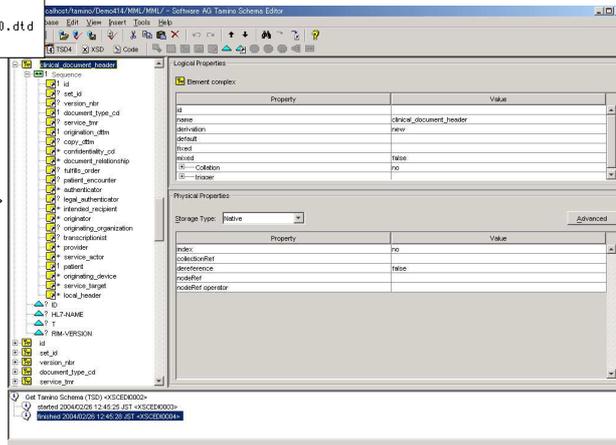
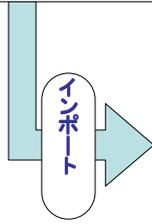
### MMLV3.0DTDs

各DTDの内容を1つの構文文書としてまとめる

MML\_3.0\_levelone\_1.0.dtd  
MML\_3.0\_header\_1.0.dtd  
v3dt\_1.0.dtd  
MML\_datatypes\_and\_header\_3.0.dtd  
MML\_Modules\_3.0.dtd

### ポイント:

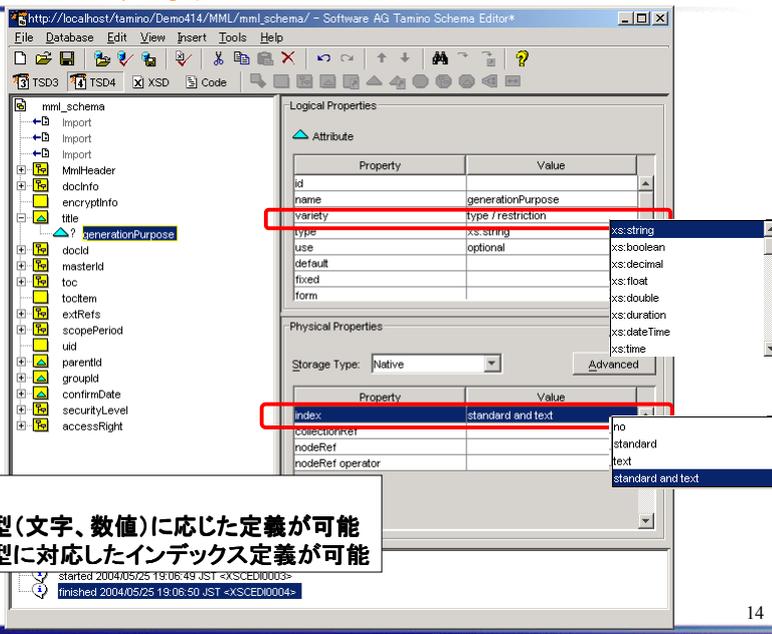
- ①複数の名前空間に対応済
- ②XMLインスタンスのバリデーションチェックとして機能



MMLVersion3.0規格書(平成15年1月付)のDTDを使用

©2004 Beacon Information Technology Inc.Japan

## インデックス定義



### ポイント:

- ①データ型(文字、数値)に応じた定義が可能
- ②データ型に対応したインデックス定義が可能

## Tamino ユーザー一覧

- 旭化成㈱
- ㈱アシックス
- 伊藤忠商事㈱
- 呉羽化学工業㈱
- サッポロビール㈱
- 三洋電機㈱
- 新光電気工業㈱
- 住友金属鉱山㈱
- 東芝セラミックス㈱
- 東武鉄道㈱
- ネットワンシステムズ㈱
- ㈱ムラウチ
- 岩手県立大学
- 大阪大学蛋白質研究所
- 北海道大学
- (独)国立国語研究所
- (独)産業技術総合研究所
- (独)防災科学技術研究所
- その他
- Sanacorp Pharmahandel AG(ドイツ)
- Novartis(スイス)
- State Street Corporation(アメリカ)
- Dresdner Bank AG(ドイツ)
- Heller Bank AG(ドイツ)
- Avon and Somerset Constabulary(イギリス)
- German Federal Armed Forces(ドイツ)
- Apollo2000(イギリス)
- Deutsche BP(ドイツ)
- City of Munich(ドイツ)
- Lear Corporation FIAT Division(イタリア)
- DaimlerChrysler AG(ドイツ)
- Kaeser Compressors(ドイツ)
- University of St. Gallen(スイス)
- Max.mobile(オーストリア)
- Elia(ベルギー)
- North American Logistics(アメリカ)
- その他

2004年1月末現在：日本国内約150社の導入実績      2003年12月末現在：海外約1,000社の導入実績

©2004 Beacon Information Technology Inc. Japan

15

## (2) パフォーマンス/スケーラビリティについて課題検討結果

# tamino

©2004 Beacon Information Technology Inc. Japan

16

## テスト前提条件

### テストデータ:

- ①2つの施設(RT)、約5,000人の患者(EX)からなる5万件のデータを作成  
MMLVersion3.0規格書(平成15年1月付)のサンプルインスタンスを例に作成
- ②1人の患者は、1つの施設に対して最大6つの文書種別を持つ  
テーブルMML0007にある24の文書種別から任意に6種類を選択
- ③1人の患者は2つの施設(RT)に跨る場合もある

### 検索条件:「PID(EX)とFID(RT)でマッチした文書種別」

((att1 = PID1 and att2 = FID1) or (att1 = PID1 and att2= FID2))  
and  
((att3 = \*P\_01\*) or (att3 = \*P\_02\*) or ... (att3 = \*P\_10\*))

att1:PID(施設内患者ID EX)

att2:FID(施設ID RT)

att3:P\_01~P\_10(文書種別属性値 generationPurpose)

テーブルMML0007にある24の文書種別から下記10種類を選択

(recordInpatient、recordAdmission、recordConsult、reportRadiology、  
recordOutpatient、summaryOutpatient、informedConsent、reportPathology、  
reportTest、legalRecord)

## テスト方法

### ■ DB作成

- ◆TaminoManagerを使用

### ■ スキーマ定義

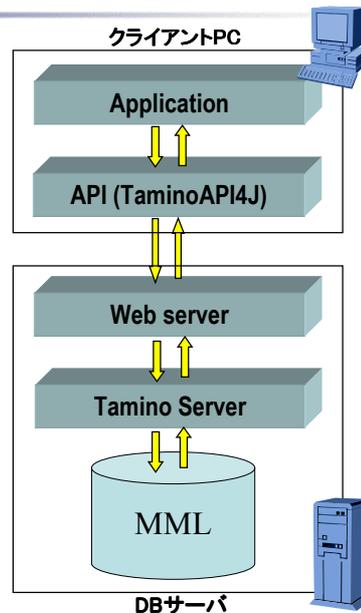
- ◆TaminoSchemaEditorを使用

### ■ データ登録

- ◆DataLoaderを使用

### ■ 計測時間

- ◆クライアントPC上のJavaアプリケーションから、クエリをDBサーバ(Tamino)へ発行し、その結果が返ってくるまでの時間を計測



## X-Query例と結果例

### クエリ例

```

/levelone[(((clinical_document_header/patient/personId/@EX='03034' and
clinical_document_header/Id/@RT='1.2.392.114319.1.5.1.1.1.1') or
(clinical_document_header/patient/personId/@EX='03034' and
clinical_document_header/Id/@RT='1.2.392.114319.1.5.1.1.1.2')) and
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='Inp' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='dAd' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='rdC' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='Rad' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='dOu' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='yOu' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='edC' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='rIP' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='rTes' or
/body/section/paragraph/content/local_markup/mml:docInfo/mml:title/@generationPurpose='alR')]
    
```

ポイント:  
数値型のデータに対しては、演算子(">","<")を使用した検索条件を指定可能

### クエリ結果例

```

<?xml version="1.0"?>
<result>
  <claim title="generationPurpose='recordingpatient'" inoId="14311" xmlns:claim="http://www.medxml.net/claim/claimModule/2.1"
  xmlns:claimA="http://www.medxml.net/claim/claimAmountModule/2.1" xmlns:mml="http://www.medxml.net/MML"
  xmlns:mmlAd="http://www.medxml.net/MML/SharedComponent/Address/1.0" xmlns:mmlBc="http://www.medxml.net/MML/ContentModule/BaseClinic/1.0"
  xmlns:mmlCi="http://www.medxml.net/MML/SharedComponent/CreatorInfo/1.0"
  xmlns:mmlCm="http://www.medxml.net/MML/SharedComponent/Common/1.0" xmlns:mmlDp="http://www.medxml.net/MML/SharedComponent/Department/1.0"
  xmlns:mmlFc="http://www.medxml.net/MML/SharedComponent/Facility/1.0"
  xmlns:mmlFcl="http://www.medxml.net/MML/ContentModule/FirstClinic/1.0" xmlns:mmlFi="http://www.medxml.net/MML/ContentModule/HealthInsurance/1.1"
  xmlns:mmlLb="http://www.medxml.net/MML/ContentModule/Inpatient/1.0"
  xmlns:mmlLs="http://www.medxml.net/MML/ContentModule/Lifestyle/1.0" xmlns:mmlMm="http://www.medxml.net/MML/SharedComponent/Name/1.0"
  xmlns:mmlPc="http://www.medxml.net/MML/ContentModule/ProgressCourse/1.0"
  xmlns:mmlPn="http://www.medxml.net/MML/SharedComponent/Phone/1.0" xmlns:mmlPp="http://www.medxml.net/MML/ContentModule/PatientInfo/1.0"
  xmlns:mmlPsl="http://www.medxml.net/MML/SharedComponent/PersonalizedInfo/1.0"
  xmlns:mmlRd="http://www.medxml.net/MML/ContentModule/RegisteredDiagnosis/1.0" xmlns:mmlRe="http://www.medxml.net/MML/ContentModule/Referral/1.0"
  xmlns:mmlRp="http://www.medxml.net/MML/ContentModule/Report/1.0" xmlns:mmlSc="http://www.medxml.net/MML/SharedComponent/Security/1.0"
  xmlns:mmlSg="http://www.medxml.net/MML/ContentModule/Surgery/1.0"
  xmlns:mmlSm="http://www.medxml.net/MML/ContentModule/Summary/1.0" xmlns:xhtml="http://www.w3.org/1999/xhtml">
    <claim title="generationPurpose='summaryOutpatient'" inoId="14313" xmlns:claim="http://www.medxml.net/claim/claimModule/2.1"
    xmlns:claimA="http://www.medxml.net/claim/claimAmountModule/2.1" xmlns:mml="http://www.medxml.net/MML"
    xmlns:mmlAd="http://www.medxml.net/MML/SharedComponent/Address/1.0" xmlns:mmlBc="http://www.medxml.net/MML/ContentModule/BaseClinic/1.0"
    xmlns:mmlCi="http://www.medxml.net/MML/SharedComponent/CreatorInfo/1.0"
    </claim>
  </result>
    
```

## ご参考: XQuery例と結果例

### クエリ式

```

for $a in input()/levelone/clinical_document_header/local_header
let $b := $a/mml:MmlHeader/mml:Ci:CreatorInfo/mml:Psi:PersonalizedInfo
let $c := $b/mml:Psi:personName/mml:Na:Name/mml:Na:fullname
let $d := $b/mml:Fc:Facility
let $e := $b/mml:Dp:Department
let $f := $b/mml:Psi:addresses/mml:Ad:Address/mml:Ad:full
let $g := $b/mml:Psi:phones/mml:Ph:Phone
let $h := $g/mml:Ph:area
let $i := $g/mml:Ph:city
let $j := $g/mml:Ph:number
where $f[containsText($e/mml:Dp:name, "内科")]
return
  <PatientList>
  <ID>{$b/mml:Ci:Id/text()}</ID>
  <Name>{$c/text()}</Name>
  <Facility>{$d/mml:Fc:name/text()}</Facility>
  <Department>{$e/mml:Dp:name/text()}</Department>
  <Address>{$f/text()}</Address>
  <Phone>{string-join(($h, $i, $j), "-")}</Phone>
  </PatientList>
    
```

ポイント:クエリ結果を任意のXML文書で取得可能

Clinical\_document\_headerの下のLocal\_header内の各要素からmmlDp:nameの値が「内科」である患者一覧(PatientList)を抽出

### クエリ結果

```

<?xml version="1.0"?>
<result>
  <PatientList>
    <ID>00000002486</ID>
    <Name>吉原 賢二</Name>
    <Facility>アボッククリニック</Facility>
    <Department>内科</Department>
    <Address>横浜市中区山下町1番地 シルクビル</Address>
    <Phone>0985-12-3456</Phone>
  </PatientList>
  <PatientList>
    <ID>00000002486</ID>
    <Name>吉原 賢二</Name>
    <Facility>アボッククリニック</Facility>
    <Department>内科</Department>
    <Address>横浜市中区山下町1番地 シルクビル</Address>
    <Phone>0985-12-3456</Phone>
  </PatientList>
</result>
    
```

## テスト環境について

### ■ クライアント(テスト実行環境)

- ◆ CPU: Pentium4(2.4GHz × 1)
- ◆ メモリ: 1GB
- ◆ OS: Windows 2000 Pro

### ■ サーバ(Taminoサーバ):

- ◆ CPU: Xeon(3.0GHz × 1)
- ◆ メモリ: 4GB
- ◆ OS: Windows 2000 Server

Property	Value
Bufferpool Size	60.00 MB
Current Used Bufferpool Size	47.52 MB
Current Number of Index Blocks	833
Current Number of Data Blocks	1325
Current Number of Temp Blocks	0
Number of Buffer Flushes	210
Logical Reads	2748189
Physical Reads	2010
Physical Writes	815

Bufferpool Size	60.00 MB
Current Used Bufferpool Size	47.52 MB

21

## まとめ:

- これからの診療情報は“XML”がベースである。
- XMLデータの容易な管理と効果的なパフォーマンスを実現するためには、それに相応しい“データコンテナ”を選択すべき。
- これからの診療情報の共有を、効率よく行うためには、MMLのDTDに完全対応でき、「交換」と「管理」の双方の機能を兼ねた「Tamino」をお勧めします。

22