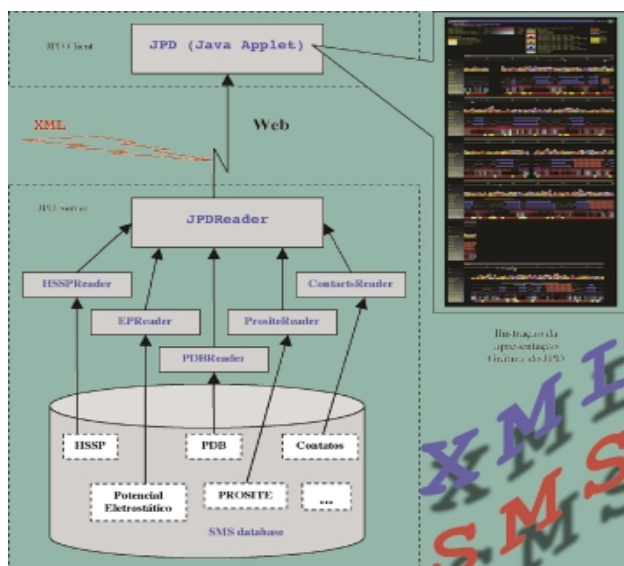


ISSN 1677-8464

## Experiência de Utilização de XML no SMS

Roberto Hiroshi Higa<sup>1</sup>  
 Christian Baudet<sup>2</sup>  
 Esther Manoela de Freitas<sup>3</sup>  
 Gabriela Félix dos Santos<sup>4</sup>  
 Aduino Luiz Mancini<sup>5</sup>  
 Paula Kuser Falcão<sup>6</sup>  
 Goran Neshich<sup>7</sup>



A *Extensible Markup Language* - XML é um padrão para documentos estruturados voltado para a Web. Ela é derivada do padrão *Standard Generalized Markup Language* - SGML, o padrão internacional para definição de estrutura e conteúdo de diferentes tipos de documentos eletrônicos (Harold, 1999). Da mesma forma que a SGML, a XML permite que o usuário defina suas próprias *tags* ou marcações, constituindo-se assim, em uma meta-linguagem para especificação de linguagens de marcação.

Originalmente projetada para solucionar problemas relacionados com a publicação eletrônica em larga-escala, XML vem ganhando importância também como um formato de troca de dados, tanto na Web quanto fora dela (World Wide Web Consortium, 2002a). Sua

disseminação tem sido impulsionada principalmente pelas possibilidades que se vislumbram para o desenvolvimento de aplicações do tipo *business to business* - B2B, o que tem feito com que diversas empresas como IBM, Oracle e Sun anunciassem algum tipo de comprometimento com essa tecnologia. Outro fator importante para a sua bem sucedida disseminação é o grande número de empresas de tecnologia da informação e projetos de código aberto que produzem e disponibilizam ferramentas para desenvolvimento de aplicações baseadas em XML, como, por exemplo, Xalan, Xerces e Cocoon do Projeto Apache (Apache Software Foundation, 2002), JAXP<sup>8</sup> e JAX-RPC<sup>9</sup> da Sun Microsystems (2002), AIX-SDK<sup>10</sup> e XROM<sup>11</sup> da IBM Corporation (2002) e Oracle XDK<sup>12</sup> da Oracle Corporation (2002).

<sup>1</sup> M.Sc. em Engenharia Elétrica, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: roberto@cbi.cnptia.embrapa.br)

<sup>2</sup> Estudante de Engenharia da Computação, Estagiário da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: christian@cbi.cnptia.embrapa.br)

<sup>3</sup> Estudante do Curso Técnico de Informática, Estagiário da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: esther@cbi.cnptia.embrapa.br)

<sup>4</sup> Estudante do Curso Técnico de Informática, Estagiário da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: gabriela@cbi.cnptia.embrapa.br)

<sup>5</sup> Bacharel em Ciência da Computação, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: adauto@cbi.cnptia.embrapa.br)

<sup>6</sup> Ph.D. em Física Aplicada, Cristalografia de Proteínas, Pesquisadora da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: paula@cbi.cnptia.embrapa.br)

<sup>7</sup> Ph.D. em Biofísica, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: neshich@cbi.cnptia.embrapa.br)

<sup>8</sup> JAXP - Java API for XML Parsing, onde API significa Application Programming Interface.

<sup>9</sup> JAX-RPC - Java API for XML based RPC, onde RPC significa Remote Procedure Call.

<sup>10</sup> AIX-SDK - Java Development Kit para sistema operacional AIX Unix da IBM.

<sup>11</sup> JROM - Java Record Object Model.

<sup>12</sup> XDK - XML Development Kit.

A comunidade científica de biologia molecular e genômica manipula uma grande quantidade de dados, trocando-os e atualizando-os freqüentemente. Hoje é impossível pensar-se em pesquisas em biologia molecular sem que se tenha acesso aos grandes bancos de dados como GeneBank (National Center for Biotechnology Information, 2002a) para acesso a seqüências anotadas de DNA, o Swiss-prot (Bairoch & Apweiler, 2000) para acesso a seqüências anotadas de proteínas e o PDB – *Protein Data Bank* (Berman et al., 2000) para acesso a estruturas tridimensionais de proteínas, todos acessíveis através da Web. Rapidamente, percebeu-se o potencial de utilização de XML para aplicações na área científica. Em particular, muitos dos provedores de dados de biologia molecular ou já disponibilizam seus dados em formato XML ou estão desenvolvendo algum projeto neste sentido.

O Núcleo de Bioinformática Estrutural – NBI da Embrapa Informática Agropecuária tem como seu principal produto o *Sting Millennium Suite – SMS* (Structural Bioinformatics Group, 2002), acessível através da Web, usado para análise dos dados produzidos e/ou disponibilizados pelo NBI, através do *SMS database* - como contatos intramoleculares, contatos intermoleculares, entropia relativa, estrutura secundária determinada por diferentes métodos, área acessível por solvente em complexo e para cada cadeia isolada, curvatura, potencial eletrostático e hidrofobicidade.

O objetivo deste trabalho é apresentar como XML foi utilizada no desenvolvimento do módulo *Java Protein Dossier – JPD* do SMS para troca de informações entre as partes cliente e servidora.

Nas próximas seções são apresentadas: a linguagem XML de maneira geral e uma rápida revisão das principais aplicações em bioinformática; a forma como XML foi utilizada no desenvolvimento do JPD; e uma discussão dos resultados obtidos e perspectivas de trabalhos futuros.

## XML e sua utilização em aplicações de bioinformática

Um documento XML é basicamente um arquivo texto com a informação estruturada através de um conjunto de elementos. Cada elemento é delimitado por uma *tag* de início e outra de final, podendo ainda ter um conjunto opcional de atributos associados, definidos por pares chave/valor. Entre a *tag* inicial e a *tag* final está o dado ou conteúdo associado ao elemento. Além disso, os elementos de um documento XML são estruturados de forma hierárquica, podendo ser representados como árvore. Sua estrutura lógica é

definida por um documento denominado DTD — *Document Type Definition*<sup>13</sup> (Harold, 1999; McLaughlin, 2000). A Fig. 1 ilustra um documento XML simples (Harold, 1999).

```
<person ID="p1100" sex="M" >
  <name>
    <given>Judson</given>
    <surname>McDaniel</surname>
  </name>
  <birth>
    <date>21 Feb 1834</date>
  </birth>
  <death>
    <date>9 Dec 1905</date>
  </death>
</person>
```

Fig. 1. Exemplos simplificados de documento XML.

Além da especificação de XML, a W3C também tem patrocinado projetos para a definição de uma série de especificações associadas, como por exemplo (World Wide Web Consortium, 2002a):

- XSL — *EXtensible Stylesheet Language*: linguagem para expressar folhas de estilo. Ela consiste de três partes distintas. A primeira é uma linguagem para transformação de XML, denominada XSLT – *XSL-Transformations*; a segunda, denominada Xpath – *XML-Path Language*, é uma linguagem usada por XSLT para acessar ou referenciar partes de um documento XML; e a terceira, denominada XSL *Formatting Objects*, é um vocabulário XML para especificação de semântica de formatação;
- Xlink – *XML Linking Language*: linguagem para expressar links em documentos XML, ou seja, elementos que criam ou descrevem ligações entre recursos. Ela utiliza XML para criar e descrever estruturas que descrevem links, que podem ser similares aos *hyperlinks* unidirecionais do HTML (World Wide Web Consortium, 2002b), mas também são capazes de expressar *links* com características bem mais sofisticadas;
- Xquery – *XML Query Language*: facilidades de *query* para extrair dados de documentos XML reais e virtuais da Web, permitindo que coleções de documentos XML sejam acessados de forma similar a um banco de dados;

<sup>13</sup> Atualmente a W3C (World Wide Web Consortium, 2002c), um consórcio formado para desenvolver e propor padrões de tecnologias interoperáveis na Web, vem trabalhando em uma nova forma mais flexível para especificação da estrutura lógica de documentos XML denominada *Schema*.

- RDF – *Resource Description Framework*: utiliza XML para definir um sistema de ontologia leve para suporte a troca de conhecimento através da Web.

Para utilizar XML, produtores de software fazem uso de processadores XML, padronizados ou não pela W3C, para realizar o *parser* e a manipulação de documentos XML. Existem dois tipos de processadores XML padronizados pela W3C (World Wide Web Consortium, 2002a):

- os dirigidos por evento, representados basicamente pela API SAX - *Simple API for XML* (McLaughlin, 2000), que lê o documento XML seqüencialmente e a cada ocorrência de *tag* de início, *tag* de final, atributos, etc. sinaliza um evento para o processador; e
- os baseados em árvore, representados pela API DOM - *Document Object Model* (Harold, 1999), que representam o documento XML como uma árvore, o que facilita o acesso aleatório às diversas partes do documento.

Outros exemplos de APIs para manipulação de documentos XML são o JDOM<sup>14</sup> (JDOM, 2002) e XOM<sup>15</sup> (XOM, 2002) e XML-RPC<sup>16</sup> (Userland Software Inc., 2002).

Aplicações de bioinformática, de maneira geral, manipulam conjuntos de dados com as seguintes características (Achard et al., 2001):

- oriundos de fontes diversas, redundantes e com formatação não padronizada, em geral como arquivos texto;
- à medida que análises sobre esses dados são realizadas, mais dados são gerados, e estes precisam ser modelados e integrados ao conjunto de dados já existentes;
- modelam um grande número de objetos complexos e que são atualizados freqüentemente;
- o seu volume cresce exponencialmente.

Isso gera uma necessidade, cada vez mais presente, de soluções para acesso e manipulação de dados heterogêneos de forma integrada. É dentro deste contexto que a utilização de XML para aplicações em bioinformática tem despertado um interesse crescente. Ao longo dos últimos anos, observou-se o aparecimento de diversas iniciativas para utilização de XML como formato para troca de dados de bioinformática. Algumas das aplicações de XML que resultaram dessas iniciativas são:

- A BSML – *BioInformatic Sequence Markup Language* (BSML..., 2002), que é uma DTD desenvolvida para representar seqüências de DNA – ácido desoxirribonucléico, RNA – ácido ribonucléico, e proteínas. Recentemente, ela foi adotada pela EMBL – *European Molecular Biology Laboratory* como um dos formatos para tornar disponível o banco de dados de seqüências de DNA *EMBL Nucleotide Sequence Database* (European Molecular Biology Laboratory, 2002);
- A BioML – *BIOpolymer Markup Language* (Biopolymer..., 2002), que é uma DTD desenvolvida para modelar a estrutura hierárquica de um organismo vivo, inclusive, permitindo que dados de fontes diferentes sejam combinados;
- GAME – *Genome Annotation Markup Elements* (GAME..., 2002), que é uma DTD desenvolvida em Berkeley durante o projeto genoma da drosophila (Berkeley Drosophila Genome Project, 2002) para anotação e distribuição de dados. Recentemente, ela foi atualizada para utilização genérica;
- AGAVE – *Architecture for Genomic Annotation, Visualization and Exchange* (LifeCode Inc., 2002), que é um formato - DTD e *Schema* - XML desenvolvida pela empresa Double Twist para manipular dados de anotação genômica;
- PROXIML – *PROtein eXtensible Markup Language* (McArthur, 2002), que é um XML Schema proposto como uma alternativa mais moderna e flexível para troca automática de dados sobre proteínas.

Além disso, recentemente o NCBI – *National Center for Biotechnology Information* passou a disponibilizar os dados do GeneBank em um formato XML próprio (National Center for Biotechnology Information, 2002b), em adição ao padrão ASN.1 – *Abstract Syntax Notation number One* (ASN.1, 2002), adotado há mais de 10 anos para este fim. Também, recentemente, a EMBL lançou um projeto denominado XEMBL (Wang et al., 2002), cujo objetivo é disponibilizar seus dados em diversos formatos XML. Atualmente, os formatos BSML e AGAVE já são suportados, mas existe a promessa de que os dados também serão disponibilizados em outros formatos como BIOML e GAME, entre outros. Projetos abertos de bioinformática como bioperl, biojava e biophyton (BioPerl..., 2002) implementam parsers XML para formatos como GAME, AGAVE e BlastXML, uma DTD para representar resultados de uma busca Blast – *Basic Local Alignment Search Tool* (Altschul et al., 1990, 1997), utilizando XML.

Revisões sobre utilização de XML em bioinformática mais completas e atualizadas podem ser encontradas em XML... (2002); Gordon (2002) e Achard et al. (2001).

<sup>14</sup> DOM **não** é acrônimo de *Java Document Object Model*. É simplesmente o nome da biblioteca idealizada por Brett McLaughlin and Jason Hunter (JDOM, 2002).

<sup>15</sup> XOM - *XML Object Model*.

<sup>16</sup> RPC - *Remote Procedure Call*.

## Utilização de XML no módulo JPD do SMS

O SMS compreende um conjunto de programas, acessíveis através da Web, para análise de estruturas de proteínas. Desenvolvido pelo NBI, ele permite a visualização simultânea da estrutura e da seqüência da proteína e apresenta de forma amigável um conjunto de propriedades físico-químicas mapeadas para cada resíduo.

De maneira simplificada, o SMS pode ser entendido como sendo composto por duas partes principais. Uma parte servidora, denominada *SMS server* e uma parte cliente, denominada *SMS client*. A parte servidora compreende as seguintes funcionalidades:

- atualizar freqüentemente bancos de dados de domínio público como o banco de dados de estruturas de proteínas PDB — *Protein Data Bank* (Berman et al., 2000), o banco de dados de estruturas secundárias derivadas por homologia HSSP — *Homology-Derived Secondary Structure of Proteins* (Sander & Schneider, 1991; Schneider et al., 1997) e o banco de dados de famílias e domínios de proteínas PROSITE (Hofmann et al., 1999);
- utilizando esses bancos de dados e programas especializados, criar informações específicas como contatos (ou interações) intramoleculares, contatos (ou interações) intermoleculares, interação proteína/DNA, área na superfície acessível ao solvente, potencial eletrostático, curvatura na superfície e hidrofobicidade na superfície.

O conjunto de dados resultante deste processamento é denominado *SMS-database*. Já o *SMS client* é responsável por apresentar, de forma amigável para os usuários, o conjunto de dados processados pela parte servidora, constituindo-se num *Front-End* do *SMS-database*.

A versão 3.0 do SMS, atualmente em estágio de beta-teste, tem como novas funcionalidades, basicamente, os módulos *Java Table of Contacts* – JTC e *Java Protein Dossier* – JPD. Especificamente o JPD tem como funcionalidade a apresentação gráfica dos diversos parâmetros físico-químicos, relativos à estrutura de proteínas, armazenados no *SMS-database*. Essa base de dados, por sua vez, é constituída por um conjunto de diferentes bases de dados, organizadas como arquivos texto. Assim, durante o desenvolvimento da versão 3.0 decidiu-se pela utilização de XML como formato para troca de dados entre a parte servidora e cliente do JPD. Desta forma, obteve-se uma solução que:

- conseguisse lidar com os diversos formatos de dados que constituem o *SMS-database*;
- fosse flexível o suficiente para suportar futuras alterações tanto na forma de armazenamento do *SMS-database* quanto na arquitetura do próprio SMS; e
- estivesse sintonizada com a atual tendência tecnológica de manipulação de dados de bioinformática.

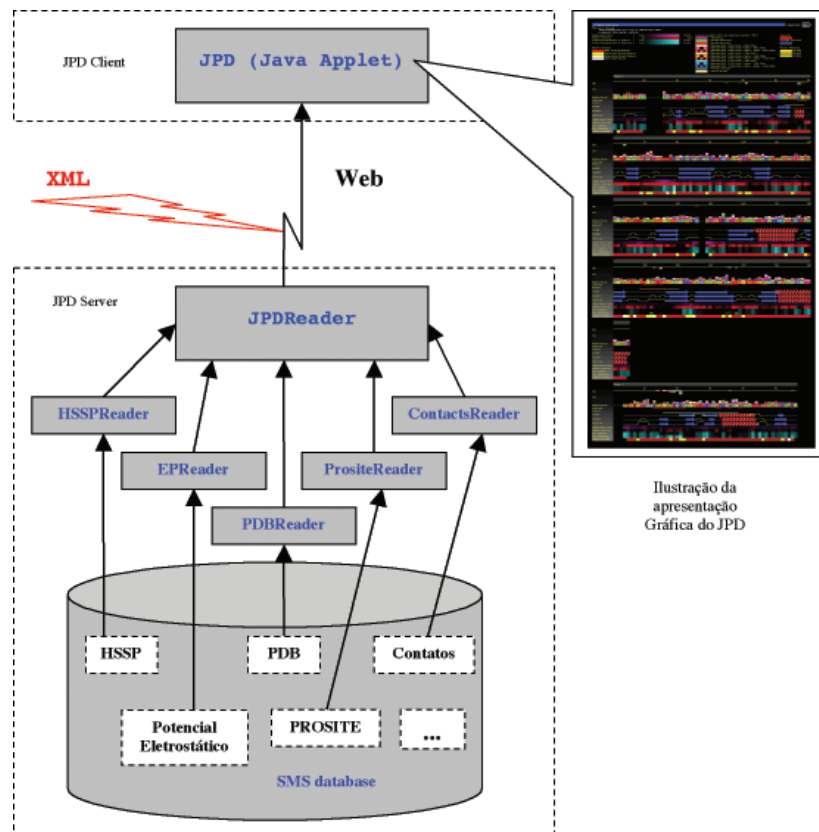


Fig. 2. Utilização de XML pelo módulo JPD do SMS.



A Fig. 2 ilustra a forma como XML foi utilizado na construção do módulo JPD do SMS. O *SMS-database* é formado por um conjunto de bases de dados de domínio público, como o PDB, o HSSP e o PROSITE, e outras com parâmetros calculados no próprio NBI, como Potencial Eletrostático e Contatos. Cada uma dessas bases de dados é organizada como um conjunto de arquivos, sendo que para cada arquivo PDB existe um arquivo HSSP, um arquivo EP (Potencial Eletrostático), um arquivo CON (Contatos), etc. Todos esses arquivos são textuais, sendo que cada um deles possui sua formatação específica.

O JPD é formado por um módulo cliente e um módulo servidor. O módulo servidor, desenvolvido em C++ (Stroustrup, 1997), é composto por diversos componentes, que são responsáveis pela leitura dos diversos tipos de arquivos textuais que formam o *SMS-database*, e um componente, denominado JPDRreader, responsável por criar uma estrutura de dados em forma de árvore em memória, de forma similar às utilizadas por *parsers* DOM (Harold, 1999). Esses dados são, então, transmitidos para o módulo cliente no formato de um documento XML. O módulo cliente, um Java Applet (Campioni & Walrath, 1996), utiliza o parser SAX 2.0 (McLaughlin, 2000), implementado na API Xerces (Apache Software Foundation, 2002), para extrair do

documento XML recebido do módulo servidor os dados da *SMS-database* utilizados para construção da apresentação gráfica.

A Fig. 3 apresenta um exemplo de um documento XML utilizado para troca de dados entre os módulos cliente e servidor do JPD. As *tags* utilizadas refletem a natureza hierárquica dos dados apresentados pelo JPD. O elemento delimitado pela *tag* <pdb> especifica que os dados são referentes ao complexo descrito pelo arquivo PDB correspondente ao código indicado pelo atributo id. O valor deste atributo corresponde ao código de quatro letras utilizado pelo PDB, sendo único para cada arquivo PDB. O atributo *numberofchains* da *tag* <complex> especifica o número de moléculas (ou *chains*) descritas no arquivo PDB. Um conjunto de elementos (ex: *tag* <temperaturefactor> para *temperature factor*) contém os valores máximo e mínimo de alguns parâmetros para o arquivo PDB correspondente e os elementos delimitados pela *tag* <chain> especificam cada molécula descrita no arquivo PDB. Para cada *chain*, um conjunto de elementos delimitados pela *tag* <residue> especifica a seqüência de resíduos que forma a cadeia protéica. Uma vez que todos os parâmetros apresentados pelo JPD são mapeados por resíduo, no documento XML, de forma similar, eles aparecem especificados sob elementos delimitados pela *tag* <residue>.

```
<?xml version="1.0"?>
<pdb id="1cho.pdb" model="0">
  <doubleoccupancy value="0"/>
  <complex numberofchains="2">
    <temperaturefactor min="2.160000" max="64.139999"/>
    <accocomplex min="0.000000" max="213.412994"/>
    <accisolation min="0.000000" max="213.412994"/>
    <electrostaticpotential min="-105.126167" max="43.930847"/>
    <chain id="E" dna="0" numberofresidues="245">
      <residue name="CYS" number="1" ifr="0">
        <pdbnumber value="0001" />
        <tempfactor tf="17.72" ca="17.58" max="19.13"/>
        <hydrophobicity value="1.28"/>
        <relentropy value="0.00"/>
        <reliability value="0.00"/>
        <accessibility complex="99.69" isolation="99.69" relative="0.37"/>
        <epotential value="-1.70"/>
        <curvature value="-0.23" lha="1"/>
        <ramachandran phi="999.00" psi="999.00" polygon="-1"/>
        <ss_dssp value="0"/>
        <ss_stride value="0"/>
        <ss_pdb value="0"/>
        <contacts numberofcontacts="3">
          <contact type="1" residuenum="121"/>
          <contact type="1" residuenum="122"/>
          <contact type="14" residuenum="122"/>
        </contacts>
      </residue>
      <residue name="GLY" number="2" ifr="0">
        <pdbnumber value="0002" />
        <tempfactor tf="16.52" ca="15.13" max="17.48"/>
        <hydrophobicity value="0.94"/>
        <relentropy value="0.00"/>
        <reliability value="0.00"/>
        <accessibility complex="0.50" isolation="0.50" relative="0.00"/>
        <epotential value="-2.73"/>
        <curvature value="0.00" lha="1"/>
        <ramachandran phi="67.09" psi="10.02" polygon="7"/>
        <ss_dssp value="0"/>
        <ss_stride value="0"/>
        <ss_pdb value="0"/>
        <contacts numberofcontacts="2">
          <contact type="4" residuenum="120"/>
          <contact type="4" residuenum="207"/>
        </contacts>
      </residue>
    </chain>
  </complex>
</pdb>
```

Fig. 3. Exemplo de documento XML utilizado para troca de dados entre o JPDRreader e o JPD.

## Discussão e trabalhos futuros

A utilização de XML como formato para troca de dados no módulo JPD do SMS permitiu isolar a parte cliente do JPD de questões relativas aos diversos formatos de dados que constituem a *SMS-database*. Isso permite que, no futuro, a forma de armazenamento dos dados utilizada possa ser trocada, por exemplo para sistemas gerenciadores de banco de dados – SGBD (Silberschatz et al., 1997), sem que parte dos programas, especialmente a parte cliente, precise ser alterada.

Além disso, como XML é um formato aberto, tornando pública a DTD, a utilização destes dados por diferentes clientes é imediata. No sentido oposto, para utilizar dados oriundos de provedores de dados que os distribuam em formato XML, utilizando um filtro, eles podem ser facilmente integrados ao conjunto de dados do SMS.

Entretanto, tanto o SMS quanto as tecnologias associadas a XML ainda estão em evolução. Atualmente, XML não tem suporte para mecanismo de herança e não suporta valores numéricos ou tabelas (Achard et al., 2001). A definição do XML *Schema* ainda não está tão difundida quanto a de DTD e as iniciativas de utilização de XML em bioinformática ainda são, de certo modo, dispersas. Ainda não existem formatos XML adotados como padrões pela comunidade de bioinformática. Por outro lado, novas funcionalidades estão previstas para as futuras versões do SMS.

Dessa forma, algumas das atividades futuras relativas à utilização de XML no SMS são:

- acompanhamento mais sistemático da evolução de XML e de suas aplicações na bioinformática, como o PROXIML —*PROtein eXtensible Markup Language* (McArthur, 2002), visando melhor integração com as propostas emergentes;
- aperfeiçoamento da DTD hoje utilizada, inclusive com a possibilidade de definição de um XML *Schema* para a *SMS-Database*;
- Adaptação dos módulos mais antigos do SMS, como *Graphical Contacts* e *IFR Contacts*, para utilização do formato XML;
- Projeto dos novos módulos a serem desenvolvidos para as futuras versões do SMS nos moldes da arquitetura do JPD.

## Referências bibliográficas

- ACHARD, F.; VAYSSEIX, G.; BARILLOT, E. Xml, bioinformatics and data integration. **Bioinformatics**, v.17, n. 2, p. 115-125, 2001.
- ALTSCHUL, S. F.; GISH, W.; MILLER, W.; MYERS, E. W.; LIPMAN, D. J. Basic local alignment search tool. **Journal of Molecular Biology**, v. 215, n. 3, p. 403-410, Oct. 1990.
- ALTSCHUL, S. F.; MADDEN, T. L.; SCHAFFER, A. A.; ZHANG, J.; ZHANG, Z.; MILLER, W.; LIPMAN, D. L. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. **Nucleic Acids Research**, v. 25, n. 17, p. 3389-3402, 1997.
- APACHE SOFTWARE FOUNDATION. **The Apache XML Project**. Disponível em: <<http://xml.apache.org>>. Acesso em: 18 dez. 2002.
- ASN.1. **ASN.1 Information site**. Disponível em: <<http://asn1.elibel.tm.fr/>>. Acesso em: 20 dez. 2002.
- BAIROCH, A.; APWEILER, R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. **Nucleic Acids Research**, v. 28, n.1, p. 45-48, 2000.
- BERKELEY DROSOPHILA GENOME PROJECT. **BDGP home**. Disponível em: <<http://www.fruitfly.org/>>. Acesso em: 20 dez. 2002.
- BERMAN, H. M.; WESTBROOK, J.; FENG, Z.; GILLILAND, G.; BHAT, T. N.; WEISSIG, H.; SHINDYALOV, I. N.; BOURNE, P. E. The Protein Data Bank. **Nucleic Acids Research**, v. 28, p. 235-242, 2000.
- BIOPERL.org. Disponível em: <<http://www.bioperl.org/>>. Acesso em: 18 dez. 2002.
- BIOPOLYMER Markup Language (BIOML) home page. Disponível em: <<http://www.bioml.com/BIOML/>>. Acesso em: 18 dez. 2002.
- BSML – Bioinformatic Sequence Markup Language. Disponível em: <<http://www.bsml.org/>>. Acesso em: 18 dez. 2002.
- CAMPIONE, M.; WALRATH, K. **The Java tutorial: object-oriented programming for the Internet**. Reading, MA: Addison-Wesley, 1997. 831 p. (The Java series).
- EUROPEAN MOLECULAR BIOLOGY LABORATORY. European Bioinformatics Institute. **The EMBL nucleotide sequence database**. Disponível em: <<http://www.ebi.ac.uk/embl/index.html>>. Acesso em: 20 dez. 2002.
- GAME - Genome Annotation Markup Elements: BioJava Wikki. Disponível em: <<http://www.biojava.org/wiki/html/BioJava/GAME.html>>. Acesso em: 18 dez. 2002.
- GORDON, Paul. **XML for molecular biology as compiled by Paul Gordon**. Disponível em: <<http://www.visualgenomics.ca/gordonp/xml/>>. Acesso em: 18 dez. 2002.
- HAROLD, E. R. **XML bible**. Foster City: IDG Books Worldwide, 1999. 1015 p.
- HOFFMANN, K.; BUCHER, P.; FALQUET, L.; BAIROCH, A. The PROSITE database, its status in 1999. **Nucleic Acids Research**, v. 27, n. 1, p. 215-219, 1999.

IBM CORPORATION. **Alphaworks**: XML. Disponível em: <<http://www.alphaworks.ibm.com/xml>>. Acesso em: 18 dez. 2002.

JDOM. Disponível em: <<http://www.jdom.org/index.html>>. Acesso em: 18 dez. 2002.

LIFECODE INC. **AGAVE XML**. Disponível em: <<http://www.animorphics.net/lifesci.html>>. Acesso em: 18 dez. 2002.

MARTHUR, D. A. **An extensible XML Schema definition for automated exchange of protein data**: PROXIML (PROtein eXtensible Markup Language). Disponível em: <<http://www.soe.ucsc.edu/~douglas/proximi/>>. Acesso em: 18 dez. 2002.

MCLAUGHLIN, B. Java and XML. Beijing: O'Reilly, 2000. 479 p.

NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION. **NCBI data in XML**. Disponível em: <<http://www.ncbi.nlm.nih.gov/IEB/ToolBox/XML/ncbixml.txt>>. Acesso em: 18 dez. 2002a.

NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION. **NCBI homepage**. Disponível em: <<http://www.ncbi.nlm.nih.gov>>. Acesso em: 18 dez. 2002b.

ORACLE CORPORATION. **XML home**. Disponível em: <<http://otn.oracle.com/tech/xml/content.html>>. Acesso em: 18 dez. 2002.

SANDER, C.; SCHNEIDER, R. Database of homology-derived protein structures and the structural meaning of sequence alignment. **Proteins: Struct. Function and Genetics**, v. 9, p. 56-68, 1991.

SCHNEIDER, R.; DE DARUVAR, A.; SANDER, C. The HSSP database of protein structure sequence alignments. **Nucleic Acids Research**, v. 25, n. 1, p. 226-230, 1997.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Database system concepts**. 3<sup>rd</sup> ed. New York: McGraw-Hill, 1997. 821 p.

STROUSTRUP, B. **The C++ programming language**. 3<sup>rd</sup> ed. Reading, MA: Addison-Wesley, 1997.

STRUCTURAL BIOINFORMATICS GROUP. **Sting Millennium Suite**. Disponível em: <<http://www.nbi.cnptia.embrapa.br/>>. Acesso em: 21 dez. 2002.

SUN MICROSYSTEMS. **Java technology and XML**. Disponível em: <<http://java.sun.com/xml/>>. Acesso em: 18 dez. 2002.

USERLAND SOFTWARE INC. **XML-RPC home page**. Disponível em: <<http://www.xmlrpc.com>>. Acesso em: 18 dez. 2002.

WANG, L.; RIETHOVEN, J. J. M.; ROBINSON, A. J. XEMBL - distributing EMBL data in XML format. **Bioinformatics**, v. 18, n. 8, p. 1147-1148, 2002.

WORLD WIDE WEB CONSORTIUM. **Extensible Markup Language (XML)**. Disponível em: <<http://www.w3.org/XML>>. Acesso em: 18 dez. 2002a.

WORLD WIDE WEB CONSORTIUM. **HTML 4.01 specification**. Disponível em: <http://www.w3.org/TR/1999/REC-html401-19991224/>>. Acesso em: 20 dez. 2002b.

WORLD WIDE WEB CONSORTIUM. **World Wide Web Consortium [home page]**. Disponível em <<http://www.w3.org/>>. Acesso em: 18 dez. 2002c.

XML examples for biology. Disponível em: <<http://www-alt.pasteur.fr/~letondal/XML/#examples/>>. Acesso em: 18 dez. 2002.

XOM. Disponível em: <<http://www.cafeconleche.org/XOM>>. Acesso em: 18 dez. 2002.

## Comunicado Técnico, 32

### Embrapa Informática Agropecuária Área de Comunicação e Negócios (ACN)

Av. André Tosello, 209  
Cidade Universitária - "Zeferino Vaz"  
Barão Geraldo - Caixa Postal 6041  
13083-970 - Campinas, SP  
Telefone (19) 3789-5743 - Fax (19) 3289-9594  
e-mail: sac@cnptia.embrapa.br

#### 1ª edição

2002 - on-line  
Todos os direitos reservados

## Comitê de Publicações

**Presidente:** José Ruy Porto de Carvalho  
**Membros efetivos:** Amarindo Fausto Soares, Ivanilde Dispatto, Luciana Alvim Santos Romani, Marcia Izabel Fugisawa Souza, Suzilei Almeida Carneiro  
**Suplentes:** Adriana Delfino dos Santos, Fábio Cesar da Silva, João Francisco Gonçalves Antunes, Maria Angélica de Andrade Leite, Moacir Pedroso Júnior

## Expediente

**Supervisor editorial:** Ivanilde Dispatto  
**Normalização bibliográfica:** Marcia Izabel Fugisawa Souza  
**Capa:** Intermídia Publicações Científicas  
**Editoração Eletrônica:** Intermídia Publicações Científicas