인터넷 데이타베이스

Chapter 22

HTML

- 간단한 마크업 언어(markup language)
- 텍스트에 태그(tags)라는 명령어로 주석을 다는데, 태그는 보통 시작 태그와 종료 태그로 짝을 이룬다.

HTML 의 예: 서적 목록

```
<HTML><BODY>
Fiction:
<UL><LI>Author: Milan Kundera</LI?
        <LI>Title: Identity</LI>
        <LI>Published: 1998</LI>
</UL>
Science:
<UL><LI>Author: Richard Feynman</LI>
        <LI>Title: The Character of Physical Law</LI>
        <LI>Hardcover</LI>
</UL></BODY></HTML>
```

웹 페이지에 DB의 내용을 넣기

- 데이타베이스 질의의 결과를 포함시켜 웹 페이지를 만드는 방법은?
  - DB와 교신할 프로그램용 프로세스를 웹 서버가 만든다.
  - 웹 서버는 CGI (Common gateway interface)를 통하여 이 프로그램과 교신한다.
  - 프로그램은 DB로부터 받은 내용을 가지고 결과 페이지를 생성한다.
  - 기타 프로토콜: ISAPI (Microsoft Internet Server API), NSAPI (Netscape Server API)

응용 서버(Application Servers)

- CGI의 경우, 페이지 호출이 발생할 때마다 매번 새로운 프로세스가 탄생된다 : 매우 비효율적
- 응용 서버: 웹 서버와 응용간을 잇는 일종의 소프트웨어
- 기능:
  - 성능을 위하여, 쓰레드나 프로세스들을 미리 만들어 놓는다.
  - 데이타베이스 접속을 폴화(pooling : 기존의 접속 집단을 재사용)
  - 이질적인 데이타 원천 통합
  - 여러 데이타 원천에 대한 트랜잭션을 관리
  - 세션 관리

기타 서버측 처리

- Java Servlet: 서버측에서 도는 Java 프로그램으로서, 잘 정의된 API를 통하여 서버와 교신.
- JavaBeans: Java로 작성된, 재사용가능형 소프트웨어 구성요소.
- Java Server Page 와 Active Server Page: 웹 페이지 내에 포함된 코드로서, 웹 서버가 인터프리트한다.

HTML 이후: XML

- Extensible Markup Language (XML): "확장가능형 HTML"
- SGML과 HTML이 합쳐진 결과: SGML의 표현력과, HTML의 단순성
- 새로운 마크업 언어 정의 가능 : document type declaration (DTD)

XML: 언어 구성

- Element(엘리먼트)
  - XML의 주요 구성 단위
  - 시작 태그와 종료 태그
  - 엄격한 내포 규칙
- Element는 상세 정보를 표현할 애트리뷰트들을 가질 수 있음.
- Entity: 매크로와 유사하며, 공통 내용물을 표현
- Comments(주석)
- Document type declaration (DTD)

XML로 표현한 서적 목록

```
<?XML version="1.0" standalone="yes"?>
<!DOCTYPE BOOKLIST SYSTEM "booklist.dtd">
<BOOKLIST>
<BOOK genre="Fiction">
  <AUTHOR>
    <FIRST>Milan</FIRST><LAST>Kundera</LAST>
  </AUTHOR>
  <TITLE>Identity</TITLE>
  <PUBLISHED>1998</PUBLISHED></BOOK>
<BOOK genre="Science" format="Hardcover">
  <AUTHOR>
    <FIRST>Richard</FIRST><LAST>Feynman</LAST>
  </AUTHOR>
  <TITLE>The Character of Physical Law</TITLE>
</BOOK></BOOKLIST>
```

XML: DTD

- DTD는 원하는 문서에서 허용될, 엘리먼트, 애트리뷰트, 개체를 정의한, 규칙 집합임.
- DTD는 없지만 내포 방식이 적절한 XML문서를 well-formed 문서라고 한다.
- DTD가 있고 그 DTD의 규칙을 따르는 XML문서를 valid문서라고 한다.

DTD의 예

```
<!DOCTYPE BOOKLIST [
  <!ELEMENT BOOKLIST (BOOK)*>
  <!ELEMENT BOOK (AUTHOR, TITLE, PUBLISHED?)>
  <!ELEMENT AUTHOR (FIRST, LAST)>
    <!ELEMENT FIRST (#PCDATA)>
    <!ELEMENT LAST (#PCDATA)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT PUBLISHED (#PCDATA)>
  <!ATTLIST BOOK genre (Science|Fiction) #REQUIRED>
  <!ATTLIST BOOK format (Paperback|Hardcover) "Paperback">
]>
```


도메인 특유(Domain-Specific) DTD

- 특정한 분야에서, 이질적인 원천간의 자료교환을 위하여 개발한 특수 DTD
- 예: Mathematical Markup Language (MathML)
  - 수학적인 문서자료를 웹상에 부호화
  - HTML에서: <IMG SRC="xysq.gif" ALT="(x+y)^2">
  - MathML에서:

```
<apply> <power/>
  <apply> <plus/> <ci>x</ci> <ci>y</ci> </apply>
  <cn>2</cn>
</apply>
```

XML-QL: XML 데이터 질의어

- 목표: XML문서 조작용, 고수준, 명세형 언어
- 아직 표준은 없음
- XML-QL 질의의 예:

```
WHERE
  <BOOK>
    <NAME><LAST>$1</LAST></NAME>
  </BOOK> in "www.booklist.com/books.xml''
CONSTRUCT <RESULT> $1 </RESULT>
```


XML-QL (계속)

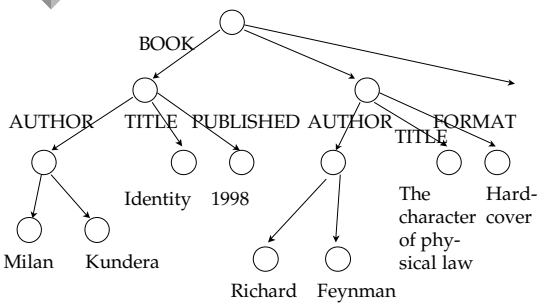
심화 예제:

```
WHERE <BOOK>  $b  <BOOK> IN
    "www.booklist.com/books.xml",
  <AUTHOR> $n </AUTHOR>
  <PUBLISHED> $p </PUBLISHED> in $e
CONSTRUCT
  <RESULT>
    <PUBLISHED> $p </PUBLISHED>
      WHERE <LAST> $l </LAST> IN $n
      CONSTRUCT <LAST> $l </LAST>
  </RESULT>
```


준 구조화된(Semi-structured) 데이타

- 일부 구조화된 데이타
- 준 구조화된 데이타용 데이타 모델이라면 모두 일종의 레이블 달린 그래프를 이용한다.
- 여기에서는 object exchange model (OEM)을 설명함:
  - 객체지향 3쌍(레이블, 타입, 값) 형태
  - 복합 객체는 더 작은 객체들로 계층적으로 분해된다.

예: OEM으로 표현한 서적 목록



덱스트 탐색을 위한 인덱싱

- 텍스트 DB: 텍스트 문서들의 모임
- '키워드 탐색'이 중요한 질의 형태임:
  - 불리언 질의(Boolean query): 질의 항들을 AND, OR ,NOT으로 연결. 결과는 불리언 식을 만족하는 문서들의 목록.
  - 순위형 질의(Ranked query): 결과는 "관련성"정도에 따라 등수를 매긴 문서목록
  - IR: 정밀도(Pricision: 검색된 것들 중 실제 관련있는 문서의 백분율)와 조회도(recall: 관련있는 것들 중 실제 검색된 문서의 백분율)

역 화일(Inverted File)

- 가능한 질의 항 별로, 그 단어를 포함하는 문서 식별자의 리스트 (역 리스트) 를 저장.
- 질의 수행법: 역 리스트들을 교집합/합집합.
- 예: Agent AND James

RID	문서
1	Agent James
2	Mobile agent

단어	역 리스트
Agent	<1,2>
James	<1>
Mobile	<2>

시그너쳐 화일(Signature File)

- 각 문서마다 데이타 엔트리를 하나씩 만든 인덱스 구조.
- 해시 함수로 단어들을 비트 벡터로 해싱함.
- 해싱한 단어들을 OR하여, 해당 문서의 데이타 엔트리(문서 시그너쳐)를 구성.
- S2&S1=S2이면 시그너쳐 S1은 시그너쳐 S2에 부합된다고 해석.

시그너쳐 파일: 질의 수행법

- 단어의 논리곱 형태 불리언 질의:
  - 질의 시그너쳐 Sq 생성.
  - 모든 문서 시그너쳐들을 검조.
  - 시그너쳐 S 가 Sq에 부합하면, 해당 문서를 검색해 와서 적중 착오인지 검사.
- k 단어의 논리합 형태 불리언 질의:
  - k개의 질의 시그너쳐 S1, …, Sk 생성.
  - S1, …, Sk중 어느 하나에도 부합하는 문서 시그너쳐 탐색.
  - 적중 착오인지 검사.

시그너쳐 화일: 예

단어	해시
Agent	1010
James	1100
Mobile	0001

RID	문서	시그너쳐
1	Agent James	1110
2	Mobile agent	1011

요약

- 웹 상에 DB 내용을 보이려면 CGI-scripts, Servlets, ASP, JSP 등의 서버측 처리 필요.
- XML은 새로 등장하고 있는 문서 기술 표준으로서, 임의의 DTD정의 가능. XML 문서용 질의어로 XQL 등이 등장중.
- 웹 상의 텍스트 데이타가 급증함에 따라 텍스트 DB의 중요도 부각.

  불리언 질의어는 역 화일로 효율적인 처리 가능. 순위형 질의 처리는 더 복잡함.