

SuperSQL처리계를 이용한 효과적인 미디어 출판

김태석^{1†} · 신상규² · 토야마모토미치²

Efficient Media Publication Using SuperSQL Processor

Kim Tai-Suk · Shin Sang-Gyu · Toyama Motomichi

ABSTRACT

In this paper, we propose an effective use of SuperSQL for efficient media publication that utilizes relational databases. SuperSQL is an enhanced SQL processing system that converts data stored in relational databases into a variety of applied formats, such as XML and HTML, as a result of query. In order to integrate various media generators in SuperSQL query processor, we discuss methods to prepare both for new media promptly and for efficient media publication processing system. This research makes two main contributions: (1) it is a study about methods to integrate media generators by combining common processes in SuperSQL query processor, and (2) it introduces the trinity data model, which divide the media generating processes into three aspects, which are value, structure and medium abstraction. We also show the good performance and feasibility of our system.

Key words : Database, Database publishing, Media publishing

요약

본 논문은 관계데이터베이스시스템을 이용하여 효율적인 미디어 출판을 위한 SuperSQL의 효과적 이용방법에 대해 논하고 있다. SuperSQL은 관계데이터베이스에서의 검색결과를 XML이나 HTML과 같은 다양한 미디어 형태로 변화시킬 수 있도록 SQL을 확장한 언어와 이를 처리하는 시스템이다. 본 논문에서는 SuperSQL의 다양한 미디어 출판시스템을 통합하기 위해 효과적인 미디어 출판이 가능한 새로운 미디어 출판 처리 방법에 대해 논의한다. 본 논문에서 제시하는 것은 첫 번째로 SuperSQL 질의처리 프로세서의 다양한 미디어 출판 프로세스들 간의 공통된 처리 프로세서를 통합함으로써 통합된 미디어 생성 프로세서를 생성하는 방법을 제안하고 있다. 다른 한 가지는 Trinity Data Model개념을 기반으로 하는 미디어 출판 프로세서를 각각의 특징에 따라 값과 구조, 미디어 추상태이블로 분리하여 처리하는 방법을 제안하였으며 또한 실험을 통해 본 제안 시스템의 유효성을 검증하였다.

주요어 : 데이터베이스시스템, 데이터베이스출판, 미디어출판, 시스템통합, 확장SQL

1. 서론

현대 정보사회에서 IT기술의 발전은 데이터의 폭발적인 증가를 가져왔다. 이러한 변화에 따라 사용자에게 제공되는 수많은 데이터로부터 실제 유용한 정보를 추출하고 관리하는 데이터 조직화와 대용량 데이터를 관리할 수 있는 시스템인 데이터베이스시스템의 중요성은 더욱 높

아지게 되었다. 데이터베이스시스템은 여러 어플리케이션 사이에 정보를 공유할 수 있는 중요한 리포지터리(repository)의 역할을 한다. 예를 들어 같은 데이터를 이용해 리포트나 웹페이지, XML문서 등의 여러 가지 형태로 활용할 수 있다. 최근의 인터넷과 디지털방송, 휴대폰과 같은 IT기술발전에 의한 세계적인 정보의 디지털화는 개인 및 기업에게 정보의 관리와 다양한 형태로의 데이터출판 기술의 중요성을 증가되고 있다.

SuperSQL^[1-5]은 데이터베이스 질의 언어인 SQL을 확장하여 데이터베이스출판을 목적으로 하는 시스템이다. SuperSQL은 데이터베이스로부터의 결과를 특정하게 그룹화한 구조로 출판할 수 있는 부호와 출판할 미디어 지

2006년 2월 7일 접수, 2006년 3월 7일 채택

¹⁾ 동의대학교 컴퓨터소프트웨어공학과

²⁾ 일본 KEIO대학 이공학연구과

주 저자 : 신상규

교신저자 : 김태석

E-mail; tskim@deu.ac.kr

정어, 특정한 효과를 줄 수 있는 태그정보를 지정함으로써 HTML, XML, LaTeX, PDF와 같은 다양한 형태의 데이터로 변환할 수 있다. 이와 같은 다양한 미디어 출판에 대응할 수 있도록 SuperSQL은 각각의 미디어 생성 처리 부분을 가지고 있다.

본 논문에서는 SuperSQL시스템을 Trinity Data Model개념을 기본으로 효율적인 미디어 생성 프로세서를 제안한다.

2. 관련연구

최근 주목받고 있는 XML에서는 AT&T 연구소의 SilkRoute^[6,7]나 XPERATO^[8,9] (IBM 연구소)프로젝트가 관계데이터베이스시스템에서 XML문서로의 효과적인 출판기술을 제안하고 있다. SQL Server나 Oracle, IBM의 DB2, Microsoft Access같은 상업 데이터베이스 시스템들도 XML문서를 출판하는 기능을 지원하고 있으며, 그 외에 많은 연구 프로젝트나 소프트웨어가 데이터베이스 시스템으로부터 HTML 문서를 출판하는 기술을 제안하고 있다. 하지만 이러한 연구의 대부분은 XML이나 HTML과 같은 특정한 미디어 포맷으로 한정된다. SuperSQL은 기존의 연구와 다음과 같은 차이점이 있다.

- 1) 다양한 미디어를 지원하고 있으며, 새로운 미디어의 추가가 가능하다.
- 2) 사용자 정의에 의해 관계데이터베이스의 검색결과를 그룹화한 구조로 변경할 수 있다.
- 3) Target Form Expression(TFE) 기술을 이용해 멀티 레벨, 멀티페이지 문서를 작성할 수 있다.

본 논문은 이러한 기능의 지원뿐만 아니라 보다 효과적인 출판 기능을 가진 시스템의 설계를 목표로 하고 있다. 또한 SuperSQL은 SQL을 확장해 One-Source Multi-Use를 실현한 질의(Query) 언어다.

SuperSQL의 질의어는 SQL의 SELECT구를 ENERATE <medium><TFE> 구문으로 확장한 GENERATE구로 바꿔놓은 것이다. 여기서 <medium>은 HTML, XML, PDF 같은 출력형식을 지정한다. <TFE>는 SQL에 있어서의 Target List의 확장인 Target Form Expression을 표현하는 것으로 Connector, Repeater등의 레이아웃을 지정하는 연산자를 가진 식이다. r_i 를 릴레이션, P를 SQL의 조건으로 하면, SuperSQL의 질의어는 다음과 같이 나타낼 수 있다.

GENERATE <medium><TFE>

FROM r_1, r_2, \dots, r_n
WHERE P

TFE(Target Form Expression)은 SQL 질의어에서 SELECT구에 입력하는 속성이름을 레이아웃 지정연산자인 연결연산자(Connector), 반복연산자(Groupier)와 조합해, 미디어 출력결과와 구조를 사용자가 지정하는 방식이다. 또한, “@{Decorator Expression}”으로 표현하는 장식자(Decorator)에 의해 표의 셀의 넓이나 배경색 등의 상세한 지정이 가능하다.

3. 미디어 출판시스템

본 논문에서는 제안하는 새로운 시스템은 각각 존재하던 미디어처리시스템을 통합하여, 통합된 시스템이 새로운 미디어에 대해 유연히 대처할 수 있도록 하는 것을 목표로 한다.

3.1 미디어 출판시스템의 문제점

SuperSQL 처리프로세스는 그림 1과 같이 파서(Parser), List Constructor, Code Generators로 구성된다.

SuperSQL 질의어는 그림 1의 Parser에서 일반적인 SQL문과 Layout Expression으로 구문 분석되어 DBMS Interface에서 검색결과를 받는다. List Constructor는 데이터베이스의 검색결과를 Layout Expression에 따라 계층적인 구조로 변화시킨다. 마지막으로 Code Generator가 유저가 지정한 미디어의 소스코드를 처리 결과로 생성

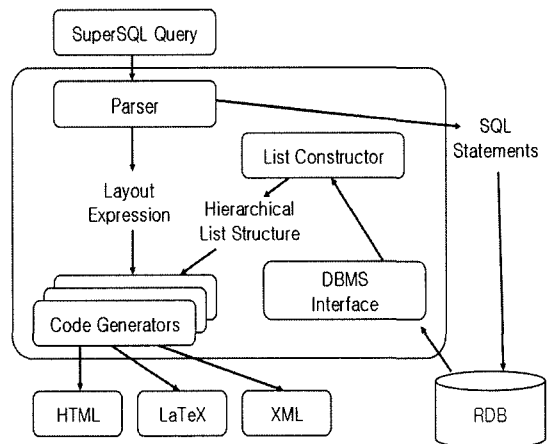


그림 1. SuperSQL 처리 시스템

하게 된다. 다양한 미디어를 생성하기 위해 각 타깃미디어에 대한 미디어 프로세서가 존재하며 또한, 유저가 새로운 미디어를 생성하기 위해서는 그 미디어에 대한 새로운 미디어 프로세서를 추가해야 하는데 이것은 각각의 미디어가 각자 특징적인 구성자(Constructor)나 장식자(Decorator)를 가지고 있기 때문이다. 이러한 구조는 각각의 미디어가 가지는 고유한 처리(예를 들어 HTML의 Hyper Link 기능)를 하기에는 좋지만, 공통된 구조 처리 부분과 같이 통합 처리할 수 있는 부분을 반복해서 사용해야 하는 비효율적인 처리 구조를 가진다. 또한 유저는 각각의 미디어 특징에 맞는 SuperSQL 질의어를 작성해야 하므로 공통적인 사용자 인터페이스를 제공할 수 없다. 즉, 같은 문자 사이즈 지정에서도 유저는 HTML을 작성하고 싶은 경우는 "A@{size=5}", LaTeX를 지정할 경우는 "A@{size=Large}"와 같이 각 미디어 고유의 Decorator를 정의할 필요가 있다. 이것은 유저가 각 미디어 고유의 Decorator를 알지 못할 경우 의도한 레이아웃으로의 출력이 불가능하게 된다.

3.2 Trinity Data Model 개념

모든 출판 미디어는 미디어 고유의 구성자(Constructor)와 값(Value) 및 구조(Structure)를 가지고 있다. 따라서 관계데이터베이스에서 타깃미디어로 변환하는 경우에 이러한 Constructor나 계층구조를 추가해야 한다. 본 논문에서 제안하는 시스템은 그림 2와 같이 관계데이터베이스에서 검색된 결과(Value)와 SuperSQL 질의어에서 추출된 구조정보(TFE), 각 타깃미디어에 종속적인 특별한 Constructor의 세 부분으로 구성되는 Trinity Data Model 개념을 기본으로 하고 있다.

이 개념을 기본으로 본 논문에서 제안한 시스템은 각 미디어 생성부의 타깃미디어 종속적인 미디어 구성자와 장식자(이후 미디어구조화정보)를 추가하는 처리프로세

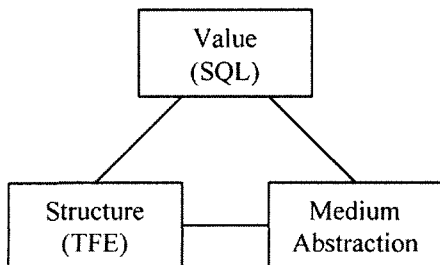


그림 2. The Trinity Data Model

서에서 미디어 구조를 구성하는 프로세서를 독립적으로 운용할 수 있는 방법을 제안한다. 즉, 각 타깃미디어별로 존재하는 각 미디어 처리 프로세서를 하나의 통합된 구조로 설계, 각 미디어 종속적인 처리는 메타데이터를 이용해 정의하는 것이다. 이러한 방법의 장점은 다음과 같다.

- 1) 통합된 미디어 생성처리부의 작성과 관리가 가능하다. TFE 표현의 추가 등에 의한 기능의 확장이 필요할 경우, 기존의 시스템에서는 각 미디어 별로 기능이 추가될 때 마다 생성처리부의 작성이 필요하지만 제안 시스템에서는 미디어 생성처리부가 하나로 통합되면서 새롭게 추가할 필요가 없어진다.
- 2) 각 미디어 종속적인 미디어구조정보를 구조생성처리계로부터 분리함으로써 새로운 미디어에 기존의 시스템보다 빨리 대응할 수 있다. 새로운 미디어가 추가될 경우 구조생성부에 Hard-Coding하지 않고 메타데이터파일에 동적으로 연결되면서 미디어의 추가, 기능의 추가변화에 효율적으로 대응할 수 있게 된다.

그림 3은 본 논문에서 제안하고 있는 미디어 출판시스템의 구성을 나타낸다. 본 논문에서 제안하는 시스템은 Trinity Data Model 개념의 구조(Structure) 부분인 미디어 생성자(Media Generator), 장식자처리계(Decorator Processor)와 TFE 처리계(TFE Processor), 미디어구성자처리(Media Constructor)로 구성되는 Media Abstraction 부분, 값(Value) 부분에 해당하는 관계데이터베이스 인터페이스(DBMS Interface)의 세 부분으로 크게 나누어져 Trinity Data Model 개념과 같이 독립적으로 처리가 가능하도록 구성되어 있다.

간단하게 처리의 흐름을 설명하면, TFE 처리계가 구조 정보(TFE 정보)를, 장식자처리계는 사용자가 지정한 장

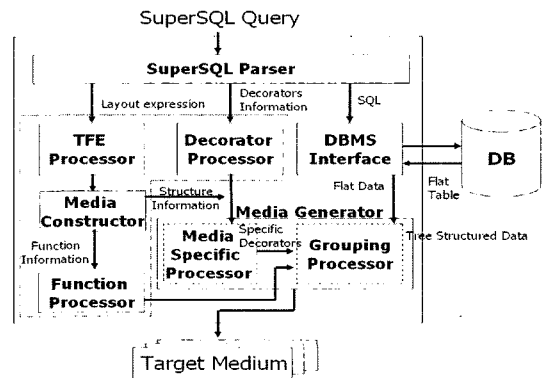


그림 3. Trinity Data Model 개념의 미디어출판 시스템의 구성

식자정보를, 데이터베이스인터페이스는 SuperSQL 질의어에서 추출된 SQL문을 데이터베이스를 통해 검색결과로 받는다. TFE 처리계는 사용자로부터 지정된 구조정보로부터 타깃미디어에 관한 레이아웃표현(Layout Expression)을 작성한다. 레이아웃표현은 그룹 처리에 참조되는 그룹정보를 가진다. 미디어구성자는 메타데이터로부터 타깃미디어 고유의 구성자(태그집합 혹은 고유의 구조정보)를 추출한다. 이때, 타깃미디어 고유의 함수(HTML의 Hyper Link등)가 발견되면 함수처리계(Function Processor)를 이용해 각 미디어 고유의 함수를 처리한다. 장식자처리계는 TFE처리계로부터의 타깃미디어 정보를 기본으로 미디어메타데이터로부터 타깃미디어 고유의 구성자를 검색해 타깃미디어를 꾸민다. 미디어생성자에 특수미디어처리계(Media Specific Processor)와 그룹화처리계(Grouping Processor)가 포함되어 있는 것을 주목할 필요가 있다. 특수미디어처리계는 타깃미디어 종속적인 특수한 처리를 담당하는 프로세스이다. 이 처리부분에서 전처리된 타깃미디어 고유의 미디어구조화정보(구성자장식자)·함수등이 통합적으로 추가된다. 그룹화처리계는 관계데이터베이스의 검색 결과로부터 TFE에 의한 그룹화(계층구조화)를 수행하는 처리부이다. 미디어생성자의 그룹화처리계는 장식자처리계와 TFE처리계, 미디어구성자로부터 처리된 정보를 통합한 특수미디어처리계의 정보를 참조하면서 데이터베이스로부터의 결과 테이블을 구조화한다.

3.3 검색결과와 그룹화 구조작성

DBMS 인터페이스를 통해서 검색된 관계데이터베이스의 테이블 형태의 결과는 사용자가 정의한 SuperSQL의 TFE에 의한 계층구조로 변경할 필요가 있다. 정의된 계층구조로 변경하기 위해 SuperSQL 프로세스에서는 그룹화처리를 한다. 그룹화처리는 SuperSQL시스템의 가장 중요한 특징의 하나이다. 다음과 같은 SuperSQL 질의어의 경우, 그림 4와 같이 계층화(그룹화)된 결과를 출판한다.

```
Generate HTML [ P.age@{size=5},
[P.name@{bgcolor=blue, size=7},
[C.model, {c.co.or}!,
M.maker@{color=blue}]]]!
```

이러한 계층화된 표를 작성하기 위해, SuperSQL프로세서에서는 SuperSQL 질의어에서 추출한 TFE표현(TFE

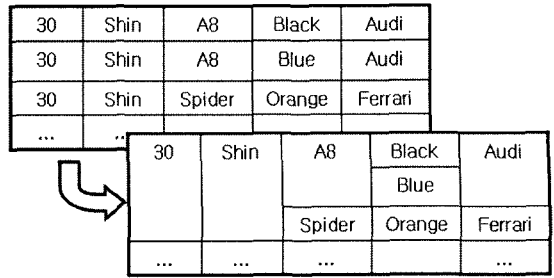


그림 4. SuperSQL의 그룹화처리

Expression)을 참조하여 레이아웃표현(Layout Expression)을 작성한다. 레이아웃표현은 TFE로부터 작성된 그룹화정보로, 프로세스는 이 레이아웃표현을 참조해 계층구조화를 수행한다. 다음은 TFE표현과 이 TFE에 의해 작성되는 레이아웃표현을 나타내고 있다. 레이아웃표현은 계층화 레벨인 Connector(C1, C2, C3), Grouper(G1, G2, G3)의 정보를 포함한 구조생성 정보 리스트이다.

```
TFE : [ A % [ B, C, D ]! ],
Layout Exp.:(G1(C3 1(G2(C1 2 3 4))))
```

이 레이아웃표현을 참조해 작성되는 실제의 계층구조리스트(Hierarchical List)와 관계데이터베이스로부터의 테이블형태의 결과를 표 1에 나타낸다.

본 시스템에서는 계층구조리스트를 XML View형태로 생성한다. 이와 같이 중간 결과에 XML 태그를 조립할 수 있는 시스템설계는 계층구조를 작성할 때 XML 태그 대신에 타깃미디어 고유의 미디어구조화정보를 조립하는 것도 가능하다. 따라서 본 논문에서는 계층구조리스트의 중간 결과를 생성할 때 논문^[4]의 XML태그 추가 위치에 타깃미디어 고유의 구조화정보를 조립하는 방법으로 중간 결과를 생성 후 다시 미디어구조화정보를 추가하는 두 단계의 처리를 한 과정으로 통합할 수 있도록 시스템을

표 1. 검색된 결과의 계층구조화

Flat List	((A1 B11 C11 D11) (A1 B12 C12 D12) (A2 B21 C21 D21) (A2 B22 C22 D22))
Hierarchical List	((A1((B11 C11 D11) (B12 C12 D12)))) (A2((B21 C21 D21) (B22 C22 D22))))

설계함으로써 처리시간의 단축을 유도했다. 본 대안시스템에서는 SuperSQL 질의어에서 XML 형태의 스키마를 지원하고 있다. 이 스키마를 기본으로 출판할 미디어 고유의 전처리된 미디어구조화정보부가 추가된 미디어 구조 스키마를 그림 5와 같이 작성한다. TFE 스키마와는 달리 미디어구조 스키마는 <CONSTRUCTOR> 태그에 전처리된 타깃미디어 고유의 미디어구조화정보가 있는 것을 알 수 있다. 그룹화처리계는 이 전처리된 미디어 구조 스키마를 참조하면서 계층구조를 수행한다.

3.4 타깃미디어 구문분석

본 논문에서 제안하는 시스템의 특징은 각각의 미디어 생성부로부터 각 타깃미디어에 종속적인 고유의 미디어 구조화정보를 분리하여 처리하는 것이다. 이 분리방법에 의해 통일된 미디어 생성부가 효과적으로 각 타깃미디어 고유의 미디어구조화정보를 추가할 수 있도록 제어하기 위해서는 각 타깃미디어의 구조의 특징을 분석해 볼 필요가 있다. 다음과 같은 SuperSQL 질의어에 의하면 표 2와 같이 계층화된 테이블이 생성된다.

```
Generate HTML [ C.name, [C.color]!]!
Form car C
```

표 2. 생성되는 테이블의 예

Ferrari	Red
	Blue
Benz	White
	Black
...	...

```
<TYPE>Horizontal_Repeater</TYPE>
- <CONSTRUCTOR>
  <![CDATA[ <td><table border = 1>
  ]]>
</CONSTRUCTOR>
- <DECORATOR>
- <CONSTRUCTOR>
  <![CDATA[ <font size="7" color="red"> ]]>
</CONSTRUCTOR>
</DECORATOR>
- <CONNECTER>
  <TYPE>Horizontal_Connector</TYPE>
- <CONSTRATOR>
  <![CDATA[ <td><table border = 1> ]]>
</CONSTRATOR>
- <COLUMN>
  <TYPE>COLUMN</TYPE>
- <DECORATOR>
  <COLOR>BLUE</COLOR>
```

그림 5. 미디어구조 스키마

이러한 테이블을 작성하기 위해서 작성되는 HTML과 LaTeX의 미디어구조화정보부가 붙여진 소스구조를 엘리먼트(ELEMENT)단위로 분석한 것이 그림 6이다. 그림 6을 보면 TFE에 의한 계층화 정의에 의해 붙여지는 구성자는 타깃미디어의 종류에 상관없이 비슷한 패턴을 가지고 있는 것을 알 수 있다. 즉, 각 미디어를 구성하기 위해 붙여지는 구성자를 보면 모든 구조는 Value(그림 6의 Value)와 이 Value를 타깃미디어 고유의 구성자가 Value의 전(Preamble)과 뒤(Trailer)로 구성된 구조화할 데이터(Value)의 범위를 알 수 있도록 하나의 쌍이 되어 붙여지고 있다(그림 6의 Pair of Constructor). 그림 6으로부터 알 수 있는 것은 각 TFE표현에 의해 붙여질 수 있는 구성자가 타깃미디어의 고유의 구성자에 의존할 뿐, 붙여지는 장소, 즉 패턴은 대체로 비슷하다는 것을 알 수 있다. 즉, 구성할 Value의 범위를 알 수 있도록 구성자를 추가하는 것이 기본 패턴이 된다. 이 기본패턴을 확장하는 방법으로 본 논문에서 제안한 시스템에서 실행되는 미디어구조화정보의 추가방법에 대해서 설명한다.

3.5 미디어 구조화 정보의 추가

그림 6의 HTML과 LaTeX의 예로부터 분석해보면 표 3과 같은 기본 구성자를 추출한 테이블을 만들 수 있다. 이러한 테이블을 본 논문에서는 추상테이블(Abstraction Matrix)이라고 말한다. 추상테이블은 Trinity Data Model 개념의 Medium Abstraction에 해당하는 부분으로, 본 대안시스템에서는 각 미디어 고유의 구조화정보를 정의하고 있다. 이 추상테이블에 정의된 각 타깃미디어를 구성하기 위해 필요한 구조화정보를 동적으로 참조하는 방법으로 미디어 출판이 실행된다. Dim은 차원으로 SuperSQL로 정의되는 Grouper(G1, G2, G3)를 나타낸다. 추상테이블은 실제, 구성자의 메타데이터로써 그림 7과 같은 형태로 저장된다. 본 논문에서 제안하는 시스템에서는 추상테이블을 가리키는 기본적인 구성자 단위를 엘리먼트(ELEMENT)라고 하며 다음과 같이 정의한다. Value를 D, 구성자 집합을 T, T의 요소를 t, t_i는 Value의 앞에 붙일 수 있는 구성자 요소(Preamble)로, te는 뒤에 붙여지는 요소(Trailer)로 정의한다. 기본적인 구성자가 붙여지는 결과 스트림 엘리먼트 E는 다음과 같이 정의한다.

$$E = \{t_i | t_i \in T\} + \{d(x, e, y) | x \in D, y \in E\} + \{t_e | t_e \in T\}$$

기본적인 엘리먼트 스트림 E는 엘리먼트 자신을 요소로서 가질 수도 있다. 실제 출판되는 미디어를 구성하기

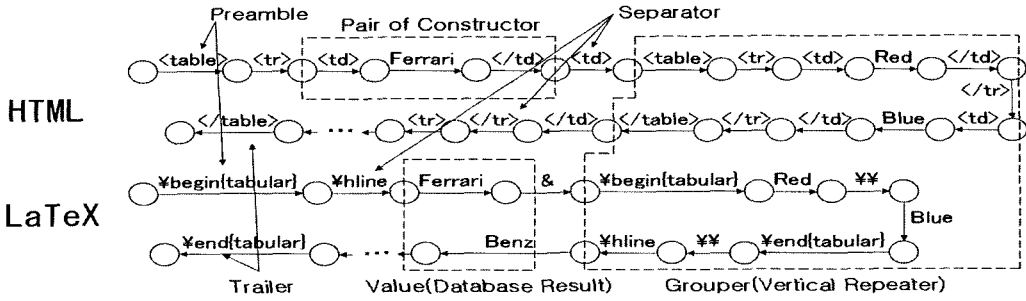


그림 6. 타깃미디어 구문분석

표 3. 추상테이블 예

1) LaTeX

Dim	Preamble	Separator	Trailer
0	\documentstyle{article} \begin{document}		\end{document}
1	\begin{tabular}{cc}	&	\end{tabular}
2	\begin{tabular}{c}	\\	\end{tabular}
3		\vfill \eject	

2) HTML

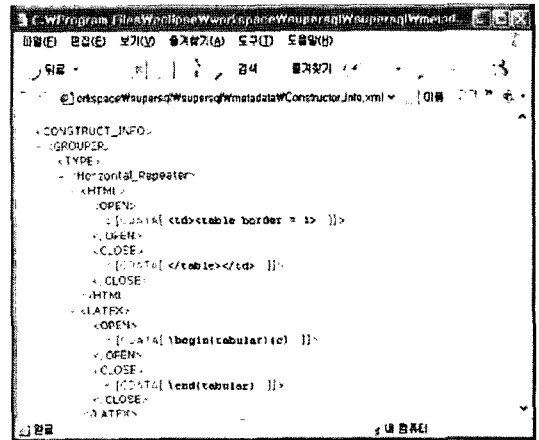
Dim	Preamble	Separator	Trailer
0	<html><body>		</body> </html>
1	<table><tr><td>	</td><td>	</td></tr> </table>
2	<table><tr><td>	</td></tr> <tr><td>	</td></tr> </table>
3		

위해서 붙여질 수 있는 구성자는 표5에 있는 분리연산자 (Separator) *s*와 엘레먼트와 엘레먼트를 연결하는 연결요소 *c*, 엘레먼트를 감싸는 Wrapper *w*등을 생각할 수 있다. 이러한 요소를 추가하기 위해 엘레먼트 *X*를 다음과 같이 정의했다.

$$X = \{w_s | w \in T\} + \{e(x) | x \in E\} + \{s | s \in T\} + \{w_e | t \in T\} + \{c | c \in T\}$$

예를 들어, 연결요소는 HTML의 경우 <hr>태그 같은 것이 해당될 수 있다. Wrapper요소는 HTML문서의

그림 7. Constructor 메타데이터파일



Hyper Link처리(SuperSQL의 "% 연산자에 의해 정의된다)를 하는 경우에 붙여지는 태그 추가에 사용될 수 있다.

SuperSQL에서 A@{Decorator Expression}에 의해 지정되는 장식자를 처리하는 방법은 HTML과 LaTeX에 붙여지는 장식자는 엘레먼트 E와 X의 정의를 확장함으로써 제어할 수 있다. 구성자의 안쪽에 반복적으로 추가되는 장식자를 처리하기 위해 요소 T를 HTML문서 구성을 기본으로 좀 더 세분화하여 제어할 수 있도록 다음과 같이 확장하였다. HTML문서작성을 예로 "<"태그를 Open Tag *t_o*, "</" 태그를 Close Tag *t_c*, "table" 같은 태그 이름을 *t_m*로, 반복되는 속성을 *a(x)*로 정의하면, 확장된 요소 T는 다음과 같이 정의할 수 있다.

$$T = \{t_o, t_c | t \in T\} + \{t_m | t \in T\} + \{a(x) | x \in A\} + \{t_c | t \in T\}$$

3.6 통합된 인터페이스의 제공

본 논문에서 제안하는 미디어 출판시스템에서는 사용자가 특정 미디어의 장식자를 알지 못하더라도 프로세스가 자동으로 출판미디어에 해당하는 장식자를 추가하도록 했다.

그림 8과 같이 유저가 HTML을 출판할 때 HTML에는 없는 “size=Large”와 같은 장식자를 사용했을 경우도, 처리프로세스는 먼저 ①과 같이 “size”가 “Large”인 테이블을 검색한다. 검색된 결과와 같은 “Large” 값을 가지는 다른 “Actual Value”칼럼을 검색 (②, ③), 그에 해당하는 “User Definition”정의 칼럼에서 HTML에 해당하는 값으로 바꿔준다(④, ⑤).

4. 실험결과

본 논문에서 제안한 시스템과 종래의 시스템의 소스코드의 양을 비교해 보면 표 4와 같이 기존시스템과 공통처리 부분에 있어서는 약간 많은 정도지만 각각의 미디어를 생성하기 위한 Generator부분은 기존시스템에 비해 10% 이하로 현저히 줄어들었음을 알 수 있다.

표 4의 소스코드의 양도 실제 프로그램에 Hard-Coding된 것이 아니라 추상테이블에 동적으로 정의된 양이다.

표 4를 통해 본 논문에서 제안하고 있는 시스템이 새로운 미디어의 추가에 보다 효율적으로 대응된다고 판단할 수 있다. 또한 그림 8에서는 통합된 처리로 인해 각각 10000 튜플, 20000 튜플, 30000 튜플의 양으로 만들어진 가상의 데이터에 대한 그룹화처리에서도 기존의 시스템보다 40%정도 빠른 실행 결과를 보여주고 있다.

User Definition		Target Media	Actual Value
SIZE	1	HTML	SIZE="1"
SIZE	1	LaTeX	¥footnotesize
...
SIZE	4	HTML	SIZE="4"
SIZE	4	LaTeX	¥Large
...
SIZE	Large	LaTeX	¥Large
...	Huge	LaTeX	¥Huge
...

그림 8. 사용자 정의 Decoration의 처리

표 4. 각 시스템의 소스코드의 양

	기존시스템	본 시스템
공통처리부분	3500	3900
HTML Gen.	2400	190
XML Gen.	1800	170
LaTeX Gen.	1800	170

5. 결론

본 논문에서는 Trinity Data Model의 개념을 기본으로, 새로운 SuperSQL의 미디어 처리계의 설계 방법을 제안했다. 이것은 보다 효율적이고 유연한 미디어 출판 처리계를 목표로 했다. 또한 제안시스템은 각 타깃미디어 처리의 통합과 추상 테이블 이용에 의해, 종래의 시스템보다 새로운 미디어에 동적이면서도 빠르게 대응하는 것을 목표로 했다. 각 타깃미디어 처리계로부터 공통으로 처리할 수 있는 부분을 모듈화 하여, 유연하게 제어할 수 있는 처리계에 재설계하는 것으로 각 타깃미디어의 통합화를 유도했다. 추상 테이블 개념을 이용해 미디어 생성부로부터 각 미디어 고유의 미디어구조화정보를 분리해 메타데이터로 취급함으로써 통합화 된 미디어생성부는, 추상 테이블의 정보를 제어하는 것으로 각 타깃미디어에 대응할 수 있게 되었다. 또한 추상 테이블을 응용해, 유저의 애매한 장식자의 정의에도 유연하게 대응할 수 있는 방법을 제안했다. 실제로 비교 시스템에서 소스 코드의 양과 처리 시간을 비교한 결과, 본 논문에서 제안한 시스템의 효율성이 높게 나타났다. 이것을 응용하면 유저가 각 미디어별로 지정한 장식자의 정의를 공통화 할 수 있으므로, 향후 연구과제로 추상테이블개념을 응용한 표준

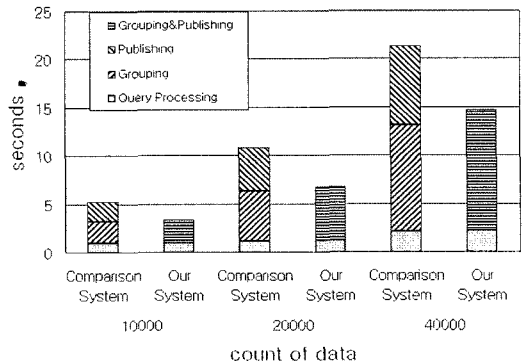


그림 9. 실행시간의 비교

의 SuperSQL 유저인터페이스에 대한 연구가 이루어져 공통인터페이스만으로 각각의 미디어출판에 대응할 수 있는 시스템의 개발과 시스템의 문제로 지적되어진 각 타깃미디어 특유의 처리에 대한 미디어 의존적인 외부처리 함수를 이용한 처리시스템의 개량과 효과적인 외부처리 함수를 개발하는 방법에 관한 논의가 필요하다.

참 고 문 헌

1. M. Toyama., : SupserSQL : An Extended SQL for Database Publishing and Presentation, Proc. ACM SIGMOD International Conference on Management of Data, pp.584-586, Seattle, USA, 1998.
2. M. Akahori, T. Arisawa, M. Toyama., : Data Integration on Relational Database and XML Using SueprSQL, IPSJ Transactions on Databases, No.SIG8(TOD 10), July. 2001.
3. <http://supersql.db.ics.keio.ac.jp>
4. S. G. Shin, T. Arisawa, M. Toyama., : The Integration of Media Generators in SuperSQL Query Processor, Third International Conference on ELPIIT, pp72-76, Ulaanbaatar, Mongolia, Jul. 2003.
5. T. Seto, T. Nagafuji, M. Toyama., : Generating HTML Sources with TFE Enhanced SQL, ACM Symposium on Applied Computing, pp.96-100, CA, USA, Feb. 1997.
6. M. F. Fernandez, W. C. Tan, D. Suci., : SilkRoute: trading between relations and XML, Computer Networks, Vol.33, No.1-6, pp.723-745, Jun. 2000.
7. M. F. Fernandez, Y. Kadiyska, D. Suci., A. Morishima, W. C. Tan., : SilkRoute: A framework for publishing relational data in XML, ACM Trans. Database Syst., Vol.27, No.4, pp.438-493, Dec. 2002.
8. M. J. Carey, D. Florescu, Z. G. Ives, Y. Lu, J. Shanmugasundaram, E. J. Shekita, S. N. Subramanian., : XPERANTO : Publishing Object-Relational Data as XML, WebDB Informal Proceedings 2000, pp.105-110, Texas, USA, May. 2000.
9. M. J. Carey, J. Kiernan, J. Shanmugasundaram, E. J. Shekita, S. N. Subramanian, : XPERANTO : A Middleware for Publishing Object-Relational Data as XML Documents, Proc. 26th Int. Conf. on Very Large Databases, pp.646-648, Cairo, Egypt, Sep. 2000.
10. Miller, R., Tsatalos, O. and Williams, J. : DataWeb: Customizable Database Publishing for the Web, IEEE MultiMedia, Vol.4, No.4, pp.14-21, 1997.
11. C. Lang, J. Chow, Database publishing on the Web and Intranets, Conriolis Group Books, 1996.



김 태 석 (tskim@deu.ac.kr)

1982 경북대학교 전자공학과 공학사 졸업
1992 일본 KEIO대학 이공학부 계산기과학전공 공학박사 졸업
1992 일본 KEIO대학 이공학부 객원연구원
2000~2003 동의대학교 전산정보원장
2003~2005 동의대학교 교무처장
1993~현재 동의대학교 컴퓨터소프트웨어공학과 교수

관심분야 : 인터넷응용, 원격강의, 자연어처리



신 상 규 (shin@db.ics.keio.ac.jp)

2000 동의대학교 컴퓨터공학과 학사 졸업
2003 일본 KEIO대학 컴퓨터과학전수 석사 졸업
2004~현재 일본 KEIO대학 컴퓨터과학전수 박사 과정

관심분야 : 데이터베이스시스템, 데이터마이닝, XML 처리



토야마 모토미치 (toyama@ics.keio.ac.kr)

1979 일본 KEIO대학 관리공학과 졸업
1981 동대학 석사 졸업
1992~1995 동대학 관리공학과 전임강사
1996 미국 OREGON대학 객원연구원
1996~현재 일본 KEIO대학 정보공학과 전임강사

관심분야 : 데이터베이스시스템, 데이터베이스출판