

# 객체관계형 데이터베이스 시스템에서의 시간 확장\*

\*윤성현, 신예호, 오광진, 이지영, 류근호

충북대학교 컴퓨터학과

## A Temporal Extension on Object Relational Database System

Sung-Hyun Yoon, Ye Ho Shin, Kwang Jin Oh, Jiyoung Lee, Keun Ho Ryu

Dept. of Computer Science Chungbuk National University

### 요 약

데이터베이스 시스템은 실세계에서의 객체들이 다양한 모델링을 통하여 구축된 데이터베이스에 대해 검색, 갱신, 분석 출력 및 논리적인 일관성 제어 등을 할 수 있는 소프트웨어 시스템이다. 그러나 전통적인 데이터베이스 시스템은 시간에 따라 변화하는 데이터의 이력을 관리 할 수 없다. 따라서 이력정보(Historical Informations)를 관리할 수 있게 하기 위해서는 시간차원의 확장이 필요하다 즉, 기존의 데이터베이스 시스템이 제공하지 못하였던 각 정보들의 이력관리와 이력을 이용한 다양한 서비스를 가능하게 하도록 시간차원을 지원할 수 있는 데이터베이스 시스템을 시간 데이터베이스 시스템이라 하며, 이를 위하여 이 논문에서는 객체관계형 데이터베이스 시스템을 기반으로 객체관계형의 특징과 레이어 개념을 이용하여 새로운 시간차원을 확장함으로써, 기존의 시스템을 현 상태로도 유지할 수 있는 시간 데이터베이스 시스템을 설계한다.

### 1. 서 론

기존의 데이터베이스 시스템에서는 시간에 따라 변하는 이력정보를 관리할 수 없었다 따라서 원자의 병력이나 추가변동과 같은 대용량 데이터베이스의 시계열 데이터 분석이나 데이터웨어하우징에 있어서, 지속적으로 변화하는 정보를 효율적으로 관리할 수 없었다. 따라서 이를 지원하기 위해서는 시간 차원을 지원해 줄 수 있는 시간 개념이 추가된 데이터모델이 필요하고, 객체에 대한 일반적인 질의 처리 이외에도 시간을 고려한 이력질의가 가능한 시간 데이터베이스 시스템(Temporal Database System)에 대한 연구가 필요하다

이 논문에서는 객체관계형 데이터베이스 시스템을 기반으로, 상속을 비롯한 객체관계형의 특징을 이용하여 시간차원을 확장하는 방법을 제시한다 이는 레이어 개념을 이용한 기법으로서, 기존의 일반 데이터베이스 시스템으로부터 시간 데이터베이스로의 전환이 용이하다 아울러 시간에 따른 객체의 변화정보를 효율적으로 처리할 수 있는 시간 연산자(Temporal Operator)를 설계한다. 이를 위해 이 논문을 다음과 같이 전개한다. 먼저 2장에서는 관련 연구를 통해 기본적인 시간에 대한 연구를 요약한다. 3장에서는 관련연구를 통한 시간 데이터모델에 대해 기술한다 4장에서는 객체관계형에서 시간확장기법을 설명하고, 마지막으로 5장에서는 결론 및 향후 연구사항을 기술한다

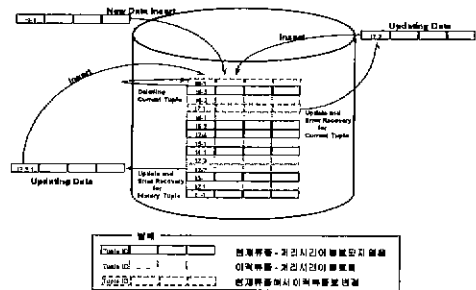
### 2. 관련연구

데이터베이스 시스템에서의 시간차원 확장에 대한 연구는 이미 20년 전부터 시행되어져 왔다[Snod92] 현재 데이터베이스 질의어의 표준인 SQL3에서는 시간지원 질의어의 표준으로 SQL/Temporal을 포함한다. 시간지원 질의어의 표준으로 TSQ22 데이터모델이 정의되어 있다. 또한 구현모델로서는 TupleTimeStamp, BackLog등이 있으며, 이들은 관계형 질의어의 확장으로 이루어진다

한편 시간 데이터베이스에서 사용되는 시간에는 객체가 실세계에서 발생한 시간을 의미하는 유효시간(valid Time)과 객체가 데이터베이스에 기록된 시간을 의미하는 처리시간(transaction time)[Jens91]이

있다. 처리시간은 시간이 단조 증가하며 과거에 저장된 이력자료에 대한 수정이 불가능하다. 반면 유효시간은 수정되기 이전의 이력자료 유지가 불가능하다. 따라서 완전한 이력을 지원하기 위해서는 유효시간과 처리시간을 동시에 지원하는 이원시간(bitemporal) 구조를 지원해야 한다[Jens94, Snod95].

시간 데이터베이스에서의 연산은 'Append-Only' 정책에 따라 데이터의 물리적 삭제는 발생하지 않고, 처리시간이 종료하는 논리적 삭제만이 발생할 뿐이다. 그리고 논리적 삭제가 발생한 튜플들도 모두 한 데이터베이스 속에서 함께 공존하게 된다 <그림 1>은 시간 데이터베이스 시스템에서 상태 변경이 발생하는 상황들을 도식적으로 표현한 것이다



<그림 1. 시간 데이터베이스의 상태전개>

### 3. 시간 데이터모델

시간 데이터 타입은 시스템에서 시간을 표현해주는 유일한 수단이므로 사용자에게 친숙하고, 의미전달이 쉬워야만 한다. 시간 데이터 타입은 크게 timepoint, interval, period로 구분할 수 있다. timepoint는 단일 시간점으로 표현하며, 이미 SQL-92에 정의된 datetime으로 표현한다. period는 시작점과 끝점 두 개의 시간점으로 구성된다. interval은 질의가 수행되기 전까지 그 값이 정해지지 않고 질의의 시간 피연산자

\*이 논문은 한국통신의 '98-99년도 정보통신 기초연구과제에 의해 수행되었음

로부터 시간개산함수에 의해 추출된다.

시간에 대한 연산은 두 시간값을 인자로 받아들여 각 시간값 사이의 관계를 유추해내는 시간관계연산자와 하나 이상의 시간 값을 받아들여 새로운 시간값을 생성하는 시간산출연산자로 나누어 정의될 수 있다. 여기서는 Allen[Alle83]의 시간 관계연산자에서 축소가 가능한 연산자들을 제외하고, TSQL2[Snod95]에서 정의된 연산자를 참조하여 5개의 시간관계연산자를 정의한다.

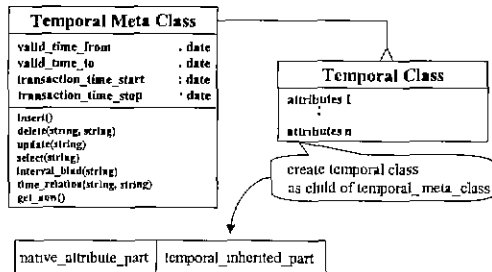
한편 시간 데이터베이스 시스템에서의 유효시간차원과 처리시간차원은 상호 직교성의 성질을 가지므로 서로 독립적이다. 따라서 이를 고려한 각각의 시간차원을 명시할 수 있는 연산자로서 각각 VALID, AS OF로 정의한다. <그림 2>에서는 이 논문에서 정의한 연산자를 정리하였다.

연산자 구분	기호	의미
시간 관계 연산자	before	
	=	
	meet	
	overlap	
시간 산출 연산자	+	시간구간 증가
	-	시간구간 감소
차원 구분자	VALID	유효시간차원 명시
	AS OF	처리시간차원 명시
AS OF	NOW	아직 정해지지 않은 시간값

<그림 2 시간 연산자 정의>

4. 객체관계형 모델의 시간확장

객체관계형 데이터베이스 시스템의 특징중 하나인 상속은 상위 클래스의 모든 속성과 메소드를 하위 클래스로 물려주게 된다. 또한 메소드는 연산을 객체로 포함한다 따라서 시간을 지원하는 최상위의 시간 클래스를 생성하고, 이를 모든 하위 클래스로 상속받게 하면 시간의 확장이 용이해진다 또한 객체관계형 시스템이 제공하는 메소드로서 모든 시간 연산들이 구성되면, 시간 연산 자체도 상속이 가능하므로 하위 클래스 각각에 모든 시간 연산들을 적용이 가능하다



<그림 3. 시간클래스의 구조>

<그림 3>은 시간 클래스의 구조를 보여준다. 이는 시간정보를 저장하는 시간속성과 시간연산을 수행할 메소드로 구성된다 시간속성의 타입은 date 타입을 사용하게 되는데 이는 사용자의 수준에서는 SQL-92에서 정의된 타입이고, 시스템 수준에서는 시간점이며 시간 연산의 기본단위가 되는 granularity의 의미를 갖게 된다

사용자가 생성하는 모든 클래스는 <그림 3>의 시간클래스를 상속받음으로써, 시간속성과 시간메소드를 사용할 수 있다. 다음은 시간클래스를 상속받는 사용자 정의 클래스 STUDENT를 생성하는 예이다

```
create class STUDENT as child of TimeClass (
    name char(10), grade char(1) )
```

객체관계형 질의에서는 관계형 질의와는 달리 객체포인터를 이용한 질의가 가능하니 따라서 이를 이용하여 시간연산을 수행할 메소드를 질의 자체에서 호출할 수 있다 <그림 4>는 STUDENT 클래스에 하나의 인스턴스를 삽입하는 질의를 관계형 질의와 비교하여 보여준다 그림에서 OR질의는 객체포인터 'ip'에 삽입된 객체의 포인터를 지정하고, 삽입연신 메소드를 호출할 때 이를 이용한다. 또한 질의에 유효시간차원이 명시되어 있지 않은 경우에는 처리시간차원과 동등한 시간값이 삽입된다.

```
SQL92 insert into STUDENT(name, grade) values('이종연', 'M'),
OR insert into STUDENT(name, grade) values('이종연', 'M') to ip;
call insert() on ip;
```

Name	Grade	ttstart	ttstop	vtbegn	vtend
'이종연'	'M'	'3/5/1995'	'12/31/9999'	'3/5/1995'	'12/31/9999'

<그림 4. 삽입연산에서의 SQL92질의와 OR질의>

4.1 삽입 연산

삽입 연산은 일반 속성 데이터의 삽입과 시간 속성 데이터의 삽입으로 나누어 고려될 수 있다. 일반 속성 데이터의 삽입은 일반적인 데이터베이스 시스템과 차이가 없다 반면 시간 속성 데이터의 삽입은 해당 객체에 해당하는 시간정보를 삽입하는 것으로서 이는 하나의 객체가 삽입될 때마다, 이미 그 값이 정해져 있다 <표 1>은 하나의 객체가 삽입될 때, 시간클래스의 각 속성에 삽입되는 정보를 보여준다.

<표 1 삽입연산시의 시간클래스 >

시간속성	삽입 값
ttstart	현재 시간(current time)이 삽입되며 not null임
ttstop	NOW가 삽입되며 not null임
vtbegn	사용자 정의 시간이 삽입되며, default는 현재시간
vtend	사용자 정의 시간이 삽입되며, default는 NOW

<그림 4>는 삽입연산의 예를 보여준다. 결국 insert() 메소드는 객체가 삽입될 때 시간속성에 해당 시간값을 저장하는 역할을 수행한다

4.2 삭제 연산

시간 데이터베이스에서는 append-only 정책에 따르면 객체의 물리적인 삭제는 없고 해당 객체의 시간정보를 변경하여, 객체가 의미를 갖는 시간 구간을 만료시키는 논리적인 삭제만이 이루어진다. 즉 삭제하려는 객체의 처리종료시간을 현재시간으로 변경함으로써, 해당객체를 더 이상 유효하지 않은 객체로 만들므로써 삭제연신이 수행된다. 따라서 삭제 연산에서는 <그림 5>에서 보여주듯이 삭제하려는 정보의 시간 적용 구간에 따라 연신의 회수가 달라지게 된다.



<그림 5 시간구간에 따른 삭제연산>

<그림 6>은 삭제연산의 예를 보여준다 temporal 질의에서 OR질의로 진원되는 과정에서도 객체포인터가 사용되는데, 그림에서 보듯이 해당 객체의 객체포인터를 추출한 후, 이에 대한 삭제연산 메소드를 호출함으로써 삭제연산이 수행된다 또한 질의에 명시된 시간구간 VALID ['3/5/1995', '3/5/1998']은 메소드의 인자로 넘겨주게 되고, 이를 <그림 5>에서처럼 시간구간 비교연산에 사용하게 된다 <그림 6>에서의 질

의는 <그림 5>에서 세 번째 경우에 해당하는 것으로서, 기존 객체의 처리종료시간을 현재시간으로 변경하고, 명시된 시간구간에 따라 새로운 객체가 삽입되게 된다

```
Delete from STUDENT where name = '이종연' VALID {3/5/1995' 3/5/1998},
OR Select STUDENT into dp from STUDENT where name = '이종연',
call delete('3/5/1995', 3/5/1998') on dp;
```

Name	Grade	lstart	lstop	vbegin	vend
'이종연'	'M'	'3/5/1995'	'5/5/1999'	'3/10/1995'	'12/31/9999'
'이종연'	'M'	'5/5/1999'	'12/31/9999'	'3/6/1998'	'12/31/9999'

<그림 6 삭제연산의 예>

4.3 갱신 연산

갱신연산은 삭제연산과 삽입연산으로 구성될 수 있다. 즉, 변경된 값을 가진 객체를 삭제하고, 새로이 변경된 값을 가진 객체를 삽입함으로써 갱신연산이 수행된다. 따라서 갱신연산은 삭제연산과 비교할 때 연산의 수행과정의 비슷하므로 삭제연산을 위한 알고리즘이 그대로 갱신연산에 사용될 수 있다 그러나 갱신연산에서는 변경된 정보를 가진 객체를 삽입하기 위해 삽입연산을 한 번 더 수행하게 된다

4.4 추출 연산

시간 데이터베이스 시스템에서의 삽입, 삭제, 갱신 연산은 시간절의기 된다. 즉, 절의 자체에 시간에 관해 아무런 단서가 없다 하더라도, 이들 연산은 결국 시간클래스와 시간연산을 필요로 하게된다. 즉, 시간 데이터베이스 시스템에서의 삽입, 삭제, 갱신 연산은 모두 입력되는 절의가 시간절의가 아니라 하더라도 결국 시간클래스에 영향을 미치는 시간절의가 된다 그러나 추출연산은 절의에 명시되는 시간에 따라 시간절의와 비시간절의의 차이가 뚜렷하며, 또한 절의에 있어서 유효시간치원과 처리시간차원을 달리 명시함으로써, 데이터베이스의 시간에 따른 다양한 상태를 표현해준다. 따라서 추출연산에서는 절의 자체에서 시간절의와 비시간절의가 명확히 구분되고, 입력되는 절의에 아무런 시간 구간이 명시되어 있지 않다면, 현재의 데이터베이스 상태를 검색하여 스냅샷 형태의 결과를 추출한다 <그림 7>은 추출연산의 예이다. 첫 번째 절의의 경우는 시간절의의 예로서 이는 OR 절의로 전환되어 수행될 수 있다 반면 절의에 시간에 대한 아무런 단서가 없다면 이는 비시간 절의로서 OR절의로 전환되지 않고 바로 수행이 가능하다.

```
Select grade from STUDENT where name = '이종연'
VALID {3/5/1996, 'NOW'},
OR Select grade into sp from STUDENT where name = '이종연',
call select('3/5/1996', 'NOW') on sp;
```

Name	Grade	lstart	lstop	vbegin	vend
'이종연'	'M'	'5/5/1999'	'12/31/9999'	'3/6/1998'	'12/31/9999'

```
Select grade from STUDENT where name = '이종연'
```

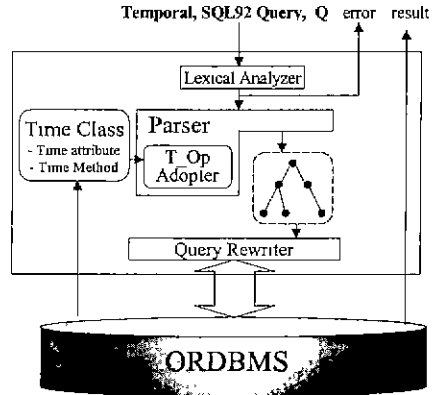
Name	Grade
'이종연'	'M'

<그림 7. 추출연산의 예>

5 시스템 구성

이 논문에서 제시하는 시스템은 기존의 객체관계형 데이터베이스 시스템을 기반으로 레이어 개념을 사용한다 따라서 절의 자체를 시간 절의와 비시간 절의로 구분할 수 있다 즉, 입력되는 절의가 시간절의인지 비시간절의인지, 각각의 경우 처리 흐름이 달라지게 된다 <그림 8>은 시스템의 전체적인 구조를 보여주고 있다. 시스템의 구성은 Lexical analyzer, parser, 시간 데이터 클래스기 레이어를 구성하고, 일반 객체관계형 데이터베이스 시스템이 실질적인 절의처리를 담당한다.

절의가 입력되면, 입력절의는 analyzer와 parser를 거치면서 절의 트리를 생성하게 된다. 이는 입력된 절의를 분석하여 절의의 유형을 판단하고, 절의의 성공적인 수행여부를 확인한다 또한 비시간절의인 경우는, 곧바로 기반 데이터베이스 시스템으로 넘겨지게 된다. 시간 절의의 경우에는 parser에 내장된 temporal operation adopter가 시간메타클래스로부터 해당연산에 필요한 시간속성 및 시간 메소드를 호출한다 이러한 과정을 거쳐 입력된 절의는 OR절의로 전환되고, 기반 시스템을 이를 수행하여 결과를 보여준다



<그림 8. 시스템 구성>

6. 결론

지금까지 객체관계형 데이터베이스 시스템에서의 시간차원 확장에 대한 연구를 전개하였다. 이 논문에서는 객체관계형 데이터베이스 시스템을 기반으로 레이어를 이용하여 이원시간차원을 지원하는 시간 데이터베이스 시스템의 프로토타입을 설계하였다 이에 관하여 관련 연구를 통해, 시간연산자와 시간 데이터타입을 설계하고 이들에 대한 제약조건을 정의하였다. 아울러 시간연산을 정의하였으며, 시스템 구조를 설계 하였다.

이 논문의 기반 플랫폼이 된 데이터베이스 시스템은 객체관계형 데이터베이스 시스템으로서, 객체관계형 공간 데이터베이스 시스템에서의 시간 확장을 향후 과제로 한다 이에 관해 시공간 연산의 시공간 연산 지 등의 연구가 예상되며, 객체관계형 데이터베이스 시스템의 장점을 살려, 이를 활용할 수 있는 시간확장기법의 연구가 필요하다

참고문헌

[Alle 83] J F Allen, 'Maintaining knowledge about temporal intervals', Comm. of the ACM vol26, no 11, pp.832-843, Nov.1983  
 [Jens94] C Jensen, J. Clifford, R. Elmasri, S. Gadia, P. Hayes, and S. Jajodia, "A Glossary of Temporal Database Concepts", ACM SIGMOD Record, 23, No 1, 1994  
 [Snod92] R. Snodgrass, "Temporal Database" in A. U. Frank, I. Campari, and U. Formentin(eds), Theories and Method of Spatio-Temporal Reasoning in Geographic Space Vol. 639 of Lecture Notes in Computer Science Springer-Verlag, 1992. pp 22-64  
 [Snod95] R Snodgrass, "The TSQL2 Temporal Query Language", Kluwer Academic Publisher, 1995  
 [Torp97] Kristian Torp, Christian S. Jensen, Michael Bohlen. "Layered Implementation of Temporal DBMSs-Concept and Techniques", TimeCenter TR-2, 1997