

2005

CXQuery XQuery

2006

# CXQuery XQuery

論文 碩士學位 論文 提出

2006年 1月

梨花女子大學校 科學技術大學院  
李 玟 英

# 李 玟 英 碩 士 學 位 論 文 認 准

指導教授 \_\_\_\_\_

審查委員 \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

梨花女子大學校 科學技術大學院

論文概要.....	vi
I. ....	1
1.1 .....	1
1.2 .....	4
II. ....	6
2.1 XML .....	6
2.1.1 XPath.....	6
2.1.2 XQuery .....	7
2.1.3 CXQuery.....	8
2.2 XML .....	9
2.2.1 XML .....	10
2.2.2 XML .....	11
2.2.3 Native XML .....	12
2.2.4 XML-Enabled .....	13

III. CXQuery	14
3.1 XML	17
3.2 XML	18
3.2.1	19
3.2.1.1 DTD	20
3.2.1.2	23
3.2.2 XQuery	25
3.2.3	29
3.3 CXQuery	32
IV.	35
4.1	35
4.2	36
4.3	37
4.3.1	37
4.3.2 CXQuery XQuery	39
4.4	44
4.4.1 DTD	44
4.4.2	45
V.	47

[ 1- 1] title, genre, actor	.....	3
[ 3- 1] XML DBMS	.....	15
[ 3- 2] CXQuery XQuery	.....	16
[ 3- 3] DTD	.....	20
[ 3- 4] DTD	.....	21
[ 3- 5]	.....	23
[ 3- 6]	.....	24
[ 3- 7] CXQuery XQuery	.....	27
[ 3- 8]	.....	28
[ 3- 9]	.....	30
[ 3- 10] CX-threshold	.....	31
[ 3- 11] CX-threshold	.....	32
[ 4- 1] movie	.....	38
[ 4- 2] CXQuery	.....	39
[ 4- 4]	.....	40
[ 4- 5] , CX-threshold	.....	41
[ 4- 6]	.....	41

[ 4- 10]	.....	45
[ 4- 11]	XQuery .....	46

[ 2- 1] RDBMS XML DBMS	.....	12
[ 3- 1] genre 가	.....	17
[ 3- 2] CXQuery XQuery	.....	19
[ 3- 3] XML	.....	21
[ 3- 4]	.....	24
[ 3- 5]	.....	26
[ 3- 6] CX - threshold XQuery	.....	32
[ 4- 1]	.....	35
[ 4- 2]	.....	36
[ 4- 3] CX - threshold XQuery	.....	42
[ 4- 4] CX - threshold XQuery	.....	43



# 論文概要

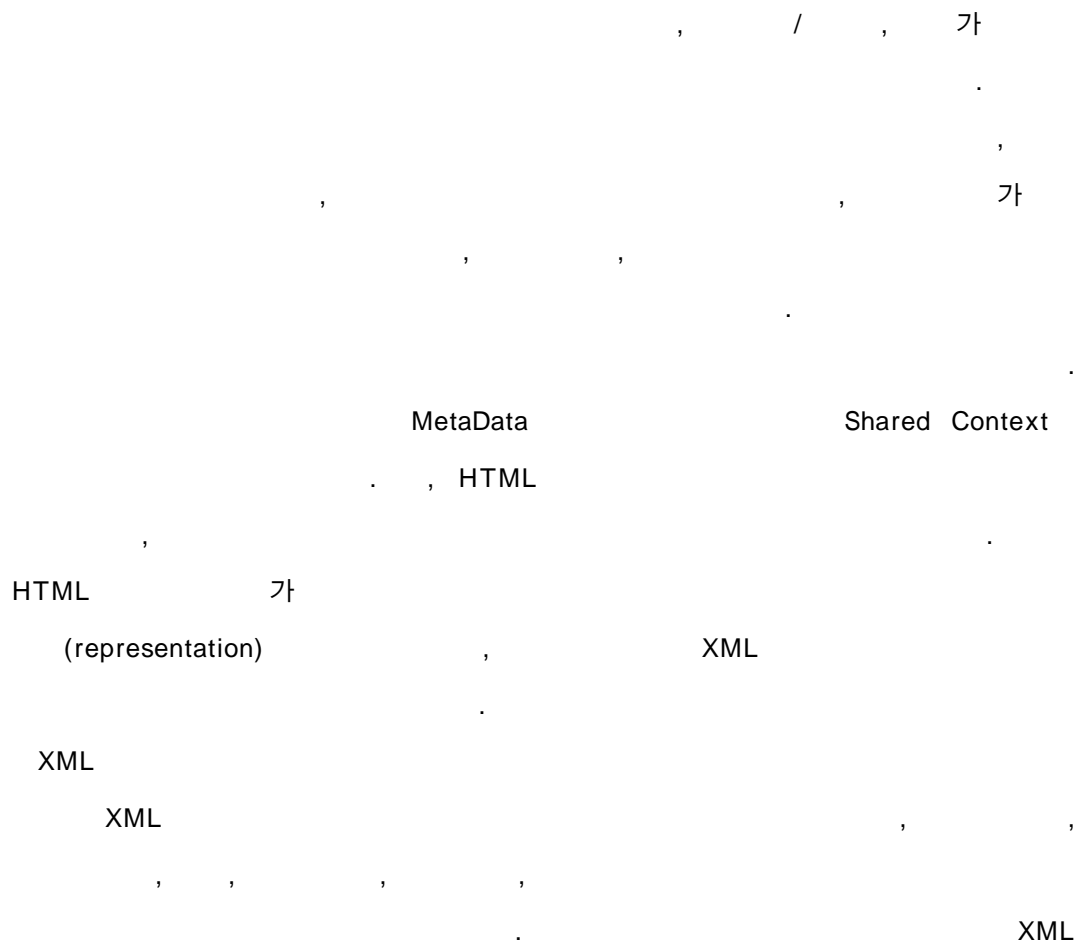
가  
XML  
XML 가 , W3C  
가 Tag  
Attribute self - desctription,  
가  
Software AG, IBM, Sun, Microsoft, Netscape, DataChannel, SAP  
XML [1].  
XML XML XML  
[2].  
XPath[3] Xquery[4] XML 가  
가 가  
가 가 XML 가  
가 ,  
가 .

가

가 가 CXQuery가  
XML DBMS CXQuery가 XQuery  
.  
CXQuery Xpath Xquery 가 XQuery  
.

# I.

## 1.1



XML  
XML  
[5]. XML

World Wide Web Consortium (W3C) HTML XML  
XML XML  
XML  
가 가  
XML 가  
가 가  
[6]. 가 XML

Q1. action 'Jean Reno' 가  
가?

가 XML  
[ 1-1]  
가

<pre> &lt;!ELEMENT movie (year, country*,                 genre*, title, director, actor+)+&gt; &lt;!ELEMENT year (#PCDATA) &gt; &lt;!ELEMENT country (#PCDATA) &gt; &lt;!ELEMENT genre (#PCDATA) &gt; &lt;!ELEMENT title (#PCDATA) &gt; &lt;!ELEMENT director (#PCDATA) &gt; &lt;!ELEMENT actor (#PCDATA) &gt; <b>1 :</b>           /  &lt;!ELEMENT movie (general_info, detail_info)+&gt; &lt;!ELEMENT general_info (year, country*,                        genre+)&gt; &lt;!ELEMENT year (#PCDATA)&gt; &lt;!ELEMENT country (#PCDATA)&gt; &lt;!ELEMENT genre (#PCDATA)&gt; &lt;!ELEMENT detail_info (title, people)+&gt; &lt;!ELEMENT title (#PCDATA)&gt; &lt;!ELEMENT people (director, actors)*&gt; &lt;!ELEMENT director (#PCDATA)&gt; &lt;!ELEMENT actors (actor)*&gt; &lt;!ELEMENT actor (#PCDATA)&gt; <b>3 :</b>           / </pre>	<pre> &lt;!ELEMENT movie (production,                 detail_info, people)+&gt; &lt;!ELEMENT production (#PCDATA)&gt; &lt;!ATTLIST production year CDATA #IMPLIED &gt; &lt;!ATTLIST production country CDATA #IMPLIED &gt; &lt;!ELEMENT detail_info (#PCDATA)&gt; &lt;!ATTLIST detail_info genre CDATA #IMPLIED&gt; &lt;!ATTLIST detail_info title CDATA #IMPLIED&gt; &lt;!ELEMENT people (#PCDATA)&gt; &lt;!ATTLIST people actor CDATA #IMPLIED&gt; &lt;!ATTLIST people director CDATA #IMPLIED&gt; <b>2:</b>           /  &lt;!ELEMENT genre (country, year)+&gt; &lt;!ATTLIST genre type CDATA #IMPLIED &gt; &lt;!ELEMENT country (#PCDATA) &lt;!ELEMENT year (movie)*&gt; &lt;!ELEMENT movie (title, people)+&gt; &lt;!ELEMENT title (#PCDATA)&gt; &lt;!ELEMENT people (#PCDATA)&gt; &lt;!ATTLIST people actor CDATA #IMPLIED&gt; &lt;!ATTLIST people director CDATA #IMPLIED&gt; <b>4:</b>           / </pre>
--	--

[ 1- 1 ] title, genre, actor

CXQuery 가 가 CXQuery가 .  
CXQuery . CXQuery  
"Leon" [7].

Genre="action" and actor="Jean Reno"

가 가 가  
가 .

## 1.2

XML

가

XQuery

가

가

가

CXQuery

XML

DBMS

DBMS

CXQuery

가

(trade-off) . XQuery 가  
XML 가  
가 .  
CXQuery  
XQuery XML 가 가 .  
CXQuery XML 가  
CXQuery CXQuery  
XQuery . CXQuery  
XQuery . XQuery  
가 XQuery  
XQuery XQuery  
XQuery 가 CXQuery가 가  
.  
. 2 XML  
, XML . 3  
CXQuery XQuery .  
4 5 .

## II.

XML XML XML  
XML XML  
XML XML  
CXQuery가  
CXQuery 가

### 2.1 XML

#### 2.1.1 XPath

XPath XML W3C XML  
path( ) . XPath  
XSLT XML , XPath XML  
(syntax rule) XPath  
XSLT . XPath XML  
(path expression) 가  
XML 가  
XPath XML (pattern expression)



XPath / XML

XPath movies movie title

XPath : /movies/movie/title

```
<movies>
<movie>
<p_year>1994</p_year>
<nation>America</nation>
<genre>drama</genre>
<m_title>Leon</m_title>
<main_actor>Jean Reno</main_actor>
</movie>
</movies>
```

XPath /

/

(node - set)

가

가

### 2.1.2 XQuery

XQuery . XQuery ,

, FOR, LET, WHERE, RETURN(FLWR) ,  
 , XQuery  
 ( : ) ( : ) .  
 [8].  
 XQuery XML Query Data Model . XQuery XML  
 ,  
 .  
 XQuery , FOR, LET, WHERE, RETURN ,  
 FLWR .  
 FLWR .  
 FOR , LET .  
 가 FOR XQuery가  
 .  
 XQuery SQL . , SQL  
 .  
 XQuery 1.0 (Update) .  
 XQuery SQL 가 .  
 (Projection) (SQL SELECT XQuery RETURN)  
 . SQL , XQuery  
 .  
 XQuery WHERE , ORDER BY . XQuery SQL

**2.1.3 CXQuery**

CXQuery XML  
 . CXQuery 가 .  
 , XML  
 CXQuery .  
 , 가  
 가  
 . CXQuery  
 가

## 2.2 XML

, XML XML  
 , , , “XML ” 가  
 ,  
 XML

DBMS XML DBMS  
 (Native) XML 가 .  
 XML XML , XML  
 XML XML . XML ,

### 2.2.1 XML

- XML 가 , XML (parsing)
- RDBMS XML RDBMS BLOB(Binary Large Object) CLOB(Character Large Object) , 가 ,
- RDBMS XML RDBMS , 가 XML 가 XML 가 (XML Enabled Database) 가 가 .

XML 가 (XML Enable Database)  
 가 XML  
 가 XML (Native XML Database)  
 (Contents Management System)

### 2.2.2 XML

, 가  
 XML 가  
 . XML  
 XML  
 XML 가  
 , XML  
 가  
 , XML  
 가 , XML  
 가 가 XML  
 XML 가 XML 가  
 . XML XML  
 가 XML

XML , 가 . [ 2-1]

	RDBMS	XML DBMS
	- - - -	-XML & - -XML -Xpath Query
	-XML Data -XML -XML 100%	- - DB ,

[ 2- 1] RDBMS XML DBMS

### 2.2.3 Native XML

XML XML XML 가 XML ( ) XML [9].

XML XML XML 가 , , API, XML .

XML XML XML  
 가 .  
 XML XML  
 . XML XML  
 XML  
 . XML DOM XML  
 . Software AG Tamino[10], X-Hive  
 X-Hive/DB[11], eXcelon eXtensible Information Server(XIS)[12], Ipedo  
 Ipedo[13] .

#### 2.2.4 XML-Enabled

XML-Enabled XML ( )  
 ) , . XML  
 XML  
 . XML  
 XML ,  
 XML LOB . IBM DB2[14],  
 Oracle[15], MS SQL [16], XML  
 XML-Enabled .

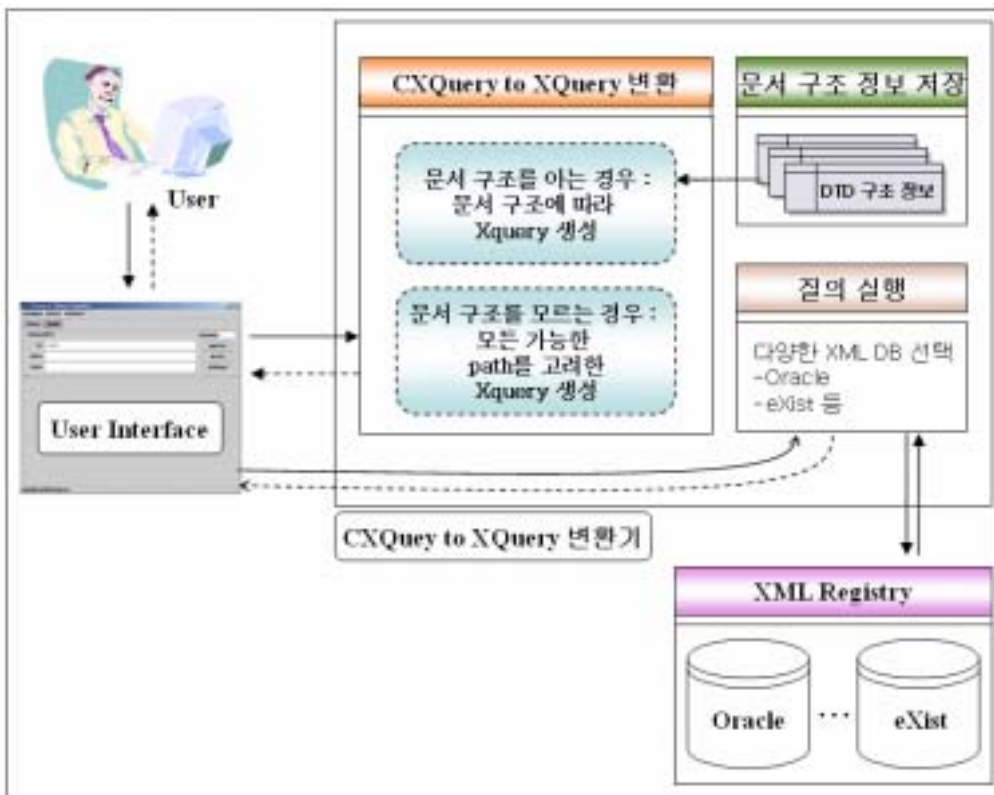
### III. CXQuery

가 CXQuery XQuery 가 CXQuery 가  
 가 XML XQuery  
 가 . CXQuery XQuery XML  
 XQuery , XML  
 가 가 가  
 가 . 가  
 가 가 가  
 threshold XQuery가 . XML  
 가 XQuery  
 CXQuery .  
 3 . XML ,  
 CXQuery XQuery , XQuery XML  
 .

- 1) User interface CXQuery .
- 2) CXQuery Xquery XQuery User Interface
- 3) Xquery 가
- 4) ( DBMS 가 )



CXQuery  
가  
가  
XQueries  
가



[ 3- 1 ] XML DBMS

[ 3- 1 ] User Interface  
CXQuery CXQuery XQuery



### 3.1 XML

CXQuery가  
 가 genre="action"  
 XQuery XML  
 가 가  
 [ 3- 1]

[ 3- 1] genre 가

1	element 가	element가	//genre="action"
2		attribute	//@genre="action"
3	element element가 가	element	//genre/*="action"
4	element element attribute가	element 가	//genre//@*="action"

가 genre 가

**//genre="action" OR //@genre="action" OR //genre/\*="action" OR  
 //genre//@\*="action"**

actor

**//actor="Jean Reno" OR //@actor="Jean Reno" OR //actor/\*="Jean Reno" OR  
//actor//@\*="Jean Reno"**

Return title  
1, 3, 4 가 . title  
return 가  
2가 .

**//title or //@title**

Q1 CXQuery XQuery 가 [ 3-2]

### 3.2 XML

XML CXQuery XQuery .  
가 DTD  
XQuery가 .  
DTD  
CXQuery  
CXQuery .

[ 3- 2] CXQuery XQuery

CXQuery	XQuery
<p>For \$c in doc() Where genre="action" AND actor="Jean Reno" Return title</p>	<p>For \$c in doc() Where (\$c//genre="action" OR \$c//genre//*="action" OR \$c//@genre="action" OR \$c//genre//@*="action") AND (\$c//actor=" Jean Reno" OR \$c//actor//*="Jean Reno" OR \$c//@actor="Jean Reno" OR \$c//actor//@*=" Jean Reno") Return \$c//title</p>
	<p>For \$c in doc() Where (\$c//genre="action" OR \$c//genre//*="action" OR \$c//@genre="action" OR \$c//genre//@*="action") AND (\$c//actor=" Jean Reno" OR \$c//actor//*="Jean Reno" OR \$c//@actor="Jean Reno" OR \$c//actor//@*=" Jean Reno") Return \$c//@title</p>

3.2.1

XML

DTD

. DTD(Document Type

Definition)

XML

, DTD

XML

가

가

가

CXQuery XQuery

### 3.2.1.1 DTD

[ 3- 3] DTD

[ 3- 4]

가

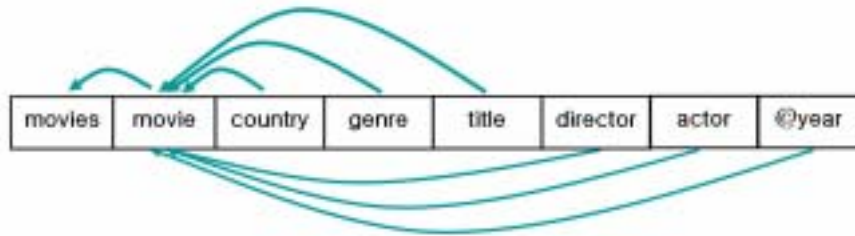
가

```

<!ELEMENT movies (movie+)>
<!ELEMENT movie (country, genre,
                 title, director, actor)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT director (#PCDATA)>
<!ELEMENT actor (#PCDATA)>
<!ATTLIST movie year CDATA #REQUIRED>

```

[ 3- 3] DTD



[ 3- 4] DTD

가 [

3-3]

[ 3- 3] XML

Type	Data type	Path	Name
ELE	P	/	movies
ELE	P	/movies	movie
ATT	L	/movies/movie	year
ELE	L	/movies/movie	country
ELE	L	/movies/movie	genre
ELE	L	/movies/movie	title
ELE	L	/movies/movie	director
ELE	L	/movies/movie	actor

[ 3- 3] DTD

가

- **Type :** 가 Element Attribute
- **Data type:** , 가  
가  
⇒ L : 가 가  
(#PCDATA ) Element 가  
⇒ P : 가 가  
Element 가 가  
⇒ C : 가 가 Element 가
- **Path :**
- **Name:**

Element

Attribute 가

Q1 CXQuery

title, genre, actor name title

/movies/movie/title, genre /movies/movie/genre actor

/movies/movie/actor



### 3.2.1.2

XML

가 . [ 3- 5]

DTD

```
<!ELEMENT movie (year, country, genre,  
title, director, actor)+>  
<!ELEMENT year (#PCDATA)>  
<!ATTLIST year yyyy CDATA #IMPLIED>  
<!ELEMENT country (#PCDATA)>  
<!ATTLIST country name CDATA #IMPLIED>  
<!ELEMENT genre (#PCDATA)>  
<!ELEMENT title (#PCDATA)>  
<!ATTLIST title name CDATA #IMPLIED>  
<!ELEMENT director (#PCDATA)>  
<!ATTLIST director name CDATA #IMPLIED>  
<!ELEMENT actor (#PCDATA)>  
<!ATTLIST actor name CDATA #IMPLIED>
```

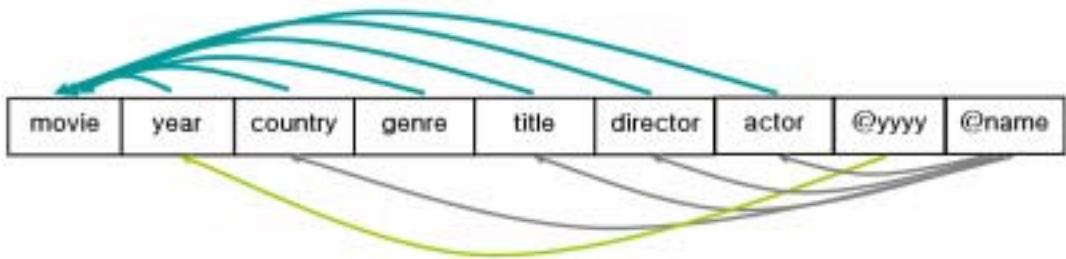
[ 3- 5]

[ 3- 5]

name

4

[ 3-6] [ 3- 4]



[ 3- 6]

[ 3- 4]

Type	Data Type	Path	Name
ELE	P	/	movie
ELE	P	/movie	Yaar
ELE	L	/movie	Country
ELE	L	/movie	Genre
ELE	L	/movie	title
ELE	L	/movie	Director
ELE	L	/movie	Actor
ATT	L	/movies/year	yyyy
ATT	L	/movie/country/@name /movie/title/@name /movie/director/@name /movie/actor/@name	name

CXQuery	name	Name
/movie/country/@name,	/movie/title/@name,	/movie/director/@name,
/movie/actor/@name	4	.

### 3.2.2 XQuery

. CXQuery XQuery

- 1) CXQuery .
  - 2) .
  - 3) .
  - 4) for doc() .
  - 5) where return
- Data type .

[ 3- 3] CXQuery XQuery

CXQuery : **for \$c in doc()**  
**Where genre="action" and actor="Jean Reno"**  
**Return title**

actor, title . [ 3-3]  
 [ 3 -5] . CXQuery가 genre,

[ 3- 5]

Type	Data type	Path	Name
ELE	L	/movies/movie	Genre
ELE	L	/movies/movie	Title
ELE	L	/movies/movie	Actor
...	...	...	....

genre : /movies/movie/genre

actor : /movies/movie/actor

title : /movies/movie/title

for \$c in doc() for \$c in /movies/movie . CXQuery  
 where  
 return

XQuery .

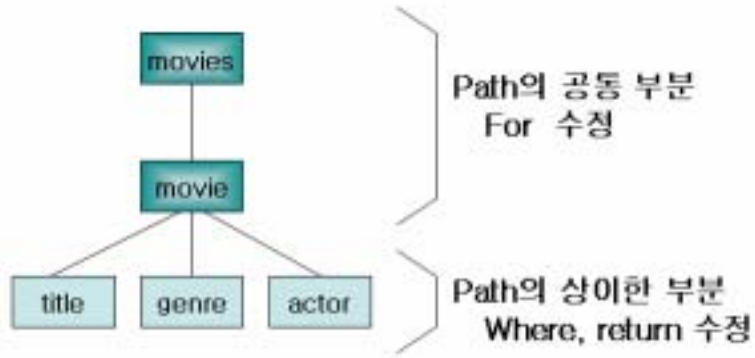
XQuery : for \$c in /movies/movie

where \$c/genre="action" AND \$c/actor=" Jean Reno"

return \$c/title

XQuery

[ 3-6]



[ 3- 7]

CXQuery XQuery

L

XQuery

가

가

가

[ 3- 8]

genre

“action”

type

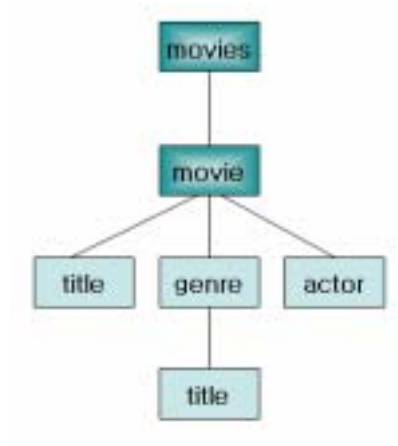
genre="action"

genre

genre//\*[="action"

가

Data type



[ 3- 8]

Data type

- Data Type L :
  - 가
    - ◆ /movies/movie/genre="action"
- Data Type P :
  - 가 가 가
    - ◆ /movies/movie/genre/@\*="action"
    - ◆ /movies/movie/genre//\*="action"
    - ◆ /movies/movie/genre//\*/@\*="action"
- Data Type C :
  - 가 가 가
    - ◆ /movies/movie/genre="action"

- ◆ /movies/movie/genre/@\*="action"
- ◆ /movies/movie/genre//\*="action"
- ◆ /movies/movie/genre//\*/@\*="action"

[ 3- 8] Data type XQuery

XQuery : **for \$c in /movies/movie  
 where (\$c/genre//\*="action" OR \$c/genre//\*/@\*="action" OR  
 /movies/movie/genre/@\*="action") AND \$c/actor=" Jean Reno"  
 return \$c/title**

### 3.2.3

CXQuery

가

가

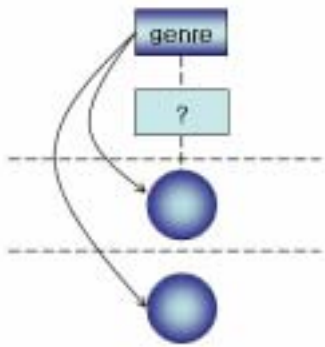
가

가

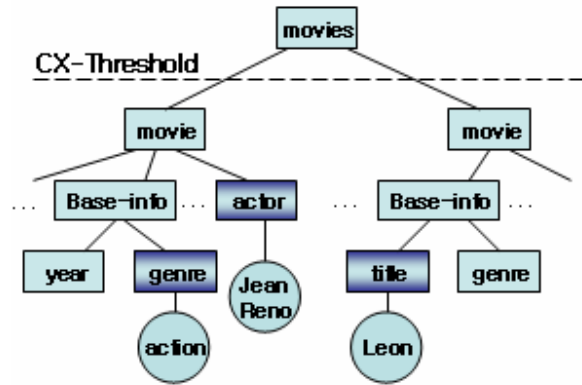
가

가

threshold



(a)



(b)

가

[ 3- 9]

CXQuery가

가 가 .

<actor>

<name> Jean Reno </name>

</actor>

[ 3- 9] (a)

. XML actor="Jean Reno"

3.2.1

Data Type

[ 3- 9] (b)

genre actor movie title movie

. (b) 가

가 가

가

threshold genre, actor,

title

가

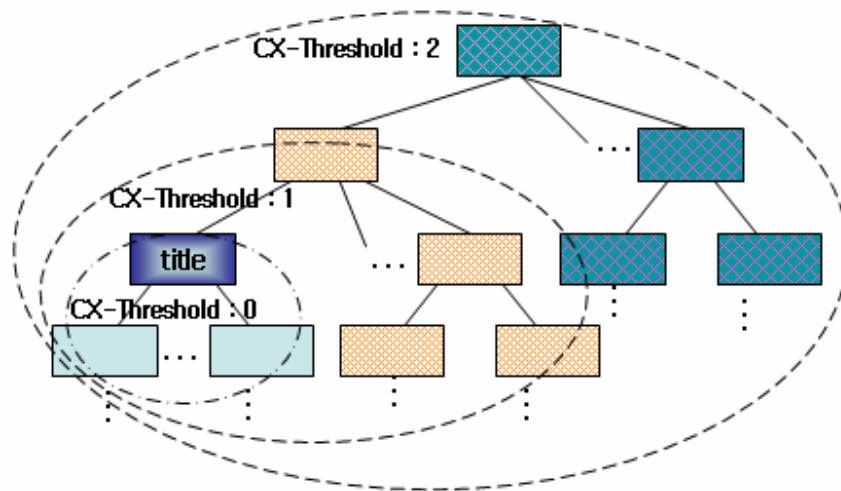
CX-threshold

가



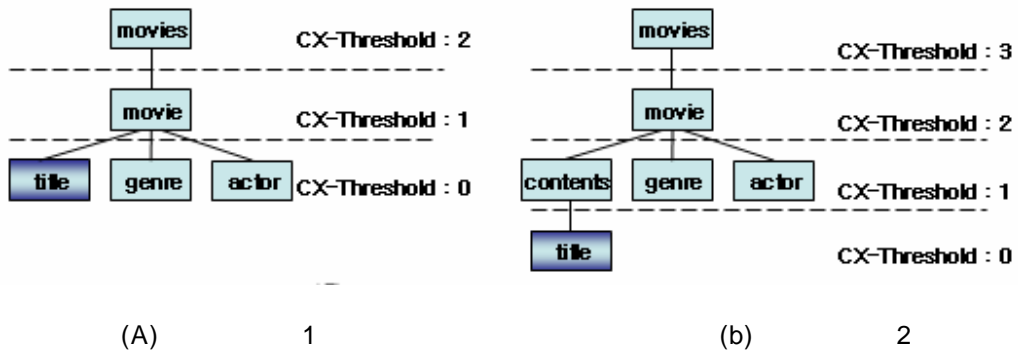
return  
CX-threshold

- CX-Threshold가 0 : where return
- CX-Threshold가 1 : where return



[ 3- 10] CX-threshold

threshold [ 3-10]  
XQuery /movies/movie/title, /movies/movie/genre,  
/movies/movie/actor return title CX-threshold가 0  
/movies/movie/title , CX-  
threshold가 1 /movies/movie가 ,  
threshold가 2 /movies가 .



[ 3- 11] CX-threshold

[ 3- 11] Threshold XQuery [ 3- 6] .

[ 3- 6] CX-threshold XQuery

CX- Threshold : 1	CX- Threshold : 2
(a) for \$c in /movies/movie Where \$c/genre="action" and \$c/actor="Jean Reno" Return \$c/title	(a) for \$c in /movies Where \$c/movie /genre="action" and \$c/movie /actor="Jean Reno" Return \$c/movie /title
(b)	(b) for \$c in /movies/movie Where \$c/genre="action" and \$c/actor="Jean Reno" Return \$c/contents/title

### 3.3 CXQuery

CXQuery

Reno” . , actor “Jean

```

<actors>
  <theater>
    <actor> Jean Reno </actor>
  </theater>
  <movie>
    <actor> Jean Reno </actor>
  </movie>
</actors>

```

movie actor

CXQuery theater actor  
가

CXQuery : for \$c in doc()  
return name

가 title name [ 3- 5] CXQuery  
name 4

- /movie/country/@name**
- /movie/title/@name**
- /movie/director/@name**
- /movie/actor/@name**

가	title	name		name
		. CXQuery가		
가	가		가	.
		' 가 !	.	
			,	' 가
!		' 가	!	' 가
가		! 가	.	
				가 ,
		가 가	.	CXQuery
				가
	XQuery			.
		CXQuery		

# IV.

CXQuery XQuery XML  
 CXQuery ,  
 XQuery가 .  
 가 XQuery  
 CXQuery가 가 가 .

## 4.1

CXQuery XQuery .

[ 4- 1 ]

	Microsoft Windows XP professional
	Oracle 10g, eXist1.0
	Java J2SDK1.4 JDBC Driver

CXQuery XQuery Microsoft Windows XP professional  
 Oracle10g XML  
 eXist1.0 . CXQuery XQuery JDK1.4

## 4.2

, CXQuery XQuery ,

[ 4-2]

[ 4- 2]

DTD Parse	DTDParse()	Directory DTD (.dtd)
	Insert()	DTD
	getCurrent()	row
	getPath()	
CXQuery to XQuery converter	XQueryNot DependDTD()	가 XQuery
	XQueryWithDTD()	가 XQuery

	makeXQuery()	XQuery
Result Store	outFile()	XQuery
XQtable	CXPanel()	CXQuery 가 XQuery
DBHandle r	DBInfo()	XQuery

### 4.3

CXQuery XQuery .

#### 4.3.1

[ 1- 1] [ 4- 1] movie

. DTD

{genre, year, title, actor, country,

director}

<pre> &lt;!ELEMENT year (country)+&gt; &lt;!ELEMENT country (genre)&gt; &lt;!ELEMENT genre (movie)*&gt; &lt;!ELEMENT movie (title, director, actor)+&gt; &lt;!ELEMENT title (#PCDATA)&gt; &lt;!ELEMENT director (#PCDATA)&gt; &lt;!ELEMENT actor (#PCDATA)&gt; <b>5 :</b>      /  &lt;!ELEMENT movie (year, country,                 genre, title, director, actor)+&gt; &lt;!ELEMENT year (#PCDATA)&gt; &lt;!ELEMENT country (name)*&gt; &lt;!ELEMENT name (#PCDATA)&gt; &lt;!ELEMENT genre (type)*&gt; &lt;!ELEMENT type (#PCDATA)&gt; &lt;!ELEMENT title (name)&gt; &lt;!ELEMENT name (#PCDATA)&gt; &lt;!ELEMENT director (name)&gt; &lt;!ELEMENT name (#PCDATA)&gt; &lt;!ELEMENT actor (name)&gt; &lt;!ELEMENT name (#PCDATA)&gt; <b>6 :</b>      / </pre>	<pre> &lt;!ELEMENT movie (year, country,                 genre, title, director, actor)+&gt; &lt;!ELEMENT year (#PCDATA)&gt; &lt;!ATTLIST year yyyy CDATA #IMPLIED&gt; &lt;!ELEMENT country (#PCDATA)&gt; &lt;!ATTLIST country name1 CDATA #IMPLIED&gt; &lt;!ATTLIST country name2 CDATA #IMPLIED&gt; &lt;!ELEMENT genre (#PCDATA)&gt; &lt;!ATTLIST genre type CDATA #IMPLIED&gt; &lt;!ELEMENT title (#PCDATA)&gt; &lt;!ATTLIST title name CDATA #IMPLIED&gt; &lt;!ELEMENT director (#PCDATA)&gt; &lt;!ATTLIST director name CDATA #IMPLIED&gt; &lt;!ELEMENT actor (#PCDATA)&gt; &lt;!ATTLIST actor name1 CDATA #IMPLIED&gt; &lt;!ATTLIST actor name2 CDATA #IMPLIED&gt; <b>7 :</b>      / </pre>
--	---

[ 4- 1] movie



가

가

가

가

CXQuery

가 가

### 4.3.2 CXQuery XQuery

User Interface

CXQuery INPUT

for, where, return

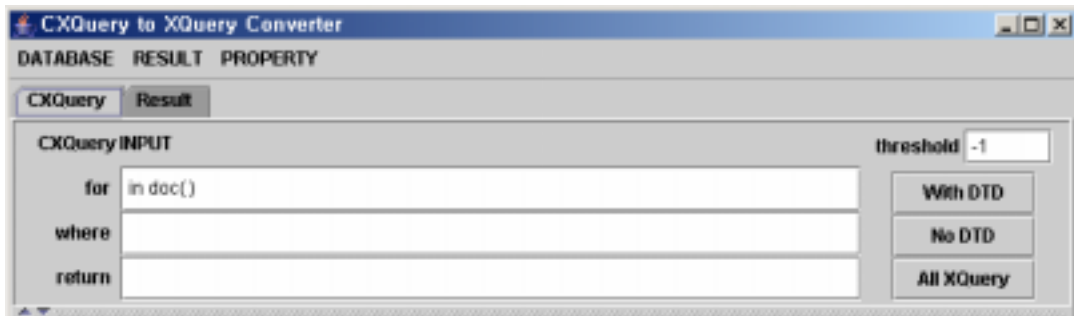
threshold

distance

With

DTD, No DTD, All XQuery

CXQuery가 XQuery



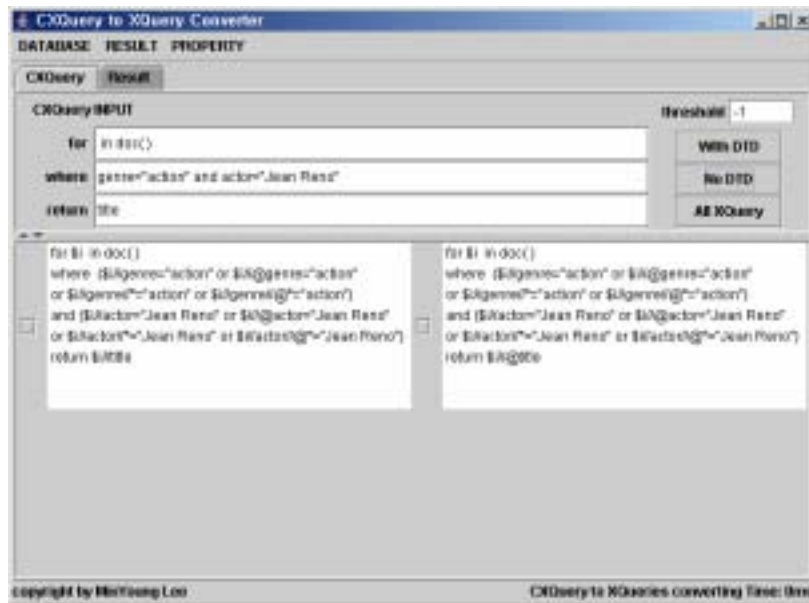
[ 4- 2] CXQuery

- With DTD : XQuery . DTD  
PROPERTY -> Set DTD Directory

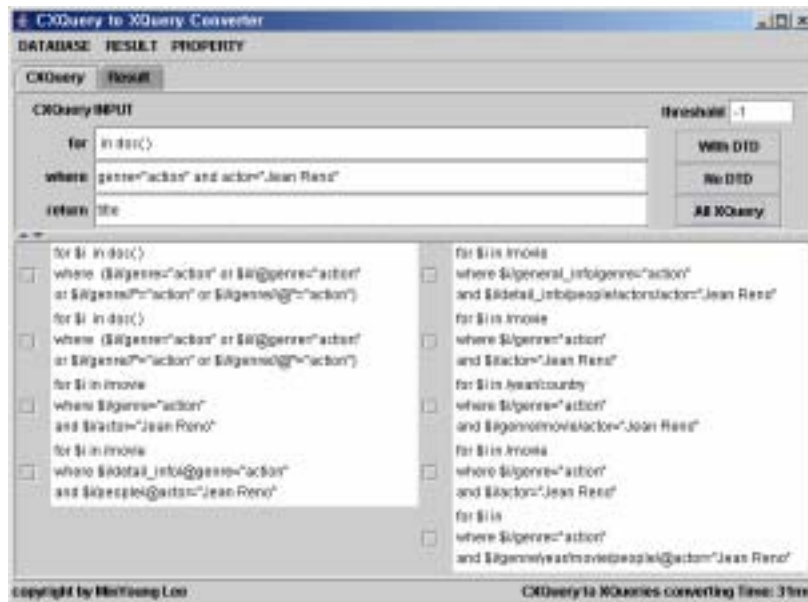
- No DTD : XQuery .
- All XQuery : 가 XQuery .

CXQuery (No DTD )  
 (With DTD) 가 [ 4 -4] [ 4-5] .  
 가 가 [ 4-6] .

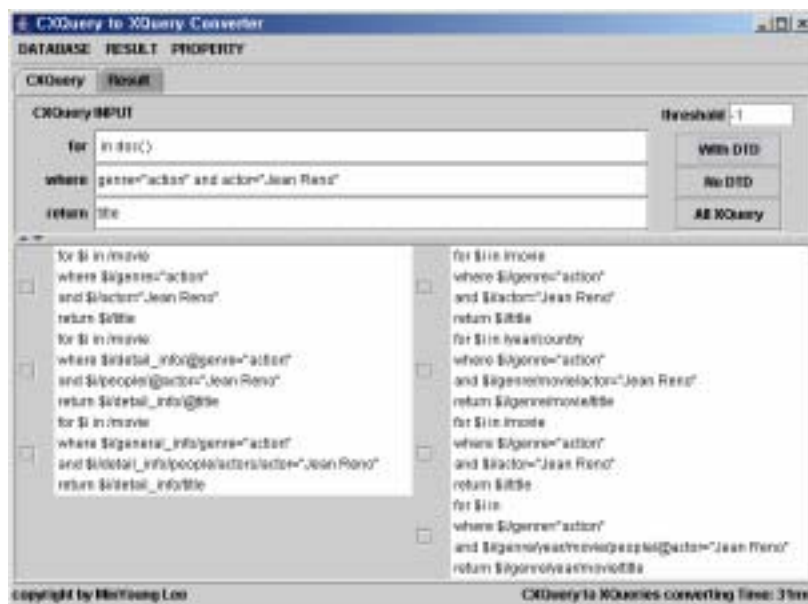
CXQuery : for \$i in doc()  
 where genre="action" and actor="Jean Reno"  
 return title



[ 4- 3]



[ 4- 4] , CX-threshold



[ 4- 5]

CX - Threshold

XQuery . XQuery CX - threshold [ 4 - ]

[ 4 - 3 ] CX - threshold XQuery

7  XQuery	<pre>for \$i in /movie where \$i/genre="action" and \$i/actor="Jean Reno" return \$i/title</pre>
	<pre>for \$i in /movie where \$i/detail_info/@genre="action" and \$i/people/@actor="Jean Reno" return \$i/detail_info/@title</pre>
	<pre>for \$i in /movie where \$i/general_info/genre="action" and       \$i/detail_info/people/actors/actor="Jean Reno" return \$i/detail_info/title</pre>
	<pre>for \$i in /movie where \$i/genre="action" and \$i/actor="Jean Reno" return \$i/title</pre>
	<pre>for \$i in /year/country where \$i/genre="action" and \$i/genre/movie/actor="Jean Reno" return \$i/genre/movie/title</pre>
	<pre>for \$i in /movie where \$i/genre="action" and \$i/actor="Jean Reno"</pre>

	return \$i/title
	for \$i in doc() where \$i/genre="action" and \$i/genre/year/movie/people/@actor="Jean Reno" return \$i/genre/year/movie/title
	for \$i in /movie where \$i/genre="action" and \$i/actor="Jean Reno" return \$i/title

CX-Threshold

XQuery [ 4- 4] .

[ 4- 4] CX-threshold

XQuery

CX- threshold	XQuery
1	for \$i in /movie where \$i/genre="action" and \$i/actor="Jean Reno" return \$i/title
	for \$i in /movie where \$i/genre="action" and \$i/actor="Jean Reno" return \$i/title
	for \$i in where \$i/movie/genre="action" and \$i/movie/actor="Jean Reno" return \$i/movie/title

2	<pre> for \$i in /movie where \$i/detail_info/@genre="action" and \$i/people/@actor="Jean Reno" return \$i/detail_info/@title </pre>
	<pre> for \$i in /movie where \$i/general_info/genre="action" and       \$i/detail_info/people/actors/actor="Jean Reno" return \$i/detail_info/title </pre>
	<pre> for \$i in where \$i/movie/genre="action" and \$i/movie/actor="Jean Reno" return \$i/movie/title </pre>
	<pre> for \$i in /year/country/genre where \$i="action" and \$i/movie/actor="Jean Reno" return \$i/movie/title </pre>
	<pre> for \$i in where \$i/movie/genre="action" and \$i/movie/actor="Jean Reno" return \$i/movie/title </pre>

## 4.4

CXQuery XQuery converter 가

### 4.4.1 DTD

DTD Parsing

[ 4-11]

. DTD

가 가

. (a) DTD

(b)

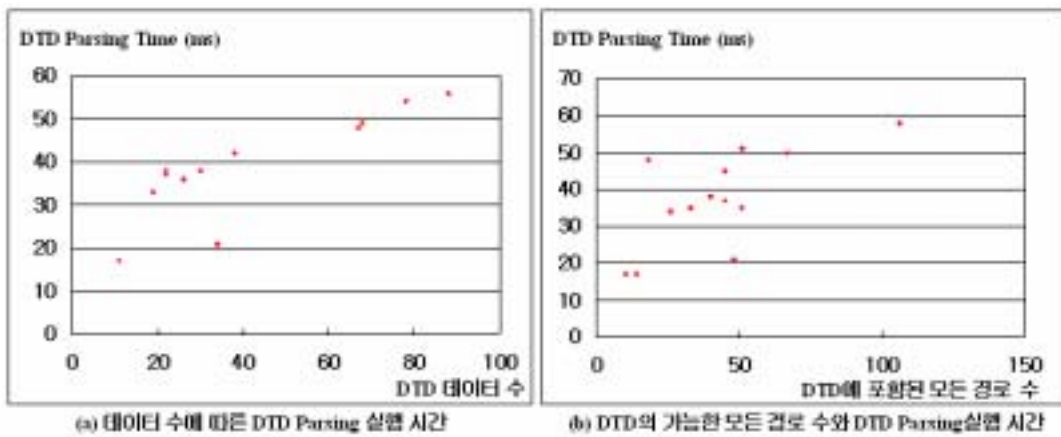
가

. DTD

Parsing

parent

가



[ 4- 6]

가 100

60ms

Parsing

Parsing

가

DTD DTD Parsing

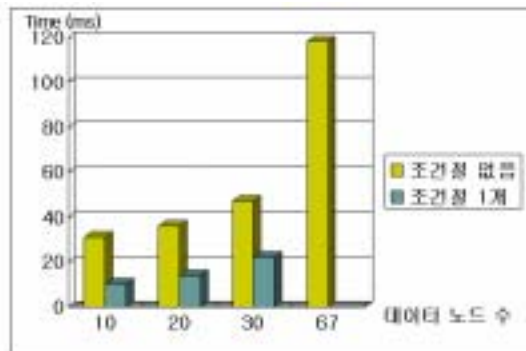
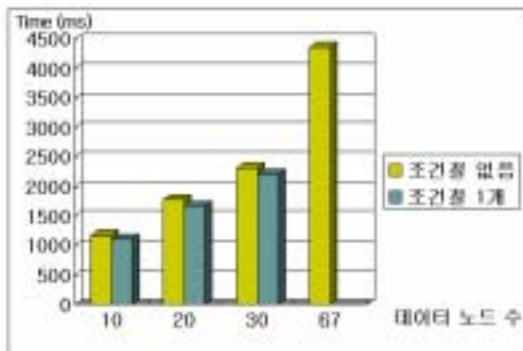
#### 4.4.2

Q1	1
Q2	2
Q3	3

CXQuery XQuery

XQuery가

가 , 가



조건절 개수에 따른 실행시간

조건절 개수에 따른 생성된 쿼리 수

[ 4- 7]

XQuery

XQuery 가 DTD 가

가

가

가

DTD

가



# V.

XML 가  
가 .  
가 . XML  
,  
XML .  
XML ,  
,  
XML , , , “XML  
” XML RDB XML  
RDB 가 XML XML DB  
XML DB XML ,  
가 가 가  
CXQuery가 가 . CXQuery XQuery  
가 가 가  
가 XQuery .  
, CXQuery가 XQuery 가  
가

CXQuery가

CXQuery XQuery

CXQuery

CXQuery

XQuery

.

- [1] XML, <http://www.xml.com>
- [2] George Feinberg, Native XML database storage and retrieval, Journal of Linux, 2005
- [3] W3C Consortium, XPath 1.0: An XML Path Language, W3C Working Draft, Nov. 1999
- [4] W3C Consortium, XQuery1.0: An XML Query Language, W3C Working Draft, Jun. 2001
- [5] Matthias Nicola, Jasmi John, XML parsing: a threat to database performance, Proc. Of the 12th Information and Knowledge Management archive, 2003 : 175-178
- [6] W3C Consortium, XQuery1.0 Requirements, W3C Working Draft, 12 Nov. 2003
- [7] , XML , , 2004. 8

- [8] W3C Consortium, XML Path Language(XPath) Version 1.0, W3C Recommendation, 16 Nov.1999
- [9] T. Fiebig, S. Helmer, C.-C. Kanne, G. Moerkotte, J. Neumann, R. Schiele, T. Westmann, Anatomy of a native XML base management system, Journal of VLDB, 2002
- [10] Tamino, <http://www.tamino.co.kr>
- [11] X-Hive/DB, <http://www.x-hive.com/products/db/index.html>
- [12] eXtensible Information Server(XIS), <http://www.xml.com/pub/p/381>
- [13] Ipedo, <http://www.ipedo.com/>
- [14] IBM DB2, <http://www-128.ibm.com/developerworks/xml/library/x-mxd4.html>
- [15] eXist, <http://www.exist-db.org>
- [16] Oracle, <http://www.oracle.com>
- [17] SQL Server 2005 Express Edition,  
<http://www.microsoft.com/sql/editions/express/default.msp>

## ABSTRACT

# Implementation of CXQuery To XQuery Converter

Department of Computer Science and Engineering  
Ewha Institute of Science and Technology  
Ewha Womans University  
Lee, Min young

XML provides simple yet flexible ways to represent the structure and contents of arbitrary documents. The simplicity and flexibility of XML have made it possible for XML to be adopted as the basis of data interchange standards in a wide variety of application areas.

Therefore, there have been many proposals for a Query language for XML. And various database provide XML Query languages. Users have to know the structure of documents for querying because most of existing Query languages require the users

to know the structure of XML documents, including all the element and attribute names, data types of the data values, and the hierarchical structure of the elements.

Otherwise, using CXQuery, users can query by specifying only the names of data and their values against many documents of various structures is likely to impose a heavy burden on the query processor in a query processing time aspect. But Users make special engine in order to supporting CXQuery.

So, I propose CXQuery to XQuery converter that provide users freedom and use existing XML database.