

# B2B, B2G 환경에서 전자문서의 상호운용을 지원하는 DTD 상호운용 구조의 설계

박상윤\*

## <목 차>

I. 서론	2. IDEP 프로토콜
II. 관련 연구	3. DTD 관리 서버
1. XML	IV. 결론
2. DTD의 상호운용	참고문헌
III. DTD 상호운용 구조와 IDEP 프로토콜	Abstract
1. DTD 상호운용 구조	

## I. 서 론

인터넷 기반 비즈니스가 활성화됨에 따라 기업 간(B2B : Business to Business) 또는 기업과 정부 간(B2G : Business to Government) 전자문서(digital document)를 통한 상거래 및 행정업무 처리의 필요성이 커지고 있다. 그러나 기업 및 정부의 전자문서의 형식 및 구조가 상이하여 전자문서 교환 후 데이터베이스에 적재하는 등의 데이터 처리는 사실상 수동적인 수준에 머물고 있어 전자문서 사용의 장점을 살리지 못하고 있는 실정이다. 이에 따라 기업 간 또는 기업과 정부 간 공용의 전자문서 형식을 사용하고 이에 대한 처리를 자동화하여 상거래 및 행정업무의 효율을 개선하기 위한 방안으로서 기업 간 또는 기업과 정부가 공용할 수 있는 전자문서의 형식을 XML(eXtensible Markup Language) 기반의 DTD(Data Type Definition)로 정의하고[1-3] 이를 상호운용할 수 있는 DTD 상호운용[8][9] 구조(interoperability architecture)의 정의가 요구된다.

\*대림대학 컴퓨터정보계열 교수

본 논문의 2절에서는 전자문서의 객체화 기술을 지원하는 차세대 웹 언어인 XML 기술과 전자문서의 구조를 정의하는 DTD 기술에 관하여 연구동향을 기술하며 3절에서는 DTD를 상호운용하는 인프라 구조인 DTD 상호운용 구조와 DTD 관리 서버 및 DTD 정보의 검색 및 교환을 지원하는 IDEP(Interoperable DTD Exchange Protocol) 프로토콜의 설계사항에 대하여 소개한다.

## II. 관련 연구

### 1. XML

차세대 웹 표준 언어로 예상되는 XML은 1998년 2월, W3C(The World Wide Web Consortium)에 의해 XML 버전 1.0 표준 명세서가 정의된 바 있다. XML은 문서의 구조를 정의하는 DTD와 DTD를 따르는 XML 문서의 생성 규칙을 명시하는 언어로써 기존 개발되어 사용되고 있는 SGML(Standard Generalized Markup Language)의 부분 집합이다[4]. XML은 SGML의 구조화된 문서의 특성과 자유로운 태그 정의 등의 장점과 인터넷상에서 하이퍼미디어 문서를 쉽게 제공할 수 있는 HTML의 장점을 모두 갖도록 설계되었다. XML은 인터넷상에서 사용할 수 있어야 하고 다양한 응용 프로그램을 지원해야 하며 SGML과의 호환성을 유지해야 하는 등의 설계 원칙을 가지며, 사용자가 작성하기 용이한 간결하고 명확한 문법을 제공하고 간결한 마크업(mark-up)을 지원하는 등의 특성을 갖는다.

XML은 XML documents라고 하는 데이터 객체와 이를 처리할 수 있는 프로그램의 행위를 명시하는데 XML documents는 파싱(parsing)되거나, 파싱되지 않은 데이터를 포함하는 엔티티(entity)라는 적재 단위로 구성되어 있다. 파싱된 데이터는 마크업과 문자 데이터 형태의 텍스트들로 구성되는데 마크업은 문서의 논리적 구조를 명시한다. 모든 XML 문서는 물리적 구조와 논리적 구조를 갖는데, 물리적으로는 엔티티로 구성되고 논리적으로는 선언(declarations), 요소(elements), 주석(comments), 문자 참조(character references) 및 처리 명령(processing instructions) 등으로 구성된다. XML 문서는 논리적 구조상의 제약사항을 정의하고 미리 정의된 적재 단위의 사용을 지원하는 DTD라는 메커니즘을 제공하는데 이는 문서의 최초 요소 이전에 직접 정의되거나 외부로부터 참

조되어야 한다.

## 2. DTD의 상호운용

DTD란 문서의 내용(contents)과 구조(structure)를 구분한 후 문서의 구조를 태그(tag) 기반의 마크업 언어로 계층적(hierarchical)으로 표현하는 기술을 의미한다. XML 문서는 DTD를 포함하는데 DTD에서 정의된 문서의 구조에 따라 문서의 내용이 작성된다. 따라서 동일한 DTD를 따르는 XML 문서들은 내용이 다르더라도 동일한 구조를 가지게 된다. DTD는 계층적으로 구성된 엘리먼트(element) 태그의 집합으로 구성되며 엘리먼트는 객체지향(object-orient) 기술과 접목되어 독립된 객체로 표현되기도 한다. DTD와 객체기술을 접목시킨 기술로는 W3C에서 정의한 DOM(Document Object Model) 기술 등이 있다[5].

문서의 구조를 표현하는 DTD 기술은 분산 환경에서 원격지 데이터베이스간의 상호운용을 위한 데이터베이스 스키마(schema)로도 사용된다[6][7]. 따라서 공유된 DTD가 제공된다면 이질적인(heterogeneous) 데이터베이스들 간의 정보 교환 시에도 수신된 데이터를 데이터베이스에 반영시키는 작업이 자동화될 수 있다. 이러한 기술이 B2B, B2G 환경에 적용될 경우 기업간 또는 기업과 정부간에 교환되는 전자문서가 XML 및 DTD 기반으로 작성되고 DTD가 이들 간에 상호 공유된다면 B2B, B2G 환경의 상거래 및 행정업무 처리가 자동화될 수 있다.

그러나 이러한 자동화된 전자문서 처리를 활성화하기 위해서는 DTD를 공유해야 하는 주체 간에 DTD 공유가 선행되어야 하며 특히 DTD의 위치를 파악하고 DTD를 상호교환하는 인프라 구조의 확립이 필수적이라고 하겠다. 따라서 본 논문에서는 B2B, B2G 환경에서 주체들 간에 DTD를 효과적으로 검색하고 이를 상호교환할 수 있는 DTD 상호운용 구조(architecture) 및 상호운용 규약(protocol)을 제시하도록 한다.

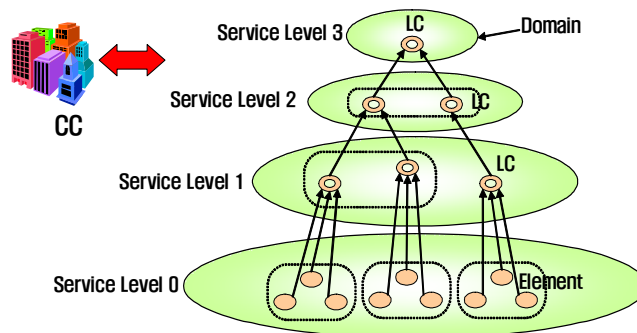
### III. DTD 상호운용 구조와 IDEP 프로토콜

DTD 상호운용 구조는 인터넷상에 위치한 전자문서들의 DTD를 검색하고 상호운용하는 체계를 정의하는 인프라 구조(infra structure)이다. DTD 상호운용

체계에서 사용되는 IDEP(Interoperable DTD Exchange Protocol) 프로토콜은 DTD 상호운용 구조의 구성요소들 간의 DTD 정보 교환을 지원하는 통신 프로토콜이다. 본 절에서는 DTD 상호운용 구조와 IDEP 프로토콜의 세부사항에 대하여 기술하기로 한다.

## 1. DTD 상호운용 구조

DTD 상호운용 구조의 구성요소로는 DTD 상호운용 구조 전체에 대한 DTD 검색과 LC들의 관리를 담당하는 중앙 조정자(coordinator)인 CC(Central Coordinator), 지역 도메인에서 DTD 검색을 지원하는 지역 조정자인 LC(Local Coordinator), DTD 관리 서버인 Element 및 DTD 정보의 교환을 지원하는 IDEP 프로토콜 등이 있다. <그림 1>은 DTD 상호운용 구조의 개념도를 예시한다.



<그림 1> DTD 상호운용 구조 개념도

<그림 1>에서 예시하는 바와 같이 DTD 상호운용 구조는 계층적인 도메인(domain)들로 구성된다. 동일한 계층의 DTD 관리 서버들의 집합을 도메인(domain)이라 하는데 각 도메인에는 서비스 레벨(service level) 번호가 부여된다. CC는 DTD 상호운용 구조 전체에 대한 DTD 검색과 각 도메인의 LC 관리를 담당하며 도메인 내에서 지역 조정자 역할을 하는 DTD 관리 서버를 LC라고 한다. 도메인 내에서 LC를 제외한 일반 DTD 관리 서버를 Element라고 하며 LC는 Element들 중에서 자동으로 선출(voting)된다. 특정 도메인의 LC는 상위 도메인의 Element가 되며 다시 상위 도메인에 속한 Element들 중에서 LC

가 선출된다. 이 과정은 최상위 도메인에서 하나의 최상위 LC가 선출될 때까지 반복되며 LC 선출은 IDEP 프로토콜을 통해 자동화 된다. 도메인 내의 Element 간 또는 Element와 LC 간에는 정기적인 Alive 메시지가 교환되는데 Element들은 이를 통해 도메인 멤버를 확인하고 LC의 존재 여부를 검사한다. 만약 LC가 사라졌을 경우 Element들은 IDEP 프로토콜을 통해 새로운 LC를 선출하고 이를 상위 도메인에 반영시킨다. 이와 같이 DTD 상호운용 구조는 Element들의 변화에 따라 동적으로 재구성된다.

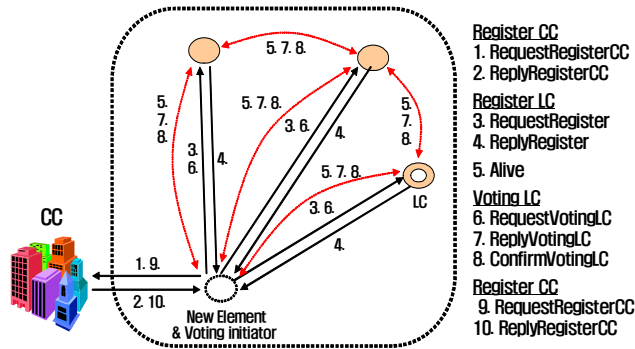
## 2. IDEP 프로토콜

IDEP 프로토콜은 DTD 상호운용 구조의 구성 및 DTD 정보 검색, DTD 정보 교환, LC 선출 등을 지원한다.

### □ Element 등록

DTD 상호운용 구조에 새로 참여하는 Element는 CC에 등록하여(RequestRegisterCC), 도메인명, 도메인 멀티캐스트(multicast) 주소, Service Level, 초기화된 타임스탬프(timestamp) 등을 배정받는다(ReplyRegisterCC). 새로운 Element가 배정받은 도메인내의 기존 Element들과 도메인 LC에게 등록 메시지(RequestRegister)를 멀티캐스팅(multicasting) 하고, 기존 Element들과 LC는 새로운 Element에게 자신의 정보를 포함하는 응답 메시지를 송신한다(ReplyRegister).

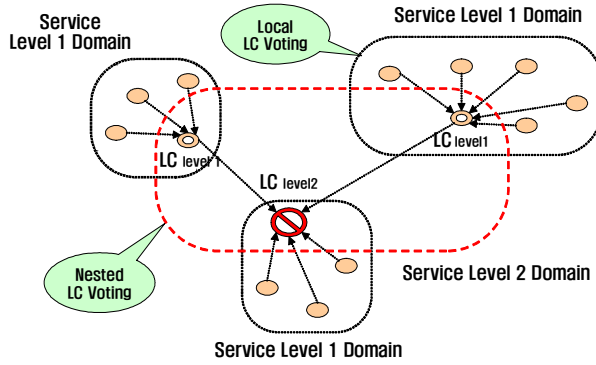
그림 2에서 예시하는 바와 같이 도메인 내의 각 Element는 도메인 내의 다른 Element들과 LC에게 주기적인 생존 메시지를 멀티캐스팅하고(Alive) 이를 수신한 Element들은 자신의 도메인 멤버 목록에서 해당 Element에 대한 정보를 갱신하며 일정시간 이상 생존 메시지가 도착하지 않는 Element에 대해서는 목록에서 해당 정보를 삭제한다.



<그림 2> Element 등록 및 LC 선출 과정

□ LC 선출

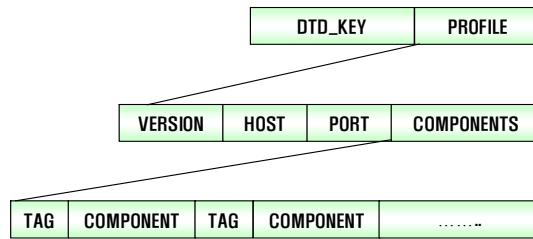
LC로부터 일정시간 이상 생존 메시지가 도착하지 않으면 이를 처음 감지한 Element가 도메인의 다른 Element들에게 LC 선출 메시지를 멀티캐스팅하고 (RequestVotingLC) 이를 수신한 각 Element는 자신의 타임스탬프 정보를 포함하는 메시지를 다른 Element들에게 멀티캐스팅 한다(ReplyVotingLC). 다른 Element들의 타임스탬프들을 수신한 각 Element는 자신의 타임스탬프와 수신된 타임스탬프들 중 가장 큰 값의 타임스탬프를 가진 Element를 확인하고 LC 확인 메시지(ConfirmVotingLC)를 통해 이를 다른 Element에게 전파한다. 도메인 내의 모든 Element가 동일한 Element를 LC로 채택했을 경우 LC 선출과정은 종료되나 그렇지 않은 경우 상기의 LC 선출 메시지(RequestVotingLC) 전송으로부터의 선출과정을 다시 반복하게 된다. LC 선출과정이 완료되면 해당 LC는 CC에게 등록 메시지(RequestRegisterCC)를 송신하여 새로 선출된 LC로서 등록하고 새로 상위 도메인에 대한 도메인명, 도메인 멀티캐스트(multicast) 주소, Service Level, 초기화된 타임스탬프 등을 다시 배정받는다 (ReplyRegisterCC). 새로 선출된 LC는 CC로부터 추가 도메인을 부여받았으므로 새로운 도메인 내에서 다시 등록 과정을 수행해야 한다. 이 과정은 상기한 이전 Element 등록 과정과 동일하게 진행되며 만약 새로운 LC가 새 도메인에서 다시 LC로 선출될 경우 LC 선출과정은 다시 상위 도메인으로 이어져 계층적 DTD 상호운용 구조에서 연속적인 LC 선출이 발생하게 된다. 그림 3은 각 도메인의 Element 등록 및 LC 선출과 계층적으로 중첩된 도메인 간에 연속적인 LC 선출과 이를 통한 DTD 상호운용 구조의 구성 과정을 예시한다.



<그림 3> 중첩 도메인 간의 연속적인 LC 선출

□ IDR 데이터베이스

IDR(Interoperable DTD Reference)이란 특정 DTD에 대한 네트워크상의 위치 정보를 유지하는 객체로서 B2B, B2G의 주체들은 IDR을 통해 원하는 DTD를 획득할 수 있다. <그림 4>는 IDR의 구조를 예시한다.



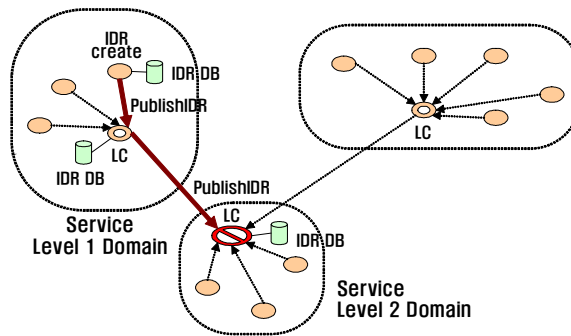
<그림 4> IDR의 구조

IDR은 DTD를 식별하는 코드인 DTD\_KEY와 DTD 위치 정보를 포함하는 PROFILE로 구성된다. IDR의 PROFILE은 특정 DTD를 가지고 있는 호스트에 대한 네트워크 정보와, 접근 정보가 포함된 COMPONENT들로 구성된다.

모든 DTD 관리 서버는 자체 IDR 데이터베이스를 유지하고 LC의 경우 하위 Element들에 대한 IDR 정보도 유지한다. IDR 데이터베이스는 캐시(cache) 형태로 유지되므로 특정 IDR에 대한 일정시간 이상의 참조가 없으면 해당 IDR은 자동으로 삭제된다.

DTD 관리 서버에 새로운 IDR이 생성되면 DTD 관리 서버는 IDR 데이터베

이스에 이를 등록하고 도메인 LC에게 IDR을 전파한다(PublishIDR). 도메인 LC는 전달받은 IDR을 IDR 데이터베이스에 적재한 후 다시 상위 도메인의 LC에게 IDR을 전달한다(PublishIDR 메시지). <그림 5>는 IDR이 생성되어 상위 LC로 전달되는 과정을 예시한다.

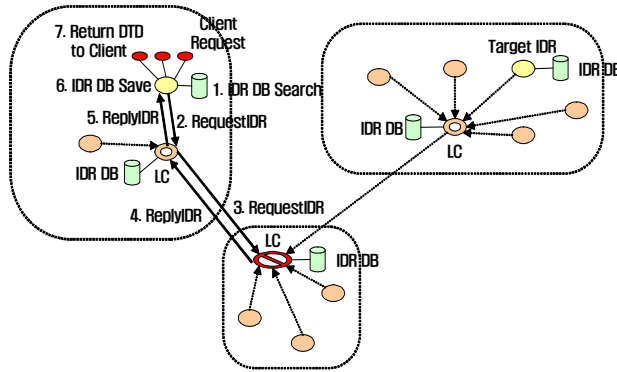


<그림 5> IDR의 등록 및 배포

#### □ IDR 검색

DTD 정보를 원하는 클라이언트가 특정 DTD 관리 서버에게 DTD 정보를 요청할 경우 DTD 관리 서버는 자체 IDR 데이터베이스를 검색하고 IDR을 찾을 경우 클라이언트에게 IDR의 정보를 제공한다. 자체 데이터베이스에 해당 IDR이 없을 경우 DTD 관리 서버는 도메인 LC에게 IDR 검색을 요청한다(RequestIDR). IDR 검색 요청을 받은 LC도 자체 IDR 데이터베이스를 검색하고 해당 IDR을 찾을 경우 요청한 DTD 관리 서버에게 IDR을 전달한다(ReplyIDR), 해당 IDR이 없을 경우 LC는 상위 도메인의 LC에게 IDR 검색을 요청한다(RequestIDR). 상위 LC도 동일한 과정을 반복한 후 IDR을 찾지 못하면 다시 상위 LC로 IDR 검색 요청을 반복하여 최종적으로 DTD 상호운용 구조의 최상위 LC에 이르기까지는 IDR을 획득할 수 있다. <그림 6>은 연쇄적인 IDR 검색 과정을 예시한다.





<그림 6> IDR의 검색

이러한 연쇄적인 IDR 검색 및 IDR 반환 과정은 참여한 DTD 관리 서버 및 LC들에게 해당 IDR을 전달하는 계기가 되므로 특정 DTD의 IDR은 수차례의 검색을 통해서 DTD 상호운용 구조의 상당 부분에 전파되게 된다. 따라서 차기에 동일한 IDR에 대한 요구가 발생하면 상기와 동일한 검색과정 없이 지역 DTD 관리 서버 또는 지역 LC가 해당 정보를 제공할 수 있게 된다. 또한, 일정 시간 이상 해당 IDR에 대한 검색 요청이 없을 경우 해당 IDR은 데이터베이스로부터 삭제되므로 IDR 데이터베이스는 IDR에 대한 신뢰성을 제공할 수 있게 되고 데이터베이스의 부피도 일정 수준으로 유지할 수 있게 된다.

□ IDEP 메시지 정의

IDEP 프로토콜은 CC 및 LC 등록, LC 선출, IDR 검색 및 관리, LC-Element-Element간의 존재 확인 등에 대한 11개 메시지를 정의한다. IDEP 메시지들은 OMG(Object Management Group) 그룹의 CORBA(Common Object Request Broker Architecture) 기술의 IDL(Interface Definition Language) 언어를 사용하여 정의하고 각 메시지의 엔코딩은 CORBA의 CDR(Common Data Representation)을 따른다.

**IDEP Message Header**

▷ IDEPHeader : IDEP 프로토콜의 모든 메시지의 선두에 위치하는 메시지 헤더

---

```

struct IDEPHeader {
    char protocol[3];
    Version IDEP_version;
    octet message_type;
    unsigned long message_size;
}

```

---

### CC 및 LC로의 등록 메시지

- ▷ RequestRegisterCC : 새로운 Element가 추가되거나 새로운 LC가 선출되었을 때 이를 CC에 통보하는 메시지

---

```

typedef unsigned long RequestID;
typedef boolean Status;

struct RequestRegisterCC {
    RequestID req_id;
    string serverName;
    string IPaddress;
    unsigned long IDRsize;
    Status value;
}

```

---

- ▷ ReplyRegisterCC : 새로운 Element 또는 LC의 CC 등록에 대하여 CC가 도메인 및 Service Level을 할당하는 메시지

---

```

typedef unsigned long DomainID;
typedef unsigned long ServiceLevel;

struct ReplyRegisterCC {
    RequestID req_id;
    DomainID domain_id;
    string IPaddress;
    ServiceLevel level;
    Status value;
}

```

---

- ▷ RequestRegister : 새로 추가된 Element가 도메인내의 Element 및 LC에게 등록하는 메시지

---

```

struct RequestRegister {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    Status value;
}

```

---

- ▷ ReplyRegisterLC : Element의 등록에 대하여 도메인내의 Element 및 LC가 응답하는 메시지

---

```

struct ReplyRegisterLC {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    Status value;
}

```

---

### LC 선출 메시지

- ▷ RequestVotingLC : 새로운 LC 선출을 요구하는 메시지

---

```

struct RequestVotingLC {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    Status value;
}

```

---

- ▷ ReplyVotingLC : 새로운 LC 선출 요구에 대하여 도메인내의 Element가 다른 Element들에게 타임스탬프를 실어 보내는 메시지

---

```

typedef unsigned long Timestamp;

struct ReplyVotingLC {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    Timestamp t_value;
    Status value;
}

```

---

- ▷ ConfirmVotingLC : 도메인내의 Element들이 새로운 LC를 승인하는 메시지

---

```

struct ReplyVotingLC {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    Status value;
}

```

---

### IDR 검색 및 관리 메시지

- ▷ PublishIDR : 새로운 IDR을 도메인 LC 및 상위 도메인 LC들에게 전파하는 메시지

---

```
struct PublishIDR {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    string IDRstream;
}
```

---

- ▷ RequestIDR : 도메인 LC 또는 상위 도메인 LC에게 IDR 검색을 요구하는 메시지

---

```
struct RequestIDR {
    RequestID req_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    sequence<octet> DTD_key;
}
```

---

- ▷ ReplyIDR : IDR을 획득한 LC가 IDR 검색을 요청한 대상에게 IDR을 반환하는 메시지

---

```
struct ReplyIDR {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
    octet ReplyStatus;
    string IDRstream;
}
```

---

### 기타

- ▷ Alive : 도메인 내의 LC, Element들이 서로에게 주기적으로 전송하는 생존 확인 메시지

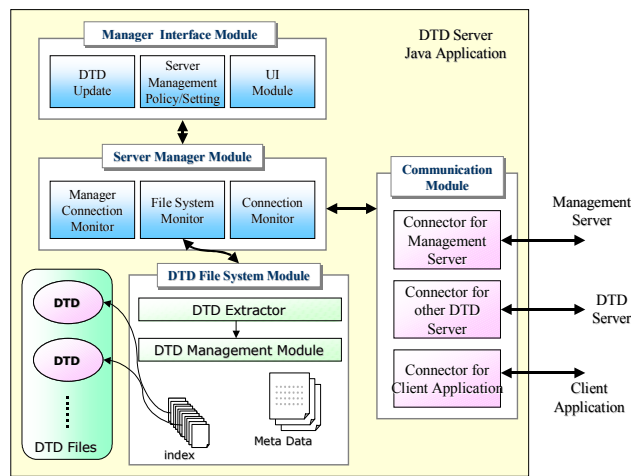
---

```
struct Alive {
    RequestID req_id;
    DomainID domain_id;
    string serverName;
    string IPaddress;
    ServiceLevel level;
}
```

---

### 3. DTD 관리 서버

DTD 관리 서버는 IDR 데이터베이스를 관리하며 IDR을 통한 DTD 검색으로 클라이언트에게 DTD 정보를 제공한다. 또한 동일 도메인에 속한 다른 Element 와 LC 선출, IDR 검색 등의 기능을 제공한다.



<그림 7> DTD 관리 서버의 구조

그림 7에서 제시하는 바와 같이 DTD 관리 서버는 DTD 관리 서버의 통합 관리를 위한 인터페이스를 제공하는 Manager Interface Module, DTD 관리 서버의 내부 모듈을 모니터링하는 Server Manager Module, IDR 데이터베이스 관리와 DTD 변환을 지원하는 DTD File System Module 및 클라이언트, LC, 타 DTD 관리 서버 등과의 통신을 담당하는 Communication Module 등으로 구성된다.

## IV. 결 론

기업 간 온라인 결제, 기업 간 ERP 시스템 연동, 디지털 정부의 사이버 행정의 도입 등 B2B 및 B2G 환경에서 전자문서의 사용이 확산되고 있다. 그러나

이러한 배경에도 불구하고 전자문서의 자동화된 처리는 매우 미진한 수준에 있다. B2B, B2G 환경에서 기업 간, 기업과 정부 간의 상거래 및 행정업무 수행 시에 교환되는 전자문서의 형식을 공용화하는데 XML DTD 기술이 사용될 경우 전자문서의 자동화된 처리가 가능하여 대상 기관의 시간과 비용 절감을 통한 큰 업무 효율의 개선이 예상된다.

본 논문에서는 B2B, B2G 환경에서 DTD의 사용을 활성화하고 각 기관에서 생성한 DTD를 효과적으로 검색하여 기관 간에 상호운용할 수 있는 DTD 상호운용 구조를 제안하였으며 DTD 상호운용 구조에서 DTD 정보의 교환을 지원하는 IDEP 프로토콜의 세부사항을 제시하였다. 본 논문에서 제시된 DTD 상호운용 인프라 구조 및 프로토콜은 DTD 기반 전자문서의 사용을 촉진하여 B2B, B2G 환경의 전자상거래 및 사이버 행정을 확산하는데 크게 기여할 것으로 기대한다.

## 참고문헌

1. The World Wide Web Consortium (W3C), Extensible Markup Language (XML) 1.0 (Fourth Edition), <http://www.w3.org/TR/2006/REC-xml-20060816/>, Aug. 2006.
2. The World Wide Web Consortium (W3C), Namespaces in XML 1.1 (Second Edition), <http://www.w3.org/TR/2006/REC-xml-names11-20060816/>, Aug. 2006.
3. The World Wide Web Consortium (W3C), Extensible Stylesheet Language (XSL) Version 1.1, <http://www.w3.org/TR/xsl/>, Dec. 2006.
4. International SGML/XML Users' Group (ISUG), A Gentle Introduction to SGML, <http://www.isgmlug.org/sgmlhelp/g-index.htm>
5. The World Wide Web Consortium (W3C), Document Object Model (DOM) Level 1 Specification, <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/> DOM.pdf, Oct. 1998.
6. The World Wide Web Consortium (W3C), XML Base, <http://www.w3.org/TR/xmlbase/>, Jun. 2001.
7. The World Wide Web Consortium (W3C), XML Schema Part 0: Primer Second Edition, <http://www.w3.org/TR/xmlschema-0/>, Oct. 2004.
8. The Object Management Group (OMG), Common Object Request Broker Architecture (CORBA/IIOP) 3.03, [http://www.omg.org/technology/documents/formal/corba\\_iiop.htm](http://www.omg.org/technology/documents/formal/corba_iiop.htm), Mar. 2004.
9. The Object Management Group (OMG), CORBA 3.0 - ORB Interoperability Architecture chapter, <http://www.omg.org/cgi-bin/apps/doclist.pl>, Feb. 2006.

## Abstract

### Design of Interoperable DTD Architecture supporting Interoperability of Digital Documents for B2B, B2G Environments

Park, Sang-yun\*

According as Internet based EC(Electronic Commerce) has been activated, More requirements for exchanging digital documents are being proposed in B2B(Business to Business) and B2G(Business to Government) environments. However, the documents can not be processed automatically, because the structures of documents in each site(companies, central/local governments) do not match.

Therefore, W3C(World Wide Web Consortium) has proposed XML(eXtensible Markup Language) DTD(Document Type Declaration) technology to support structuring documents. The XML DTD technology can support interoperability between EC sites. And, XML DTD based documents can be processed automatically in every sites. They will improve efficiency of services in B2B and B2G environments.

In this paper, we propose interoperable DTD architecture and IDEP (Interoperable DTD Exchange Protocol) protocol which supports searching DTDs and exchanging DTDs. This technology can popularize DTDs and digital documents.

Key Words : Interoperability, XML, DTD, EC(Electronic Commerce), B2B,  
B2G

---

\*Division of Computer Science & Information, Daelim College