



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2012년07월03일  
(11) 등록번호 10-1130386  
(24) 등록일자 2012년03월19일

(51) 국제특허분류(Int. Cl.)  
G06F 17/21 (2006.01)  
(21) 출원번호 10-2005-0103811  
(22) 출원일자 2005년11월01일  
심사청구일자 2010년10월25일  
(65) 공개번호 10-2006-0070418  
(43) 공개일자 2006년06월23일  
(30) 우선권주장  
11/018,405 2004년12월20일 미국(US)  
(56) 선행기술조사문헌  
EP1452966 A  
US20020129058 A1  
JP2004265402 A

(73) 특허권자  
마이크로소프트 코포레이션  
미국 워싱턴주 (우편번호 : 98052) 레드몬드 윈  
마이크로소프트 웨이  
(72) 발명자  
비숍, 앤드류 케이.  
미국 98052 워싱턴주 레드몬드 윈 마이크로소프  
트 웨이마이크로소프트 코포레이션 내  
존스, 브라이언 엠.  
미국 98052 워싱턴주 레드몬드 윈 마이크로소프  
트 웨이마이크로소프트 코포레이션 내  
(뒷면에 계속)  
(74) 대리인  
제일특허법인

전체 청구항 수 : 총 19 항

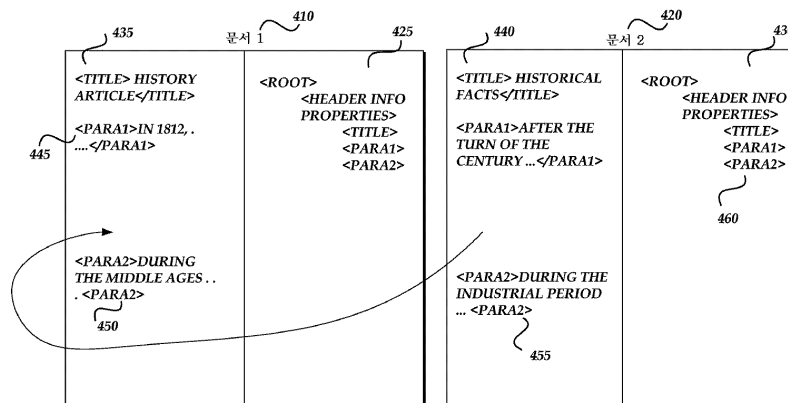
심사관 : 신유철

(54) 발명의 명칭 교체 포맷들을 가진 컨텍스트-프리 문서 부분들

(57) 요약

컴퓨터 생성 문서와의 연관 또는 컴퓨터 생성 문서 내의 삽입을 위해 텍스트 또는 데이터의 자기 기술 부분들이 제공된다. 개별 단락 또는 단락 그룹 또는 텍스트 조각과 같은 문서의 부분들에는 그 부분과 연관된 파일 포맷 타입, 스타일, 폰트, 리스트 타입 등과 같은 자기 기술 속성이 제공된다. 이러한 문서의 일부가 후속적으로 제2 문서 또는 제1 문서의 다른 위치에 링크 또는 복사 또는 이동되는 경우, 그 부분에 제공된 자기 기술 속성은 그 부분과 함께 제2 문서 또는 제1 문서의 다른 위치로 이동하게 된다. 콘텐츠 부분을 수신하는 애플리케이션은 그 부분에 제공된 속성에 따라 그 부분을 소비하고 표시할 수 있다.

대표도



(72) 발명자

**로스스칠러, 채드 비.**

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이마이크로소프트 코포레이션 내

**위커, 찰스 에스.**

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이마이크로소프트 코포레이션 내

**리틀, 로버트 에이.**

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이마이크로소프트 코포레이션 내

**빌라론, 손 알란**

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이마이크로소프트 코포레이션 내

---

**특허청구의 범위**

**청구항 1**

컴퓨터에 의해 실행될 때, 수신 문서 내의 콘텐츠 부분과 연관된 자기 기술(self-describing) 속성에 기초하여 상기 콘텐츠 부분을 통합하는 방법을 수행하는 명령들이 저장된 컴퓨터 판독 가능 기록 매체에 있어서,

상기 방법은

콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 가진 콘텐츠 부분을 수신하는 단계;

상기 콘텐츠 부분과 컴퓨터 생성 문서를 통합하기 위해 수신 애플리케이션에서 상기 콘텐츠 부분을 수신하는 단계;

상기 수신 애플리케이션에서, 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 파싱하는 단계 - 상기 자기 기술 속성은 적어도 하나의 마크업 언어 요소를 포함함 - ;

상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리하는 단계;

상기 컴퓨터 생성 문서에 첨부된 적어도 하나의 스키마 파일에 대해 상기 적어도 하나의 마크업 언어 요소를 검증하는(validating) 단계 - 상기 스키마 파일은 상기 적어도 하나의 스키마 파일에 리스트된 마크업 언어 요소들에 의한 상기 컴퓨터 생성 문서의 유효한 주석들을 정의하는 규칙들을 포함함 - ; 및

상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계 - 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계는 상기 적어도 하나의 마크업 언어 요소를 포함하는 상기 자기 기술 속성에 의해 상기 컴퓨터 생성 문서에 주석 달기하는 단계를 포함함 -

를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 2**

제1항에 있어서, 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 가진 콘텐츠 부분을 수신하는 단계는 상기 수신 애플리케이션에서 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 위한 선택의 표시를 수신하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 3**

제1항에 있어서, 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 파싱하는 단계는 상기 콘텐츠 부분을 포맷하는 파일 포맷을 결정하는 단계 및 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 위한 임의의 처리 요건을 결정하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 4**

제3항에 있어서, 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 위한 임의의 처리 요건을 결정하는 단계는 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 전에 상기 콘텐츠 부분이 다른 파일 포맷으로 변환되어야 하는지를 결정하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 5**

제4항에 있어서, 상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리하는 단계는 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 전에 상기 콘텐츠 부분을 상기 다른 파일 포맷으로 변환하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 6**

제3항에 있어서, 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 위한 임의의 처리 요건을 결정하는 단계는 상기 수신 애플리케이션이 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 위한 추가적인 기능을 요구하는지를 결정하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 7**

제6항에 있어서, 상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리하는 단계는,  
 상기 수신 애플리케이션에 의해 추가 기능에 액세스하는 단계; 및  
 상기 추가 기능으로 상기 콘텐츠 부분을 처리하는 단계  
 를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 8**

제1항에 있어서, 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계는 상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리한 후에 상기 콘텐츠 부분을 상기 컴퓨터 생성 문서 내로 복사하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 9**

제1항에 있어서, 상기 방법은 전자 파일 컨테이너 내에 상기 컴퓨터 생성 문서 및 상기 콘텐츠 부분을 저장하는 단계; 및 상기 콘텐츠 부분이 상기 컴퓨터 생성 문서와 어떻게 관련되는지를 나타내는 관계 표현을 제공하는 단계를 더 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 10**

제9항에 있어서, 상기 콘텐츠 부분이 상기 컴퓨터 생성 문서와 어떻게 관련되는지를 나타내는 관계 표현을 제공하는 단계는 상기 콘텐츠 부분이 상기 컴퓨터 생성 문서와 통합되는 상기 컴퓨터 생성 문서 내의 위치를 나타내는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 11**

제10항에 있어서, 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계는,  
 상기 콘텐츠 부분을 상기 컴퓨터 생성 문서 내의 주어진 위치에 링크하는 단계; 및  
 상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리한 후에 상기 주어진 위치에 상기 콘텐츠 부분을 제공하기 위해 상기 콘텐츠 부분을 호출하는 단계  
 를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 12**

제1항에 있어서, 상기 방법은 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 가진 콘텐츠 부분을 수신하기 전에,  
 수신 애플리케이션에 의한 사용을 위한 콘텐츠 부분을 생성하는 단계; 및  
 상기 콘텐츠 부분을 포맷하는 파일 포맷을 상기 수신 애플리케이션에 대하여 식별하기 위해 상기 콘텐츠 부분에 자기 기술 속성을 적용하는 단계  
 를 더 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 13**

수신 문서 내의 콘텐츠 부분과 연관된 자기 기술 속성에 기초하여 상기 콘텐츠 부분을 통합하는 방법에 있어서,  
 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 가진 콘텐츠 부분을 수신하는 단계;  
 상기 콘텐츠 부분과 컴퓨터 생성 문서를 통합하기 위해 수신 애플리케이션에서 상기 콘텐츠 부분을 수신하는 단계;  
 상기 수신 애플리케이션에서, 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 파싱하여 상기 콘텐츠 부분을 포맷하는 파일 포맷을 결정하고 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서 사이의 관계를 결정하는 단계 - 상기 자기 기술 속성은 적어도 하나의 마크업 언어 요소를 포함함 - ;  
 전자 파일 컨테이너 내에 상기 컴퓨터 생성 문서 및 상기 콘텐츠 부분을 저장하는 단계;

상기 콘텐츠 부분이 상기 컴퓨터 생성 문서와 어떻게 관련되는지를 나타내는 관계 표현을 제공하는 단계;

상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리하는 단계;

상기 컴퓨터 생성 문서에 첨부된 적어도 하나의 스키마 파일에 대해 상기 적어도 하나의 마크업 언어 요소를 검증하는 단계 - 상기 스키마 파일은 상기 적어도 하나의 스키마 파일에 리스트된 마크업 언어 요소들에 의한 상기 컴퓨터 생성 문서의 유효한 주석들을 정의하는 규칙들을 포함함 - ; 및

상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계 - 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계는 상기 적어도 하나의 마크업 언어 요소를 포함하는 상기 자기 기술 속성에 의해 상기 컴퓨터 생성 문서에 주석 달기하는 단계를 포함함 -

를 포함하는 콘텐츠 부분 통합 방법.

#### 청구항 14

제13항에 있어서, 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계는

상기 컴퓨터 생성 문서 내의 주어진 위치에 상기 콘텐츠 부분을 링크하는 단계; 및

상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리한 후에 상기 주어진 위치에 상기 콘텐츠 부분을 제공하기 위해 상기 콘텐츠 부분을 호출하는 단계

를 포함하는 콘텐츠 부분 통합 방법.

#### 청구항 15

컴퓨터에 의해 실행될 때, 수신 문서 내의 콘텐츠 부분과 연관된 자기 기술 속성에 기초하여 상기 콘텐츠 부분을 통합하는 방법을 수행하는 명령들이 저장된 컴퓨터 판독 가능 기록 매체에 있어서,

상기 방법은

콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 갖는 콘텐츠 부분을 수신하는 단계;

상기 콘텐츠 부분과 컴퓨터 생성 문서를 통합하기 위해 수신 애플리케이션에서 상기 콘텐츠 부분을 수신하는 단계;

상기 수신 애플리케이션에서, 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 파싱하는 단계 - 상기 자기 기술 속성은 적어도 하나의 마크업 언어 요소를 포함함 - ;

상기 콘텐츠 부분을 포맷하는 파일 포맷에 따라 상기 콘텐츠 부분을 처리하는 단계;

상기 컴퓨터 생성 문서에 첨부된 적어도 하나의 스키마 파일에 대해 상기 적어도 하나의 마크업 언어 요소를 검증하는 단계 - 상기 적어도 하나의 스키마 파일은 상기 적어도 하나의 스키마 파일에 리스트된 마크업 언어 요소들에 의한 상기 컴퓨터 생성 문서의 유효한 주석들을 정의하는 규칙들을 포함하고, 상기 검증하는 단계는 상기 적어도 하나의 마크업 언어 요소가 상기 적어도 하나의 스키마 파일의 규칙들을 위반하는지를 결정하는 단계를 포함하며, 상기 규칙들은 텍스트 입력의 최대 문자 길이와 무효 마크업 언어 요소의 삽입 중 적어도 하나를 포함함 - ;

상기 컴퓨터 생성 문서에서 무효 마크업 언어의 수신을 판정할 때에 에러 메시지를 생성하는 단계; 및

상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계 - 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하는 단계는 검증된 상기 적어도 하나의 마크업 언어 요소를 포함하는 상기 자기 기술 속성에 의해 상기 컴퓨터 생성 문서에 주석 달기하는 단계를 포함함 -

를 포함하는 컴퓨터 판독 가능 기록 매체.

#### 청구항 16

제15항에 있어서, 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 가진 콘텐츠 부분을 수신하는 단계는 상기 수신 애플리케이션에서 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 위한 선택의 표시를 수신하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 17**

제15항에 있어서, 상기 콘텐츠 부분을 포맷하는 파일 포맷을 식별하는 자기 기술 속성을 파싱하는 단계는 상기 콘텐츠 부분을 포맷하는 파일 포맷을 결정하는 단계 및 상기 콘텐츠 부분과 상기 컴퓨터 생성 문서를 통합하기 위한 임의의 처리 요건을 결정하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 18**

제17항에 있어서, 상기 콘텐츠 부분을 상기 컴퓨터 생성 문서와 통합하기 위한 임의의 처리 요건을 결정하는 단계는 상기 콘텐츠 부분을 상기 컴퓨터 생성 문서와 통합하기 전에 상기 콘텐츠 부분이 다른 파일 포맷으로 변환되어야 하는지 여부를 결정하는 단계를 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 19**

제15항에 있어서,

상기 방법은,

상기 컴퓨터 생성 문서 및 상기 콘텐츠 부분을 전자 파일 컨테이너에 저장하는 단계; 및

상기 콘텐츠 부분이 상기 컴퓨터 생성 문서에 어떻게 관련되는지를 나타내는 관계 표현을 제공하는 단계를 더 포함하는 컴퓨터 판독 가능 기록 매체.

**청구항 20**

삭제

**명세서**

**발명의 상세한 설명**

**발명의 목적**

**발명이 속하는 기술 및 그 분야의 종래기술**

- [0023] <관련 출원의 상호 참조>
- [0024] 본 발명은 CIP(continuation-in-part)으로서, 본 명세서에서 충분히 설명하는 바와 같이 본 명세서에 참조로 반영된, "컨텍스트-프리 문서 부분들"이라는 제목으로 2003년 12월 9일자로 출원된 미국 특허 출원 번호 10/731,900에 대한 우선권을 주장한다.
- [0025] <저작권 고시>
- [0026] 이 특허 명세서의 일부는 저작권의 보호를 받는 자료를 포함하고 있다. 저작권 소유자는 미국 특허 상표청 특허 파일 또는 레코드에 나타내 있는 대로의 특허 문서 또는 특허 명세서의 어느 누군가에 의한 팩시밀리 재생에 반대하지 않지만, 그 외의 경우에는 모든 저작권은 보유된다.
- [0027] 본 발명은 일반적으로 상이한 컴퓨터 생성 문서들의 데이터 관리에 관한 것이다. 구체적으로, 본 발명은 교체 포맷들 및 단일 컴퓨터 생성 문서와 연관된 다른 속성들을 가진 컨텍스트-프리 문서 부분들의 이용에 관한 것이다.
- [0028] 컴퓨터 시대의 도래와 함께, 컴퓨터 및 소프트웨어 사용자들은 작성, 계산, 체계화, 프리젠테이션의 준비, 전자 메일의 송수신, 작곡 등을 돕는 사용자 친화 소프트웨어 애플리케이션에 점차 익숙해져 왔다. 예를 들어, 현대의 전자 워드 프로세싱 애플리케이션들은 사용자들이 다양한 유용한 문서를 준비하는 것을 가능하게 한다. 현대의 스프레드시트 애플리케이션들은 사용자들이 데이터를 입력, 조작 및 체계화하는 것을 가능하게 한다. 현대의 전자 슬라이드 프리젠테이션 애플리케이션들은 사용자들이 텍스트, 픽처, 데이터 또는 다른 유용한 객체를 포함하는 다양한 슬라이드 프리젠테이션을 생성하는 것을 가능하게 한다.
- [0029] 다양한 포맷팅 타입에 따라 이러한 애플리케이션들에 의해 생성된 문서들 및 관련 속성들을 표현하기 위한 방

법 및 시스템이 개발되어 왔다. 예를 들어, 문서들은 바이너리 포맷, 하이퍼텍스트 마크업 언어(HTML) 포맷, 리치 텍스트 포맷(RTF), 확장형 마크업 언어 포맷(XML) 등으로 표현될 수 있다. 또한, 단락, 테이블, 스타일, 폰트, 리스트 등을 포함하는 많은 문서 구성 컴포넌트가 존재한다. 문서의 몇몇 컴포넌트는 제1 컴포넌트에 구조적인 제한을 제공하기 위해 다른 컴포넌트를 참조한다. 예를 들어, 문서 내의 단락은 이 단락이 문서에서 어떻게 보일 것인지를 정의하는 특정 포매팅 타입 또는 스타일 설정을 참조할 수 있다. 다른 예로서, 문서의 단락 컴포넌트는 문서 내의 특정 리스트 구조의 일부일 수 있다. 일반적으로, 스타일과 같이 문서의 다른 객체들이 참조할 수 있는 다수의 속성 및 객체가 존재하므로, 문서 내에는 스타일 및 폰트 정의와 같은 다양한 속성 및 객체가 배치되는 헤더가 필요하다.

[0030] 불행히도, 사용자가 예를 들어 단락, 텍스트 부분, 이미지, 차트 등과 같은 추가 콘텐츠를 특정 문서에 추가하려고 할 때, 추가 콘텐츠가 추가될 문서의 포매팅과 다른 포매팅에 따라 포맷되는 문제가 발생한다. 예를 들어, 주어진 애플리케이션은 HTML 포맷으로 모든 리치 텍스트를 저장할 수 있다. 이러한 애플리케이션에 의해 생성된 문서로부터의 콘텐츠를 non-HTML 애플리케이션으로 가져가기 위하여는 HTML 포맷 콘텐츠는 다른 포맷으로 변환되어야 한다. 그렇지 않고, 예를 들어 HTML 포맷 콘텐츠가 예를 들어 XML과 같은 다른 포매팅에 따라 포맷된 문서에 추가되는 경우, 추가된 콘텐츠의 원래의 포매팅이 손상되거나, 추가 콘텐츠 모두를 로딩하지 못할 수 있다.

[0031] 본 발명은 이들 및 다른 사항들을 고려하여 이루어졌다.

### 발명이 이루고자 하는 기술적 과제

[0032] 본 발명은 문서 내에 텍스트 또는 데이터의 자기 기술 부분들을 제공함으로써 상기 및 다른 문제들을 해결한다. 본 발명의 일 실시예에 따르면, 개별 단락 또는 단락 그룹 또는 텍스트 조각과 같은 문서의 부분들에는 그 부분과 연관된 파일 포맷 타입, 스타일, 폰트, 리스트 타입 등과 같은 자기 기술 속성이 제공된다. 이러한 문서의 일부가 후속적으로 제2 문서 또는 제1 문서의 다른 위치에 복사 또는 이동되는 경우, 그 부분에 제공된 자기 기술 속성은 그 부분과 함께 제2 문서 또는 제1 문서의 다른 위치로 이동하게 된다. 결과적으로, 제2 문서 또는 제1 문서를 준비하고 표시하는 애플리케이션은 그 부분에 대해 제공된 속성에 따라 그 부분을 소비하고 표시할 수 있다.

[0033] 다른 실시예에 따르면, 문서를 표시하고 편집하기 위하여 문서 컨테이너가 설정된다. 문서의 자기 기술 부분들 또는 컴포넌트들은 문서 컨테이너 내에 유지되고 편집될 수 있다. 대안으로, 문서의 자기 기술 부분들 또는 컴포넌트들은 문서 컨테이너와 별개로 유지되고 편집될 수 있으나, 문서 컨테이너 내로부터 지시될 수 있다. 문서의 개방 시간에 문서 컨테이너 내로부터 지시되는 모든 문서 부분들 또는 컴포넌트들은, 상이한 문서 부분들 또는 컴포넌트들의 개별 파일 포맷 타입 또는 다른 속성에 따라 이들 각각을 소비할 수 있는 애플리케이션에 의해 병합되고 표시될 수 있다.

[0034] 본 발명의 상기 및 다른 특징들, 이점들 및 양태들은 개시된 실시예들에 대한 아래의 상세한 설명의 검토로부터, 그리고 첨부된 도면 및 청구범위를 참조하여 보다 명확히 이해될 수 있다.

### 발명의 구성 및 작용

[0035] 간략하게 전술한 바와 같이, 본 발명의 실시예들은 텍스트 또는 데이터의 부분들을 문서 내에 삽입하거나 문서에 의해 참조하기 위한 방법 및 시스템에 관한 것으로서, 삽입 또는 참조되는 텍스트 또는 데이터의 부분들은 텍스트 또는 데이터의 부분들이 삽입되거나 수신 문서와의 병합을 위해 참조되는 문서와 상이한 파일 포맷 타입 및 다른 속성에 따라 포맷된다. 이들 실시예는 조합될 수 있고, 다른 실시예들이 이용될 수 있으며, 본 발명의 사상 또는 범위를 벗어나지 않고 구조적 변경이 이루어질 수 있다. 따라서, 아래의 상세한 설명은 제한적인 의미로 간주되어서는 안 되며, 본 발명의 범위는 첨부된 청구범위 및 그 균등물에 의해 정의된다.

[0036] 이제, 여러 도면을 통해 동일 부호가 동일 요소를 나타내는 도면을 참조하여 본 발명의 양태들 및 예시적인 동작 환경이 설명된다. 도 1 및 다음 설명은 본 발명이 구현될 수 있는 적절한 컴퓨팅 환경의 간단하고 일반적인 설명을 제공하는 것을 목적으로 한다. 본 발명은 개인용 컴퓨터의 운영 체제에서 실행되는 애플리케이션 프로그램과 함께 실행되는 프로그램 모듈과 관련하여 설명되지만, 이 분야의 전문가들은 본 발명이 다른 프로그램 모듈과 조합하여서도 구현될 수 있다는 것을 인식할 것이다.

[0037] 일반적으로, 프로그램 모듈은 루틴, 프로그램, 컴포넌트, 데이터 구조, 및 특정 태스크를 수행하거나 특정 추상 데이터 타입을 구현하는 다른 타입의 구조를 포함한다. 더욱이, 이 분야의 전문가들은 본 발명이 핸드헬드



장치, 멀티프로세서 시스템, 마이크로프로세서 기반 또는 프로그래머블 소비자 전자 장치, 미니 컴퓨터, 메인 프레임 컴퓨터 등을 포함하는 다른 컴퓨팅 시스템 구성과 함께 실시될 수 있다는 것을 이해할 것이다. 본 발명은 통신 네트워크를 통해 링크된 원격 프로세싱 장치들에 의해 태스크가 수행되는 분산 컴퓨팅 환경에서도 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 양자에 위치할 수 있다.

[0038] 이제 도 1을 참조하여, 본 발명의 다양한 실시예를 실시하기 위한 개인용 컴퓨터(2)의 예시적인 컴퓨터 아키텍처가 설명된다. 도 1에 도시된 컴퓨터 아키텍처는 중앙 처리 유닛(CPU; 4), RAM(8) 및 ROM(10)을 포함하는 시스템 메모리(6), 및 메모리를 CPU(4)에 결합시키는 시스템 버스(12)를 포함하는 통상의 개인용 컴퓨터를 나타낸다. 예를 들어 시동 중에 컴퓨터 내의 요소들 간의 정보 전송을 돕는 기본 루틴을 포함하는 기본 입출력 시스템이 ROM(10)에 저장되어 있다. 개인용 컴퓨터(2)는 또한 운영 체제(16), 애플리케이션 프로그램(305)과 같은 애플리케이션 프로그램 및 데이터를 저장하기 위한 대용량 저장 장치(14)를 포함한다.

[0039] 대용량 저장 장치(14)는 버스(12)에 접속된 대용량 저장 장치 제어기(도시되지 않음)를 통해 CPU(4)에 접속된다. 대용량 저장 장치(14) 및 관련 컴퓨터 판독 가능 매체는 개인용 컴퓨터(2)의 비휘발성 저장을 제공한다. 여기에 포함된 컴퓨터 판독 가능 매체에 대한 설명은 하드 디스크 또는 CD-ROM 드라이브와 같은 대용량 저장 장치를 지칭하지만, 컴퓨터 판독 가능 매체는 개인용 컴퓨터(2)에 의해 액세스될 수 있는 임의의 이용 가능 매체일 수 있다는 점을 이 분야의 전문가가 이해하여야 한다.

[0040] 예를 들어, 컴퓨터 판독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있으나 이에 한하지 않는다. 컴퓨터 저장 매체는 컴퓨터 판독 가능 명령, 데이터 구조, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 분리식 및 비분리식 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EPROM, EEPROM, 플래시 메모리 또는 다른 반도체 메모리 기술, CD-ROM, DVD 또는 다른 광학 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 다른 자기 저장 장치, 또는 원하는 정보를 저장하는 데 사용할 수 있고 컴퓨터로 액세스할 수 있는 임의의 다른 매체를 포함하지만 이에 한하지 않는다.

[0041] 본 발명의 다양한 실시예에 따르면, 개인용 컴퓨터(2)는 인터넷과 같은 TCP/IP 네트워크(18)를 통한 원격 컴퓨터들에 대한 논리적 접속을 이용하여 네트워크 환경에서 동작할 수 있다. 개인용 컴퓨터(2)는 버스(12)에 접속된 네트워크 인터페이스 유닛(2)을 통해 TCP/IP 네트워크(18)에 접속할 수 있다. 네트워크 인터페이스 유닛(20)은 다른 타입의 네트워크 및 원격 컴퓨터 시스템에 접속하는 데에도 사용될 수 있다는 것을 이해하여야 한다. 개인용 컴퓨터(2)는 또한 키보드 또는 마우스(도시되지 않음)를 포함하는 다수의 장치로부터 입력을 수신하고 처리하기 위한 입출력 제어기(22)를 포함할 수 있다. 유사하게, 입출력 제어기(22)는 표시 스크린, 프린터 또는 다른 타입의 출력 장치로 출력을 제공할 수 있다.

[0042] 간단히 전술한 바와 같이, 워싱턴 레드먼드의 마이크로소프트사의 윈도우 XP 운영 체제와 같이 네트워크된 개인용 컴퓨터의 동작을 제어하는 데 적합한 운영 체제(16)를 포함하는 다수의 프로그램 모듈 및 데이터 파일이 개인용 컴퓨터(2)의 대용량 저장 장치(14) 및 RAM(8)에 저장될 수 있다. 대용량 저장 장치(14) 및 RAM(8)은 또한 하나 이상의 애플리케이션 프로그램을 저장할 수 있다. 구체적으로, 대용량 저장 장치(14) 및 RAM(8)은 전자 문서(310)를 생성하고 편집하기 위한 애플리케이션 프로그램(305)을 저장할 수 있다. 예를 들어, 애플리케이션 프로그램(305)은 워드 프로세싱 애플리케이션 프로그램, 스프레드시트 애플리케이션, 콘택 애플리케이션 등을 포함할 수 있다. 다른 타입의 전자 문서를 생성하고 편집하기 위한 애플리케이션 프로그램도 본 발명의 다양한 실시예에서 이용될 수 있다. 후술하는 스키마 파일(330) 및 네임 스페이스/스키마 라이브러리(400)도 도시되어 있다.

[0043] 본 발명의 예시적인 실시예들은 객체 지향 프로그래밍 환경에서 상이한 소프트웨어 객체들 간의 통신에 의해 구현된다. 본 발명의 실시예들에 대한 아래의 설명을 위해, 객체 지향 프로그래밍 환경의 컴포넌트들을 간단히 설명하는 것이 유용하다. 도 2는 객체 지향 프로그래밍 모델에 따른 소프트웨어 객체들 간의 상호작용을 나타내는 간단한 블록도이다. 객체 지향 프로그래밍 환경에 따르면, 제1 객체(210)는 소프트웨어 코드, 실행 가능 메소드, 속성 및 파라미터를 포함할 수 있다. 마찬가지로, 제2 객체(220)도 소프트웨어 코드, 실행 가능 메소드, 속성 및 파라미터를 포함할 수 있다.

[0044] 제1 객체(210)는 제2 객체(220)와 통신하여 메시지 호(230)를 통해 제2 객체(220)를 호출함으로써 제2 객체(220)로부터 정보 또는 기능을 얻을 수 있다. 객체 지향 프로그래밍 환경 분야의 전문가에게 잘 알려져 있듯이, 제1 객체(210)는 서로로부터 정보 및 기능을 얻기 위하여 2개의 상이한 소프트웨어 객체(210, 220)가 서



로 통신하는 것을 가능하게 하는 애플리케이션 프로그래밍 인터페이스(API)를 통해 제2 객체(220)와 통신할 수 있다. 예를 들어, 제1 객체(210)가 제2 객체(220)에 포함된 메소드에 의해 제공되는 기능을 요구하는 경우, 제1 객체(210)는 제2 객체(220)로 메시지 호(230)를 전송할 수 있는데, 여기에서 제1 객체는 요구된 메소드를 식별하며, 식별된 메소드를 동작시키기 위해 제2 객체에 의해 요구되는 임의의 필요한 파라미터를 제2 객체로 전송한다. 제2 객체(220)가 제1 객체로부터 호를 수신하면, 제2 객체는 제공된 파라미터에 기초하여 호출된 메소드를 실행하고, 실행된 메소드로부터 얻은 값을 포함하는 리턴 메시지(250)를 다시 제1 객체(210)로 전송한다.

[0045] 예를 들어, 본 발명의 실시예들과 관련하여, 그리고 후술하는 바와 같이, 제1 객체(210)는, 제1 객체가 문서 내의 지정된 XML 요소의 검증을 요구하는 메소드를 식별하는 확장형 마크업 언어 스키마 검증 객체와 같은 제2 객체에 메시지를 전달하는 제3자 주문 애플리케이션(a third party customized application)일 수 있는데, 상기 지정된 XML 요소는 식별된 메소드와 함께 제1 객체에 의해 전달되는 파라미터이다. 이 예에 따르면, 제1 객체로부터의 호 수신시, 스키마 검증 객체는 지정된 XML 요소 상에서 식별된 메소드를 실행하여, 검증된 XML 요소와 연관된 결과 또는 값의 형태로 제1 객체로 메시지를 리턴한다. 간단히 전송한 객체 지향 프로그래밍 환경의 동작은 이 분야의 전문가에게 공지되어 있다.

[0046] 후술하는 바와 같이, 본 발명의 실시예들은 확장형 마크업 언어(XML)의 컴포넌트들의 사용, 주문화 및 적용에 있어서 소프트웨어 객체들의 상호작용을 통해 구현된다. 도 3은 문서, 첨부 스키마 파일 및 스키마 검증 기능 모듈 사이의 상호작용을 나타내는 블록도이다. 이 분야의 전문가에게 공지되어 있듯이, 확장형 마크업 언어(XML)는 사용자가 문서 내의 텍스트 또는 데이터에 적용되고 또한 관련 태그들이 적용되는 텍스트 또는 데이터를 정의하는 태그 네임들을 생성하는 것을 허용함으로써 문서 내의 텍스트 또는 데이터를 기술하는 메소드를 제공한다. 예를 들어, 도 3을 참조하면, 애플리케이션(305)으로 생성한 문서(310)는 XML 태그들(315, 320, 325)로 마크업된 텍스트를 포함한다. 예를 들어, 텍스트 "Greetings"에는 XML 태그 <title>이라는 주석이 달린다. 텍스트 "My name is Sarah"에는 <body> 태그가 달린다. XML에 따르면, <title> 및 <body> 태그의 생성자는 이들 태그가 적용될 태그들을 기술하기 위한 그 자신의 태그들을 자유롭게 생성한다. 이어서, 임의의 다운스트림 소비 애플리케이션 또는 컴퓨팅 머신에 텍스트에 적용된 태그들의 정의에 관한 명령들이 제공되는 한, 이 애플리케이션 또는 컴퓨팅 머신은 이 태그들에 따라 데이터를 이용할 수 있다. 예를 들어, 다운스트림 애플리케이션이 이 애플리케이션에 의해 처리된 기사 또는 출판물의 제목으로서 정의된 텍스트를 추출하도록 프로그래밍된 경우, 이 애플리케이션은 도 3에 도시된 바와 같이 문서(310)를 파싱하고 텍스트 "Greetings"를 추출할 수 있는데, 이는 그 텍스트에 태그 <title>이라는 주석이 달려 있기 때문이다. 도 3에 도시된 문서(310)에 대한 특정 XML 태그 네이밍의 생성자는 텍스트 또는 데이터에 적용된 태그들과 연관된 정의를 제3자(third parties)가 제공받는 한 제3자에 의해 이용될 수 있는 문서(310)에 포함된 텍스트 또는 데이터에 대한 유용한 설명을 제공한다.

[0047] 본 발명의 실시예들에 따르면, 문서(310)에 입력된 텍스트 및 XML 마크업은 다양한 상이한 파일 포맷에 따라, 그리고 문서(310)를 생성한 애플리케이션(305)의 원시 프로그래밍 언어(native programming language)에 따라 저장될 수 있다. 예를 들어, 텍스트 및 XML 마크업은 워드 프로세싱 애플리케이션, 스프레드시트 애플리케이션 등에 따라 저장될 수 있다. 대안으로, 문서(310)에 입력된 텍스트 및 XML 마크업은 XML 포맷으로 저장될 수 있음으로써 텍스트 또는 데이터, 임의 적용된 XML 마크업, 및 폰트, 스타일, 단락 구조 등과 같은 임의의 포맷팅이 XML 표현으로 저장될 수 있다. 따라서, XML로서 저장된 데이터를 이해할 수 있는 다운스트림 또는 제3자 애플리케이션은 이렇게 XML 표현으로 저장된 텍스트 또는 데이터를 열어 소비할 수 있다. 문서(310)의 텍스트 및 XML 마크업 및 관련 포맷팅 및 다른 속성을 XML로 저장하는 것에 대한 상세한 설명을 위하여, 본 명세서에 충분히 설명된 바와 같이 본 명세서에 참조로 반영된, 2002년 6월 28일자로 출원된 "Word Processing Document Stored in a Single XML File that may be Manipulated by Applications that Understanding XML"이라는 제목의 미국 특허 출원 10/187,060을 참조하라. 본 발명에 따른 예시적인 스키마는 본 명세서에 그 전체가 참조로 반영된, 2003년 12월 2일자로 출원된 "Mixed Content Flexibility"라는 제목의 미국 특허 출원 10/726,077의 11페이지에서부터 개시되어 있다.

[0048] 도 3에 도시된 바와 같이 텍스트 또는 데이터에 적용된 XML 마크업 요소들(태그들)에 대한 정의 프레임워크를 제공하기 위하여, 마크업되고 저장된 데이터의 사용자 및 소비자가 문서의 생성자에 의해 설계된 XML 태깅 정의를 이해하는 것을 가능하게 하기 위해 필요한 정보를 포함하는 XML 스키마 파일이 생성된다. 이 분야에서 네임 스페이스 또는 XSD 파일로도 지칭되는 각각의 스키마 파일은 바람직하게는 주어진 스키마 파일에 따라 문서에 적용될 수 있는 모든 XML 요소들(태그들)의 리스팅을 포함한다. 예를 들어, 도 3에 도시된 스키마 파일(330)은 XML 요소들의 속성들 또는 스키마 파일에 따른 XML 요소들이 주석으로 첨부될 수 있는 텍스트 또는

데이터와 연관된 제한 및/또는 규칙을 포함하는 문서(310)에 적용될 수 있는 소정의 XML 요소들의 정의를 포함하는 스키마 파일일 수 있다. 예를 들어, 도 3에 도시된 스키마 파일(330)을 참조하면, 이 스키마 파일은 네임 스페이스 "intro"에 의해 식별되며, <intro card>의 루트 요소를 포함한다.

[0049] 스키마 파일(330)에 따르면, <intro card> 요소는 스키마 파일의 루트 요소로서, 또한 두 자식 요소 <title> 및 <body>에 대한 부모 요소로서 기능한다. 이 분야에 공지되어 있듯이, 단일 루트 요소 아래에 다수의 부모 요소가 정의될 수 있으며, 각각의 부모 요소 아래에 다수의 자식 요소가 정의될 수 있다. 그러나, 일반적으로 주어진 스키마 파일(330)은 하나의 루트 요소만을 포함한다. 도 3을 계속 참조하면, 스키마 파일(330)은 또한 <title> 및 <body> 요소들 각각에 대한 속성들(340, 350)을 포함한다. 속성들(340, 345)은 문서(310) 내의 텍스트 또는 데이터에 각각의 요소들을 적용하는 것과 연관된 추가적인 정의 또는 규칙을 제공할 수 있다. 예를 들어, 속성(345)은 <title> 요소가 주석으로 첨부된 텍스트는 길이가 25 문자 이하이어야 한다는 것을 정의한다. 따라서, 길이가 25 문자를 넘는 텍스트에 <title> 요소 또는 태그가 주석으로 첨부된 경우에는, 그 텍스트에 대해 시도된 주석 첨부는 스키마 파일(330)에 포함된 정의에 따라 무효로 될 것이다.

[0050] XML 요소들에 속성들과 같은 정의 또는 규칙을 적용함으로써, 스키마 생성자는 주어진 스키마 파일과 연관된 문서에 포함된 데이터의 구조를 지시할 수 있다. 예를 들어, 이력서에 적용되는 XML 마크업을 정의하기 위한 스키마 파일(330)의 생성자가 이력서의 경력 부분이 4개보다 많은 현재 또는 이전 직업 엔트리를 포함하는 것을 원하지 않는 경우, 스키마 파일(330)의 생성자는 예를 들어 스키마 파일(330)에 따라 경력 텍스트가 유효하게 하기 위하여 4개보다 많은 현재 또는 과거 직업 엔트리가 <experience> 태그들 사이에 입력되는 것을 허용하지 않도록 <experience> 요소의 속성을 정의할 수 있다. 이 분야의 전문가에게 공지되어 있듯이, 스키마 파일(330)은 주어진 문서(310)에 첨부되거나, 첨부된 스키마 파일에 정의된 허용 가능한 XML 마크업의 주어진 문서(310)에 대한 적용을 위해 주어진 문서(310)와 연관될 수 있다. 일 실시예에 따르면, 첨부 또는 연관된 스키마 파일(330)의 XML 요소들로 마크업된 문서(310)는 첨부 또는 연관된 스키마 파일(330)을 식별하는 네임 스페이스와 연관된 균일 자원 식별자(URI)를 지시함으로써 첨부 또는 연관된 스키마 파일을 지시할 수 있다.

[0051] 본 발명의 실시예들에 따르면, 문서(310)는 다수의 첨부 스키마 파일을 가질 수 있다. 즉, 문서(310)의 생성자는 하나 이상의 스키마 파일로부터 XML 마크업의 주석 첨부를 위한 프레임워크를 제공하기 위하여 문서(310)에 하나 이상의 스키마 파일(330)을 연관시키거나 첨부할 수 있다. 예를 들어, 문서(310)는 회계 데이터와 연관된 텍스트 또는 데이터를 포함할 수 있다. 문서(310)의 생성자는 다수의 회계 제도와 연관된 XML 마크업 및 정의를 포함하는 XML 스키마 파일들(330)을 연관시키기를 원할 수 있다. 따라서, 문서(310)의 생성자는 하나 이상의 회계 제도로부터의 XML 스키마 파일(330)을 문서(310)와 연관시킬 수 있다. 또한, 주어진 XML 스키마 파일(330)은 회계 데이터를 원하는 포맷으로 배열하기 위한 템플릿과 같은 특정 문서 구조와 연관될 수 있다.

[0052] 본 발명의 실시예들에 따르면, XML 스키마 파일들 및 관련 문서 솔루션들의 집합이 문서(310)와 별개로 배치된 네임 스페이스 또는 스키마 라이브러리에 유지될 수 있다. 문서(310)는 문서(310)에 첨부되거나 연관된 하나 이상의 스키마 파일과 연관된 네임 스페이스 또는 스키마 라이브러리에 URI에 대한 포인터를 포함할 수 있다. 문서(310)가 하나 이상의 관련 스키마 파일로부터 정보를 요구할 때, 문서(310)는 네임 스페이스 또는 스키마 라이브러리를 지시하여 원하는 스키마 정의를 얻는다. 네임 스페이스 또는 스키마 라이브러리의 동작의 이용에 대한 상세한 설명을 위하여, 본 명세서에 충분히 설명된 바와 같이 참조로 반영된, 2002년 6월 27일자로 출원된 "System and Method for Providing Namespace Related Information"이라는 제목의 미국 특허 출원 10/184,190, 및 2002년 6월 27일자로 출원된 "System and Method for Obtaining and Using Namespace Related Information for Opening XML Documents"라는 제목의 미국 특허 출원 10/185,940을 참조하라. 네임 스페이스 또는 스키마 라이브러리로부터 XML 스키마 파일 및 관련 솔루션과 같은 소프트웨어 컴포넌트를 다운로드하기 위한 메카니즘에 대한 상세한 설명을 위하여, 2002년 6월 5일자로 출원된 "Mechanism for Downloading Software Components from a Remote Source for Use by a Local Software Application"라는 제목의 미국 특허 출원 10/164,260을 참조하라.

[0053] 도 3을 계속 참조하면, 전술한 바와 같이 문서(310)에 첨부되거나 연관된 XML 스키마 파일(330)에 대해 문서(310)에 적용된 XML 마크업을 검증하기 위한 스키마 검증 기능 모듈(350)이 도시되어 있다. 전술한 바와 같이, 스키마 파일(330)은 허용 가능한 XML 요소들 및 관련 속성들을 설명하고, 연관된 스키마 파일(330)로부터의 XML 마크업에 의한 문서(310)의 유효한 주석 첨부를 위한 규칙을 정의한다. 예를 들어, 스키마 파일(330)에 나타난 바와 같이, 2개의 자식 요소 <title> 및 <body>는 루트 또는 부모 요소 <intro card> 아래에 정의된다. 자식 요소들 <title> 및 <body>와 관련된 텍스트의 허용가능한 스트링 길이를 정의하는 속성들(340,

345)도 도시되어 있다. 전술한 바와 같이, 사용자가 스키마 파일(330)에 포함된 XML 마크업 정의를 위반하여 문서에 첨부되거나 연관된 스키마 파일(330)로부터의 XML 마크업에 의한 문서(310)에 주석을 첨부하려고 시도하는 경우, 무효 또는 에러 상태가 표시된다. 예를 들어, 사용자가 25 문자를 초과하는 제목 스트링을 입력하려고 시도하는 경우, 그 텍스트 엔트리는 스키마 파일(330)의 <title> 요소의 최대 문자 길이 속성을 위반하게 된다. 연관된 스키마 파일(330)에 대해 문서(310)에 적용된 XML 마크업을 검증하기 위하여, 스키마 검증 모듈(350)이 이용된다. 이 분야의 전문가가 이해하듯이, 스키마 검증 모듈은 XML 마크업 및 관련 텍스트가 문서(310)에 입력될 때 문서(310)에 입력된 XML 마크업 및 관련 텍스트를 연관되거나 첨부된 XML 스키마 파일(330)과 비교하기에 충분한 컴퓨터 실행 가능 명령들을 포함하는 소프트웨어 모듈이다.

[0054] 본 발명의 실시예들에 따르면, 스키마 검증 모듈(350)은 문서(310)에 적용된 각각의 XML 마크업 요소 및 관련 텍스트 또는 데이터를 첨부 또는 연관된 스키마 파일(330)과 비교하여 각각의 요소 및 관련 텍스트 또는 데이터가 첨부된 스키마 파일(330)에 설명된 규칙 및 정의를 따르는지를 결정한다. 예를 들어, 사용자가 <title> 요소들(320)이 주석으로 첨부된 25 문자를 초과하는 문자 스트링을 입력하려고 시도하는 경우, 스키마 검증 모듈은 그 텍스트 스트링을 첨부 스키마 파일(330)의 텍스트 스트링 속성(340)과 비교하여, 사용자에게 의해 입력된 텍스트 스트링이 최대 허용 가능한 텍스트 스트링 길이를 초과한 것으로 결정한다. 따라서, 에러 메시지 또는 다이얼로그가 사용자에게 표시되어, 사용자가 입력하고 있는 텍스트 스트링이 첨부 스키마 파일(330)에 따른 최대 허용 가능 문자 길이를 초과하였음을 사용자에게 알린다. 또한, 사용자가 <title> 및 <body> 요소들 사이에 XML 마크업 요소를 추가하려고 시도하는 경우, 스키마 검증 모듈(350)은 사용자에게 의해 적용된 XML 마크업 요소가 첨부 스키마 파일(330)에 따라 <title> 및 <body> 요소들 사이에 허용되는 유효 요소가 아닌 것으로 결정한다. 따라서, 스키마 검증 모듈(350)은 사용자에게 대한 에러 메시지 또는 다이얼로그를 생성하여 사용자에게 무효 XML 마크업임을 알린다.

[0055] 컨텍스트-프리 문서 텍스트 및 데이터 부분들

[0056] 간단히 전술한 바와 같이, 본 발명의 실시예들은 텍스트 또는 데이터의 일부를 문서에 삽입하기 위한 방법 및 시스템을 제공하는데, 텍스트 또는 데이터의 삽입 부분은 지정된 스타일, 폰트, 리스트 타입 등과 같은 그 자신의 특정 포매팅 및 스타일 속성을 갖는다. 도 4는 본 발명의 실시예들에 따라 텍스트의 일부가 제2 문서에서 제1 문서로 복사 또는 이동되는 제1 문서와 제2 문서 간의 상호작용을 나타내는 블록도이다. 도 5는 본 발명의 실시예들에 따라 텍스트의 일부가 제2 문서에서 제1 문서로 복사 또는 이동되는 제1 문서와 제2 문서 간의 상호작용을 나타내는 블록도이다. 하나의 문서에서 다른 문서로 텍스트 부분이 복사 또는 이동하는 것에 의해 본 발명을 설명하는 것은 단지 예시적이라는 것을 이해하여야 한다. 하나의 문서에서 다른 문서로 텍스트를 복사 또는 이동하는 것 외에, 텍스트 부분의 자기 기술 속성들은 파일을 파싱하고 조작하기를 원하거나 필요로 하는 사용자 및 소프트웨어 애플리케이션이 전체 문서 또는 파일과 연관된 속성들을 관리하기 위해 전체 문서를 파싱할 필요 없이 파일 또는 문서의 부분들에 리치 편집(rich editing)을 적용할 수 있게 한다.

[0057] 도 4를 참조하면, 문서(410) 및 문서(420)는 본 발명의 실시예들에 따라 문서(420)에서 문서(410)로 텍스트의 일부가 어떻게 복사될 수 있는지를 나타내기 위해 설명된다. 제1 문서(410)는 하나의 제목과 2개의 단락을 갖는다. 또한, 문서(410)에 대해 문서(435)에 적용된 XML 구조를 나타내는 XML 구조 창(XML structure pane, 425)이 도시되어 있다. 예를 들어, <title> 요소는 문서(435)의 제목에 적용되고, <paragraph 1> 요소는 문서(435)의 제1 단락에 적용되며, <paragraph 2> 요소는 문서(435)의 제2 단락에 적용된다. 제2 문서(420)는 XML 구조로 유사하게 마크업된 문서(440)를 포함한다. 문서(420)에 대한 XML 구조 창(430)이 도시되어 있다.

[0058] 본 발명의 실시예들에 따르면, 제2 문서로부터의 텍스트 또는 데이터 부분이 제1 문서로 복사 또는 이동될 수 있어서, 스타일 및 다른 포매팅 속성들이 텍스트 또는 데이터 부분과 함께 하나의 문서에서 다른 문서로 전달된다. 따라서, 제1 문서와 연관된 스타일 또는 다른 포매팅 속성들에 대해 제1 문서에 삽입된 텍스트 부분과 연관된 스타일 또는 포매팅 속성들을 리졸브(resolve)할 필요가 없다. 예를 들어, 도 4에 도시된 바와 같이, 제1 문서는 제1 문서와 연관된 스타일 및 다른 포매팅 속성을 참조하기 위한 헤더 정보 속성 요소를 가질 수 있다. 즉, 제1 문서의 사용자 또는 생성자는 제1 문서에 대한 스타일 또는 포매팅 속성 세트를 설정할 수 있다. 전술한 바와 같이, 종래의 시스템에 따르면, 사용자가 제2 문서(420)에서 제1 문서(410)로 제2 단락(455)을 복사하기를 원하는 경우, 제2 문서(420) 내의 제2 단락(455)과 연관된 스타일 또는 포매팅 정보는 제1 문서(410)의 문서 레벨에 포함된 스타일 또는 포매팅 정보와 충돌할 수 있다.

- [0059] 예를 들어, 제1 문서(410)는 문서(435)에 굵은 이탤릭체의 포매팅을 적용하게 하는 "Header 1"이라고 하는 스타일을 가질 수 있다. 반면, 제2 문서(420)의 생성자는 제2 문서에 포함된 제2 단락(455)을 이탤릭체는 아니지만 굵은 글자체를 포함하도록 포매팅하기 위해 스타일 표시자 "Header 1"을 사용할 수도 있다. 제2 문서(420)의 제2 단락(455)이 제1 문서(410)로 복사되는 경우, 제2 문서에 적용된 스타일은 제1 문서(410) 내의 문서에 적용된 스타일과 충돌하게 된다. 따라서, 제1 문서(410)의 사용자는 2개의 스타일의 충돌을 방지할 수 있도록 삽입 텍스트 또는 데이터에 적용된 스타일들을 변경하기 위해 제1 문서(410)에 적용된 XML 구조의 헤더 정보 속성 요소를 수동으로 파싱하는 것이 요구될 것이다.
- [0060] 본 발명의 실시예들에 따르면, 제2 문서(420)의 제2 단락(455)과 같은 문서의 지정된 텍스트 및 데이터 부분들과 같이 스타일 마크업을 적용하기 위하여 XML 스키마가 제공되어 주어진 텍스트 또는 데이터 부분에 대한 특정 스타일 또는 포매팅 적용이 텍스트 또는 데이터 부분과 함께 제1 문서로 전달된다. 특정 텍스트 또는 데이터 선택에 XML 마크업을 적용함으로써, 제1 문서는 삽입된 텍스트 또는 데이터 부분에 적용된 스타일 또는 포매팅 마크업에 액세스하는 한은 삽입된 텍스트 또는 데이터 부분에 적용된 스타일 또는 포매팅 속성과 디컨플릭트(de-conflict)할 필요가 없다.
- [0061] 도 5를 참조하면, 제1 문서(410)로 삽입할 텍스트 부분(510) 및 텍스트 부분(520)이 도시되어 있다. 본 발명의 실시예들에 따르면, 각각의 텍스트 부분(510, 520)은 문장, 단락, 다수의 단락, 문장들 또는 단락들의 조각, 영숫자 데이터 또는 임의의 다른 원시 객체들일 수 있다. 제2 문서(420)의 생성자는 후술하는 바와 같이 그 자신과 함께 그 자신의 스타일 또는 다른 포매팅 속성을 전달하는 특정 텍스트 또는 데이터 부분들을 지정하기 위해, XML 요소들을 정의하는 컨텍스트-프리 텍스트 부분 스키마에 따라 제2 문서를 XML 데이터로 마크업한다는 것을 이해해야 한다. 결과적으로, 텍스트 부분들(510, 520) 각각은 제1 문서(410)에 삽입될 수 있고 문서(410) 상에서 동작하는 XML 파싱 애플리케이션에 의해 주 문서(410)와 별개로 처리될 수 있는 그 자신의 스타일 및 포매팅 속성을 가진 "미니(mini)" 문서가 된다.
- [0062] 제2 단락(455)과 같은 텍스트 또는 데이터 부분 또는 도 5에 도시된 텍스트 부분들(510, 520) 중 어느 하나가 다양한 소프트웨어 애플리케이션으로부터 이용 가능한 통상의 절단/복사 및 붙이기 방법에 따라 제2 문서에서 제1 문서로 복사 및 붙이기 또는 절단 및 붙이기될 수 있다는 것을 이해해야 한다. 대안으로, 다양한 XML 파싱 애플리케이션에 따라, 사용자는 XML 구조 창(430)으로부터 적절한 XML 노드를 선택하고 그 XML 노드를 제1 문서의 XML 구조에 복사함으로써 텍스트 또는 데이터 부분을 제2 문서에서 제1 문서로 삽입할 수도 있다. 즉, 제2 문서(420)의 XML 구조(430)로부터 <paragraph 2> 노드(460)를 선택하고 이 노드를 제1 문서(410)의 XML 구조로 이동함으로써 전술한 바와 같이 제2 단락(455)의 XML 마크업 및 관련 텍스트 또는 데이터가 그 자신의 스타일 및 다른 포매팅 속성들과 함께 제1 문서에 삽입될 수 있다.
- [0063] 본 발명의 실시예들의 동작을 더 설명하기 위하여, 다음의 XML 구조는 다음의 두 단락을 가진 간단한 문서와 연관된다.
- [0064] 단락 1: The quick brown fox jumps over the lazy dog.
- [0065] 단락 2: The quick brown fox jumps over the lazy dog.
- [0066] 다음의 XML 구조에서 알 수 있듯이, "Times New Roman"의 폰트 및 "Default Paragraph Font"의 스타일 식별이 두 단락에 적용된다. 두 단락의 실제 텍스트도 다음의 XML 구조에 나타나 있다.
- [0067] <?xml version="1.0" encoding="EUC-KR" standalone="yes" ?>
- [0068] <w:wordDocument
- [0069] xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml"  
 xmlns:wx="http://schemas.microsoft.com/office/word/2003/auxHint" w:macrosPresent="no"  
 w:embeddedObjPresent="no" w:ocxPresent="no" xml:space="preserve">
- [0070]
- [0071] <w:styles>
- [0072] <w:versionOfBuiltInStylenames w:val="4" />
- [0073] <w:latentStyles w:defLockedState="off" w:latentStyleCount="156" />



[0074]     <w:style w:type="paragraph" w:default="on" w:styleId="Normal">

[0075]         <w:name w:val="Normal" />

[0076]     <w:rPr>

[0077]         <wx:font wx:val="Times New Roman" />

[0078]         <w:sz w:val="24" />

[0079]         <w:sz-cs w:val="24" />

[0080]         <w:lang w:val="EN-US" w:fareast="EN-US" w:bidi="AR-SA" />

[0081]     </w:rPr>

[0082]     </w:style>

[0083]     <w:style w:type="character" w:default="on" w:styleId="DefaultParagraphFont">

[0084]         <w:name w:val="Default Paragraph Font" />

[0085]         <w:semiHidden />

[0086]     </w:style>

[0087]     <w:style w:type="list" w:default="on" w:styleId="NoList">

[0088]         <w:name w:val="No List" />

[0089]         <w:semiHidden />

[0090]     </w:style>

[0091]     </w:styles>

[0092]     <w:body>

[0093]     <w:p>

[0094]     <w:r>

[0095]         <w:t>The quick brown fox jumps over the lazy dog.</w:t>

[0096]     </w:r>

[0097]     </w:p>

[0098]     <w:p>

[0099]     <w:r>

[0100]         <w:t>The quick brown fox jumps over the lazy dog.</w:t>

[0101]     </w:r>

[0102]     </w:p>

[0103]     <w:p />

[0104]     </w:body>

[0105]     </w:wordDocument>

[0106]     본 발명의 실시예들에 따라, 예를 들어 사용자가 "The quick brown fox jumps over the lazy dog"과 같이 제3 단락을 문서에 삽입하여 삽입된 단락이 아래와 같은 최종 문서를 제공하도록 하고자 하는 경우를 고려하자.

[0107]     단락 1: The quick brown fox jumps over the lazy dog.

[0108]     단락 2: The quick brown fox jumps over the lazy dog.

- [0109] 단락 3: The quick brown fox jumps over the lazy dog.
- [0110] 아래는 표준 디폴트 스타일이 적용된 문서의 처음 두 단락을 나타내는 XML 구조의 일례이다. 이 예에 따르면, 다음의 XML 구조에 나타난 바와 같이, 사용자는 "My Style"이라는 스타일 하에서 문서에 제3의 이탤릭체 단락을 삽입한다. 이 단락은 원시 워드 프로세싱 애플리케이션 외부의 환경에서 삽입되고 있기 때문에, "My Style" 스타일이 존재하는 것을 검사하는 로직은 수동으로 행해져야 한다. XML 파일은 스타일 선언의 적절한 위치를 찾기 위해 파싱되어야 하며, 이어서 스타일 선언은 충돌이 없음을 보장하기 위해 파싱되어야 한다. 이것이 완료되면, "My Style" 선언이 입력될 수 있다. 이제, "My Style" 선언이 입력된 후, 단락 자체에 대한 적절한 위치를 찾기 위하여 XML 파일은 다시 파싱되어야 한다. "My Style" 선언과의 충돌이 없으므로, 삽입되고 있는 단락은 "My Style" 스타일을 계속 참조할 수 있다. 그렇지 않은 경우, 단락 자체는 적절한 스타일을 참조하도록 변경되어야 한다.
- [0111] <?xml version="1.0" encoding="EUC-KR" standalone="yes" ?>
- [0112] <?mso-application progid="Word.Document" ?>
- [0113] = <w:wordDocument
- [0114] xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml"  
 xmlns:wx="http://schemas.microsoft.com/office/word/2003/auxHint" w:macrosPresent="no"  
 w:embeddedObjPresent="no" w:ocxPresent="no" xml:space="preserve">
- [0115] = <w:styles>
- [0116] <w:versionOfBuiltInStylenames w:val="4" />
- [0117] <w:latentStyles w:defLockedState="off" w:latentStyleCount="156" />
- [0118] = <w:style w:type="paragraph" w:default="on" w:styleId="Normal">
- [0119] <w:name w:val="Normal" />
- [0120] = <w:rPr>
- [0121] <wx:font wx:val="Times New Roman" />
- [0122] <w:sz w:val="24" />
- [0123] <w:sz-cs w:val="24" />
- [0124] <w:lang w:val="EN-US" w:fareast="EN-US" w:bidi="AR-SA" />
- [0125] </w:rPr>
- [0126] </w:style>
- [0127] = <w:style w:type="character" w:default="on" w:styleId="DefaultParagraphFont">
- [0128] <w:name w:val="Default Paragraph Font" />
- [0129] <w:semiHidden />
- [0130] </w:style>
- [0131] = <w:style w:type="table" w:default="on" w:styleId="TableNormal">
- [0132] <w:name w:val="Normal Table" />
- [0133] <wx:uiName wx:val="Table Normal" />
- [0134] <w:semiHidden />
- [0135] = <w:rPr>

```

[0136]     <wx:font wx:val="Times New Roman" />
[0137]     </w:rPr>
[0138]  = <w:tblPr>
[0139]     <w:tblInd w:w="0" w:type="dxa" />
[0140]  = <w:tblCellMar>
[0141]     <w:top w:w="0" w:type="dxa" />
[0142]     <w:left w:w="108" w:type="dxa" />
[0143]     <w:bottom w:w="0" w:type="dxa" />
[0144]     <w:right w:w="108" w:type="dxa" />
[0145] </w:tblCellMar>
[0146] </w:tblPr>
[0147] </w:style>
[0148] = <w:style w:type="paragraph" w:styleId="myStyle">
[0149]     <w:name w:val="myStyle" />
[0150]     <w:basedOn w:val="Normal" />
[0151]     <w:rsid w:val="000874C3" />
[0152] = <w:pPr>
[0153]     <w:pStyle w:val="myStyle" />
[0154] </w:pPr>
[0155] = <w:rPr>
[0156]     <wx:font wx:val="Times New Roman" />
[0157]     <w:color w:val="FF0000" />
[0158] </w:rPr>
[0159] </w:style>
[0160] = <w:style w:type="list" w:default="on" w:styleId="NoList">
[0161]     <w:name w:val="No List" />
[0162]     <w:semiHidden />
[0163] </w:style>
[0164] </w:styles>
[0165] = <w:body>
[0166] = <w:p>
[0167] = <w:r>
[0168]     <w:t>The quick brown fox jumps over the lazy dog.</w:t>
[0169] </w:r>
[0170] </w:p>
[0171] = <w:p>

```



[0172]       = <w:r>  
 [0173]           <w:t>The quick brown fox jumps over the lazy dog.</w:t>  
 [0174]           </w:r>  
 [0175]           </w:p>  
 [0176]       = <w:p>  
 [0177]       = <w:pPr>  
 [0178]           <w:pStyle w:val="myStyle" />  
 [0179]           </w:pPr>  
 [0180]       = <w:r>  
 [0181]           <w:t>The quick brown fox jumps over the lazy dog.</w:t>  
 [0182]           </w:r>  
 [0183]           </w:p>  
 [0184]           <w:p />  
 [0185]           </w:body>  
 [0186]           </w:wordDocument>

[0187]       전술한 바와 같이, 본 발명의 실시예들에 따르면, 문서의 특정 텍스트 또는 데이터 부분을 마크업하고 기술하는 데 사용될 수 있는 XML 요소를 제공하여 텍스트 또는 데이터 부분에 적용된 스타일 또는 다른 포매팅 속성들이 텍스트 또는 데이터 부분이 삽입되는 문서와 연관된 나머지 스타일 또는 포매팅 속성들과 별개로 처리되도록 하는 XML 스키마가 제공된다. 텍스트 또는 데이터 부분이 삽입되는 문서가 문서에 삽입되는 텍스트 또는 데이터 부분과 연관된 XML 스키마를 이해할 수 있는 애플리케이션에 의해 조작되는 한, 삽입 텍스트 또는 데이터 부분을 래핑하는 XML 요소가 이해될 것이다.

[0188]       아래는 본 발명의 실시예들에 따라 제3의 이탤릭체 단락 "The quick brown fox jumps over the lazy dog"의 추가를 보이는 XML 구조의 샘플이다. 아래의 XML 구조에 나타난 바와 같이, 처음 두 단락은 "Times New Roman" 및 "Default Paragraph Font"의 스타일 식별을 갖는 문서에 포함된다. 그러나, 아래의 XML 구조에 나타난 바와 같이, 문서에 삽입되고 있는 세 번째 단락은 <cfChunk>라고 하는 XML 요소 내에 래핑되어 있다. 요소 <cfChunk>는 이 특정 XML 구조의 생성자에 의해 본 발명의 실시예들에 따른 요소에 제공될 수 있는 무한수의 이름을 나타낸다는 것을 이해해야 한다. 요소는 유사하게 컨텍스트-프리 부분, cfPortion, 컨텍스트-프리 데이터, cfData 등으로 지칭될 수 있다는 것을 이해해야 한다. 요소 <cfChunk> 내에 "My Style"의 스타일 식별을 갖고 "italics"의 폰트 식별을 갖는 세 번째 단락 "The quick brown fox jumps over the lazy dog"이 포함된다.

[0189]       본 발명의 실시예들에 따르면, 세 번째 단락 "The quick brown fox jumps over the lazy dog"이 제2 문서에서 제1 문서로 복사 또는 이동될 때, 이동된 텍스트 또는 데이터 부분은 텍스트 또는 데이터 부분을 래핑하고 텍스트 또는 데이터 부분과 함께 그 자신의 개별 스타일들 또는 다른 포매팅 속성들을 전달하는 데 사용되는 <cfChunk> 요소를 포함하는 스키마 파일에 따라 XML 요소들과 연관된 문법 및 다른 정의 규칙들에 대한 제1 문서 액세스와 연관된 XML 파싱 애플리케이션을 제공하는 XML 스키마 파일을 지시하거나 참조한다. 따라서, 제1 문서의 XML 파싱 애플리케이션이 요소 <cfChunk>를 만날 때, 이 애플리케이션은 연관된 XML 스키마를 참조하여, <cfChunk> 요소 내에 래핑된 구조에 포함된 스타일 및 다른 포매팅 지시들이 특히 제1 문서에 삽입되고 있고 <cfChunk> 요소와 연관되고 있는 텍스트 또는 데이터 부분에 대해 처리된다는 것을 학습할 수 있다. 따라서, 본 발명의 실시예들에 따르면, 아래의 XML 구조의 샘플에 나타난 바와 같이, 세 번째 단락이 제1 문서에 삽입될 때, 텍스트 부분과 연관된 스타일 및 다른 포매팅 속성들은 텍스트 부분이 삽입되는 문서와 연관된 스타일 또는 다른 포매팅 속성들과의 잠재적인 충돌과 관계 없이 삽입된 텍스트 또는 데이터 부분을 수신하는 애플리케이션에 의해 허가된다. 애플리케이션은 임의의 충돌을 리졸브하고 단락에 대한 스타일 참조를

적절한 스타일과 적절히 연관시키기 위한 필요한 작업을 행한다.

```
[0190] <?xml version="1.0" encoding="EUC-KR" standalone="yes" ?>
[0191] =                                     <w:wordDocument
[0192] xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml"
xmlns:wx="http://schemas.microsoft.com/office/word/2003/auxHint"           w:macrosPresent="no"
w:embeddedObjPresent="no" w:ocxPresent="no" xml:space="preserve">
[0193] = <w:styles>
[0194]     <w:versionOfBuiltInStylenames w:val="4" />
[0195]     <w:latentStyles w:defLockedState="off" w:latentStyleCount="156" />
[0196] = <w:style w:type="paragraph" w:default="on" w:styleId="Normal">
[0197]     <w:name w:val="Normal" />
[0198] = <w:rPr>
[0199]     <wx:font wx:val="Times New Roman" />
[0200]     <w:sz w:val="24" />
[0201]     <w:sz-cs w:val="24" />
[0202]     <w:lang w:val="EN-US" w:fareast="EN-US" w:bidirectional="AR-SA" />
[0203] </w:rPr>
[0204] </w:style>
[0205] = <w:style w:type="character" w:default="on" w:styleId="DefaultParagraphFont">
[0206]     <w:name w:val="Default Paragraph Font" />
[0207]     <w:semiHidden />
[0208] </w:style>
[0209] = <w:style w:type="list" w:default="on" w:styleId="NoList">
[0210]     <w:name w:val="No List" />
[0211]     <w:semiHidden />
[0212] </w:style>
[0213] = <w:style w:type="table" w:default="on" w:styleId="TableNormal">
[0214]     <w:name w:val="Normal Table" />
[0215]     <wx:uiName wx:val="Table Normal" />
[0216]     <w:semiHidden />
[0217] = <w:rPr>
[0218]     <wx:font wx:val="Times New Roman" />
[0219] </w:rPr>
[0220] = <w:tblPr>
[0221]     <w:tblInd w:w="0" w:type="dxa" />
```

[0222]     = <w:tblCellMar>  
 [0223]         <w:top w:w="0" w:type="dxa" />  
 [0224]         <w:left w:w="108" w:type="dxa" />  
 [0225]         <w:bottom w:w="0" w:type="dxa" />  
 [0226]         <w:right w:w="108" w:type="dxa" />  
 [0227]     </w:tblCellMar>  
 [0228]     </w:tblPr>  
 [0229]     </w:style>  
 [0230]     </w:styles>  
 [0231]     = <w:body>  
 [0232]     = <w:p>  
 [0233]     = <w:r>  
 [0234]         <w:t>The quick brown fox jumps over the lazy dog.</w:t>  
 [0235]         </w:r>  
 [0236]         </w:p>  
 [0237]     = <w:p>  
 [0238]     = <w:r>  
 [0239]         <w:t>The quick brown fox jumps over the lazy dog.</w:t>  
 [0240]         </w:r>  
 [0241]     = <w:p>  
 [0242]         </w:p>  
 [0243]     = <w:cfChunk>  
 [0244]     = <w:styles>  
 [0245]     = <w:style w:type="paragraph" w:styleId="myStyle">  
 [0246]         <w:name w:val="myStyle" />  
 [0247]         <w:basedOn w:val="Normal" />  
 [0248]         <w:rsid w:val="000874C3" />  
 [0249]     = <w:pPr>  
 [0250]         <w:pStyle w:val="myStyle" />  
 [0251]         </w:pPr>  
 [0252]     = <w:rPr>  
 [0253]         <wx:font wx:val="Italics" />  
 [0254]         <w:color w:val="FF0000" />  
 [0255]         </w:rPr>  
 [0256]     </w:style>  
 [0257]     </w:styles>

[0258]       <w:p>  
 [0259]       <w:pPr>  
 [0260]           <w:pStyle w:val="myStyle" />  
 [0261]       </w:pPr>  
 [0262]       <w:r>  
 [0263]           <w:t>The quick brown fox jumps over the lazy dog.</w:t>  
 [0264]       </w:r>  
 [0265]       </w:p>  
 [0266]       </w:cfChunk>  
 [0267]       <w:p />  
 [0268]       </w:body>  
 [0269]       </w:wordDocument>

[0270]       <u>교체 파일 포맷을 가진 컨텍스트-프리 문서 부분</u>

[0271]       전술한 바와 같이, 종래의 방법 및 시스템에 따르면, 데이터 또는 콘텐츠의 일부가 수신 문서로서 상이한 파일 포맷을 가진 문서에 추가될 때, 수신 문서는 인커밍 부분이 스타일, 폰트 등을 포함하는 적절한 포맷팅 속성들과 함께 적절히 표시되는 방식으로 인커밍 부분을 처리하지 못할 수 있다. 또는, 인커밍 부분이 수신 문서와 다른 포맷팅 타입을 갖는 경우, 수신 문서는 인커밍 부분을 전혀 로딩하지 못할 수 있다. 예를 들어, HTML에 따라 포맷된 데이터의 일부가 XML에 따라 포맷된 문서로 복사되는 경우, HTML 부분은 로드에서 실패할 수 있는데, 이는 문서의 처리 및 표시를 담당하는 애플리케이션이 콘텐츠의 인커밍 부분이 무엇인지를 이해하지 못할 수 있기 때문이다. 도 4 및 5를 참조하여 전술한 바와 같이, 텍스트 또는 데이터의 인커밍 부분들은 <cfChunk> XML 태그 내에 랩핑될 수 있고, 텍스트 또는 데이터의 일부에 적용된 XML 주석은 텍스트 또는 데이터의 인커밍 부분과 연관된 스타일, 폰트 등과 같은 포맷팅 정보를 식별하여, 수신 XML 표현 문서가 론칭되고 텍스트 또는 데이터의 인커밍 부분이 론칭된 문서와 통합(즉, 병합, 복사, 이동, 링크)될 때 텍스트 또는 데이터의 인커밍 부분에 적용될 포맷팅 속성들을 확인 또는 리졸브할 수 있다.

[0272]       본 발명의 다른 실시예에 따르면, 텍스트, 데이터 또는 다른 콘텐츠의 인커밍 부분에 XML <cfChunk> 주석이 적용되며, 이것은 인커밍 부분이 수신 문서에 사용되는 것과 다른 파일 포맷 타입인 것을 수신 XML 표현 문서에 알리기 위하여 텍스트, 데이터 또는 다른 콘텐츠의 인커밍 또는 참조 부분과 연관된 교체 파일 포맷 타입들에 관하여 수신 문서 및 애플리케이션에 알리기 위하여 자기 기술 속성들을 제공한다. 도 6은 본 발명의 실시예들에 따라 자기 기술 교체 파일 포맷(self-described alternative file formats)을 가진 컨텍스트-프리 콘텐츠 부분의 XML 기반 문서로의 삽입을 나타내는 간단한 블록도이다. 도 6에 도시된 바와 같이, XML 표현 문서(610), 예를 들어 메모랜덤 또는 기사와 같은 워드 프로세싱 문서가 도시되어 있다. XML 문서(610)는 다양한 XML 구조가 문서에 적용된 것으로 도시되어 있다. 예를 들어, 문서에 입력될 헤딩 정보를 구조화하기 위하여 <heading> 태그들(615)이 문서에 적용된다. 문서(610)의 본문에 삽입될 텍스트, 데이터 또는 다른 콘텐츠를 구조화하기 위해 <Body> 태그들(620)이 문서에 적용된다. 도 6에 도시된 XML 구조 샘플은 양호하게 구성된 XML 태그들 및 관련 구조를 나타내는 것을 의미하지 않는다는 것을 이 분야의 전문가가 이해해야 한다. 도 6에 도시된 샘플 XML 구조 및 정보는 단지 예시적인 것이다.

[0273]       도 6을 계속 참조하면, <body> 태그(620) 내의 XML 문서(610)에 세 개의 <cfChunk> 태그(625, 630, 635)가 적용되었다. 본 발명의 실시예들에 따르면, <cfChunk> 태그들(625, 630, 635)은 인커밍 부분들이 XML 문서(610)에 대해 자기 기술하는 것을 허용하는 컨텍스트-프리 방식으로 텍스트, 데이터 또는 다른 콘텐츠의 인커밍 부분들에 주석을 첨부하는 데 사용된다. 도 4 및 5를 참조하여 전술한 바와 같이, XML 문서(610)를 생성 및/또는 편집하는 애플리케이션은 사용시에 그 애플리케이션과 통합되거나 애플리케이션에 의해 호출되는 XML 파싱 애플리케이션을 호출하여, XML 문서(610)에 적용된 XML 구조를 파싱한다.

[0274]       본 발명의 실시예들에 따르면, XML 파싱 애플리케이션이 부분들(640, 645, 650)과 각각 연관된 XML <cfChunk>

태그들(625, 630, 635)을 파싱할 때, 파싱 애플리케이션은 콘텐츠의 각각의 인커밍 부분에 적용된 지정 파일 포맷 타입 및 포매팅 정보를 결정한다. 예를 들어, XML 문서(610)의 본문에 삽입될 제1 인커밍 부분(640)은 <cfChunk 1> 태그와 연관되어 있고, 인커밍 부분의 파일 포맷 타입을 HTML로서 식별한다. "style:1"의 스타일 타입도 식별된다. 따라서, XML 문서(610)를 생성, 편집 및 표시하기 위한 애플리케이션, 예를 들어 워드 프로세싱 애플리케이션(125)은 문서(610)에 적용된 XML 구조의 파싱을 통해, 삽입된 콘텐츠 부분(640)이 HTML에 따라 포맷되고 "style:1"의 포매팅 스타일을 갖는다는 것을 학습한다. 따라서, XML 표현 문서(610)를 생성 및/또는 편집하기 위해 사용되는 애플리케이션은 인커밍 부분과 연관된 파일 포맷 타입 및 스타일에 따라 인커밍 콘텐츠 부분(640)을 처리할 수 있다. 예를 들어, 인커밍 부분(640)을 참조하여, 수신 애플리케이션은 인커밍 부분이 HTML에 따라 처리, 판독 및 표시되어야 한다는 것, 또는 인커밍 부분이 HTML 파일 포맷에서 수신 애플리케이션에 의해 처리하기에 적절한 다른 파일 포맷 타입으로 변환되어야 한다는 것을 알게 된다. 유사하게, 수신 애플리케이션은 인커밍 부분이 "style:1"의 식별 포매팅 속성과 연관된 포매팅 속성들, 예를 들어 폰트, 스타일 등과 함께 표시되어야 한다는 것을 알게 된다.

[0275] 도 6에 도시된 바와 같이, 제2 인커밍 콘텐츠 부분(645)이 도시되어 있다. 제2 인커밍 부분(645)은 "style:4"로서 식별되는 스타일의 리치 텍스트 포매팅(RTF)의 파일 포맷 타입을 가진 것으로 도시되어 있다. 인커밍 부분(640)에서와 같이, 콘텐츠의 인커밍 부분을 자기 기술하는 데 사용되는 정보는 수신 또는 소비 애플리케이션에 의해 인커밍 부분과 연관된 파일 포맷 타입을 이해하고 인커밍 콘텐츠에 적용된 임의의 관련 포매팅 속성들을 이해하기 위해 사용될 수 있다. 유사하게, 관련 바이너리 파일 포맷 타입 및 "style:2"의 관련 스타일을 가진 제3 인커밍 부분(650)이 도시되어 있다. 다른 인커밍 콘텐츠 부분에서와 같이, 수신 또는 소비 애플리케이션은 인커밍 콘텐츠 부분을 수신하고 처리하기 위하여 식별된 스타일 속성들과 같은 식별된 파일 포맷 타입 및 다른 정보를 이용할 수 있다.

[0276] 이 실시예에 따르면, 수신 또는 소비 애플리케이션, 예를 들어 워드 프로세싱 애플리케이션은 인커밍 부분들과 연관된 파일 포맷 타입에 비추어 콘텐츠의 인커밍 부분들을 이용할 수 있다. 종래의 방법 및 시스템에 따르면, 인커밍 부분들은 수신 애플리케이션에 의해 로딩되지 못하거나, 인커밍 부분들이 수신 애플리케이션에 의해 성공적으로 로딩된 경우에도 관련 텍스트, 데이터 또는 다른 콘텐츠가 잘못 처리될 수 있다. 예를 들어, 인커밍 부분은 만족스럽지 못한 방식으로 표시될 수 있다. 수신 애플리케이션은 인커밍 콘텐츠의 원래의 파일 포맷 타입을 확인할 수 있으므로, 수신 애플리케이션은 인커밍 콘텐츠의 변환, 또는 인커밍 콘텐츠의 이해, 소비 및 이용에 필요한 소프트웨어 프로그래밍의 호출과 같은 적절한 동작을 취할 수 있다. 예를 들어, 인커밍 콘텐츠가 HTML에 따라 포맷되는 경우, 수신 애플리케이션은 인커밍 콘텐츠를 이용하기 전에 인커밍 콘텐츠를 변환하거나, 인커밍 콘텐츠를 이용, 예를 들어 표시하는 데 필요한 코드에 액세스할 수 있다. 따라서, 본 발명의 실시예들에 따르면, 도 4 및 5를 참조하여 전술한 바와 같이, 수신 애플리케이션이 인커밍 콘텐츠에 대한 관련 포매팅 속성들을 리졸브할 때, 수신 애플리케이션은 인커밍 콘텐츠와 연관된 교체 파일 포맷 타입들도 리졸브할 수 있다.

[0277] 본 발명의 일 실시예에 따르면, 전술한 콘텐츠 부분들의 수신 또는 소비를 담당하는 소프트웨어 애플리케이션, 예를 들어 워드 프로세싱 애플리케이션, 스프레드시트 애플리케이션, 슬라이드 프리젠테이션 애플리케이션, 탁상 출판 애플리케이션 등도 다른 애플리케이션들에 의해 사용하기 위한 컨텍스트-프리 데이터 부분들(640, 645, 650)을 생성할 수 있다. 이러한 애플리케이션에 의해 데이터의 일부가 저장될 때, 또는 다른 문서에 의해 조작되는 문서에 붙이기 위해 데이터의 일부가 복사될 때, 컨텍스트-프리 부분을 생성하기 위해 사용되는 애플리케이션은 데이터에 <cfChunk> 주석을 적용할 수 있으며, 데이터의 일부와 연관된 파일 포맷은 물론 포매팅과 같은 다른 속성들(예를 들어, 굵은 글자체, 이탤릭체, 폰트 등)을 식별하기 위해 데이터의 일부에 대한 속성들을 저장할 수 있다. 예를 들어, 컨텍스트-프리 부분(640, 645, 650)을 생성하는 애플리케이션이 HTML 포맷에 따라 데이터를 생성하는 경우, 다른 애플리케이션들에 의한 사용을 위해 생성되는 데이터의 컨텍스트-프리 부분들은 데이터의 컨텍스트-프리 부분과 연관된 파일 포맷으로서 HTML을 식별하기 위해 주석이 첨부될 수 있다. 전술한 바와 같이, 소비 애플리케이션이 데이터를 수신할 때, 소비 애플리케이션은 생성 애플리케이션에 의해 데이터의 일부에 적용된 관련 파일 포맷 및 다른 포매팅 속성들에 따라 컨텍스트-프리 부분을 판독하고 사용할 수 있다.

[0278] 이제, 도 7을 참조하고, 본 발명의 다른 실시예에 따르면, 수신 문서(710) 및 하나 이상의 인커밍 콘텐츠 부분(730, 735, 740)은 전자 파일 컨테이너(700)에 넣어질 수 있다. 예를 들어, 파일 컨테이너(700)는 .ZIP 파일 포맷과 같은 파일 포맷을 나타낼 수 있다. 실시예들에 따르면, 파일 컨테이너(700)가 선택되거나 열리는 경우, XML 수신 문서(710) 및 관련 콘텐츠 부분들(730, 735, 740)의 각각은 개별 파일들로서 식별되고 가시적으로 표현될 수 있다. 이 실시예에 따르면, 각각의 인커밍 콘텐츠 부분(730, 735, 740)은 도 6을 참조하여

전술한 바와 같이 수신 문서(710) 내에 일렬로 삽입되거나 채워지지 않으나, 그 대신에 수신 문서(710)는 관계 파일(750)을 참조하는데, 이 참조 파일은 인커밍 콘텐츠 부분들을 지시하며, 수신 문서(710)에, 그리고 인커밍 콘텐츠 부분들을 이용하기 위해 수신 문서(710)의 처리를 담당하는 애플리케이션에 정보를 제공한다.

[0279] 도 7에 도시된 바와 같이, XML 문서(710)는 저자/편집자에 의해 생성되고 있는 수신 문서를 나타내며, 콘텐츠 부분들(730, 735, 740)은 수신 문서(710)와 병합하기 위한 인커밍 콘텐츠 부분들을 나타낸다. 수신 문서(710)는 3개의 콘텐츠 참조(715, 720, 725)를 포함한다. 본 발명의 실시예들에 따르면, 파일 컨테이너(700)는 수신 문서 애플리케이션, 예를 들어 워드 프로세싱 애플리케이션, 스프레드시트 애플리케이션, 슬라이드 프리젠테이션 애플리케이션 등에 의한 로드 및/또는 론치 시에 수신 문서(710)와 병합되는 인커밍 콘텐츠 부분들과 수신 문서(710) 사이의 관계를 나타내는 관계 트리를 포함하는 관계 파일(750)에 링크된다.

[0280] 관계 파일(750)을 참조하면, XML 문서(710)에서 식별된 각각의 콘텐츠 참조는 관련 인커밍 콘텐츠 부분을 지시한다. 본 발명의 실시예들에 따르면, 수신 문서(710)의 조작을 담당하는 소프트웨어 애플리케이션, 예를 들어 워드 프로세싱 애플리케이션, 스프레드시트 애플리케이션, 슬라이드 프리젠테이션 애플리케이션 등이 수신 문서(710)를 로딩할 때, 문서(710)는 관련 파싱 애플리케이션, 예를 들어 XML 파서(parser)에 의해 수신 문서(710)에 적용된 구조에 대해 파싱된다. 각각의 참조(715, 720, 725)가 파싱될 때, 수신 애플리케이션은 어떠한 인커밍 또는 관련 콘텐츠 부분이 식별된 참조에 의해 지시되는지를 결정하기 위해 관계 파일(750)을 참조한다. 예를 들어, XML 문서(710)가 파싱될 때, 제1 참조(715)는 예를 들어 HTML의 파일 포맷 타입을 가진 콘텐츠 부분인 제1 콘텐츠 부분(730)을 지시하는 것으로 발견된다. 제2 참조(720)가 XML 문서(710)에서 파싱될 때, 관계 파일(750)에 대한 참조는 제2 참조(720)가 RTF의 파일 포맷을 가진 제2 인커밍 콘텐츠 부분(735)을 지시하는 것으로 결정한다. 유사하게, 제3 참조(725)가 수신 문서(710)에서 파싱될 때, 관련된 관계 파일(750)에 대한 참조는 제3 참조(725)가 바이너리 파일 포맷 타입을 가진 제3 인커밍 부분(740)을 지시하는 것을 보여준다. 참조된 콘텐츠와 연관된 파일 포맷 타입들과 같은 모든 정보가 리졸브되면, 수신 애플리케이션은 병합 문서를 표시하기 위해 지시된 콘텐츠 부분들과 수신 문서를 병합할 수 있다. 도 7을 참조하여 설명된 바와 같은 문서의 컴포넌트들의 포함 및 연관성에 대한 상세한 설명을 위해, 본 명세서에 충분히 설명된 바와 같이 참고로 반영된, 2004년 12월 20일자로 미국출원, "Improved Management And Use Of Data In A Computer-Generated Document", 대리인 관리번호 No.60001.0440US01, 출원인 관리번호 309861.01, 일련 번호 \_\_\_\_\_를 참조하라.

[0281] 본 명세서에 설명된 바와 같이, 텍스트 또는 데이터의 일부를 문서에 삽입하거나 텍스트 또는 데이터의 일부를 문서와 연관시키기 위한 방법 및 시스템이 제공되는데, 여기에서 삽입 또는 연관된 텍스트 또는 데이터의 부분은 그 자신과 함께 그 파일 포맷 타입 및 지정 스타일, 폰트, 리스트 타입 등과 같은 다른 포매팅 속성들을 운반하여, 수신 애플리케이션이 텍스트 또는 데이터의 부분을 적절히 처리할 수 있게 한다. 본 발명의 범위 또는 사상으로부터 벗어나지 않고 본 발명에서 다양한 수정 또는 변형이 이루어질 수 있다는 것이 이 분야의 전문가에게 자명할 것이다. 본 명세서에 개시된 본 발명의 명세서 및 실시를 고려할 때 본 발명의 다른 실시예들은 이 분야의 전문가에게 자명할 것이다.

**발명의 효과**

[0282] 본 발명에 따르면, 텍스트 또는 데이터의 부분들을 문서 내에 삽입하거나 문서에 의해 참조하기 위한 방법 및 시스템이 제공된다.

**도면의 간단한 설명**

[0001] 도 1은 본 발명을 위한 예시적인 동작 환경을 제공하는 컴퓨팅 시스템 및 관련 주변 장치 및 네트워크 장치의 간단한 블록도.

[0002] 도 2는 객체 지향 프로그래밍 모델에 따른 소프트웨어 객체들 간의 상호작용을 나타내는 간단한 블록도.

[0003] 도 3은 문서, 첨부 스키마 파일 및 스키마 검증 기능 모델 간의 상호작용을 나타내는 블록도.

[0004] 도 4는 본 발명의 실시예들에 따라 텍스트의 일부가 제2 문서에서 제1 문서로 복사 또는 이동되는 제1 문서와 제2 문서 간의 상호작용을 나타내는 블록도.

[0005] 도 5는 본 발명의 실시예들에 따라 텍스트의 일부가 제2 문서에서 제1 문서로 복사 또는 이동되는 제1 문서와 제2 문서 간의 상호작용을 나타내는 블록도.

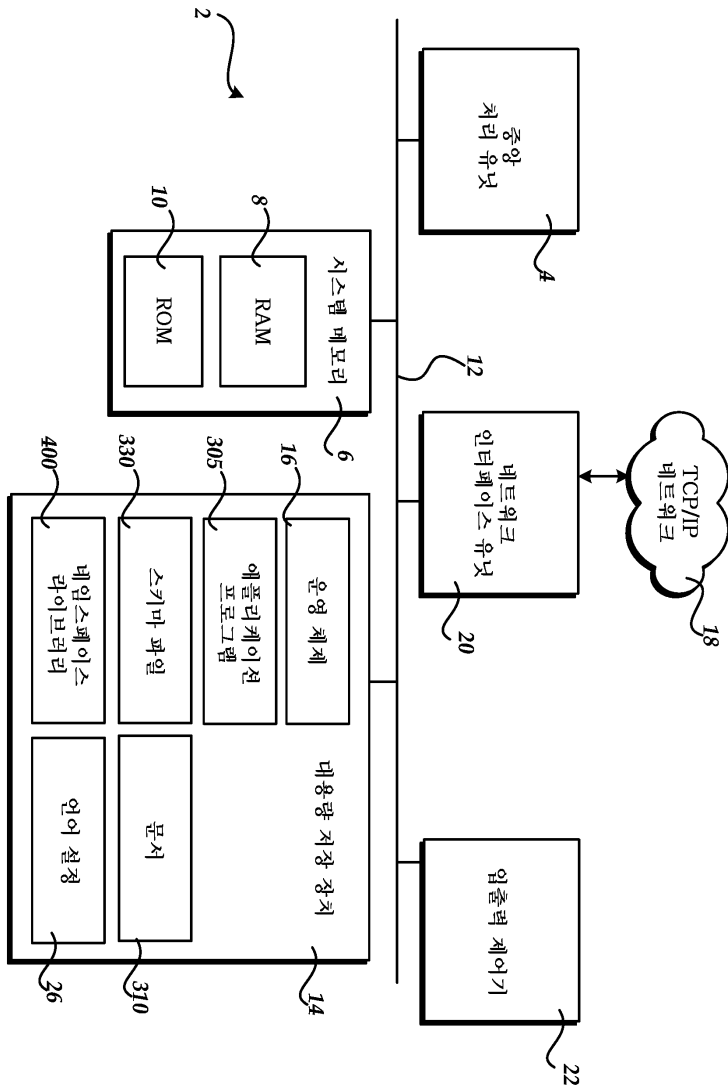


- [0006] 도 6은 본 발명의 실시예들에 따라 자기 기술 교체 파일 포맷들을 가진 컨텍스트-프리 문서 부분들 또는 컴포넌트들의 XML 기반 문서 내로의 삽입을 나타내는 간단한 블록도.
- [0007] 도 7은 본 발명의 실시예들에 따라 XML 기반 문서와의 병합을 위해 상이한 또는 교체 파일 포맷들을 가진 컨텍스트-프리 문서 부분들 또는 컴포넌트들을 참조하는 XML 기반 문서를 나타내는 간단한 블록도.
- [0008] <도면의 주요 부분에 대한 간단한 설명>
- [0009] 4: 중앙 처리 유닛
- [0010] 6: 시스템 메모리
- [0011] 8: RAM
- [0012] 10: ROM
- [0013] 14: 대용량 저장 장치
- [0014] 16: 운영 체제
- [0015] 18: TCP/IP 네트워크
- [0016] 20: 네트워크 인터페이스 유닛
- [0017] 22: 입출력 제어기
- [0018] 26: 언어 설정
- [0019] 305: 애플리케이션 프로그램
- [0020] 310: 문서
- [0021] 330: 스키마 파일
- [0022] 400: 네임 스페이스 라이브러리

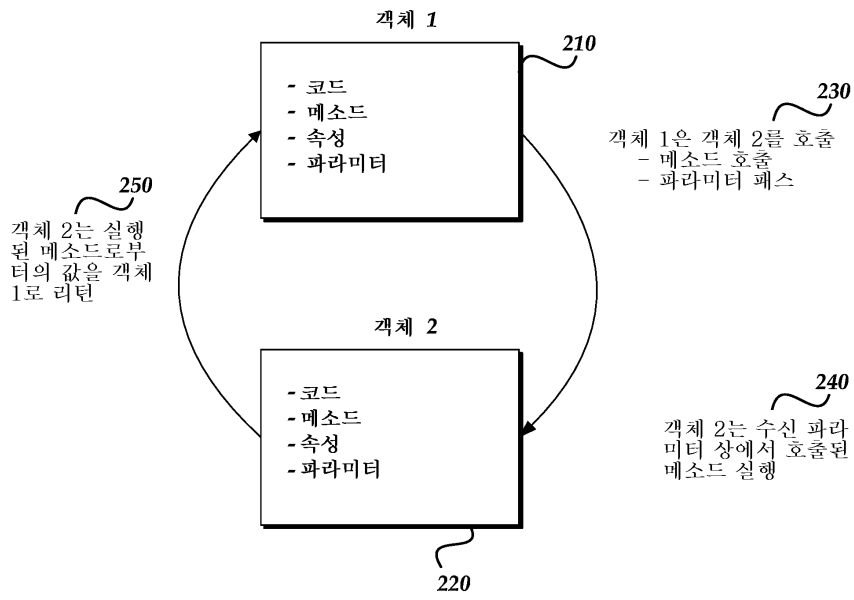


도면

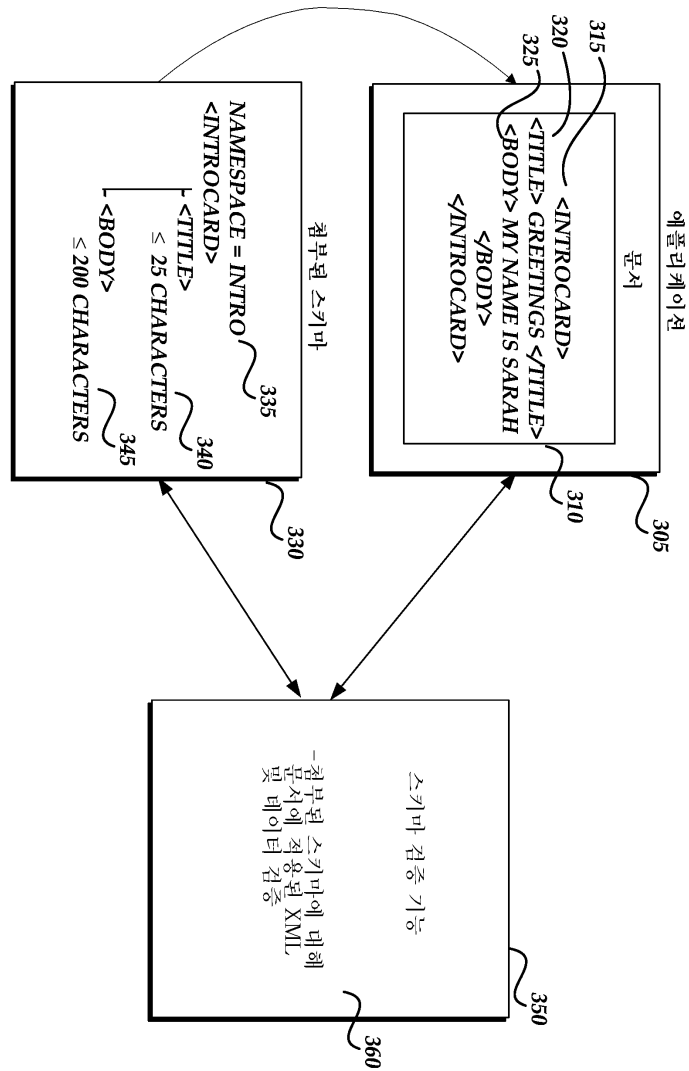
도면1



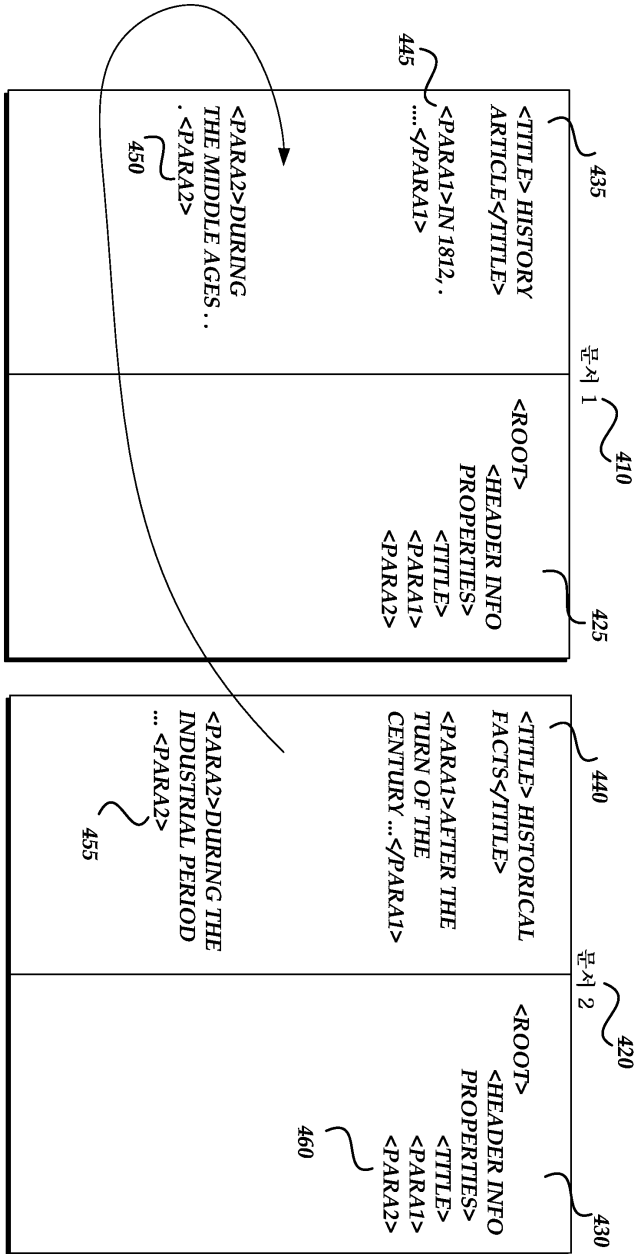
도면2



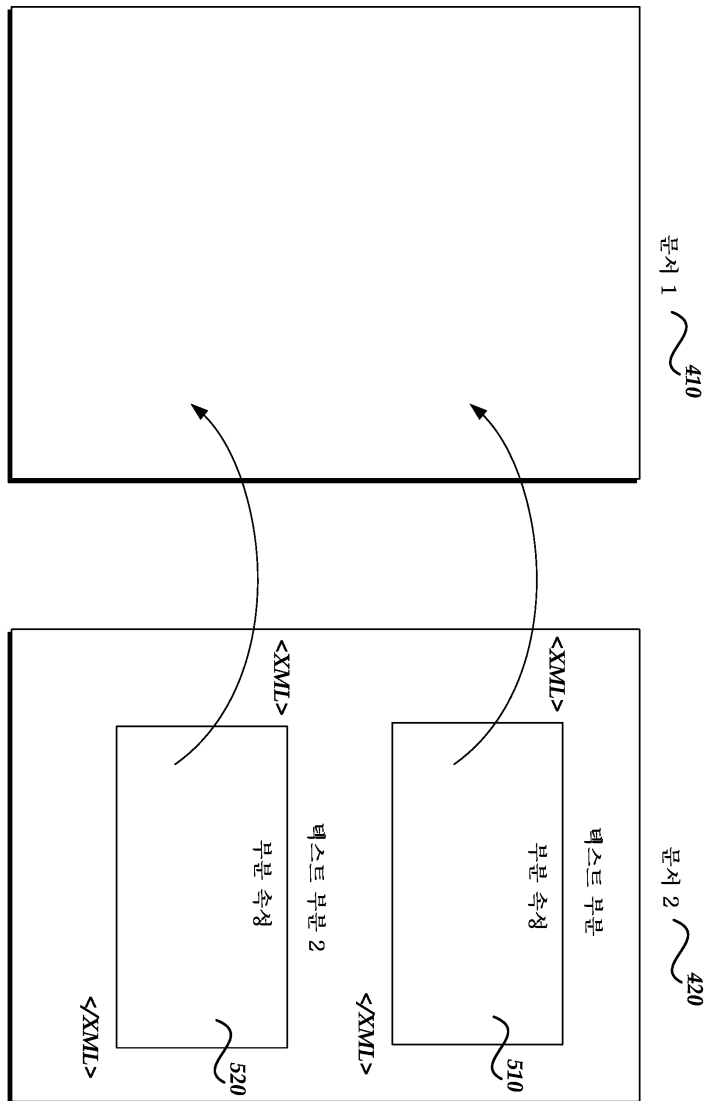
도면3



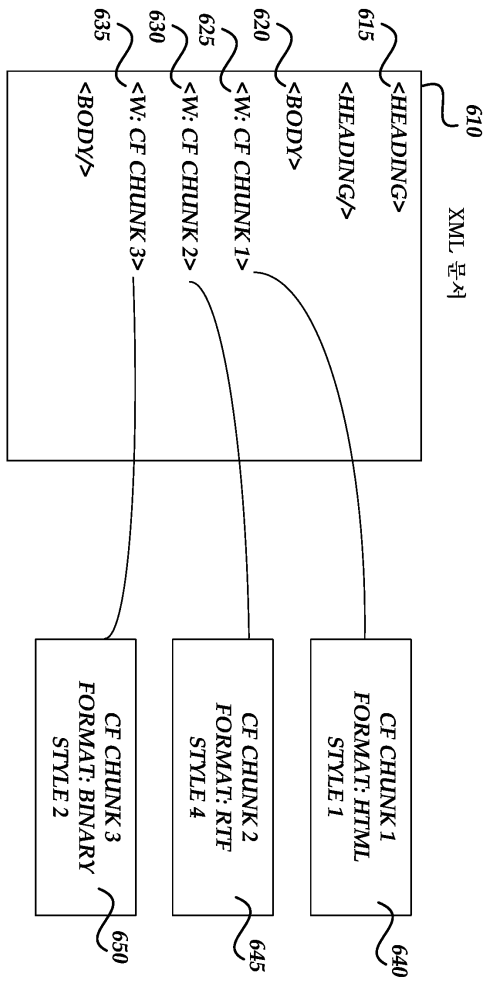
도면4

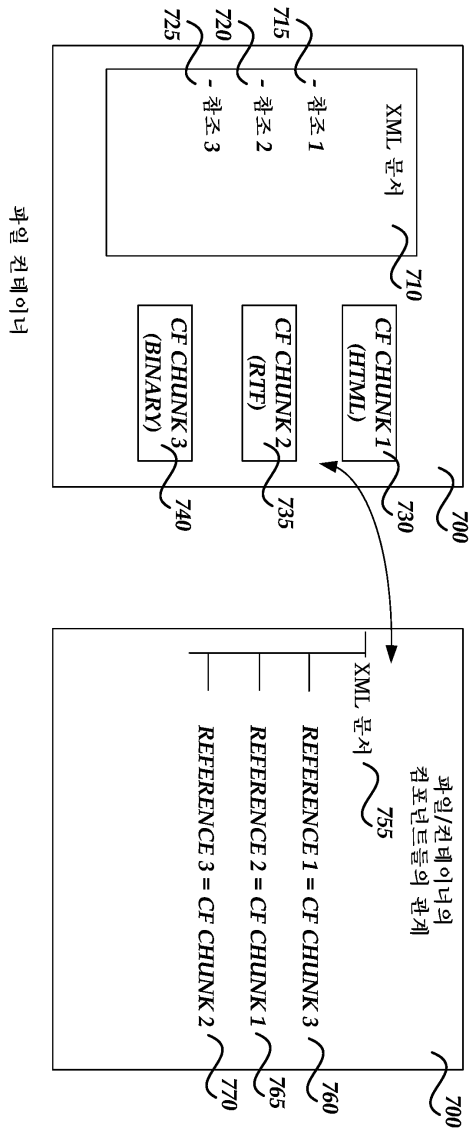


도면5



도면6





도면7