



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년04월17일
(11) 등록번호 10-0893829
(24) 등록일자 2009년04월10일

(51) Int. Cl.

G06F 17/21 (2006.01)

(21) 출원번호 10-2002-7013369
(22) 출원일자 2002년10월05일
심사청구일자 2007년01월30일
번역문제출일자 2002년10월05일
(65) 공개번호 10-2002-0087482
(43) 공개일자 2002년11월22일
(86) 국제출원번호 PCT/IB2002/000336
국제출원일자 2002년01월30일
(87) 국제공개번호 WO 2002/63494
국제공개일자 2002년08월15일

(30) 우선권주장

01/01530 2001년02월05일 프랑스(FR)

01402815.3 2001년10월30일

유럽특허청(EPO)(EP)

(56) 선행기술조사문헌

JP12222275 A*

JP12222276 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

코닌클리케 필립스 일렉트로닉스 엔.브이.

네덜란드왕국, 아인드호펜, 그로네보르스베그 1

(72) 발명자

아미에르-카프리-오그리오마이리암

네덜란드, 아아아인드호펜5656, 홀스트란6

챔피온매튜

네덜란드, 아아아인드호펜5656, 홀스트란6

데빌레스실바인

네덜란드, 아아아인드호펜5656, 홀스트란6

(74) 대리인

이병호, 장훈

전체 청구항 수 : 총 13 항

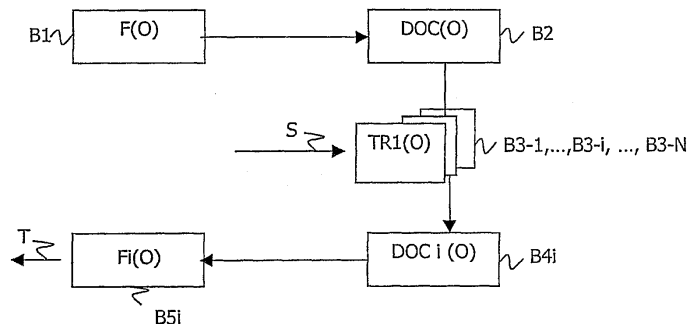
심사관 : 안병일

(54) 포맷 개작을 이용한 오브젝트 전송 방법

(57) 요약

본 발명의 목적은 소스 엔티티(예를 들면, 서버)에서 목적지 엔티티(예를 들면, 클라이언트)로 임의 버전의 오브젝트(예를 들면 이미지, 오디오 또는 비디오와 같은 멀티미디어 오브젝트)를 포함하는 파일을 전송하는 방법을 제공하는 것이며, 상기 버전은 목적지 엔티티의 프로파일의 함수(스크린, 계산 용량, 기억...)로서 소스 엔티티의 레벨로 생성된다. 본 발명에 따르면, 기본 버전의 오브젝트는 마크업 언어(예를 들면, XML)로 기록되고 기본 문서로 칭해지는 문서에 기술된다. 하나 이상의 변환은 (예를 들면, XSL 포맷으로) 규정된다. 기본 문서에 관한 변환의 애플리케이션은 변환 문서를 생성하도록 한다. 개작 버전의 오브젝트는 상기 변환 문서로부터 생성된다. 적용: 예를 들면, 인터넷을 통한 멀티미디어 오브젝트 다운로드.

대표도 - 도1



(81) 지정국

국내특허 : 중국, 일본, 대한민국, 미국

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 독일,
덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드,
이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투
갈, 스웨덴, 핀란드, 사이프러스, 터키

특허청구의 범위

청구항 1

개작 버전(adapted version)의 오브젝트(O)를 포함하는 파일을 소스 엔티티(source entity; 1)에서 목적지 엔티티(destination entity; 2)로 전송하는 방법으로서,

상기 소스 엔티티는, 마크업 언어(markup language)로 기록되고 기본 버전의 상기 오브젝트를 포함하는 기본 순차 파일(basic progressive file; F(O), CS)을 기술하는 기본 문서(DOC(O))를 저장하기 위한 메모리(10)를 포함하는, 상기 파일 전송 방법에 있어서:

- 상기 개작 버전의 함수로서 미리 규정된 변환(Tri(O))을 선택하는 단계;
- 변환된 문서(DOCi(O))를 생성하기 위해 상기 기본 문서에 상기 선택된 변환을 적용하는 단계를 포함하는 변환 단계로서, 상기 기본 문서는 상기 기본 문서에 적용 가능한 상기 미리 규정된 변환을 규정하기 위해 상기 기본 문서에서 상기 기본 순차 파일의 구조가 보일 수 있도록 상기 구조를 기술하고, 상기 변환 동작은 상기 기본 순차 파일을 또 다른 포맷으로 재코딩하기 위해 상기 기본 순차 파일을 디코딩해야 하는 것을 회피하는, 상기 변환 단계;
- 상기 변환된 문서에 기초하여 상기 개작 버전의 상기 오브젝트를 포함하는 파일(Fi(O))을 생성하는 단계;
- 상기 생성된 파일을 전송하는 단계를 포함하는 것을 특징으로 하는, 파일 전송 방법.

청구항 2

마크업 언어로 기록되고, 기본 버전의 오브젝트를 포함하는 기본 순차 파일을 기술하는 기본 문서로부터 개작 버전의 상기 오브젝트를 포함하는 파일을 생성하는 방법에 있어서:

상기 개작 버전의 함수인 미리 규정된 변환을 상기 기본 문서에 적용함으로써 변환된 문서를 생성하는 변환 단계로서, 상기 기본 문서는 상기 기본 문서에 적용 가능한 상기 미리 규정된 변환을 규정하기 위해 상기 기본 문서에서 상기 기본 순차 파일의 구조가 보일 수 있도록 상기 구조를 기술하고, 상기 변환 동작은 상기 기본 순차 파일을 또 다른 포맷으로 재코딩하기 위해 상기 기본 순차 파일을 디코딩해야 하는 것을 회피하는, 상기 변환 단계; 및

상기 변환된 문서로부터 상기 개작 버전을 포함하는 상기 파일을 생성하는 단계를 포함하는, 파일 생성 방법.

청구항 3

제 2 항에 있어서, 상기 기본 문서는, 태그들에 의해 한계가 정해지고 하나 이상의 속성들을 포함할 수 있는 하나 이상의 요소들을 포함할 때, 상기 변환은 하나 이상의 요소들을 억제하고 하나 이상의 속성들 및/또는 하나 이상의 요소들의 값을 변경하는 단계를 포함하는 것을 특징으로 하는, 파일 생성 방법.

청구항 4

제 2 항에 있어서, 상기 미리 규정된 변환은, 상기 마크업 언어로 기록된 문서를 상기 마크업 언어로 기록된 또 다른 문서로 변환하는 규칙들을 규정하도록 허용하는 변환 언어로 기록되는 것을 특징으로 하는, 파일 생성 방법.

청구항 5

마크업 언어로 기록되고 기본 순차 파일을 기술하는 문서를 포함하는 메모리와, 제 2 항에 청구된 파일 생성 방법을 구현하는 수단을 포함하는 전자 장비.

청구항 6

적어도 소스 엔티티와 하나 이상의 목적지 엔티티들을 포함하는 전송 시스템으로서, 상기 엔티티들은 제 1 항에 청구된 파일 생성 방법을 구현하도록 의도된, 전송 시스템.

청구항 7

미리 결정된 문자들의 세트를 이용하는 마크업 언어로 기록되는 문서를 저장하는 컴퓨터 판독 가능 기록 매체로서, 상기 문서는 콘텐츠를 포함하는 요소들 및/또는 하나 이상의 속성들 및/또는 하나 이상의 서브-요소들을 포함하고, 상기 문서는 2진 데이터, 마커들 및 하나 이상의 마커들과 연관된 파라미터들을 포함하는 2진 파일을 기술하는, 상기 컴퓨터 판독 가능 기록 매체에 있어서,

상기 문서에서, 요소가 각각의 마커와 연관되고, 상기 마커와 연관된 가능한 파라미터들이 상기 요소의 서브-요소 또는 속성을 형성하며, 하나 이상의 요소들의 상기 콘텐츠는 상기 2진 데이터와 링크되는 상기 문자들의 상기 세트의 문자들에 의해 형성되는 것을 특징으로 하는, 컴퓨터 판독 가능 기록 매체.

청구항 8

박스들로 포맷화되는 파일을 기술하는 문서를 저장하는 컴퓨터 판독 가능 기록 매체에 있어서, 상기 파일은:

- 순차 파일을 포함하는 메인 박스와,
- 마크업 언어로 기록되고 상기 순차 파일을 기술하는 문서를 포함하는 추가 박스를 적어도 포함하는, 컴퓨터 판독 가능 기록 매체.

청구항 9

프로그램을 저장하는 컴퓨터 판독 가능 기록 매체에 있어서, 상기 프로그램이 프로세서에 의해 실행될 때, 상기 프로그램이 제 1 항에 청구된 파일 전송 방법의 단계들을 실행하는 명령들을 포함하는, 컴퓨터 판독 가능 기록 매체.

청구항 10

프로그램을 저장하는 컴퓨터 판독 가능 기록 매체에 있어서, 상기 프로그램이 프로세서에 의해 실행될 때, 상기 프로그램이 제 2 항에 청구된 파일 생성 방법의 단계들을 실행하는 명령들을 포함하는, 컴퓨터 판독 가능 기록 매체.

청구항 11

제 1 항에 있어서, 상기 소스 엔티티(1)와 목적지 엔티티(2)가 서로 떨어져 있을 때, 상기 목적지 엔티티는 오브젝트 전송에 관련된 요청을 상기 소스 엔티티에 송신하도록(F1) 의도되며, 상기 목적지 엔티티는:

- 요청들 및 요청들에 대한 응답들을 수신 및 송신하도록 의도된 서버 모듈(30);
- 오브젝트 전송에 관련되는 상기 요청을 수신하고(F2), 상기 요청에 대응하는 기본 문서를 찾고, 상기 기본 문서에 관련되는 요청을 상기 서버 모듈에 송신하고(F3), 전송될 개작 버전의 오브젝트를 포함하는 파일을 생성하고, 상기 생성된 파일이 상기 목적지 엔티티에 송신되는 것(F8)을 고려하여 상기 서버 모듈에 상기 생성된 파일을 송신하도록(F7) 의도된 프로세싱 모듈(32);

상기 서버 모듈로부터 상기 기본 문서에 관련되는 요청을 수신하고(F4), 변환을 선택하고, 변환된 문서를 상기 프로세싱 모듈에 송신하는 것(F6)을 고려하여 상기 서버 모듈에 상기 변환된 문서를 송신하도록(F5) 의도된 변환 모듈(34)을 포함하는 것을 특징으로 하는, 파일 전송 방법.

청구항 12

제 11 항에 있어서, 상기 요청은 상기 소스 엔티티가 상기 목적지 엔티티의 프로파일(PP)을 결정하도록 허용하고, 상기 변환 모듈에 의해 적용되는 변환(TRi(0))은 상기 프로파일(PP)의 함수인 것을 특징으로 하는, 파일 전송 방법.

청구항 13

제 12 항에 있어서, 상기 요청(F1)은, 상기 소스 엔티티가 상기 목적지 엔티티의 프로파일(PP)을 결정하도록 허용하기 위해 상기 소스 엔티티(1)와 상기 목적지 엔티티(2) 사이에 협상 단계(S2)를 포함하는 것을 특징으로 하는, 파일 전송 방법.

명세서

기술 분야

- <1> 본 발명은 소스 엔티티에서 목적지 엔티티로 개작 버전의 오브젝트를 포함하는 파일을 전송하는 방법에 관한 것이다. 또한, 본 발명은 개작 버전의 오브젝트를 포함하는 파일을 생성하는 방법에 관한 것이다.
- <2> 본 발명은 또한,
- <3> - 이러한 파일 생성 방법을 구현하는데 적절한 장비와,
- <4> - 이러한 파일 전송 방법을 구현하는데 적절한 전송 시스템
- <5> 에 관한 것이다.
- <6> 본 발명은 또한 이러한 방법들의 단계들을 실행하기 위한 명령들을 포함하는 프로그램들에 관한 것이다.
- <7> 본 발명은 마지막으로:
- <8> ·마크업 언어(markup language)로 기록되는 문서와,
- <9> ·순차 파일(progressive file)을 포함하는 적어도 메인 박스(main box)와, 마크업 언어로 기록되고 상기 순차 파일을 기술하는 문서를 포함하고 적어도 사이드 박스(side box)를 포함하는 박스들로 포맷화되는 파일과,
- <10> 이러한 문서 또는 이러한 파일이 저장되는 기억 매체
- <11> 에 관한 것이다.

배경 기술

- <12> 본 발명은 인터넷 분야에서 특히 흥미 있는 애플리케이션들을 갖는다. 보다 많은 장비들이 인터넷에 액세스할 수 있다. 예를 들면, 개인용 컴퓨터들, 이동 전화기들, 개인 휴대 정보 단말기들, 휴대용 컴퓨터들, 포켓 컴퓨터들 등을 이용하는 경우가 그것이다. 이러한 장비들은 다소 제한된 자원들(스크린, 계산 용량, 기억 용량...)을 가지며, 가변 레이트들을 제공하는 링크들(무선 링크들, 유선 링크들, 광 링크들...)에 의해 인터넷에 접속된다. 전송될 오브젝트들은 수신기의 프로파일의 함수로서 개작되도록 전송될 필요가 있다. 이는 특히 수신기가 사용할 수 없는 데이터의 헛된 전송을 피하게 할 수 있으며, 따라서 통과 대역이 절약된다.
- <13> 1999년 8월 19일 공개된, 국제 특허 출원 제W099/41734호에는 순차 섹션들로 구성되는 그래픽 파일과 이러한 파일을 선택적으로 전송하는 방법에 관하여 기술되어 있다. 이 방법은 수신기가 순차 파일에 필요한 섹션들을 나타내는 것을 포함한다.
- <14> 따라서, 이 방법은 수신기가 그래픽 파일의 구조를 알고 있고, 획득하고자 하는 해상도 또는 품질의 함수로서 복구될 섹션들을 결정하는 방법을 알고 있음을 의미한다.

발명의 상세한 설명

- <15> 특히, 본 발명의 목적은 개작 버전의 오브젝트를 포함하는 파일을 전송하는 방법을 제공하는 것이며, 이 방법은 사용자로부터 특정한 지능을 요구하지 않는다.
- <16> 이 목적은 서두에서 기술된 바와 같은 전송 방법으로 달성되며, 상기 전송 방법에 있어서, 상기 소스 엔티티는 마크업 언어로 기록되고 기본 버전의 상기 오브젝트를 포함하는 기본 순차 파일을 기술하는 기본 문서를 저장하는 메모리를 포함하는 것에 특징이 있으며, 상기 방법은:
- <17> ·상기 개작 버전의 함수로서 미리 규정된 변환을 선택하는 단계와,
- <18> ·변환된 문서를 생성하기 위해 상기 기본 문서에 상기 선택된 변환을 적용하는 단계를 포함하는 변환 단계와,
- <19> ·상기 변환된 문서에 기초하여 상기 개작 버전의 상기 오브젝트를 포함하는 파일을 생성하는 단계와,
- <20> ·상기 생성된 파일을 전송하는 단계
- <21> 를 포함한다.

- <22> 따라서, 버전의 개작은 소스 엔티티 측에서 발생한다. 따라서, 목적지 엔티티들 측에는 어떠한 특정 지능도 필요하지 않다.
- <23> 본 발명에 따르면, 소스 엔티티는 마크업 언어로 된 기본 문서를 사용하며, 그 문서는 전송될 기본 버전의 오브젝트를 포함하는 기본 순차 파일을 기술한다. 소스 엔티티는 또한 하나 이상의 변환된 문서들을 얻기 위해 이 기본 문서에 적용되도록 의도된 하나 이상의 미리 규정된 변환들을 갖는다. 전송될 파일은 얻어진 변환 문서로부터 생성된다. 이는 전송될 개작 버전의 오브젝트를 포함한다. 이 개작 버전은 기본 버전과 상이하다.
- <24> 본 발명은 특히 마크업 언어로 오브젝트들을 기술하는 단계와 상기 획득된 기술들에 변환들을 적용하는 단계를 포함한다. 이하에서, 이 기술들은 문서들로 칭해진다. 이러한 마크업 언어로 기본 순차 파일의 구조를 기술할 수 있다. 이것은 기본 순차 파일의 구조가 얻어진 문서들에서 볼 수 있음을 의미한다. 따라서 이는 이 문서들에 적용하는 변환들을 규정할 수 있다. 이러한 변환들은 간단한 방법으로 표현되며, 그들의 사용은 아주 적은 계산을 요구한다.
- <25> 기본 순차 파일의 구조를 볼 수 있는 문서에 대하여 변환들을 실행함으로써, 기본 순차 파일들을 다른 포맷으로 기록하기 위해 기본 순차 파일들을 디코딩할 필요가 없게 된다. 따라서, 계산 부하는 상당한 범위로 제한된다.
- <26> 또한, 본 발명에 따르면, 미리 규정된 변환들뿐만 아니라, 소스 엔티티에서 기본 문서와, 어떤 실시예들에서 초대되는 기본 순차 파일을 충분히 저장할 수 있다. 개작된 변환을 적용함으로써 기본 문서로부터 요구가 발생할 때 전송될 파일이 생성된다. 따라서, 소스 엔티티의 메모리에 그것을 저장할 필요가 없다. 따라서, 소스 엔티티의 레벨에서 저장될 정보의 품질은 상당히 제한된다.
- <27> 본 발명의 바람직한 실시예에서, 기본 문서는 하나 이상의 속성들을 포함할 수 있는 태그에 의해 범위가 정해지는 하나 이상의 요소들을 포함하고, 상기 변환은 하나 이상의 요소들을 억제하는 단계와, 하나 이상의 속성들 및/또는 하나 이상의 요소들의 값을 변경하는 단계를 포함한다.
- <28> 미리 규정된 변환은 상기 마크업 언어로 기록된 문서를 상기 마크업 언어로 기록된 다른 문서로 변환하기 위한 규칙들을 규정할 수 있는 변환 언어로 기록되는 것이 바람직하다.
- <29> 예로서, 마크업 언어는 XML 언어(eXtensible Markup Language)이고, 상기 변환 언어는 XSL(eXtensible Style sheet Language)이며, 이 2개의 언어는 W3C 컨소시엄에 의해 규정된다.
- <30> 본 발명의 다른 목적은, 주어진 문자들의 세트를 사용하여 마크업 언어로 기록되고, 2진 데이터, 마커들 및 하나 이상의 마커들과 연관된 파라미터들을 포함하는 2진 파일을 기술하는 문서를 제공하는 것이다. 본 발명에 따르면, 이러한 문서는:
- <31> · 콘텐츠들을 포함하는 요소들, 및/또는 하나 이상의 속성들, 및/또는 하나 이상의 서브-요소들을 포함하고,
- <32> · 요소는 각각의 마커와 연관되고 가능한 파라미터들은 상기 요소의 속성 또는 서브-요소를 구성하는 상기 마커와 연관되고 하나 이상의 요소들의 콘텐츠들은 상기 2진 데이터와 연결되는 상기 문자들의 상기 세트의 문자들의 형태이다.
- <33> 제 1 실시예에서, 적어도 상기 문서의 콘텐츠의 일부는 상기 세트의 문자들로 변환된 상기 2진 파일의 2진 데이터에 대응한다.
- <34> 이 제 1 실시예에서, 문서는 2진 파일의 데이터를 자체적으로 포함하고 있기 때문에 2진 파일에 독립적이다. 그러나, 문자들로의 2진 데이터의 변환은 이 데이터의 임의 확장을 의미한다.
- <35> 제 2 실시예에서, 적어도 상기 문서의 콘텐츠의 일부는 상기 2진 파일의 상기 2진 데이터에 관한 하나 이상의 포인터들에 대응한다.
- <36> 이 제 2 실시예에서, 상기 문서는 데이터에 액세스할 수 있는 2진 파일로 되돌아온다. 따라서, 사용하기 위해 상기 문서는 그것을 기술하는 2진 파일에 의해 수반되는 것이다. 따라서 이 실시예에서, 어떠한 데이터 확장은 없다.
- <37> 본 발명의 다른 목적은, 본 발명에 따른 파일 전송 방법을 구현하는데 유리하게 사용되는 특정 2진 파일을 제공하는 것이다.
- <38> 이러한 2진 파일은 박스들로 포맷화되며, 적어도 이하들:

- <39> · 순차 파일을 포함하는 메인 박스,
- <40> · 상기 순차 파일을 기술하는 마크업 언어로 기록된 문서를 포함하는 적어도 추가 박스를 포함한다.
- <41> 상기 추가 박스는 상기 문서에 적용되도록 의도된 하나 이상의 변환들을 포함하는 것이 바람직하다.
- <42> 따라서, 오브젝트에 관련되는 모든 정보는 단일 파일로 재그룹화되며, 이것은 파일 관리의 시점에서 볼 때 바람직하다.
- <43> 이러한 파일은, 예를 들면 JPEG 2000파일이다. 이 경우에 추가 박스는, 예를 들면 JPEG 2000 표준에 규정되는 선택적 "XML" 박스로 형성된다. 이는 또한 새로운 전용 박스에 의해서 구성될 수 있다.
- <44> 상기 소스 및 목적지 엔티티들이 서로 떨어져 있을 때, 목적지 엔티티는 오브젝트 전송에 관련되는 요청을 소스 엔티티에 전송하는 것이 바람직하며, 상기 목적지 엔티티는:
 - <45> · 요청들 및 요청들에 관한 응답들을 수신 및 송신하도록 의도된 서버 모듈과,
 - <46> · 오브젝트 전송에 관련되는 상기 요청을 수신하고, 상기 요청에 대응하는 기본 문서를 찾고, 상기 기본 문서에 관련되는 요청을 상기 서버 모듈에 송신하고, 전송될 개작 버전의 오브젝트를 포함하는 파일을 생성하여, 그 생성된 파일이 목적지 엔티티에 송신되는 것을 고려하여 상기 서버 모듈에 상기 생성된 파일을 송신하도록 의도된 프로세싱 모듈과,
 - <47> · 상기 서버 모듈로부터 초래되고, 변환을 선택하기 위해 상기 기본 문서에 관련되는 상기 요청을 수신하고, 상기 프로세싱 모듈에 변환된 문서를 송신하는 것을 고려하여 상기 서버 모듈에 변환된 문서를 송신하도록 의도된 변환 모듈
- <48> 을 포함한다.
- <49> 또한, 목적지 엔티티에 의해 송신된 요청은 소스 엔티티가 목적지 엔티티의 프로파일을 결정하게 하며, 상기 변환 모듈에 의해 적용된 변환은 상기 프로파일의 함수이다. 예를 들면, 상기 요청은 소스 엔티티가 목적지 엔티티의 프로파일을 결정할 수 있게 하는 소스 엔티티와 목적지 엔티티 사이에 협상 단계를 포함한다.

실시예

- <56> 본 발명은 특히 소스 엔티티에서 목적지 엔티티로 임의 버전의 오브젝트를 포함하는 파일을 전송하는 방법에 관한 것이다. 본 발명에 따른 이 전송 방법은 상기 전송에 사용될 버전을 생성하는 단계를 포함한다. 이러한 생성은 마크업 언어로 기록되고 기본 순차 파일을 기술하는 기본 문서에 기초하여 실행되며, 상기 기본 순차 파일은 기본 버전의 상기 오브젝트를 포함한다. 이러한 단계를 통해 변환된 문서가 생성될 수 있다. 마지막으로, 이러한 변환된 문서는 개작 버전으로 칭해지는 또다른 버전의 오브젝트를 포함하는 파일을 생성하는데 사용된다. 이 파일은 그 다음에 목적지 엔티티에 전송된다.
- <57> 도 1은 본 발명에 따른 파일 전송 방법의 단계들을 요약한 블록도이다. 블록(B1)은 임의 버전의 오브젝트 0에 대응하는 기본 순차 파일 F(0)를 나타낸다. 블록(B2)은 마크업 언어로 기록되고 기본 순차 파일을 기술하는 기본 문서 DOC(0)를 나타낸다. 블록들(B3-1) 내지 (B3-N)은 변환된 문서를 생성하기 위해 기본 문서 DOC(0)에 적용되도록 의도된 변환들 TRi(0)를 나타낸다. 선택 단계 S를 통해, 오브젝트(0)의 버전과, 또 선택된 버전의 함수인 적용될 변환을 선택할 수 있다. 블록(B4-i)은 기본 문서 DOC(0)에 변환 TRi(0)를 적용함으로써 생성되는 변환된 문서 DOC(0)를 나타낸다. 마지막으로, 블록(B5-i)은 변환된 문서 DOCi(0)로부터 생성되고 다른 버전의 오브젝트(0)를 포함하는 파일 F(i)를 나타낸다. 끝으로, 단계 T에서, 파일 Fi(0)는 목적지 엔티티에 전송된다.
- <58> 이 파일 전송 방법은 기본 파일의 순차 특성에 이점이 있다. 이 방법은, 개작 버전들의 오브젝트를 생성하기 위해 그 특성을 조작할 수 있도록 문서에 이 순차 기본 파일의 신택스를 기술하는 단계로 구성된다.
- <59> 이러한 방법은 소스 엔티티와 목적지 엔티티가 서로 떨어져 있는 본 발명에 따른 전송 시스템에 사용되는 것이 바람직하다. 이러한 전송 시스템의 일례는 도 2에 도시된다. 이 시스템은 소스 엔티티와, 액세스들이 4 및 5로 각각 언급되는 각각이 인터넷에 액세스할 수 있는 적어도 목적지 엔티티를 포함한다.
- <60> 사실상, 소스 엔티티는 하나 또는 다양한 기계들로 확장될 수 있다. 다양한 기계들로 확장됨으로써 태스크들을

분배할 수 있고, 소스 엔티티의 레벨에서 임의 과부하를 방지할 수 있게 된다. 이하에서는, 간단하게 하기 위하여, 소스 엔티티는 서버로 칭해지는 단일 기계에 수용된다.

- <61> 서버(1)는 메모리(10) 및 프로세서 어셈블리(12)를 구비한다. 메모리(10)는, 예를 들면, 판독 전용 메모리(ROM), 랜덤 액세스 메모리(RAM), 하드 디스크(HD), 및 그 기능이 상기 매체를 판독하기 위한 것인 판독기에 삽입되는 플로피 디스크 형태의 제거할 수 있는 기억 매체(FD), CD-ROM, DVD...이다. 메모리(10)는 특히:
 - <62> · 기본 순차 파일 F(0),
 - <63> · 이 기본 순차 파일을 기술하는 기본 문서 DOC(0),
 - <64> · 기본 순차 파일에 적용되도록 의도된 적어도 변환 TRi(0),
 - <65> · 본 발명에 따른 파일 전송 방법의 구현을 위해 프로세서(12)에 의해 실행되는 프로그램들의 세트 P10,
 - <66> 를 포함한다.
- <67> 클라이언트 장비(2)는 메모리(20), 프로세서 어셈블리(22), 스크린(23), 및 예를 들면 키보드, 보이스 컨트롤, 마우스...를 포함하는 사용자 인터페이스(24)를 구비한다. 메모리(22)는, 예를 들면, 판독 전용 메모리(ROM) 및 랜덤 액세스 메모리(RAM)로 형성된다. 어떤 경우에, 메모리는 또한 하드 디스크(HD), 및/또는 그 기능이 상기 매체에 저장된 데이터를 판독하기 위한 것인 판독기에 삽입되는 디스크 형태의 제거할 수 있는 기억 매체 FD, CD-ROM, DVD...이다. 메모리(22)는 특히 본 발명에 따른 전송 방법을 구현하기 위해 프로세서(22)에 의해 실행되는 공지된 프로그램들의 세트 P20을 포함한다.
- <68> 클라이언트 장비(2)는 통상적으로 인터넷에 접속되고, 자원 식별자를 포함하는 HTTP 형태의 요청을 출력한다(HTTP는 IETF의 RFC2616에 규정되는 네트워크 프로토콜이다; 자원 지시자의 포맷은 IETF의 RFC2396에 규정된다). 이 자원을 내장한 장비는 HTML 문서를 상기 클라이언트에 전송함으로써 응답하며, 이 문서는 다음에 클라이언트 장비에 의해 해석되어 장비의 스크린에 상기 문서의 콘텐츠를 디스플레이한다(HTML은 인터넷상에서 현재 사용되고 W3C 컨소시엄에 의해 규정되는, 콘텐츠를 기술하기 위한 언어이다). HTML 문서는 스크린상에 디스플레이되는 페이지를 기술하고, 이러한 페이지는 사용자가 임의 오브젝트를 포함하는 파일을 다운로드하기 위해 선택하는 링크(예를 들면, 버튼 또는 아이콘 형태)를 포함하는 것이 통상적이다. 사용자가 이러한 링크를 선택할 경우, 클라이언트 장비는 상기 파일의 전송을 요구하기 위해 HTTP 형태의 요청을 전송한다.
- <69> 도 3에서, 소스 엔티티(1)의 소프트웨어 구조에 관한 일례를 도시하고 있다. 도 3에 따르면, 소스 엔티티(1)는 3개의 소프트웨어 모듈들; 서버 모듈(30), 프로세싱 모듈(32) 및 변환 모듈(34)을 포함한다. 서버 모듈(30)은, 예를 들면, APACHE(APACHE는 아파치 소프트웨어 기초(foundation)에 의해 개발된 http 서버이다)로 명명되는 모듈 형태이다. 다른 모듈들에 임의 요청을 전송하도록 구성된다. 여기서, 서버 모듈(30)은 프로세싱 모듈(32)에 순차 포맷(예를 들면, jpeg 2000 파일)을 가지는 파일에 관련된 요청을 전송하고, 변환 모듈(34)에 마크업 언어(예를 들면, XML 파일)로 기록되는 파일에 관련된 요청을 전송한다. 변환 모듈(34)은, 예를 들면 COCOON(COCOON은 APACHE XML 프로젝트의 테두리안에서 JAVA 언어로 개발된 애플리케이션이다)으로 명명되는 모듈로 형성된다.
- <70> 특히, 프로세싱 모듈(32)의 기능은 목적지 엔티티에 의한 전송 요청에 나타나는 순차 파일 포맷에 대응하는 기본 문서 DOC(0)를 찾는 것과, 서버 모듈(30)에 상기 기본 문서 DOC(0)에 관련된 요청을 송신하는 것이다. 상기 요청을 수신한 경우, 서버 모듈(30)은 변환 모듈(34)에 의해 처리되는 생성 포맷(예를 들면, XML)을 인식하고, 변환 모듈(34)에 그 요청을 송신한다.
- <71> 변환 모듈(34)의 기능은 기본 문서 DOC(0)를 복구하고, 변환 문서 DOCi(0)를 생성하기 위해 그 기본 문서 DOC(0)에 적용되는 변환 TRi(0)를 선택하는 것이다. 프로세싱 모듈(32)과 동일한 것을 송신하는 서버 모듈(30)에 그 변환된 문서 DOCi(0)를 되돌려 보낸다.
- <72> 프로세싱 모듈(32)의 기능은 전송될 개작 버전의 오브젝트를 포함하는 2진 파일 Fi(0)를 생성하는 것과, 목적지 엔티티(1)에 오브젝트를 송신하기 위해 서버 모듈(30)에 오브젝트를 되돌려 보낸다.
- <73> 도 3에는 소스 엔티티(1)에서 목적지 엔티티(2)로 번호가 부여된 화살표 형태로 개작 버전의 전송에 필요한 여러 단계가 도시되어 있다:
- <74> · 화살표 F1은 목적지 엔티티(2)에 의한 요청 전송을 나타내며, 이 요청은 오브젝트 전송에 관련된다.

- <75> · 단계 F2에서, 서버 모듈(30)은 목적지 엔티티에 의해 송신되는 요청을 수신하고, 순차 포맷을 인식하고 프로세싱 모듈(32)에 상기 요청을 송신한다.
- <76> · 단계 F3에서, 프로세싱 모듈은 순차 포맷에 대응하는 기본 문서를 찾고 상기 문서에 관련되는 요청 http GET를 실행시킨다. 기본 문서를 복구하기 위해, 암시적 규칙(예를 들면, 전송될 오브젝트를 포함하는 파일이 image.jp2로 칭해지는 경우, 관련된 XML 문서는 image.xml로 칭해진다고 판정할 수 있다) 또는 명시 규칙(예를 들면, 각각이 기본 순차 파일에 관련된 기본 문서의 명칭을 포함하는 대응 테이블이 이용 가능하다)을 적용한다.
- <77> · 단계 F4에서, 서버 모듈(30)은 변환 모듈(34)에 기본 문서와 관련되는 요청을 송신한다.
- <78> · 단계 F5에서, 변환 모듈(34)은 기본 문서를 복구하고, 변환을 선택하고 서버 모듈(30)에 되돌려 보내는 변환된 문서를 생성하기 위해 선택된 변환을 적용한다.
- <79> · 단계 F6에서, 프로세싱 모듈(32)에 변환 모듈(34)에 의해 되돌려 보내지는 변환된 문서를 송신한다.
- <80> · 단계 F7에서, 프로세싱 모듈(32)은 변환 문서에 기초하여 전송될 개작 버전의 오브젝트를 포함하는 2진 파일을 생성하고, 서버 모듈(30)에 그 생성된 2진 파일을 되돌려 보낸다.
- <81> · 마지막으로, 단계 F8에서, 서버 모듈(30)은 목적지 엔티티(2)에 개작 버전의 오브젝트를 포함하는 파일을 송신한다.
- <82> 3개의 모듈이 상이한 기계들에 내장되어 있는 경우, 모두 본 발명에 따른 전송 방법을 실행하는데 필요한 데이터(즉, 보다 구체적으로 이하와 같음)가 저장된 동일 디스크에 도달하는 것이 바람직하다:
- <83> · 기본 순차 파일 F(0),
- <84> · 상기 기본 순차 파일을 기술하는 기본 문서 DOC(0),
- <85> · 상기 기본 순차 파일에 적용되는 변환 TRi(0).
- <86> 프로세싱 모듈(32) 및 변환 모듈(34)은, 예를 들면 "서브렛들(servlets)", 즉 서버(예를 들면, "코콘(cocoon)"은 "서브렛"임)에서 실행되는 JAVA 언어로 개발된 애플리케이션들이다.
- <87> 본 발명에 따른 파일 전송 방법의 다양한 애플리케이션에 대하여 이제 기술되며, 도 3을 참조하여 기술되는 클라이언트-서버 구조가 사용된다.
- <88> 도 4에 도시되는 애플리케이션의 제 1 일레에서, 클라이언트(2)에 의해 전송되는 요청 J1에 의해 서버(1)는 클라이언트(2)의 프로파일(PP)을 결정할 수 있다. 제 1 실시예에서(도시되지 않음), 요청 J1은 클라이언트(2)의 프로파일(PP)을 직접 표시한다. 도 4에 도시된 제 2 실시예에서, 서버(1)가 요청 J1을 수신한 경우, 협상 단계 J2를 개시함으로써 응답한다. 이 협상 단계를 통해서, 클라이언트(2)의 프로파일(PP)을 알 수 있다. 서버(1) (및 보다 정확하게는 서버(1)의 변환 모듈(34))은 클라이언트의 프로파일(PP)의 함수로서 적절한 파일을 생성하도록 기본 문서 DOC(0)에 적용되는 변환을 선택한다. 일례로서, 이러한 협상 단계는 "Transparent Content Negotiation in HTTP"이라는 제목하에 1998년 IETF에 의해 공개된 RFC2295, 또는 "1999년 11월 10일, "Wireless Application Protocol Forum"에 의해 공개된 문서 "Wireless Application Group, User Agent Profile Specification"에 기술되는 형태일 수 있다.
- <89> 애플리케이션에 관한 제 2 일레는, 동일 오브젝트의 완벽한 해상도를 다운로드하기 위해 클라이언트가 클릭할 수 있는 낮은 해상도의 오브젝트(중중, 이미지)가 웹 페이지에 포함되는 경우에 관련된다. 그 경우에, 서버(1)의 변환 모듈(34)이 클라이언트가 이미 사용하고 있는 데이터의 보수 데이터만을 포함하는 파일을 생성하기 위해 특정 변환을 적용한다.
- <90> 애플리케이션에 관한 제 3 일레는, 동일 웹 사이트에 이미지가 상이한 포맷들로 또는 상이한 해상도로 몇 시간 동안(예를 들면, 상이한 페이지에) 제공되는 경우에 관한 것이다. 클라이언트가 페이지들 사이에서 항해하는 경우, 서버는 연속적으로 다양한 버전들의 동일 이미지를 송신하게 한다. 변환 모듈(34)이 소망의 형태를 생성하도록 적절한 변환을 적용할 때마다, 송신될 수 있는 다양한 이미지 형태를 서버에 저장할 필요가 없다.
- <91> 본 발명은 도 3에 기술되는 형태의 클라이언트-서버 구조의 사용에 한정되지 않는다. 본 발명은 또한 국부적으로는 사용자에 의해 사용되는 개작 버전의 오브젝트를 포함하는 파일을 생성하는 방법에 관련된다. 예를 들면, 이러한 파일 생성 방법은 대형 이미지에 기초하여 소형 이미지를 생성하고자 하는 사용자에 의해 사용되는 것이

바람직하다.

- <92> 본 발명은, 예를 들면 멀티미디어 오브젝트에, 즉 고정 이미지 오디오 또는 비디오에 적용된다.
- <93> 상기 형태의 오브젝트 형태를 포함하는 파일들은, 예를 들면 JPEG 2000 포맷, GIF 포맷 또는 MPEG4 포맷의 파일들이다. 이하의 기술에서는, 파일들은 2진 파일로 한정된다. 왜냐하면, 파일들은 물리적으로 임의 비트 시퀀스로 나타나기 때문이다.
- <94> 순차 파일은, 파일의 일부만을 복구함으로써 저하된 형태의 오브젝트를 획득하도록 구성되는 파일이다. 다양한 형태의 진행들이 있다: 예를 들면, 보다 많은 데이터를 복구함으로써 신호의 품질을 향상시켜서 대형 이미지를 얻게 되고, 그레이-레벨 이미지가 아닌 컬러 이미지를 얻게 되고, 비디오의 프레임 주파수를 증가시키게 된다. 예를 들면, JPEG2000 파일들은 순차 파일들이다.
- <95> 이미지 파일들의 임의 포맷들에 의해, 이미지 파일에 포함되는 저하된 형태(종종 소형 형태)의 이미지인 "섬네일(thumbnail)"을 이미지 파일에 포함할 수 있다. 이것은, 예를 들면 Exif2.1 파일들("Digital Still Camera Image File Format Standard; Exchangeable image file format for Digital Still Cameras: Exif, Version 2.1", 12 June 1998, Japan Electronic Industry Development Association(JEIDA)). 이미지 파일 포맷을 디지털 스틸 마을 갖는 경우이다. 또한, 이러한 형태의 파일은 본 발명의 의미에서 순차 파일들을 구성한다. 상기 형태의 파일에 관련된 기본 문서는 섬네일을 기술하고, 다른 한편으로는 이미지를 기술한다. 예를 들면, 섬네일을 기술하는 변환 문서를 생성하도록 기본 문서로부터 섬네일에 관련된 부분을 추출할 수 있는 변환이 규정되게 되며, 상기 변환 문서는 상기 섬네일을 포함하는 2진 파일을 생성하는데 사용된다.
- <96> 통상적인 방법에서, 마크업 언어는, 태그를 이용하고 데이터 세트의 선택스를 기술하기 위한 이러한 태그의 사용 규칙을 규정하는 언어를 의미하는 것으로 이해할 수 있다. 따라서, 이러한 언어는 데이터 세트를 구성하고, 그 콘텐츠와 데이터 세트의 구조를 분리시킬 수 있도록 한다.
- <97> XML은 이러한 마크업 언어의 통상적인 일례이다. XML은 주로 인터넷 월드에 사용되는 이점을 제공한다. XML의 또다른 이점은, 임의 개수의 톨, 특히 XML 문서에 적용될 수 있는 변환을 규정할 수 있는 XSLT로 칭해지는 톨을 제공하는 것에 있다. 변환에 관한 규정은, XSLT 스타일 시트를 통해 만들어진다. XSLT 변환에서, XSLT 프로세서는 또다른 문서, 예를 들면 XML 문서를 생성하도록 XML 문서 및 XSLT 스타일 시트를 판독한다. 따라서, XML 및 XSLT는 특히 본 발명의 실행에 알맞다.
- <98> 이하의 기술에서, 구체적인 일례를 제공하기 위해, XML 언어는 JPEG 2000 포맷의 파일을 기술하는 문서를 생성하는데 사용된다. 이것은 한정적이지 않다. 또다른 형태의 마크업 언어 및 다른 순차 파일 포맷이 사용될 수 있다.
- <99> JPEG 2000 표준은 "JPEG 2000 Final Draft International Standard"라는 제목의 문서 ISO/IEC FCD 15444-1에 기술되어 있다.
- <100> 메인 JPEG 2000 코딩 단계는 이하와 같다:
- <101> a) 스페이스 Y-Cr-Cb에 이미지의 표시를 얻기 위한 입력 이미지의 선택적인 변환 단계, 여기서 Y는 루미넌스 성분이고, Cr은 크로미넌스 성분 레드이고, Cb는 크로미넌스 성분 블루이다;
- <102> b) 이미지 성분의 웨이브렛(wavelet) 변환 단계;
- <103> c) 획득된 계수의 수량화 단계;
- <104> d) 코딩 단계;
- <105> e) 임의 순차 구성에 따른 비트 스트림에 얻어지는 데이터를 삽입하는 단계.
- <106> 코딩될 이미지가 매우 큰 경우, 일차적인 단계를 통해 서로 무관하게 코딩되는 여러 타일들로 이미지를 분배할 수 있다. 이하의 기술에서, 간단하게 하기 위해, 이미지는 싱글 타일을 포함하는 경우로 한정된다.
- <107> JPEG 2000은 4개의 기본적인 순차 방식: 해상도, 품질, 공간적 위치 및 성분의 사용을 제공한다. JPEG 2000은 또한 상기 기본 방식의 5개 결합을 제공한다. 각각의 순차 방식은 비트 스트림의 데이터를 주문하는 특정 방법에 대응한다.
- <108> 비트 스트림은 패킷 헤더와 코딩으로부터 초래되는 유용한 데이터를 포함하는 패킷을 포함한다. 각 패킷의 유용한 데이터는 성분(i), 레이어(j), 해상도 레벨(k) 및 분할 위치(m)에 대응한다. 비트 스트림은 4개의 루프를

사용하여 구성된다. 루프의 순서는 사용되는 순차 방식을 결정한다.

- <109> 성분들을 통해 진행하는 비트 스트림에 대하여, 4개 루프의 순서는, 예를 들면 다음과 같다: 각 성분에 대하여, 각 해상도에 대하여, 각 레이어에 대하여, 각 분할 위치에 대하여 ⇒ 비트 스트림에의 패킷의 기록. 해상도를 통해 진행하는 비트 스트림에 대하여, 4개 루프의 순서는, 예를 들면, 각 해상도에 대하여, 각 성분에 대하여, 각 레이어에 대하여, 각 분할 위치에 대하여 ⇒ 비트 스트림에의 패킷의 기록. 이하의 기술에서, 상기 2개의 진행에 관한 일례로 한정되지만, 본 발명은 임의 버전의 진행에 적용 가능하다.
- <110> JPEG 2000 전문 용어에 따르면, 이미지에 관련된 모든 정보는 코드 스트림에 결합된다. 코드 스트림은 메인 헤더, 하나 이상의 타일 부분 헤더들 및 이 헤더들 다음의 비트 스트림을 포함하고, 종료 마커들에 의해 종료된다. 메인 헤더 및 타일 부분 헤더는 마커들 및 마커 세그먼트들로 구성된다. 마지막으로, 코드 스트림 자신은 2진 파일에 포함된다.
- <111> 도 5는 JPEG2000 코드 스트림의 일례에 관한 구조를 도시하고 있다. 이 코드 스트림 CS는 메인 헤더(100)에서 개시하며, 이 헤더(100)는 마커 SOC(Start Of Code stream), 이 마커 다음의 마커 세그먼트 SIZ(image and tile SIZE), COD(Coding style Default) 및 QCD(Quantization Default)를 포함한다. 그 다음, 이는 마커 세그먼트 SOT(Start Of Tile part), 다음의 마커 SOD(Start Of Data)를 포함하는 타일 부분 헤더(110)를 포함한다. 마지막으로, 이는 패킷들의 개시를 표시하는데 사용되는 마커 SOP(Start Of Packet)에 의해 선택적으로 처리되는 다양한 패킷 형태의 비트 스트림(120)을 포함한다. 그리고, 이는 종료 마커 EOC(End Of Code stream)에 의해 종료된다. 여기에서 사용되는 명칭은 JPEG 2000 권고의 명칭들이다. 다른 선택적인 마커는 JPEG 2000 권고에 규정된다. 보다 구체적으로, 마커는 2-옥텟 코드이다. 그리고, 마커 세그먼트는 마커 및 이 마커에 관련되는 파라미터들을 포함한다.
- <112> XML 언어 베이스에 대하여 이제 설명한다. 이 언어는 1998년 2월 10일에 공개된 REC-xml-19980210 제목하의 W3C 컨소시엄의 권고에 기술되어 있다.
- <113> 물리적인 관점에서, XML 문서는 분석될 수 있는 엔티티들과 분석될 수 없는 엔티티들을 포함할 수 있다. 분석될 수 있는 엔티티는 텍스트, 즉 미리 규정된 문자 어셈블리에 속하는 문자들의 시퀀스를 포함하며, 이들은 태그들 또는 텍스트 데이터를 나타낸다. 분석될 수 없는 엔티티는 텍스트, 즉 XML 텍스트가 아닌 텍스트 이외의 어떤 것을 포함할 수 있다.
- <114> 논리적인 관점에서, XML 문서는 그 한계가 개방 태그 및 폐쇄 태그로 표시되는 하나 이상의 요소를 포함한다. 요소들은 서로 인터리빙될 수 있다. 각 요소는 명칭으로 식별된다. 그것은 콘텐츠를 포함한다. 그것에 속성 명세 세트를 연관시킬 수 있다. 각 속성 명세는 명칭 및 값을 포함한다. 어떠한 속성 명칭도 동일 개시 태그에 다양한 타임을 나타낼 수 없다.
- <115> XML 언어를 이용하여 JPEG 2000 코드 스트림을 구성하는 방법에 대하여 이제 기술한다.
- <116> 이 예에서, 이하의 규칙을 적용함으로써 JPEG 2000의 각 마커 세그먼트의 XML 요소를 생성하기 위해 선택이 행해진다.
- <117> · 요소는 마커의 3-문자(letter) 코드의 이름을 따서 명명된다,
- <118> · 마커의 길이, 그 마커에 관련되는 값 및 다른 파라미터들은 요소의 속성으로 규정된다,
- <119> · 파라미터 또는 파라미터 그룹이 반복되는 경우, XML 언어가 동일 개시 태그에 동일 명칭을 갖는 하나 이상의 속성들을 가지지 못하기 때문에, 서브-요소가 속성 대신에 생성된다(이것은, 예를 들면 이미지의 성분 각각에 대하여 반복되는 3개 파라미터(Ssiz, XRsiz, YRsiz)들의 그룹에 의해 종료되는 마커 세그먼트 SIZ를 갖는 경우이다),
- <120> · 비트 스트림의 패킷들은 요소의 콘텐츠 형태의 문서에 포함된다.
- <121> 그러나, 비트 스트림이 단지 2진 데이터로 구성되는 경우, XML 문서(임의 규정된 문자들만을 허용함)에 그것을 직접 포함할 수 없다.
- <122> 제 1 실시예에서, 비트 스트림은 문자들로 변환된다. 이를 위해, "베이스 64"로 명명되고 IETF에 의해 공개된 문서 RFC2045의 문단6.8에 기술되는 코딩 방법이 사용되는 것이 바람직하다. 이 방법은, 6비트 그룹으로 비트 스트림을 분할하는 단계와 문자 알파벳으로부터의 6 비트 문자의 각 그룹에 관련시키는 단계를 포함한다. 따라서, 이 방법은 33%의 데이터 확장을 초래하는 단점을 제공한다.

- <123> 제 2 실시예에서, XML 문서에 삽입되는 문자로 비트 스트림을 변환하는 대신에, 포인터는 XML 문서에 도입되며, 2진 데이터를 포함하는 기본 파일의 영역을 가리킨다. 이 실시예에서, XML 문서는 기본 파일에 의존하게 된다.
- <124> 부록 1(Annex 1)에서는 한정되지 않은 일례를 통해서 성분들을 통한 순차 도면의 도 5에 나타나는 JPEG2000 코드 스트림을 기술하는 XML 문서가 제공된다. 이 일례에서, 패킷의 콘텐츠는 단일 라인만을 점유하도록 절단된다.
- <125> 개선된 형태의 이미지를 기술하는 변환된 XML 문서를 생성하도록 XML 문서에 적용될 수 있는 변환에 대하여 2가지 예가 제공된다. 통상적으로, 변환은 기본 XML 문서의 임의 요소의 억제, 코드 스트림 세트의 코히런스를 유지하기 위한 임의 속성들 및/또는 임의 요소들의 값의 변경을 포함한다.
- <126> 변환에 관한 제 1 예는 컬러 이미지에서 그레이-레벨 이미지로 통과하도록 하는 성분 진행으로 코딩되는 이미지에 적용될 수 있는 변환이다. 이러한 변환은 크로미넌스 성분에 대응하는 비트 스트림의 패킷에 관한 억제를 포함한다. 이를 위해, 이 변환은 이하의 단계들을 포함한다:
 - <127> · 크로미넌스 패킷에 대응하는 요소들 SOP의 억제,
 - <128> · 크로미넌스 성분에 관련되는 속성들(Ssiz, XRsiz, YRsiz)을 포함하는 요소의 억제,
 - <129> · 성분의 개수를 나타내는 요소 SIZ의 속성 Csiz의 변경(이전 값 3 ⇒ 새로운 값 1),
 - <130> · 마커 세그먼트 SIZ의 크기를 나타내는 요소 SIZ의 속성 MarkSegLen의 변경(이전 값 38+3.3=47 ⇒ 새로운 값 38+3.1=41).
- <131> 이러한 변환은 XSL-스타일 시트에 규정되는 것이 바람직하다. 부록 2에서는 한정되지 않은 일례를 통해서, 부록 1에 기술되는 XML 문서에 적용되는 XSL-스타일 시트가 제공된다.
- <132> 변환에 관한 제 2 예는 해상도에 의해 점진적으로 코딩되는 이미지에 적용될 수 있는 변환이며, 이것에 의해 대형 이미지에서 소형 이미지로 변경할 수 있게 된다. 이러한 변환은 획득되는 해상도의 개수를 규정하는 입력 파라미터 Nresout를 사용한다. 변환은 Nresout 보다 높은 해상도에 대응하는 비트 스트림의 패킷에 관한 억제를 포함한다. 이를 위해, 변환은 이하의 단계들을 포함한다:
 - <133> · 비트 스트림의 (Nre-Nresout).Ncomp.Nlayers 최종 패킷에 대응하는 요소 SOP의 억제, 여기서 Nres, Ncomp, Nlayers는 순차 기본 파일에 사용되는 해상도, 성분 및 레이어 각각의 개수이다.
 - <134> · 이미지의 크기에 관련되는 요소 SIZ의 속성에 관한 변경: 속성들(Xsiz, XTsiz, Ysiz, YTsiz)에 포함되는 값을 $2^{N(res-Nresout)}$ 로 나눔,
 - <135> · 분해 레벨의 개수에 관련되는 요소 COD의 속성에 관한 변경(nDecompLevel=Nresout),
 - <136> · 요소 QCD의 속성 Sqcd로 표시되는 수량 스타일의 함수로서, 최종 요소 SPqcd의 억제,
 - <137> · 속성 markSegLen을 요소 QCD에 따르게 하는 단계.
- <138> 상기에 기술되는 변환에 관한 2가지 일례는 한정적이지 않다. 본 발명은 다른 형태의 변환에도 적용될 수 있다. 예를 들면, 또한 레이어에 의해 점진적으로 코딩되는 이미지에 적용될 수 있는 변환이 규정될 수도 있으며, 다양한 신호 대 잡음 비를 얻을 수 있다.
- <139> 도 6은 본 발명에 따른 JPEG 2000의 2진 파일을 도시하고 있다. 도 6에 따르면, JPEG 2000 포맷의 2진 파일 FF는 본 발명의 의미에서 기본 순차 파일을 구성하는 코드 스트림 CS를 포함하는 메인 박스 BX1을 포함한다. 또한, DOC로 칭해지는 XML 문서를 포함하는 추가 박스 BX2를 포함한다. 문서 DOC는 코드 스트림 CS를 기술한다. 추가 박스 B2는 또한 문서 DOC에 적용되는 TR로 칭해지는 변환 XSL을 포함할 수도 있다.

ANNEX 1 page 1

```

<?xml version="1.0" ?>
<Codestream>
  <MainHeader>
    <SOC markSegLen="-1" markerCodeStr="ff4f">
      </SOC>
      <SIZ Csize="3" RsizeStr="JPEG 2000 - Part I" Xsize="0" XTOsize="0" XTSize="515"
        Ysize="515" YTOsize="0" YTSize="512" Ysize="512" markSegLen="47"
        markerCodeStr="ff51">
        <Comp_siz Ssize="7" XRsiz="1" YRsiz="1">
        </Comp_siz>
        <Comp_siz Ssize="7" XRsiz="1" YRsiz="1">
        </Comp_siz>
        <Comp_siz Ssize="7" XRsiz="1" YRsiz="1">
        </Comp_siz>
      </SIZ>
      <COD codeBlockHeightExp="4" codeBlockWidthExp="4" ephUsed="false"
        markSegLen="12" markerCodeStr="ff52" mct="1" nDecompLevel="5"
        numLayers="1" optByPass="false" optErTerm="false" optRegTerm="false"
        optResetMQ="false" optSegMarkers="false" optVertStrCausal="false"
        precinctPartitionIsUsed="false" progType="4" progTypeStr="Component"
        sopUsed="true" wavTrans="0" wavTransStr="9-7">
      </COD>
      <QCD Sgcd="2" markSegLen="35" markerCodeStr="ff5c" nGuardBits="2">
        <SPgcd spgcd_val="28440">
        </SPgcd>
        <SPgcd spgcd_val="28394">
        </SPgcd>
        <SPgcd spgcd_val="28394">
        </SPgcd>
        <SPgcd spgcd_val="28348">
        </SPgcd>
        <SPgcd spgcd_val="26368">
        </SPgcd>
        <SPgcd spgcd_val="26368">
        </SPgcd>
        <SPgcd spgcd_val="26338">
        </SPgcd>
        <SPgcd spgcd_val="24396">
        </SPgcd>
        <SPgcd spgcd_val="24396">
        </SPgcd>
        <SPgcd spgcd_val="24420">
        </SPgcd>
        <SPgcd spgcd_val="18435">
        </SPgcd>
        <SPgcd spgcd_val="18435">
        </SPgcd>
        <SPgcd spgcd_val="18501">
        </SPgcd>
        <SPgcd spgcd_val="20434">
        </SPgcd>
        <SPgcd spgcd_val="20434">
        </SPgcd>
        <SPgcd spgcd_val="20321">
        </SPgcd>
      </QCD>
    </MainHeader>
  </Codestream>

```

ANNEX 1 page 2

```

<TilePartHeader>
  <SOT Isot="0" Psot="32722" TNsot="1" TPspot="0" markSegLen="10"
    markerCodeStr="ff90">
  </SOT>
  <SOD markSegLen="-1" markerCodeStr="ff93">
  </SOD>
  <SOP Lsop="0" Nsop="0" markSegLen="4" markerCodeStr="ff91">
    zokxfjAav/OeNh0KcGQ=
  </SOP>
  <SOP Lsop="0" Nsop="1" markSegLen="4" markerCodeStr="ff91">
    vih3XvSgAxdN7qp7Aubt+hlASIAi7DNKpOQYs1JGcF+kax9m
  </SOP>
  <SOP Lsop="0" Nsop="2" markSegLen="4" markerCodeStr="ff91">
    lon5YDUL4HqwaIzDAg==
  </SOP>
  <SOP Lsop="0" Nsop="3" markSegLen="4" markerCodeStr="ff91">
    v5HtMkNwzWEIhKL7bEcpU5ZrpLIezrl8kZkNLcm1Cw==
  </SOP>
  <SOP Lsop="0" Nsop="4" markSegLen="4" markerCodeStr="ff91">
    o3a7iCv0m1V/xPkZz6Ks7HXbJYUyLNny18Jhj0OeYgvi
  </SOP>
  <SOP Lsop="0" Nsop="5" markSegLen="4" markerCodeStr="ff91">
    4opHeHv9L+qiu/UplNkyVlPIWoc8umfk14Z6eDIzKNGtEVBkCMaknWAcYXjmf
  </SOP>
  <SOP Lsop="0" Nsop="6" markSegLen="4" markerCodeStr="ff91">
    EnEF62yGwcrTY2ehQYEpUQ==
  </SOP>
  <SOP Lsop="0" Nsop="7" markSegLen="4" markerCodeStr="ff91">
    ldWatY7LLQ==
  </SOP>
  <SOP Lsop="0" Nsop="8" markSegLen="4" markerCodeStr="ff91">
    b3qu8E6vTqqK4IIItml9Cifij0SVY2qqIMswd7eSOQhyps5bIStPlwM0sB6i2b/j8JMo80vg=
  </SOP>
  <SOP Lsop="0" Nsop="9" markSegLen="4" markerCodeStr="ff91">
    83mg5+Jv95wRIRm61swVvYBhw7eAeb2BNrmBsC1XrTjvzGFL5rY1BCyAn8CS
  </SOP>
  <SOP Lsop="0" Nsop="10" markSegLen="4" markerCodeStr="ff91">
    7psE6fJW2/G3ohpvNaT+ovEFscTFj++9sdVip0XwnPn25oWq2yZYNh+DWr3T7H+V
  </SOP>
  <SOP Lsop="0" Nsop="11" markSegLen="4" markerCodeStr="ff91">
    XaILyXsuKvRI+C4p
  </SOP>
  <SOP Lsop="0" Nsop="12" markSegLen="4" markerCodeStr="ff91">
    W6qLBSuVSfJQRnGVfO27J97dmNDaUYjLttBepCvUXQu71TWIWBzRRR0mPgpPBg==
  </SOP>
  <SOP Lsop="0" Nsop="13" markSegLen="4" markerCodeStr="ff91">
    3s+ECWT/ZjkqNaHdcQ==
  </SOP>
  <SOP Lsop="0" Nsop="14" markSegLen="4" markerCodeStr="ff91">
    +eQE2Ek04vydGK+tx9kkpj2L6TeiR2nc4crNQ4zHDurQg==
  </SOP>
  <SOP Lsop="0" Nsop="15" markSegLen="4" markerCodeStr="ff91">
    lcQa4O/p/tH7FOjY0FjYefzcUtdrynh7RRVhBftsTu0F6Y5qXhD1D4G2uBd/a7
    uw49QaaPKtfJb8XiaVpXVqBW9LwoqQ==
  </SOP>
  <SOP Lsop="0" Nsop="16" markSegLen="4" markerCodeStr="ff91">
    ZjJalwvKx0SiIQUosc/Rkj04wqeX6qq3miqkFQ==
  </SOP>
  <SOP Lsop="0" Nsop="17" markSegLen="4" markerCodeStr="ff91">
    gA==
  </SOP>
</TilePartHeader>
<EOC markSegLen="-1" markerCodeStr="ffd9">
</EOC>
</Codestream>

```

ANNEX 2 page 1

```

<?xml version="1.0"?>
<!--
  Remove color components :

  For this, this stylesheet performs the following transformation:
  - check that SOP markers are used
  - check progression type
  - check that multiple component transform is used
  - check that nCompOut <= Csiz

  - replace the Csiz attribute value by the wanted value (nCompOut)
  - delete Comp_siz elements accordingly
  - update markSegLen value accordingly
  - delete SOP elements accordingly

  Input parameter : nCompOut, default value = 1
-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:import href="tplCheckSOPUsed.xsl"/>
  <xsl:output method="xml" indent="yes"/>

  <!-- Parameter: number of output components - Default value = 1 -->
  <xsl:param name="nCompOut">1</xsl:param>

  <!-- Match all: default template -->
  <xsl:template name="tplAll" match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>

  <!-- Match COD/@sopUsed - Overrides tplAll -->
  <xsl:template name="tplSopUsed" match="//COD/@sopUsed" priority="1">
    <!-- Check /COD/sopUsed value -->
    <xsl:call-template name="tplCheckSOPUsed"/>
  </xsl:template>

  <!-- Match COD - Overrides tplAll -->
  <xsl:template name="tplCOD" match="COD" priority="1">
    <!-- Check progType value -->
    <xsl:if test="@progType!=4">
      <xsl:message terminate="yes">
        Error: progression type is <xsl:value-of select="@progType"/>
        Should be 4 (progression by color component)
      </xsl:message>
    </xsl:if>

    <!-- Check mct value -->
    <xsl:if test="@mct=0 and //SIZ/@Csiz=3">
      <xsl:message>
        Warning: no multi-component transf. was applied to input image.
        If input was RGB, output will be the Red component.
      </xsl:message>
    </xsl:if>

    <!-- Copy COD -->
    <xsl:copy>
      <xsl:apply-templates select="@*"/>
      <!-- Set mct value to 0 -->
      <xsl:attribute name="mct">0</xsl:attribute>
    </xsl:copy>
  </xsl:template>

```

```
ANNEX 2 page 2
```

```

<!-- Match SIZ - Overrides tplAll -->
<xsl:template name="tplSIZ" match="SIZ" priority="1">

  <!-- Check nCompOut value : should be <= Csiz -->
  <xsl:if test="(($nCompOut > @Csiz) or ($nCompOut < 1))">
    <xsl:message terminate="yes">
      Error: number of output color components should be >@Csiz and <1. Exit...
    </xsl:message>
  </xsl:if>

  <xsl:copy>
    <!-- Copy attributes -->
    <xsl:apply-templates select="@*" />

    <!-- Update Csiz value -->
    <xsl:attribute name="Csiz">
      <xsl:value-of select="$nCompOut" />
    </xsl:attribute>

    <!-- Update markSegLen value accordingly -->
    <xsl:attribute name="markSegLen">
      <xsl:value-of select="38 + 3 * $nCompOut" />
    </xsl:attribute>

    <!-- Copy only relevant Comp_siz elements -->
    <xsl:apply-templates select="Comp_siz[position() &lt;= $nCompOut]" />
  </xsl:copy>
</xsl:template>

<!-- Match TilePartHeader - Overrides tplAll -->
<xsl:template name="tplTilePartHeader" match="TilePartHeader" priority="1">

  <!-- Calculate number of output packets -->
  <xsl:variable name="nSOPout">
    <xsl:value-of select="
      (//COD/@nDecompLevel + 1) * //COD/@numLayers * $nCompOut" />
  </xsl:variable>

  <xsl:copy>
    <!-- Copy attributes -->
    <xsl:apply-templates select="@*|SOP|SOD" />

    <!-- Remove SOP packets accordingly -->
    <xsl:apply-templates select="SOP[position() &lt;= $nSOPout]" />
    <xsl:message>
      Initial number of SOP : <xsl:value-of
select="count(SOP)" />
      Output number of SOP : <xsl:value-of
select="$nSOPout" />
    </xsl:message>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>

```

<143>

```
ANNEX 2 page 3
```

```

<?xml version="1.0"?>

<!--
  This stylesheet performs the following transformation:
  - check than "sopUsed" attribute of "COD" element is set to yes,
  otherwise, exit
-->

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

  <!-- Template tplCheckSOPUsed -->
  <xsl:template name="tplCheckSOPUsed">

    <xsl:variable name="sopUsed">
      <xsl:value-of select="//COD/@sopUsed" />
    </xsl:variable>

    <xsl:if test="$sopUsed!='true'">
      <xsl:message terminate="yes">
        Error: SOP markers are not used in this codetsream.
        Cannot process. Exit...
      </xsl:message>
    </xsl:if>
    <xsl:copy/>
  </xsl:template>

</xsl:stylesheet>

```

<144>

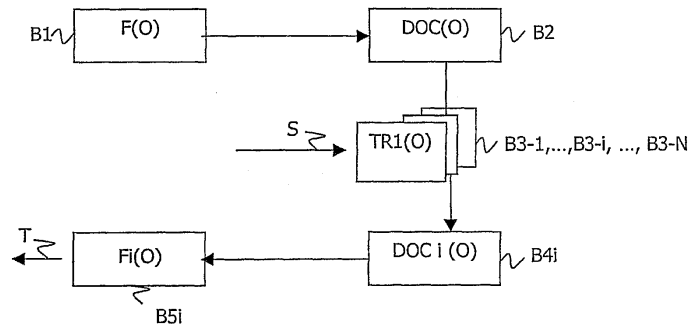
도면의 간단한 설명

- <50> 도 1은 본 발명에 따른 파일 전송 방법의 다양한 단계들을 설명하는 블럭도.
- <51> 도 2는 본 발명에 따른 전송 시스템의 일례를 도시하는 도면.

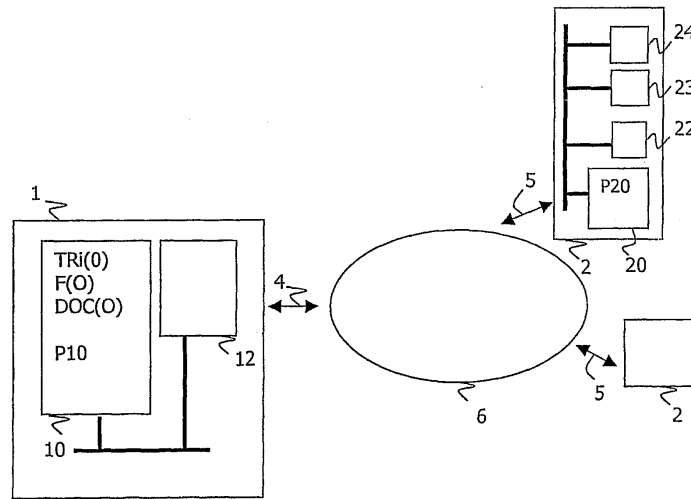
- <52> 도 3은 본 발명에 따른 전송 시스템의 서버의 소프트웨어 구조의 일례를 도시하는 도면.
- <53> 도 4는 도 3에 제공되는 클라이언트-서버 구조를 이용하는 본 발명에 따른 파일 전송 방법의 애플리케이션의 일례를 도시하는 도면.
- <54> 도 5는 JPEG 2000 코드 스트림의 개략도.
- <55> 도 6은 본 발명에 따른 2진 파일의 일례에 관한 개략도.

도면

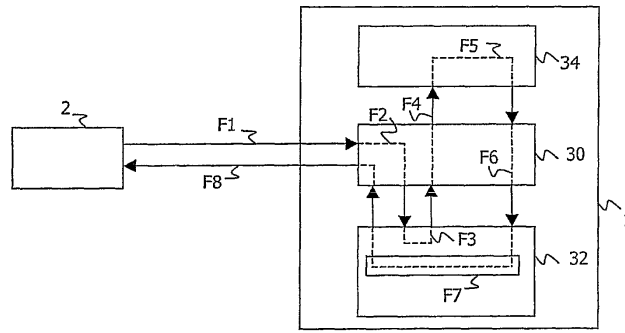
도면1



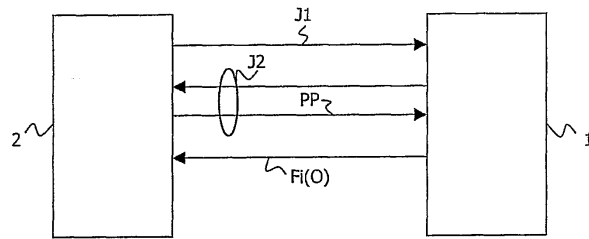
도면2



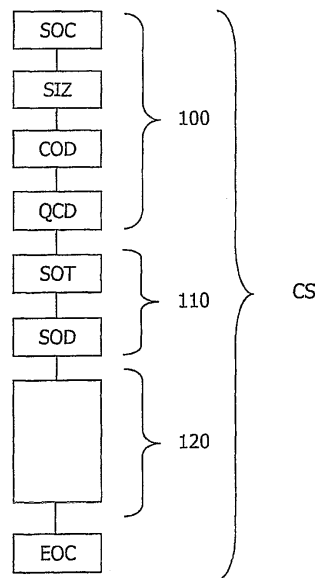
도면3



도면4



도면5



도면6

