



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년02월28일
(11) 등록번호 10-0807651
(24) 등록일자 2008년02월20일

- (51) Int. Cl.
H04L 29/06 (2006.01) G06F 15/177 (2006.01)
H04B 7/26 (2006.01)
- (21) 출원번호 10-2005-7012148
- (22) 출원일자 2005년06월27일
심사청구일자 2006년02월13일
번역문제출일자 2005년06월27일
- (65) 공개번호 10-2005-0091027
- (43) 공개일자 2005년09월14일
- (86) 국제출원번호 PCT/CA2003/001981
국제출원일자 2003년12월24일
- (87) 국제공개번호 WO 2004/059939
국제공개일자 2004년07월15일
- (30) 우선권주장
60/436,012 2002년12월26일 미국(US)
60/503,955 2003년09월17일 미국(US)
- (56) 선행기술조사문헌
인용발명(A Java Based XML Browser for Consumer Devices, 2002.03.11)

- (73) 특허권자
리서치 인 모션 리미티드
캐나다 온타리오 워털루 필립 스트리트 295 (우편번호 엔2엘 3더블유8)
- (72) 발명자
웬필드 마이클
캐나다 온타리오 엘4씨 3에스9 리치몬드 힐 38 스톡데일 크레센트
비브르 비에라
캐나다 온타리오 엘0피 1취0 킬브라이드 박스 828 6479 맥니븐로드
(뒷면에 계속)
- (74) 대리인
김태홍, 신정건

전체 청구항 수 : 총 44 항

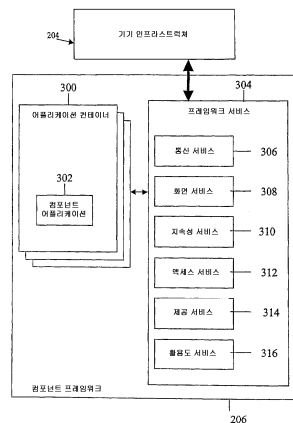
심사관 : 문성돈

(54) 컴포넌트 기반의 무선 애플리케이션을 생성하여 통신하는시스템 및 방법

(57) 요약

무선 컴포넌트 애플리케이션을 생성하는 통신하는 시스템 및 방법이 제공된다. 이동 통신 기기는 무선 네트워크, 인터넷 및 옵션의 메시지-맵 서비스와 통신한다. 각각의 무선 통신 애플리케이션은 이동 통신 기기 중의 한 기기 상의 컴포넌트 프레임워크에 의해 수행되며, 데이터 컴포넌트, 프리젠테이션 컴포넌트, 메시지 컴포넌트 및 워크플로우 컴포넌트를 포함한다. 컴포넌트 프레임워크는 통신 서비스, 화면 서비스, 지속성 서비스, 액세스 서비스, 제공 서비스 및 활용도 서비스를 포함하는 프레임워크 서비스에 대한 액세스를 제공하는 애플리케이션 컨테이너 내의 컴포넌트 애플리케이션을 실행한다. 메시지는 컴포넌트 애플리케이션에서 메시지-맵 서비스로 발송될 수 있으며, 메시지-맵 서비스는 메시지의 각각을 애플리케이션 메시지-맵을 이용하여 웹 서비스에 의해 요구된 포맷으로 변환하며, 그 후 메시지를 웹 서비스에 발송한다. 웹 서비스는 응답을 메시지-맵 서비스에 발송하며, 메시지-맵 서비스는 이 응답을 무선 컴포넌트 애플리케이션에 의해 요구된 포맷으로 변환하고, 응답을 무선 컴포넌트 애플리케이션에 발송한다.

대표도 - 도4



(72) 발명자

프리취 브린두사 엘

캐나다 온타리오 엠9씨 2제이9 토론토 9 체리 포스트 크레센트

고링 브라이언 알

캐나다 온타리오 엠9티 5브이4 밀톤 1542 모스 플레이스

클라인 로버트

캐나다 온타리오 엠4비 2브이9 리치몬드힐 68 체드윅 크레센트

비타노브 카멘 비

캐나다 온타리오 엠6피 2피4 토론토 아파트 413 66 퍼시픽 애비뉴

특허청구의 범위

청구항 1

이동 통신 기기에 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전을 프로비저닝(provisioning)하는 방법에 있어서,

상기 컴포넌트 애플리케이션 프로그램을 나타내는 복수의 컴포넌트들을 네트워크를 통해 서버로부터 수신하는 단계로서, 상기 컴포넌트들은 상기 컴포넌트 애플리케이션 프로그램의 구성 정보를 정의하기 위한 구조적 정의 언어로 표현된 메타데이터 디스크립터들 및 상기 컴포넌트 애플리케이션 프로그램의 워크플로우를 정의하기 위한 메타데이터 디스크립터들에 연결된 스크립팅된(scripted) 명령들을 포함하는 것인, 상기 수신 단계;

상기 메타데이터 디스크립터들 및 상기 스크립팅된 명령들을, 상기 프로그램에 대한 클라이언트 런타임 환경을 호스팅하는 이동 통신 기기의 애플리케이션 컨테이너에 로드하는 단계와;

상기 이동 통신 기기의 클라이언트 런타임 환경에서 후속 실행을 위해 상기 메타데이터 디스크립터들 및 상기 스크립팅된 명령들로부터 실행 가능한 버전을, 프로비저닝된(provisioned) 컴포넌트 애플리케이션 프로그램으로서 생성하는 단계

를 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 2

제1항에 있어서, 상기 애플리케이션 컨테이너는 컴포넌트 프레임워크에 의해 제공되며, 컴포넌트 프레임워크는 프로세서 및 메모리를 포함하는 상기 이동 통신 기기의 인프라스트럭처에 대하여 실행 가능한 버전을 위한 인터페이스를 제공하는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 3

제2항에 있어서, 상기 애플리케이션 컨테이너는 상기 기기 인프라스트럭처의 운영체제(operating system)의 일부인 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 4

제2항에 있어서, 각각의 제2 클라이언트 런타임 환경에서 제2 컴포넌트 애플리케이션 프로그램을 호스팅하기 위해 상기 컴포넌트 프레임워크에 의해 제2 애플리케이션 컨테이너를 생성하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 5

제2항에 있어서, 상기 애플리케이션 컨테이너의 클라이언트 런타임 환경은 상기 실행 가능한 버전을 상기 서버의 클라이언트로서 나타내는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 6

제5항에 있어서, 상기 컴포넌트들의 메타데이터 디스크립터들로 파플레이팅(populating)하도록 구성된 다수의 사전 정의된 템플릿들을 채용하는 템플릿 기반 실행 모델에 의해 실행 가능한 버전을 원시 코드로 실행하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 7

제6항에 있어서, 상기 메타데이터 디스크립터들과 관련된 메타데이터 정의 파라미터들로 사전 정의된 템플릿들을 파플레이팅(populating)하고, 상기 실행 가능한 버전을 상기 원시 코드로 구축하기 위해 상기 파플레이팅(populating)된 템플릿들을 사용하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 8

제6항에 있어서, 상기 사전 정의된 템플릿들은 데이터 컴포넌트들, 메시지 컴포넌트들 및 프리젠테이션 컴포넌

트들을 포함하는 군에서 선택된 복수의 컴포넌트들에 제공되는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 9

제5항에 있어서, 실행 동안의 구문분석(parsing)을 위해 메타데이터 정의들을 구조적 정의 언어로 유지하도록 구성된 메타데이터 기반 실행 모델에 의해 상기 실행 가능한 버전을 실행하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 10

제5항에 있어서, 실행 동안 구조적 정의 언어 노드들의 원시 표현을 사용하도록 구성된 메타데이터 기반 실행 모델에 의해 상기 실행 가능한 버전을 실행하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 11

제5항에 있어서, 상기 애플리케이션 컨테이너에 의해 서버로부터 수신된 메시지 데이터를 분석하고, 상기 이동 통신 기기의 메모리 내의 메타데이터의 표현을 업데이트하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 12

제11항에 있어서, 상기 실행 가능한 버전과 상기 컴포넌트 프레임워크 사이의 통신을 상기 애플리케이션 컨테이너에 의해 모니터링하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 13

제12항에 있어서, 상기 애플리케이션 컨테이너는 상기 메시지 데이터의 메모리 할당, 입출력 동안의 상기 메시지 데이터의 관리 및 상기 이동 통신 기기의 사용자 인터페이스 상의 메시지 데이터 프리젠테이션을 포함하는 군에서 선택된 동작들을 포함하는 상기 메시지 데이터의 처리를 모니터링하는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 14

제8항에 있어서, 네트워크를 통해 상기 컴포넌트 애플리케이션 프로그램의 워크플로우 컴포넌트를 수신하는 단계를 더 포함하며, 상기 워크플로우 컴포넌트는 구조적 정의 언어로 표현된 상기 메타데이터 디스크립터들을 갖는 컴포넌트들의 동작을 조정(coordinating)하기 위한 명령 세트를 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 15

제14항에 있어서, 애플리케이션 컨테이너가 상기 명령 세트에 상기 클라이언트 런타임 환경을 제공하는 단계를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 16

제15항에 있어서, 상기 실행 가능한 버전은 원시 코드로 변환된 명령 및 원시 코드 리다이렉터(redirector)를 갖는 랭귀지 인터프리터를 이용하여 해석된 명령들을 포함하는 군에서 선택된 실행 가능한 명령 세트를 포함하는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 17

제15항에 있어서, 상기 구조적 정의 언어는 XML로 표현되며, 상기 명령 세트는 ECMA 스크립트로 표현되는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 18

제4항에 있어서, 상기 실행 가능한 버전을 상기 이동 통신 기기의 프레임워크 서비스들의 세트에 연결하는 단계

를 더 포함하는 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 19

제18항에 있어서, 상기 프레임워크 서비스들은 상기 컴포넌트 프레임워크에 의해 제공된 프레임워크 서비스들의 표준 세트로서 사전 정의되는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 20

제19항에 있어서, 상기 사전 정의된 프레임워크 서비스는 통신, 프리젠테이션, 지속성(persistence), 액세스, 제공(provisioning) 및 유틸리티(utility)를 포함하는 군에서 선택되는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 21

제20항에 있어서, 상기 프레임워크 서비스들은 상기 복수의 컴포넌트 애플리케이션 프로그램들로부터의 상기 컴포넌트들을 공유하도록 구성되는 것인 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전 프로비저닝(provisioning) 방법.

청구항 22

네트워크를 통해 서버로부터 수신된 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전을 프로비저닝(provisioning)하는 이동 통신 기기로서, 상기 수신된 컴포넌트 애플리케이션 프로그램은 상기 컴포넌트 애플리케이션 프로그램을 나타내는 복수의 컴포넌트를 구비하고, 상기 컴포넌트들은 상기 컴포넌트 애플리케이션 프로그램의 구성 정보를 정의하기 위해 구조적 정의 언어로 표현된 메타데이터 디스크립터들 및 상기 컴포넌트 애플리케이션 프로그램의 워크플로우를 정의하는 메타데이터 디스크립터들에 연결된 스크립팅된(scripted) 명령들을 포함하는 것인, 이동 통신 기기에 있어서,

상기 실행 가능한 버전을 실행하기 위해 프로세서 및 관련 메모리를 포함하는 상기 이동 기기를 동작시키는 기기 인프라스트럭처와;

상기 기기 인프라스트럭처에 접속되고, 상기 실행 가능한 버전과 통신하도록 구성된 입력 장치 및 출력 장치를 갖는 사용자 인터페이스와;

기기 인프라스트럭처에 접속되고, 상기 네트워크와 통신하기 위해 무선 송수신기에 접속하도록 구성된 통신 기기와;

상기 메타데이터 디스크립터들 및 상기 스크립팅된 명령들로부터 실행가능한 버전을 프로비저닝된(provisioned) 컴포넌트 애플리케이션 프로그램으로서 생성하고, 상기 실행가능한 버전을 상기 기기 인프라스트럭처에 인터페이스하도록 구성된 컴포넌트 프레임워크

를 포함하는 이동 통신 기기.

청구항 23

제22항에 있어서, 상기 컴포넌트 프레임워크에 의해 제공되고, 상기 실행 가능한 버전에 상기 클라이언트 런타임 환경을 제공하는 애플리케이션 컨테이너를 더 포함하는 이동 통신 기기.

청구항 24

제23항에 있어서, 상기 애플리케이션 컨테이너는 상기 기기 인프라스트럭처의 운영체제의 일부인 것인 이동 통신 기기.

청구항 25

제23항에 있어서, 상기 컴포넌트 프레임워크는 각각의 클라이언트 런타임 환경에서 상기 컴포넌트 애플리케이션 프로그램들 각각을 호스팅하기 위해 복수의 애플리케이션 컨테이너에 대하여 구성되는 것인 이동 통신 기기.

청구항 26

제23항에 있어서, 상기 애플리케이션 컨테이너의 클라이언트 런타임 환경은 상기 실행 가능한 버전을 상기 서버의 클라이언트로 나타내는 것인 이동 통신 기기.

청구항 27

제26항에 있어서, 상기 컴포넌트들의 메타데이터 디스크립터들로 파플레이팅(populating)하도록 구성된 다수의 사전 정의된 템플릿들을 채용하는 템플릿 기반 실행 모델에 의해, 상기 실행 가능한 버전을 원시 코드로 실행하도록 구성된 애플리케이션 컨테이너를 더 포함하는 이동 통신 기기.

청구항 28

제27항에 있어서, 상기 사전 정의된 템플릿들은 상기 메타데이터 디스크립터들과 관련된 메타데이터 정의 파라미터들로 파플레이팅(populating)하도록 구성되며, 상기 파플레이팅(populating)된 템플릿들은 상기 실행 가능한 버전을 상기 원시 코드로 구축하도록 구성되는 것인 이동 통신 기기.

청구항 29

제27항에 있어서, 상기 사전 정의된 템플릿들은 데이터 컴포넌트들, 메시지 컴포넌트들 및 프리젠테이션 컴포넌트들을 포함하는 군에서 선택된 복수의 컴포넌트들에 제공되는 것인 이동 통신 기기.

청구항 30

제23항에 있어서, 실행 동안의 구문분석(parsing)을 위해 상기 메타데이터 정의들을 구조적 정의 언어로 유지하는 메타데이터 기반 실행 모델에 의해 상기 실행 가능한 버전을 실행하도록 구성된 애플리케이션 컨테이너를 더 포함하는 이동 통신 기기.

청구항 31

제23항에 있어서, 실행 동안 구조적 정의 언어 노드들의 원시 표현을 사용하는 메타데이터 기반 실행 모델에 의해 상기 실행 가능한 버전을 실행하도록 구성된 애플리케이션 컨테이너를 더 포함하는 이동 통신 기기.

청구항 32

제23항에 있어서, 상기 애플리케이션 컨테이너는 상기 서버로부터 수신된 메시지 데이터를 분석하고, 상기 이동 통신 기기의 메모리 내의 메타데이터의 표현을 업데이트하는 것인 이동 통신 기기.

청구항 33

제32항에 있어서, 상기 애플리케이션 컨테이너는 상기 실행 가능한 버전과 상기 컴포넌트 프레임워크 사이의 통신을 모니터링하는 것인 이동 통신 기기.

청구항 34

제33항에 있어서, 상기 애플리케이션 컨테이너는 상기 메시지 데이터의 메모리 할당, 입출력 동안의 상기 메시지 데이터의 관리 및 상기 이동 통신 기기의 사용자 인터페이스 상의 메시지 데이터 프리젠테이션을 포함하는 군에서 선택된 동작들을 포함하는 상기 메시지 데이터의 처리를 모니터링하는 것인 이동 통신 기기.

청구항 35

제29항에 있어서, 구조적 정의 언어로 표현된 상기 메타데이터 디스크립터들을 갖는 컴포넌트들의 동작을 조정하기 위해 명령 세트를 포함하는 워크플로우 컴포넌트를 갖는 프로그램을 더 포함하는 이동 통신 기기.

청구항 36

제35항에 있어서, 상기 애플리케이션 컨테이너가 명령 세트에 각각의 클라이언트 런타임 환경을 제공하는 것인 이동 통신 기기.

청구항 37

제36항에 있어서, 상기 실행 가능한 버전은 원시 코드로 변환된 명령들 및 원시 코드 리다이렉터와 함께 랭귀지

인터프리터를 이용하여 해석된 명령들을 포함하는 군에서 선택된 실행 가능한 명령 세트를 포함하는 것인 이동 통신 기기.

청구항 38

제36항에 있어서, 상기 구조적 정의 언어는 XML로 표현되며, 명령 세트는 ECMA 스크립트로 표현되는 것인 이동 통신 기기.

청구항 39

제25항에 있어서, 상기 이동 통신 기기 상의 프레임워크 서비스들의 세트에 연결된 실행 가능한 버전을 더 포함하는 이동 통신 기기.

청구항 40

제39항에 있어서, 상기 프레임워크 서비스들은 상기 컴포넌트 프레임워크에 의해 제공된 프레임워크 서비스들의 표준 세트로서 사전 정의되는 것인 이동 통신 기기.

청구항 41

제40항에 있어서, 상기 사전 정의된 프레임워크 서비스들은 통신, 프리젠테이션, 지속성, 액세스, 프로비저닝 (provisioning) 및 유틸리티를 포함하는 군에서 선택되는 것인 이동 통신 기기.

청구항 42

제41항에 있어서, 상기 프레임워크 서비스들은 복수의 컴포넌트 애플리케이션 프로그램들로부터의 컴포넌트들을 공유하도록 구성되는 것인 이동 통신 기기.

청구항 43

네트워크를 통해 서버로부터 수신된 컴포넌트 애플리케이션 프로그램-상기 프로그램은 복수의 컴포넌트를 포함하고, 상기 컴포넌트들은 상기 컴포넌트 애플리케이션 프로그램의 구성 정보를 정의하기 위한 구조적 정의 언어로 표현된 메타데이터 디스크립터들 및 상기 컴포넌트 애플리케이션 프로그램의 워크플로우를 정의하기 위한 메타데이터 디스크립터들에 연결된 스크립팅된(scripted) 명령들을 구비함-의 실행 가능한 버전을 프로비저닝 (provisioning)하는 이동 통신 기기에 있어서,

상기 실행 가능한 버전을 실행하기 위한 인프라스트럭처 수단과;

상기 인프라스트럭처 수단에 연결되고, 이동 통신 기기의 사용자에게 의해 실행 가능한 버전과 통신하도록 구성된 인터페이스 수단과;

상기 인프라스트럭처 수단에 연결되고, 상기 네트워크와 통신하도록 구성된 통신 수단과;

상기 메타데이터 디스크립터들 및 상기 스크립팅된 명령들로부터 실행가능한 버전을, 프로비저닝된 (provisioned) 컴포넌트 애플리케이션 프로그램으로서 생성하고, 상기 실행 가능한 버전을 상기 인프라스트럭처 수단에 인터페이스하고 상기 실행 가능한 버전에 클라이언트 런타임 환경을 제공하는 프레임워크 수단을 포함하는 이동 통신 기기.

청구항 44

실행 가능한 버전의 컴포넌트 애플리케이션 프로그램-상기 프로그램은 복수의 컴포넌트를 포함하고, 상기 컴포넌트들은 상기 컴포넌트 애플리케이션 프로그램의 구성 정보를 정의하기 위한 구조적 정의 언어로 표현된 메타데이터 디스크립터들 및 상기 컴포넌트 애플리케이션 프로그램의 워크플로우를 정의하기 위한 메타데이터 디스크립터들에 연결된 스크립팅된(scripted) 명령들을 구비함-을 이동 통신 기기에서 프로비저닝 (provisioning) 하기 위한 프로그램이 기록된 컴퓨터 판독가능한 기록매체에 있어서,

컴퓨터 판독 가능한 매체에 저장되고, 상기 실행 가능한 버전을 프로세서 및 관련 메모리를 포함하는 기기 인프라스트럭처에 인터페이스하는 컴포넌트 프레임워크 모듈과;

상기 컴포넌트 프레임워크 모듈에 연결되며, 상기 메타데이터 디스크립터들 및 상기 스크립팅된 명령들로부터

실행가능한 버전을, 프로비저닝(provisioning)된 컴포넌트 애플리케이션 프로그램으로서 생성하고, 생성된 실행 가능한 버전에 대한 클라이언트 런타임 환경을 호스팅하는 애플리케이션 컨테이너 모듈을 포함하는 프로그램이 기록된 컴퓨터 판독가능한 기록매체.

명세서

기술분야

<1> 본 출원은 전반적으로 무선 통신에 관한 것으로, 구체적으로는 이동 통신 기기용 소프트웨어에 관한 것이다.

배경기술

<2> 휴대 전화, 무선 통신 기능을 구비한 PDA 및 양방향 페이지 등의 현재 사용되고 있는 이동 통신 기기의 수가 지속적으로 증가하고 있다. 이들 이동 통신 기기 상에서 실행되는 소프트웨어 애플리케이션은 이들 기기의 활용도를 증가시킨다. 예를 들어, 도시 지역에 대한 기후를 검색하는 애플리케이션이 휴대 전화에 포함되거나, 사용자가 보석을 구입할 수 있도록 해주는 애플리케이션이 PDA에 포함될 수도 있을 것이다. 이들 소프트웨어 애플리케이션은 이들 기기의 이동성 및 무선 네트워크에의 접속성을 이용하여 사용자가 어디에 있는지에 관계없이 사용자에게 적시에 그리고 유용한 서비스를 제공한다. 그러나, 메모리 등의 이동 통신 기기의 제한된 자원으로 인해, 그리고 데이터를 이동 통신 기기에 무선으로 전달하는 복잡성으로 인해, 이동 통신 기기를 위한 애플리케이션을 개발하는 것은 용이하지 않으면서 시간이 많이 소요되는 작업으로 남아 있다.

<3> 현재, 이동 통신 기기는 인터넷 기반 브라우저 및/또는 원시(native) 애플리케이션을 통해 웹 서비스와 통신하도록 구성된다. 브라우저는 여러 상이한 기기에 대해 크로스-플랫폼(cross-platform)을 기반으로 하여 동작하도록 적합화될 수 있는 장점을 갖지만, 웹 서비스로부터 페이지(HTML에서의 화면 정의)를 요청하는 것이 화면에 포함된 데이터의 지속성을 방해한다는 단점을 갖는다. 브라우저의 또 다른 단점은 화면이 런타임으로 제공되고 이에 의해 자원이 순간적으로 집중될 수 있다는 점이다. 원시 애플리케이션은 이동 기기의 유형에 대해 특별하게 개발될 수 있어서 각각의 런타임 환경에 대해 상대적으로 최적화된 애플리케이션 프로그램을 제공할 수 있다는 장점을 갖는다. 그러나, 원시 애플리케이션은 플랫폼에 독립적이지 않아서 복수 버전의 동일한 애플리케이션의 개발을 필요로 할 뿐만 아니라 비교적 크기가 커져 그 만큼 이동 기기의 메모리 자원을 차지하는 단점을 갖는다. 따라서, 매우 다양한 런타임 환경을 갖는 웹 서비스 클라이언트 상에서 실행될 수 있을 뿐만 아니라 이동 기기 자원의 소비를 감소시킬 수 있는 애플리케이션 프로그램이 요망된다.

<4> 본 명세서에 개시된 시스템 및 방법은 전술된 단점의 적어도 일부를 제거 또는 경감시키기 위해 컴포넌트 기반 애플리케이션 환경을 제공한다.

발명의 상세한 설명

<5> 현재의 애플리케이션 프로그램은 매우 다양한 런타임 환경을 갖는 클라이언트 상에서 실행되도록 적합화될 수 없으며, 이동 기기 자원을 지나치게 과도하게 소비하여 바람직하지 않을 수 있다. 브라우저는 웹 서비스로부터 페이지(HTML에서의 화면 정의)를 요청하여 이에 의해 화면에 포함된 데이터의 지속성이 방해되는 단점을 갖는 애플리케이션 프로그램이다. 브라우저의 또 다른 단점은 화면이 런타임으로 제공된다는 것이고, 이것은 자원 집중적일 수 있다. 원시 애플리케이션은 플랫폼에 독립적이지 않아서 복수 버전의 동일한 애플리케이션의 개발을 필요로 할뿐만 아니라 비교적 크기가 커져 그 만큼 이동 기기의 메모리 자원을 차지한다는 단점을 갖는 현재의 애플리케이션 프로그램의 또 다른 예이다. 현재의 애플리케이션 프로그램과 대비하여, 무선 컴포넌트 애플리케이션을 생성하여 통신하는 시스템이 제공된다. 본 시스템은 무선 네트워크, 인터넷 및/또는 옵션으로 메시지-맵 서비스를 통한 웹 서비스와 통신하는 이동 통신 기기를 포함한다. 각각의 무선 컴포넌트 애플리케이션은 구조적 정의 언어(structured definition language)로 표현된 일련의 컴포넌트들과, 이동 통신 기기 중의 한 기기 상의 컴포넌트 프레임워크에 의해 실행되는 명령어 세트를 갖는다. 컴포넌트들은 하나 이상의 데이터 컴포넌트, 프리젠테이션 컴포넌트, 메시지 컴포넌트 및/또는 워크플로우 컴포넌트를 포함할 수 있다. 컴포넌트 프레임워크는 일반적인 프레임워크 서비스에 대한 액세스를 제공하는 애플리케이션 컨테이너 내의 컴포넌트 애플리케이션을 실행하고, 이 일반적인 프레임워크 서비스는 통신 서비스, 화면 서비스, 지속성 서비스, 액세스 서비스, 공급 서비스 및 유틸리티 서비스 중의 하나 이상을 포함한다. 메시지는 컴포넌트 애플리케이션으로부터 메시지-맵 서비스로 발송될 수 있으며, 이 메시지-맵 서비스는 메시지를 애플리케이션 메시지-맵을 이용하여 웹 서비스에 의해 요구된 포맷으로 변환하고, 그 후 이 메시지를 웹 서비스에 발송한다. 메시지에 대한 응답은 메시지-맵 서비스에 송신될 수 있고, 컴포넌트 애플리케이션에 의해 요구된 포맷으로 변환되며, 컴포넌트 애플리

케이션에 발송된다.

- <6> 또한, 무선 컴포넌트 애플리케이션을 생성하여 이 무선 컴포넌트 애플리케이션과 통신하는 시스템은 무선 컴포넌트 애플리케이션을 배치(deploy)하여 이 무선 컴포넌트 애플리케이션과 통신하는 방법을 제공한다. 상기 방법은 이동 통신 기기에 무선 컴포넌트 애플리케이션을 배치하는 단계와, 메시지-맵 서비스에 메시지-맵을 배치하는 단계를 포함한다. 본 방법은 일부 예에서 메시지를 메시지-맵 서비스에 발송하는 단계를 더 포함하며, 이 메시지-맵 서비스는 메시지를 메시지-맵을 이용하여 웹 서비스에 의해 요구된 포맷으로 변환하여 웹 서비스에 발송한다. 본 방법은 또한 응답을 웹 서비스로부터 메시지-맵 서비스로 발송하는 단계를 더 포함하며, 이 메시지-맵 서비스는 메시지를 무선 컴포넌트 애플리케이션에 의해 요구된 포맷으로 변환하여 이 메시지를 무선 컴포넌트 애플리케이션에 발송한다.
- <7> 본 명세서에는 네트워크를 통해 서버로부터 수신된 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전을 프로비저닝(provisioning) 하도록 구성된 이동 통신 기기가 개시되어 있다. 프로그램은 구조적 정의 언어로 표현된 메타데이터 디스크립터(metadata descriptor)를 포함하는 복수의 컴포넌트를 갖는다. 이동 통신 기기는 실행 가능한 버전을 실행하기 위한 인프라스트럭처 수단과, 인프라스트럭처 수단에 접속되고 기기의 사용자에게 의한 실행 가능한 버전과의 통신을 위해 구성된 인터페이스 수단과, 인프라스트럭처 수단에 접속되고 네트워크와 통신하기 위해 구성된 통신 수단과, 실행 가능한 버전을 인프라스트럭처 수단에 인터페이스하도록 구성되고 또한 실행 가능한 버전에 클라이언트 런타임 환경을 제공하도록 구성된 프레임워크 수단을 포함한다.
- <8> 또한, 이동 통신 기기에 관한 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전을 프로비저닝(provisioning) 하기 위한 컴퓨터 프로그램 제품이 개시되어 있다. 본 프로그램은 구조적 정의 언어로 표현된 메타데이터 디스크립터를 포함하는 복수의 컴포넌트를 갖는다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능한 매체와, 실행 가능한 버전을 프로세서 및 관련 메모리를 포함하는 기기 인프라스트럭처에 인터페이스하기 위해 상기 컴퓨터 판독 가능한 매체 상에 저장된 컴포넌트 프레임워크 모듈과, 컴포넌트 프레임워크 모듈에 접속되고 메타데이터 디스크립터로부터 실행 가능한 버전을 생성하고 그 결과의 실행 가능한 버전을 위한 클라이언트 런타임 환경을 호스팅(hosting)하는 애플리케이션 컨테이너 모듈을 포함한다.
- <9> 또한, 본 명세서에는 이동 통신 기기에 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전을 프로비저닝(provisioning) 하는 방법이 개시되어 있다. 상기 방법은 컴포넌트 애플리케이션 프로그램을 나타내고 구조적 정의 언어로 표현된 메타데이터 디스크립터를 포함하는 복수의 컴포넌트를 네트워크를 통해 서버로부터 수신하는 단계와, 메타데이터 디스크립터를, 프로그램을 위한 클라이언트 런타임 환경을 호스팅하는 기기의 애플리케이션 컨테이너에 로드하는 단계와, 기기의 클라이언트 런타임 환경에서 후속 실행을 위해 메타데이터 디스크립터로부터 실행 가능한 버전을 생성하는 단계를 포함한다.
- <10> 또한, 네트워크를 통해 서버로부터 수신된 컴포넌트 애플리케이션 프로그램의 실행 가능한 버전을 프로비저닝(provisioning) 하도록 구성된 이동 통신 기기가 개시된다. 프로그램은 구조적 정의 언어로 표현된 메타데이터 디스크립터를 포함하는 복수의 컴포넌트를 갖는다. 이동 기기는 실행 가능한 버전을 실행하기 위해 프로세서 및 관련 메모리를 포함하는 이동 기기를 작동시키는 기기 인프라스트럭처와, 상기 기기 인프라스트럭처에 접속되고 실행 가능한 버전과 통신하도록 구성된 입력 장치 및 출력 장치를 갖는 사용자 인터페이스와, 기기 인프라스트럭처에 접속되고, 네트워크와 통신하기 위해 무선 송수신기에 접속하도록 구성되는 통신 장치와, 실행 가능한 버전을 기기 인프라스트럭처에 인터페이스하도록 구성되고, 실행 가능한 버전에 클라이언트 런타임 환경을 제공하도록 구성된 컴포넌트 프레임워크를 포함한다.
- <11> 본 발명의 상기한 특징 및 기타 다른 특징은 첨부 도면을 참조하는 다음의 상세한 설명으로 더욱 명확해지게 될 것이다.

실시예

- <21> 도 1을 참조하면, 통신 시스템(10)은 접속된 무선 네트워크(102) 및 인터넷(104)을 경유하여 웹 서버(106)에 의해 제공된 하나 이상의 웹 서비스와 인터페이스하는 이동 통신 기기(100)를 포함한다. 이동 기기(100)는 웹 서버(106)의 웹 서비스와 통신하고 있을 때 각각 요청/응답 메시지(105)를 송신 및 수신한다. 이동 기기(100)는 메시지 헤더 정보 및 관련 데이터 콘텐츠의 형태를 갖는 요청/응답 메시지(105)를 이용하여 예를 들어 온라인 상점으로부터 제품 가격 및 용도를 요청 및 수신함으로써 웹 서비스의 웹 클라이언트로써 동작한다. 적합한 요청/응답 메시지(105)를 충족하기 위해, 웹 서버(106)는 관련 비즈니스 로직(메소드)을 이동 기기(100)의 클라이언트 애플리케이션 프로그램(302)(도 2 참조)에 노출시키기 위해 각종의 프로토콜(HTTP 및 컴포넌트 API 등, 그

러나 이러한 것으로만 제한되지는 않음)을 통해 애플리케이션 서버(110)와 통신한다. 이동 기기(100)의 애플리케이션 프로그램(302)은 객체(또는 함수)에 관해 메소드를 호출하는 것과 유사하게 애플리케이션 서버(110)의 비즈니스 로직을 사용할 수 있다. 이동 기기(100)는 무선 네트워크(102)를 경유하여 하나 이상의 웹 서버(106) 및 관련된 애플리케이션 서버(110)와 통신할 수 있는 것으로 알려져 있다. 또한, 이동 기기(100)는 애플리케이션 서버(110)에 직접 접속되어 요구시 웹 서버(106)를 바이패스할 수도 있는 것으로 알려져 있다.

<22> 도 2를 참조하면, 이동 통신 기기(100)는 이동 전화, PDA, 양방향 페이지 또는 듀얼-모드 통신 기기(도 9를 참조) 등의 기기이며, 이러한 기기뿐만 아니라 제한되지는 않는다. 이동 기기(100)는 접속부(218)를 경유하여 기기 인프라스트럭처(204)에 접속된 무선 송수신기(200)를 포함한다. 무선 송수신기(200)는 이동 기기(100)의 작동 동안 RF 또는 IR 링크 등의 적합한 무선 채널에 의해 무선 네트워크(102)에 접속 가능하며, 이 무선 채널은 이동 기기(100)가 무선 네트워크(102)를 경유하여 서로 그리고 외부 시스템(웹 서버(106) 등)과 통신하도록 하고, 클라이언트 애플리케이션 프로그램(302)과 서버(106, 110)(도 1 참조)간의 요청/응답 메시지(105)를 조정(coordinating)하도록 한다. 무선 네트워크(102)는 기기와, 무선 네트워크(102)에 접속되어 있는 외부 시스템간의 요청/응답 메시지(105)에서의 데이터의 송신을 지원한다. 무선 네트워크(102)는 이동 통신 기기(100)와, 무선 네트워크(102)에 대해 외부에 있는 기기간의 전화 통화를 위해 음성 통신을 제공할 수도 있을 것이다. 무선 데이터 통신 프로토콜은 DataTAC, GPRS 또는 CDMA 등의 무선 네트워크(102)에 의해 사용될 수 있으며, 이러한 프로토콜뿐만 아니라 제한되지는 않는다.

<23> 다시 도 2를 참조하면, 이동 기기(100)는 사용자(도시되지 않음)와 인터페이스하기 위해 접속부(222)에 의해 기기 인프라스트럭처(204)에 접속된 사용자 인터페이스(202)를 갖는다. 사용자 인터페이스(202)는 QWERTY 키보드, 키패드, 트랙휠, 스타일러스 등의 하나 이상의 사용자 입력 장치 및 LCD 화면 디스플레이 장치 등의 사용자 출력 장치를 포함한다. 화면이 터치를 감지할 수 있다면, 디스플레이는 기기 인프라스트럭처(204)에 의해 제어될 때 사용자 입력 장치로써 사용될 수도 있다. 사용자 인터페이스(202)는 이동 기기(100)의 사용자에 의해 채용되어 아래에 추가로 설명된 컴포넌트 프레임워크(206)를 통해 요청/응답 메시지(105)를 조정한다.

<24> 다시 도 2를 참조하면, 이동 통신 기기(100)의 동작은 기기 인프라스트럭처(204)에 의해 인에이블된다. 기기 인프라스트럭처(204)는 컴퓨터 프로세서(208) 및 관련 메모리 모듈(210)을 포함한다. 컴퓨터 프로세서(208)는 메모리 모듈(210)에 위치한 운영체제 및 클라이언트 애플리케이션 프로그램(302)에 의해 제공된 관련 명령을 실행함으로써 무선 통신 기기(100)의 무선 송수신기(200), 사용자 인터페이스(202) 및 컴포넌트 프레임워크(206)의 동작을 처리한다. 또한, 기기 인프라스트럭처(204)는 프로세서에 명령을 제공하기 위해 프로세서(208)에 접속되고 및/또는 메모리 모듈(210) 내의 클라이언트 애플리케이션 프로그램(302)을 로드/업데이트하기 위해 컴퓨터 판독 가능한 저장 매체(212)를 포함한다. 컴퓨터 판독 가능 매체(212)는 예를 들어, 자기 디스크, 자기 테이프, CD/DVD ROM 등의 광학적 판독 가능 매체 및 메모리 카드 등의 하드웨어 및/또는 소프트웨어를 포함한다. 각각의 경우에, 컴퓨터 판독 가능 매체(212)는 소형 디스크, 플로피 디스크, 카세트, 하드 디스크 드라이브, 고체 상태 메모리 카드 또는 메모리 모듈(210)에 제공된 RAM의 형태를 취할 수도 있을 것이다. 상기 나열된 예의 컴퓨터 판독 가능 매체(212)는 독립적으로 사용되거나 조합하여 사용될 수 있음을 주지하기 바란다.

<25> 다시 도 2를 참조하면, 이동 기기(100)의 컴포넌트 프레임워크(206)는 접속부(220)에 의해 기기 인프라스트럭처(204)에 접속된다. 컴포넌트 프레임워크(206)는 클라이언트 애플리케이션 프로그램(302)에 대한 윈시 런타임 환경을 제공하고, 프로세서(208)의 기능성 및 기기 인프라스트럭처(204)의 관련 운영체제를 위한 이동 기기(100)에 대한 인터페이스이다. 컴포넌트 프레임워크(206)는 컴포넌트 애플리케이션 프로그램(302)이 애플리케이션 컨테이너(300)에서 실행하는 이동 기기(100)에 관한 런타임 환경을 제공하며, 바람직하게는 제어되고 보안되며 안정된 환경을 제공한다. 애플리케이션 컨테이너(300)는 클라이언트 애플리케이션 프로그램(302)에 대한 호스트 컨테이너로써 지칭될 수 있으며, 메시지 메타-데이터(예를 들어, 메시지(105)의 메타-데이터, 도 1 참조)를 분석하고, 메모리 모듈(210) 내의 메타-데이터의 표현을 업데이트하는 것을 담당하게 될 것이다.

<26> 컴포넌트 프레임워크(206)는 또한 클라이언트 애플리케이션 프로그램(302)에 대한 프레임워크 서비스(304)(통신, 화면, 데이터 지속성, 보안 등의 표준 세트의 일반적인 서비스, 그러나 이러한 것으로만 제한되지는 않음)를 제공한다. 애플리케이션 프로그램(302)은 프레임워크 서비스(304)와의 통신(216)을 조정하는 애플리케이션 컨테이너(300)와의 통신(214)을 갖는다. 컴포넌트 프레임워크(206)의 프레임워크 서비스(304)는 접속부(220)를 경유하여 기기 인프라스트럭처(204)와의 접속을 통해 통신을 조정한다. 따라서, 기기 인프라스트럭처(204), 사용자 인터페이스(202) 및 무선 송수신기(200)에 대한 액세스는 컴포넌트 프레임워크(206)에 의해 클라이언트 애플리케이션 프로그램(302)에 제공된다. 또한, 클라이언트 애플리케이션 프로그램(302)은 애플리케이션 컨테이너(300)가 클라이언트 애플리케이션 프로그램(302)에 의 그리고 클라이언트 애플리케이션 프로그램

램(302)으로부터의 컴포넌트 프레임워크(206)의 통신(214, 216)의 모든 액세스를 제어하고 검증할 수 있기 때문에 적합하게 바이러스에 대한 저항성을 나타낼 수 있다. 기기 인프라스트럭처(204)(도 2 참조)의 운영체제의 일부분이 애플리케이션 컨테이너(300)를 나타낼 수 있는 것으로 알려져 있다.

- <27> 다시 도 2를 참조하면, 컴포넌트 프레임워크(206)의 클라이언트 런타임 환경은 메타-데이터 정의로부터 클라이언트 애플리케이션 프로그램(302)(컴포넌트 애플리케이션의 형태로 있는 - 아래 참조)을 생성, 호스팅 및 실행할 수 있는 것이 바람직하다. 또한, 컴포넌트 프레임워크(206)의 특정 기능은 랭귀지에 대한 지원, 메모리 할당의 조정, 네트워킹, I/O 동작 동안의 데이터의 관리, 기기 인프라스트럭처(204)를 통한 사용자 인터페이스(202)의 출력 장치 상의 그래픽 조정, 핵심적(core) 객체 지향 클래스에 대한 액세스 제공 및 파일/라이브러리 지원을 포함한다. 컴포넌트 프레임워크(206)에 의해 구현되는 런타임 환경의 예는 Microsoft에 의한 Common Language Runtime(CLR) 및 Sun Microsystems에 의한 Java Runtime Environment(JRE) 등을 포함할 수 있으며, 이러한 것으로만 제한되지는 않는다. 컴포넌트 프레임워크(206)의 런타임 환경은 이동 기기(100) 상의 클라이언트 애플리케이션 프로그램(302)을 위해 다음의 기능과 같은 기본 기능을 지원하며, 이것들에 제한되는 것은 아니다:
- <28> 메시지(105)를 웹 서버(106)(도 1 참조)의 웹 서비스에 발송하기 위한 통신 능력을 제공;
- <29> 웹 서버(106)(도 1 참조)의 웹 서비스의 인출 메시지(105)(서비스에의 메시지)를 위한 데이터 파트를 지원하기 위해 입력 장치를 통한 사용자에게 의한 데이터 입력 능력을 제공;
- <30> 출력 장치를 통한 웹 서버(106)(도 1 참조)의 웹 서비스의 응답(인입 메시지) 또는 비상관(uncorrelated) 통보를 위한 데이터 표현 또는 출력 능력을 제공;
- <31> 메모리 모듈(210) 내의 로컬 클라이언트 데이터를 유지하기 위해 데이터 저장 서비스를 제공; 및
- <32> 클라이언트 애플리케이션 프로그램(302)의 애플리케이션 컴포넌트(408)(도 3 참조)의 동작을 조정하기 위해 스크립트 언어를 위한 실행 환경을 제공.
- <33> 따라서, 도 2를 참조하면, 컴포넌트 프레임워크(206)는 런타임 환경 내에서 클라이언트 애플리케이션 프로그램(302)(웹 서비스 클라이언트 애플리케이션)을 실행하기 위해 사용되고, 요청/응답 메시지(105)를 통해 웹 서비스(106) 및 관련 애플리케이션 서버(110)(도 1 참조)의 웹 서비스 동작에 대한 액세스를 지원한다. 컴포넌트 애플리케이션 프로그램(302)은 컴포넌트 프레임워크(206)에 의해 수행되는 소프트웨어 애플리케이션을 포함한다. 컴포넌트 프레임워크(206)는 컴포넌트 애플리케이션 프로그램(302)이 실행될 때마다 애플리케이션 프로그램(302)의 각각의 컴포넌트(예를 들어, 400, 402, 404, 406 - 도 3 참조)에 대한 애플리케이션 컨테이너(300)를 생성한다. 애플리케이션 컨테이너(300)는 애플리케이션 프로그램(302)의 컴포넌트(예를 들어, 400, 402, 404, 406)를 로드하며, 기기 인프라스트럭처(204)에서 프로세서(208)에 의해 수행되는 원시 코드를 생성할 수 있다. 컴포넌트 프레임워크(206)는 따라서 도면 부호 "400", "402", "404", "406" 등의 컴포넌트의 정의를 제공하기 위한 애플리케이션 컨테이너(300)를 제공하여 통신 기기(100)의 각각의 기기 인프라스트럭처(204)에 고유한 실제적인 웹 클라이언트를 생성한다.
- <34> 도 3을 참조하면, 컴포넌트 애플리케이션 프로그램(302)의 블록도는 데이터 컴포넌트(400), 프리젠테이션 컴포넌트(402) 및/또는 메시지 컴포넌트(404)를 포함할 수 있으며, 이들은 애플리케이션 컨테이너(300)와의 통신(214)을 통해 워크플로우 컴포넌트(406)에 의해 조정된다.
- <35> 데이터 컴포넌트(400)는 컴포넌트 애플리케이션 프로그램(302)에 의해 사용되는 데이터 엔티티를 정의한다. 데이터 컴포넌트(400)가 기술할 수도 있는 데이터 엔티티의 예로는 오더(order), 사용자 및 재무 업무(financial transaction)이 있다. 데이터 컴포넌트(400)는 데이터 엔티티를 기술하기 위해 어떠한 정보가 요구되는지와, 정보가 어떠한 포맷으로 표현되는지를 정의한다. 예를 들어, 데이터 컴포넌트(400)는 숫자 형식으로 된 오더에 대한 고유 식별자로 이루어진 오더, 스트링(string) 형식으로 된 아이템의 리스트, 일자-시간 형식을 갖는 오더가 작성되는 시간, 스트링 형식으로 된 오더의 상황, 및 데이터 컴포넌트(400)의 또 다른 것의 정의에 따른 형식으로 된 오더를 발생시킨 사용자 등을 정의할 수도 있을 것이다.
- <36> 메시지 컴포넌트(404)는 웹 서비스(106) 등의 외부 시스템과 통신하기 위해 컴포넌트 애플리케이션 프로그램(302)에 의해 사용된 메시지의 포맷을 정의한다. 예를 들어, 메시지 컴포넌트(404) 중의 하나는 오더에 대한 고유 식별자, 오더의 상황 및 오더와 관련된 노트를 포함하는 오더를 발생시키기 위한 메시지 등을 기술할 수도 있을 것이다.

- <37> 프리젠테이션 컴포넌트(402)는 사용자 인터페이스(202)에 의해 디스플레이되는 것과 같이 컴포넌트 애플리케이션 프로그램(302)의 외형 및 행동을 정의한다. 프리젠테이션 컴포넌트(402)는 GUI 화면 및 제어와, 사용자가 사용자 인터페이스(202)를 이용하여 컴포넌트 애플리케이션(302)과 상호작용할 때에 실행될 동작을 정의한다. 예를 들어, 프리젠테이션 컴포넌트(402)는 화면, 라벨, 편집 박스, 버튼 및 메뉴와, 사용자가 편집 박스에 타이핑하거나 버튼을 누를 때 취해질 동작을 정의할 것이다.
- <38> 컴포넌트 애플리케이션 프로그램(302)의 워크플로우 컴포넌트(406)는 전술된 바와 같이 프리젠테이션 컴포넌트(402)에 의해 특정된 동작과 같은 동작이 수행되려고 할 때에 발생하는 처리를 정의하거나, 또는 시스템(10)으로부터 메시지(105)(도 1을 참조)가 도달할 때에 수행될 동작을 정의한다. 화면 워크플로우 및 메시지 처리는 워크플로우 컴포넌트(406)에 의해 정의된다. 워크플로우 컴포넌트(406)는 ECMAScript 등의 프로그래밍 언어 또는 스크립팅 언어로 작성되며, 원시 코드로 컴파일되어 전술된 바와 같이 애플리케이션 컨테이너(300)에 의해 실행될 수 있다. 워크플로우 컴포넌트(406)의 예는 데이터에 값을 할당하거나, 화면을 조작하거나 또는 메시지를 발송하는 것이 될 것이다.
- <39> 전술된 클라이언트 컴포넌트 애플리케이션 프로그램(302) 정의 호스팅 모델에서는 프리젠테이션 컴포넌트(402)(도 3을 참조)는 이동 기기(100)(도 1을 참조)의 클라이언트 플랫폼 및 환경에 좌우되어 변할 수도 있을 것이다. 예를 들어, 일부의 경우에 웹 서비스 소비자는 시각적 프리젠테이션을 요구하지 않는다. 컴포넌트 애플리케이션 프로그램(302)의 컴포넌트(400, 402, 404, 406)의 애플리케이션 정의는 다양한 사전 정의된 클라이언트 런타임(즉, 특정 컴포넌트 프레임워크(206))에 대한 플랫폼-특정(platform-specific) 프리젠테이션 컴포넌트(402)의 세트와 함께 플랫폼-중립(neutral) 데이터(400), 메시지(404), 워크플로우(406)의 번들로서 웹 서비스 레지스트리에 호스팅될 수 있다. 발견 또는 배치 요청 메시지(105)(도 1 참조)가 발행될 때, 클라이언트 타입은 이 요청 메시지(105)의 일부로써 특정되어야 한다. 통신 기기(100)의 상이한 클라이언트 플랫폼을 위해 컴포넌트 애플리케이션 프로그램(302)을 패키징하는 동안 데이터, 메시지 및 워크플로우 메타데이터를 복사하지 않기 위해, 애플리케이션 정의는 프리젠테이션 컴포넌트(402)의 상이한 세트와 함께 링크된 플랫폼-중립 컴포넌트 정의의 번들로서 애플리케이션 서버(110)(예를 들어) 상에 호스팅될 수 있다. 사용자가 발견 또는 다운로드 요청 메시지(105)를 만들 때, 통신 기기(100)의 클라이언트 런타임 타입이 검증되고, 적합한 번들이 무선 네트워크(102)를 통해 기기(100)에의 웹 서버(106)에 의한 전달을 위해 구성된다. 어떠한 경우, 워크플로우 컴포넌트(106)는 플랫폼-특정 표현으로 관리 및 채용될 수 있으며, 이러한 방식으로, 특정 플랫폼의 알려진 처리 능력이 강화되어 특정의 플랫폼에 대해 더욱 효율적이거나 및/또는 향상된 컴포넌트 애플리케이션 프로그램(302)을 제공한다.
- <40> 따라서, 도 3을 참조하면, 클라이언트 애플리케이션 프로그램(302)은 플랫폼-중립 컴포넌트 정의의 세트, 즉 데이터(400) 및 메시지(404) 컴포넌트와, XML(또는 다른 적합한 구조적 정의 언어)을 이용하는 프리젠테이션 컴포넌트(402)로써 정의될 수 있다. 워크플로우 컴포넌트(406)는 ECMAScript(또는 다른 적합한 플랫폼-중립 스크립팅 언어)를 이용하여 정의될 수 있다. 컴포넌트 프레임워크(206)(도 2를 참조)의 클라이언트 런타임 환경은 컴포넌트 애플리케이션 프로그램(302)의 컴포넌트(400, 402, 404, 406)가 통신 기기(100) 상에 제공될 때에 추가로 하술되는 바와 같이 메타-정의에 기초한 컴포넌트 템플릿(template)을 생성할 수 있다. 매우 다양한 런타임 환경과 함께, XML 또는 ECMAScript 등의 크로스-플랫폼 표준이 컴포넌트 애플리케이션 프로그램(302)을 사전 구축하는 대신에 애플리케이션 컴포넌트 메타데이터를 정의하기 위해 사용된다. 이 지연된 묶음은 컴포넌트 애플리케이션 프로그램(302)의 일반적인 애플리케이션 정의가 다양한 상이한 통신 기기(100)에 의해 표현된 매우 다양한 시스템 환경 상에서 실행되도록 할 수 있다.
- <41> XML 또는 그 변형체를 이용하여 데이터(4), 메시지(404) 및 프리젠테이션(402) 컴포넌트를 표현하고 ECMAScript 언어 또는 그 서브세트를 이용하여 워크플로우 컴포넌트(406)를 표현함으로써 애플리케이션 개발자가 어떠한 특정 플랫폼 또는 환경으로부터 웹 서비스 클라이언트를 요약할 수 있게 되고 "한번 개발하면 어느 곳에서도 실행가능(develop once run everywhere)" 애플리케이션이라는 원칙으로 실시할 수 있게 된다. 다음의 예는 XML 등의 구조적 정의 언어 및 컴포넌트를 정의한 ECMAScript 등의 플랫폼 중립 스크립팅/프로그래밍 언어를 이용하여 웹 서비스 클라이언트 애플리케이션 프로그램(302)이 표현되는 방법을 나타낸다:

<42> 일레의 XML 데이터 컴포넌트(400)

```
<data name="Order">
    <item name="orderId" type="Number" key="true"/>
    <item name="items" type="String" array="true"/>
    <item name="user" comp="true" compName="User"/>
    <item name="orderStatus" type="String"/>
</data>
```

...

<43>

<44> 일레의 XML 메시지 컴포넌트(404)

```
<msg name="ordConfirmation" type="response" action="mhConfirmation">
    <part name="orderId" type="String" />
    <part name="status" type="String" />
</msg>
```

...

<45>

<46> 일레의 XML 프리젠테이션 컴포넌트(402)

```
<screen name="scrConfirmation" title="Order Confirmation" param="Order">
    <layout type="vertical">
        <widget type="label" value="Order Confirmation Result:"/>
        < widget type="edit" value="@Order.orderStatus"/>
    </layout>
```

...

```
<menu>
    <item label="Continue" navigate="@scrMain"/>
    ...
</menu>
```

```
</screen>
```

...

<48>

<49> 일레의 ECMAScript 워크플로우 컴포넌트(406)

```

<actions>

  <function name="mbConfirmation">

    key = ordConfirmation.orderId;

    order = Order.gct(key);

    order.orderStatus = ordConfirmation.status;

    scrConfirmation.display(order);

  </function>

  ...

</actions>

```

<50>

<51> 도 4를 참조하면, 애플리케이션 프로그램(302)의 전술된 컴포넌트(400, 402, 404, 406)는 통신 기기(100)에 제공될 때에 컴포넌트 프레임워크(206)의 애플리케이션 컨테이너(300)에 의해 프레임워크 서비스(304)의 소정 세트에 대한 액세스가 제공된다. 프레임워크 서비스(304)는 통신 서비스(306), 화면 서비스(308), 지속성 서비스(310), 액세스 서비스(312), 프로비저닝(provisioning) 서비스(314) 및 유틸리티 서비스(316) 등을 포함하며, 이러한 것으로만 제한되지는 않는다.

<52>

도 1 및 도 4를 참조하면, 통신 서비스(306)는 컴포넌트 애플리케이션 프로그램(302)과 웹 서비스 등의 외부 시스템(10)간의 접속성을 관리한다. 예를 들어, 메시지(105)는 컴포넌트 애플리케이션(302) 대신에 통신 서비스(306)에 의해 웹 서비스(106)에 발송된다. 통신 서비스(306)는 또한 웹 서비스(106)로부터 메시지(105)의 데이터를 수신하고, 그 데이터를 컴포넌트 애플리케이션(302)에 전달한다. 통신 서비스(306)에 의해 수신된 데이터는 컴포넌트 애플리케이션 프로그램(302)에 의해 이루어진 요청에 대한 동기적 응답 및 웹 서비스(106)에 의해 이동 통신 기기(100)에 푸시(push)된 비동기적 데이터를 포함할 수 있다. 통신 서비스(306)는 또한 이동 통신 기기(100)가 무선 네트워크(102)로부터 접속 해제될 때에 접속성을 관리한다. 이동 통신 기기(100)는 무선 통신 기기가 무선 네트워크(102)에 의해 커버되지 않는 지역 내에 있을 때 또는 무선 송수신기가 디스플레이되지 않을 때 무선 네트워크로부터 접속해제된다. 이동 통신 기기(100)가 접속해제 모드에 있을 때, 컴포넌트 애플리케이션(302)에 의해 발송된 메시지는 통신 서비스(306)에 의해 큐잉(queue)되고, 이동 통신 기기(100)가 무선 네트워크(102)에 재접속될 때에 발송될 수 있다.

<53>

다시 도 1 및 도 4를 참조하면, 화면 서비스(308)는 컴포넌트 애플리케이션 프로그램(302)이 사용자 인터페이스(202)(도 2를 참조)의 출력 장치 상에 디스플레이될 때에 컴포넌트 애플리케이션 프로그램(302)의 시각적인 표현을 관리한다. 시각적 표현은 이미지와, 그래픽 사용자 인터페이스(GUI) 제어 및 윈도우와, 텍스트를 포함할 수 있다. 화면 서비스(308)는 기기 인프라스트럭처(204)의 출력 장치 상에서 사용자가 시청하는 것을 제어하는 스크린 스택(screen stack)을 관리한다.

<54>

다시 도 1 및 도 4를 참조하면, 지속성 서비스(310)는 컴포넌트 애플리케이션 프로그램(302)이 기기 인프라스트럭처(204)의 메모리 모듈(210)(도 2를 참조)에 데이터를 저장하도록 한다. 데이터베이스 조작용 지속성 서비스(310)에 의해 투명한 양상으로 컴포넌트 애플리케이션 프로그램(302)에 제공된다. 또한, 컴포넌트 프레임워크(206)의 런타임 환경은 메시지(105)의 메타데이터(즉, XML 데이터) 콘텐츠에 관해 수행된 작용을 용이하게 하여, 데이터의 지속성을 제공한다.

<55>

다시 도 1 및 도 4를 참조하면, 액세스 서비스(312)는 이동 통신 기기(100)에 제공되는 다른 소프트웨어 애플리케이션에 대한 액세스를 컴포넌트 애플리케이션 프로그램(302)에 제공한다. 예를 들어, 액세스 서비스(312)는 컴포넌트 애플리케이션 프로그램(302)으로 하여금 e-mail을 발송하기 위해 소프트웨어 애플리케이션에 액세스하거나 전화 통화를 시행하거나, 또는 메모리 모듈(210)(도 2를 참조)에 저장되어 있는 콘택트 정보를 액세스하도록 할 것이다. 액세스는 보안 방식으로 통신 기기(100) 상의 다른 소프트웨어 애플리케이션에 제공될 수 있다. 액세스 서비스(312)는 또한 이동 통신 기기(100) 상에 제공되는 다른 소프트웨어 애플리케이션으로 하여금 컴포넌트 애플리케이션 프로그램(302)을 액세스하도록 할 수도 있다. 다른 소프트웨어 프로그램은 기기 인프라스트

력처(204)(도 2를 참조)의 운영체계의 일부가 될 수도 있을 것이다.

- <56> 다시 도 1 및 도 4를 참조하면, 프로비저닝(provisioning) 서비스(314)는 이동 통신 기기(100) 상의 소프트웨어 애플리케이션의 프로비저닝(provisioning)을 관리한다. 애플리케이션 프로비저닝(provisioning)은 신규 및 업데이트된 컴포넌트 애플리케이션 프로그램(302)을 요청하여 수신하고, 무선 네트워크(102)를 통해 액세스 가능한 서비스에 대한 액세스를 위해 컴포넌트 애플리케이션 프로그램(302)을 구성하고, 컴포넌트 애플리케이션 프로그램(302) 및 서비스의 구성을 수정하고, 컴포넌트 애플리케이션 프로그램(302) 및 서비스를 제거하는 것을 포함한다. 제공 서비스(314)는 컴포넌트 애플리케이션 프로그램(302)을 대신하여 소프트웨어 애플리케이션을 제공하기 위해 요구된 단계를 수행한다.
- <57> 다시 도 1 및 도 4를 참조하면, 컴포넌트 애플리케이션 프로그램(302)은 다양한 공통 태스크를 달성하기 위해 유틸리티 서비스(316)를 이용한다. 예를 들어, 유틸리티 서비스(316)는 컴포넌트 애플리케이션 프로그램(302)을 대신하여 상이한 포맷으로의 스트링의 변환 등의 데이터 조작을 수행할 수 있다.
- <58> 통신 기기(100)의 프레임워크 서비스(304)는 컴포넌트 애플리케이션 프로그램(302)에 대한 기능성을 제공하는 것으로 알려져 있으며, 이 기능성은 전술된 서비스를 포함할 수 있다. 그 결과, 컴포넌트 애플리케이션 프로그램(302)은 그 기능성을 실행할 필요없이 통신 기기(100)의 기능성에 대한 액세스를 갖는다. 프레임워크 서비스(304)에 의해 제공된 기능성은 가장 대표적인 무선 애플리케이션에서 나타나는 핵심적인 기능성이기 때문에, 바람직하지 않게도, 각각의 현재 이용 가능한 하드 코딩된 무선 애플리케이션은 전술된 서비스의 일부 또는 전부를 실시하는 코드를 포함한다. 예를 들어, 대표적인 이동 기기 상에 10개의 하드 코딩된 무선 애플리케이션이 존재한다면, 현재의 공지 기술은 GUI를 디스플레이하거나 무선 네트워크를 액세스하는 등의 서비스를 수행하기 위해 동일한 코드의 10개의 카피가 존재하게 되도록 한다.
- <59> 반대로, 이동 통신 기기(100)(도 2 참조)의 컴포넌트 프레임워크(206)는 제공되는 컴포넌트 애플리케이션 프로그램(302)의 수에 상관없이 프레임워크 서비스(304)에 제공된 이들 서비스를 실시하는 코드의 한 카피만을 갖는 것이 바람직하다. 프레임워크 서비스(304)의 코드 복제가 최소화되기 때문에, 현재 이용 가능한 하드 코딩된 애플리케이션에 비해 컴포넌트 애플리케이션 프로그램(302)의 사이즈가 감소될 수 있다. 그러므로 컴포넌트 애플리케이션 프로그램(302)은 더 적은 메모리를 소비하고, 무선 네트워크(102)를 통해 전송하기 위한 시간을 더 적게 요구할 수 있다. 이에 의해 컴포넌트 애플리케이션 프로그램(302)이 통상적으로 메모리 및 처리 용량에 있어서 제약을 받는 이동 통신 기기(100) 상에서의 사용을 위해 최적화될 수 있다. 또한, 소프트웨어 애플리케이션 개발자는 컴포넌트 프레임워크(206)의 프레임워크 서비스(304)에 의해 제공되는 기능성 중의 임의의 기능성을 구현하는데 시간과 노력을 기울일 필요가 없게 된다.
- <60> 컴포넌트 프레임워크(206)의 클라이언트 런타임은 컴포넌트(400, 402, 404, 406) 정의에 포함된 메타데이터를 로드하고, 애플리케이션 컨테이너(300)를 통해 통신 기기(100)에 애플리케이션 프로그램(302)의 실행 가능한 버전을 구축한다. 예를 들어, 클라이언트 런타임에 대해 2가지의 동작 모델, 즉 템플릿 기반 원시 실행 및 메타데이터 기반 실행이 존재한다. 템플릿 기반 원시 실행 모델을 이용하면, 런타임은 원시 코드를 이용하여 통신 기기(100) 상에 사전 구축된 데이터, 메시지 및 화면 템플릿을 호스팅한다. 애플리케이션 프로그램(302) 정의가 로드될 때, 컴포넌트 프레임워크(206)에 의해 제공된 클라이언트 환경은 템플릿을 메타데이터-정의 파라미터로 채우고, 실행 가능한 클라이언트 애플리케이션 프로그램(302)을 원시 포맷으로 구축한다. 워크플로우 컴포넌트(406)의 워크플로우 스크립트(예를 들어, ECMAScript)는 원시 코드로 변환되거나 또는 인터프리터를 이용하여 원시 코드 리다이렉터(native code redirector)(예를 들어, 원시 코드에 대한 ECMAScript)로 실행될 수 있으며, 여기서 리다이렉터는 스크립팅 랭귀지에 대한 호출을 원시 컴포넌트 상에 관한 조작으로 해석한다. 메타데이터 기반 실행을 이용하면, 컴포넌트 프레임워크(206)의 런타임 환경은 컴포넌트(400, 402, 404, 406) 정의를 실행 시간 동안 구문분석되는 XML 등의 표현 포맷으로 유지하거나 또는 XML(예를 들어) 노드의 원시 표현을 사용한다. 실행 동안, 원시 런타임 엔진은 원시 컴포넌트 엔티티에 관해서보다는 컴포넌트(400, 402, 404, 406)의 정의에 관해 작용한다. 템플릿 기반 방식은 메타데이터 기반 실행에 비해 성능에 있어서 더 효율적이 될 수 있지만, 더 복잡한 실행 환경 및 더 많은 메모리 자원을 요구할 수 있다.
- <61> 각각의 컴포넌트 애플리케이션(302)이 이동 통신 기기(100) 상에 로드되기 전에, 워크플로우 컴포넌트(406)는 원시 코드 또는 Java™ 바이트 코드 등의 중간 형태로 컴파일될 수 있다. 중간 형태는 이동 통신 기기(100) 상에서 원시 코드로 변환된다. 워크플로우 컴포넌트(406)가 이동 통신 기기(100) 상에 로드되기 전에 이들을 컴파일하는 것은 워크플로우 컴포넌트(406)만을 포함하는 코드가 이동 통신 기기(100) 상에서 한 레벨로 해석되도록 하는데 도움을 줄 수 있다. 또는, 워크플로우 컴포넌트(406)는 해석된 언어로 작성된 코드로써 이동 통신

기기(100) 상에 로드될 수도 있으며, 해석된 언어는 원시 코드로 변환되기 전에 이동 통신 기기(100) 상에서 인터프리터에 의해 처리된다. 또는, 워크플로우 컴포넌트(406)는 임의의 컴파일된 또는 해석된 언어로 작성된 코드로 구성될 수도 있다.

<62> 전술된 프레임워크 서비스(304)에 대한 액세스를 갖고 그리고 사용자에게 활용도를 제공하기 위해 이동 환경에서 동작하는 이점을 이용하는 컴포넌트 애플리케이션(302)을 생성하기 위해, 애플리케이션 개발자는 전술된 바와 같은 컴포넌트를 생성할 수 있다. 데이터 컴포넌트(400), 프리젠테이션 컴포넌트(402) 및 메시지 컴포넌트(404)를 위한 익스텐서블 마크업 랭귀지(XML) 및 워크플로우 컴포넌트(406)를 위한 ECMAScript를 포함하는 표준 기술이 사용될 수 있다. 컴포넌트들은 그 후 컴포넌트 프레임워크(206)에 의해 애플리케이션 컨테이너(300)에서 실행되며, 컴포넌트 프레임워크는 워크플로우 컴포넌트(406)를 실행하고, 전술된 프레임워크 서비스(304)를 이용하여 프리젠테이션 컴포넌트(402), 데이터 컴포넌트(400) 및 메시지 컴포넌트(404)를 해석한다. 이 애플리케이션 개발 모델은 애플리케이션 프로그래머에 대해 요구된 전문성을 최소화하고 이동 통신 기기(100)용의 소프트웨어 애플리케이션을 작성하기 위해 요구된 시간의 양을 최소화하는데 도움을 준다. 또한, 컴포넌트가 생성되면, 이들은 복수의 컴포넌트 애플리케이션(302)에서 재사용될 수 있으며, 이로써 개발을 더욱 용이하게 한다.

<63> 도 1 및 도 5를 참조하면, 컴포넌트 애플리케이션 프로그램(302) 모델의 동작(800)이 도시되어 있다. 이동 기기(100)의 사용자는 네트워크(102)를 통해 웹 서비스에 접속하고, 그 후 요청 메시지(105)를 발송함으로써 웹 서비스에 선택된 애플리케이션 프로그램(302)을 요청한다(802). 적합한 웹 서버(106)가 요청 메시지(105)를 적합한 애플리케이션 서버(110)에 보내고, 컴포넌트 정의의 세트를 응답 메시지(105)로 네트워크(102)를 통해 이동 기기(100)에 발송한다(804). 응답 메시지(105)는 컴포넌트 애플리케이션 프로그램(302)을, XML 등의 구성된 정의 언어로 표현된 메타데이터 디스크립터를 포함하는 일련의 데이터 컴포넌트(400), 메시지 컴포넌트(404) 및 프리젠테이션 컴포넌트(402)(도 3 참조)의 형태로 포함한다. 응답 메시지(105)는 또한 ECMAScript 등의 프로그래밍/스크립팅 랭귀지로 표현된 디스크립터를 포함하는 워크플로우 컴포넌트(406)를 포함한다. 이동 기기(100)는 컴포넌트 프레임워크(206)(도 2 참조)를 통해 애플리케이션 프로그램(302)을 위한 런타임 환경을 제공하고(806), 접속된 웹 서비스에 대해 이동 기기(100) 상에 대응 웹 클라이언트를 생성하기 위해 실행 모델을 이용하여 애플리케이션 프로그램(302)의 컴포넌트(400, 402, 404, 406) 정의를 제공한다(808). 제공된 애플리케이션 프로그램(302)은 네트워크(102)를 경유한 메시지(105)를 통해 이동 기기(100)의 액세스를 조정하기 위해 프레임워크 서비스(304)(도 2 참조)를 이용한다. 이동 기기(100)의 사용자는 제공된 애플리케이션 프로그램(302)과 사용자 인터페이스(202)를 통해 상호작용한다.

<64> 전술된 컴포넌트 애플리케이션 모델은 컴포넌트 애플리케이션(302)이 도시되지 않은 일체형 개발 환경(IDE)을 이용하여 시각적으로 생성될 수 있기 때문에 무선 애플리케이션의 개발을 더욱 용이하게 한다. IDE는 애플리케이션 개발자가 컴포넌트의 시각적 표현을 개발자에게 제공하는 GUI를 이용하여 전술된 컴포넌트를 신속하게 생성할 수 있도록 하는 소프트웨어 애플리케이션이다. 프리젠테이션 컴포넌트(402), 데이터 컴포넌트(400) 및 메시지 컴포넌트(404)는 트리형-화면으로 디스플레이된다. 개발자는 컴포넌트를 드래그 및 드롭하여 이들을 컴포넌트 애플리케이션(302)에 추가한다. 프리젠테이션 컴포넌트(402)를 설계하기 위해 편집기를 열기 위해서는 프리젠테이션 컴포넌트(402)를 나타내는 아이콘을 클릭한다. 프리젠테이션 컴포넌트(402)간의 링크는 시각적으로 표현되며, 스크린 워크플로우를 정의하는 워크플로우 컴포넌트(406)를 편집하기 위해 클릭된다. 프리젠테이션 컴포넌트(402)는 컴포넌트 애플리케이션(302)이 실행할 이동 통신 기기(100)의 표시를 포함하는 인터페이스를 이용하여 설계된다. 메뉴 아이템이 표현되고, 메뉴 아이템이 선택될 때에 실행되는 워크플로우 컴포넌트(406)를 정의하기 위해 클릭된다. 메시지 컴포넌트(404)는 데이터 컴포넌트(400)에 시각적으로 맵핑될 수 있다. 메시지 컴포넌트(404)는 또한 웹 서비스(106) 메시지와 시각적으로 관련될 수 있으며, 메시지 컴포넌트(404)는 WSDL로 정의된 바와 같은 웹 서비스(106) 메시지에 기초하여 생성될 수 있다.

<65> 도 6 및 도 7을 참조하면, 시스템(10)의 또 다른 예는 메시지-맵 서비스(108)를 이용하는 메시징 시스템을 포함한다. 또한, 웹 서버(106)에 관한 웹 서비스, 통신 서비스(306), 컴포넌트 프레임워크(206) 및 이동 통신 기기(100)가 도시되어 있다. 간략화를 위해 다른 구성요소는 도시되어 있지 않다.

<66> 무선 네트워크(102)는 이동 통신 기기(100)가 메시지(107)의 데이터를 인터넷(104)에 접속되어 있는 시스템에 전송하거나 시스템으로부터 전송될 수 있도록 인터넷(104)에 접속되어 있다. 무선 네트워크(102)와 인터넷(104)간의 접속은 데이터가 무선 네트워크(102)에 접속되어 있는 이동 통신 기기(100) 등의 기기로부터 인터넷(104)에 접속되어 있는 웹 서비스 등의 시스템으로 진행하도록 하는 인터넷 게이트웨이(도시 생략)를 포함한다.

- <67> 웹 서버(106) 상의 웹 서비스는 이동 통신 기기(100) 상의 소프트웨어 애플리케이션(302)(도 2 참조)에 의해 사용되는 정보를 제공한다. 대안적으로 또는 이에 추가하여, 웹 서비스는 이동 통신 기기(100) 상의 소프트웨어 애플리케이션에 의해 제공된 정보를 수신하여 이용하거나 또는 이동 통신 기기(100) 상의 소프트웨어 애플리케이션을 대신하여 태스크를 수행한다. 웹 서비스는 웹 서비스 디스크립션 랭귀지(WSDL)를 이용하여 표현될 수 있는 이동 통신 기기(100) 상의 소프트웨어 애플리케이션(302)과의 인터페이스를 시행하고, 심플 오브젝트 액세스 프로토콜(SOAP) 등의 적합한 통신 프로토콜을 이용하여 웹 클라이언트 시스템과 통신한다. 또는, 웹 서비스는 다른 공지의 프로토콜을 사용할 수도 있다.
- <68> 웹 서비스는 이동 통신 기기(100)의 사용자에게 활용도를 제공하기 위해 무선 네트워크(102) 및 인터넷(104)을 통해 이동 통신 기기(100) 상의 소프트웨어 애플리케이션이 통신하는 시스템의 일레이다. 이동 통신 기기(100)와 웹 서비스간에 발송된 메시지(107)는 메시지-맵 서비스(108)를 중단한다. 메시지-맵 서비스(108)는 메시지를 이동 통신 기기(100) 상의 소프트웨어 애플리케이션에 의해 사용 가능한 포맷에서 웹 서비스에 의해 요구되는 포맷으로 변환하여 메시지(105)를 생성한다. 이동 통신 기기(100) 중의 하나로부터의 메시지(107)가 적합한 포맷으로 변환되면, 메시지-맵 서비스(108)는 메시지(105)를 웹 서비스 상으로 전송한다. 유사하게, 웹 서비스로부터의 메시지 또는 응답(105)이 요구된 포맷으로 변환될 때, 메시지-맵 서비스는 메시지(107)를 이동 통신 기기(100) 중의 하나에 전송한다. 메시지-맵 서비스(108)는 도 7에 상세히 설명되어 있다.
- <69> 도 7을 참조하면, 웹 서비스와 상호작용하여 정보를 변경하기 위해, 컴포넌트 애플리케이션(302)(도 2 참조)은 무선 메시지(502)를 무선 네트워크(102) 및 인터넷(104)을 통해 통신 서비스(306)를 이용하여 발송한다. 예를 들어, 무선 메시지(502)는 컴포넌트 애플리케이션(302)에 정의된 메시지 컴포넌트(404)(도 3 참조) 중의 하나에 따라 포맷화된다. 웹 서비스를 액세스하기 위해, 예를 들어, SOAP 메시지(504)가 사용된다. 상이한 메시지 유형을 해결하기 위해, 메시지-맵 서비스(108)는 무선 메시지(502)를 수신하고, 대응하는 SOAP 메시지(504)를 구축하며, SOAP 메시지(504)를 적합한 SOAP 프로토콜을 이용하여 인터넷(104)을 통해 웹 서버(106) 상의 웹 서비스에 발송한다. 유사하게, 웹 서비스에 의해 생성된 응답 메시지(504)는 SOAP를 이용하여 인터넷(104)을 통해 메시지-맵 서비스(108)에 발송되어, 이 서비스에서 메시지가 메시지 컴포넌트(404) 중의 하나에 따른 포맷으로 된 대응 무선 메시지(502)로 변환되며, 이동 통신 기기(100)에 발송되어 컴포넌트 애플리케이션(302) 대신에 통신 서비스(306)에 의해 수신된다.
- <70> 다시 도 7을 참조하면, 메시지-맵 서비스(108)는 각각의 컴포넌트 애플리케이션(302)(도 2 참조)에 대한 애플리케이션 메시지-맵(500)을 포함한다. 애플리케이션 메시지-맵(500)은 컴포넌트 애플리케이션(302)으로부터 수신된 각각의 무선 메시지(502)가 SOAP 메시지(504) 등의 대응 웹 서비스 메시지로 어떻게 변환되는지와, 컴포넌트 애플리케이션(302)에 발송될 웹 서비스로부터 수신된 SOAP 메시지(504) 등의 각각의 웹 서비스 메시지가 대응 무선 메시지(502)로 어떻게 변환되는지를 정의한다. 전송된 바와 같이, 각각의 무선 메시지(502)는 컴포넌트 애플리케이션(302) 내의 메시지 컴포넌트(404) 중의 하나에 따라 포맷화될 수 있는 한편, 각각의 웹 서비스 메시지(예를 들어, SOAP 메시지(504))는 웹 서비스의 정의에 따라 포맷화될 수 있다. 메시지-맵(500)은 정의에 정의된 어떤 정보가 애플리케이션 프로그램(302)의 메시지 컴포넌트(404)에 정의된 정보에 대응하는지를 특정한다. 이들 정의는 WSDL 등의 적합한 포맷으로 인코딩될 수 있다. 이 맵핑은 그 후 메시지(502)를 2개의 포맷간에서 변환하기 위해 사용된다.
- <71> SOAP 메시지(504) 내의 정보의 전부가 대응 무선 메시지(502) 내의 정보로 맵핑될 필요는 없는 것으로 알려져 있다. 예를 들어, 무선 메시지(502)에서 SOAP 메시지(504)로의 전환에서, 애플리케이션 메시지-맵(500)은 무선 메시지(502)에 대응하는 정보가 존재하지 않는 SOAP 메시지(504)에 제공하기 위해 디폴트 값을 특정할 수도 있을 것이다. 유사하게, SOAP 메시지(504)에서 무선 메시지(502)로 변환할 때, 무선 메시지(502) 내에 대응 정의가 존재하지 않는 SOAP 메시지(504) 내의 정보는 폐기된다. 따라서, 컴포넌트 애플리케이션(302)에 의해 활용되지 않는 웹 서비스로부터의 응답에 포함되지 않는 정보는 무선 네트워크(102)를 통해 전송되지 않을 수도 있다. 유사하게, 메시지(504)가 컴포넌트 애플리케이션(302)에 의해 발송될 때마다 동일한 웹 서비스에 발송된 SOAP 메시지(504)에서 요구되는 정보는 디폴트로 되어, 중복(redundant) 정보가 무선 네트워크(102)를 통해 전송되지 않는다. 그 결과, 무선 네트워크(102)를 통해 더 적은 데이터가 전송되어 무선 네트워크의 혼잡을 감소시킬 수 있으며, 이동 통신 기기(100) 상에서 메시지(502)를 처리하기 위해 요구된 자원의 양이 감소될 수 있다.
- <72> 따라서, 메시지-맵 서비스(108)는 컴포넌트 애플리케이션(302)이 이동 통신 기기(100) 상에서 구현되는 웹 메시징 서비스 프로토콜(예를 들어, SOAP)를 갖지 않고서도 웹 서비스를 통합할 수 있게 한다. 컴포넌트 애플리케이션(302)은 또한 웹 서비스에 의한 사용을 위해 메시지(502)를 포맷화할 필요가 없거나, 기존의 웹 서비스

(106)와 통합하기 위해 어떠한 다른 추가 처리를 수행할 필요가 없다.

- <73> 메시지-맵 서비스(108)를 사용하는 것에 대한 대안으로써, 메시지(502, 504)는 이동 통신 기기(100) 상의 통신 서비스(306)에 의해 패핑될 수도 있을 것이다. 이 예에서, SOAP 프로토콜 정의를 갖는 kSOAP 등의 웹 메시징 서비스 프로토콜을 갖는 소프트웨어가 이동 통신 기기(100)에 설치되어, 통신 서비스(306)가 웹 서비스(106)에 직접 통신할 수 있다. 메시지(502)의 해석은 메시지-맵 서비스(108)를 구현하는 중간의 제3자를 이용하는 것보다 이동 기기(100)에서 발생하는 것이 효율적이다.
- <74> 도 8은 무선 컴포넌트 애플리케이션(600)을 채용하여 통신하는 방법을 예시하는 흐름도이다. 무선 컴포넌트 애플리케이션(600)은 데이터 컴포넌트(400), 프리젠테이션 컴포넌트(402), 메시지 컴포넌트(404) 및 워크플로우 컴포넌트(406)(도 3 참조)를 포함하는 전송된 바와 같은 컴포넌트 애플리케이션(302)이다.
- <75> 도 6 및 도 8을 참조하면, 본 방법은 단계 "602"에서 개시하며, 이 단계에서는 무선 컴포넌트 애플리케이션(600)이 이동 통신 기기(100)에 배치된다. 무선 컴포넌트 애플리케이션(600)은 무선 네트워크(102)를 경유하여 무선으로 이동 통신 기기(100)에 전송된다. 또는, 무선 컴포넌트 애플리케이션(600)은 시리얼 접속, USB 링크 또는 Bluetooth™ 혹은 802.11 네트워크(도시되지 않음) 등의 단거리 통신 시스템을 통해 이동 통신 기기(100)에 로드될 수도 있을 것이다.
- <76> 다시 도 6 및 도 8을 참조하면, 본 방법은 애플리케이션 메시지-맵(108)을 배치하는 단계 "604"로 진행한다. 전송된 바와 같이, 애플리케이션 메시지-맵(108)은 메시지를 무선 컴포넌트 애플리케이션(600)에 의해 사용된 포맷에서 무선 컴포넌트 애플리케이션(600)이 통신하고 있는 웹 서비스에 의해 요구된 포맷으로 변환하기 위해 사용된다. 애플리케이션 메시지-맵(500)(도 7 참조)은 인터넷(104)을 통해 액세스 가능한 전용 메시지-맵 서비스(108)에 배치된다. 또는, 메시지-맵(108)은 이동 통신 기기(100)에 직접 배치될 수도 있을 것이다. 단계 "606"은 이동 통신 기기(100) 상의 무선 컴포넌트 애플리케이션(600)을 실행한다. 전송된 바와 같이, 이동 통신 기기(100) 상의 애플리케이션 프레임워크(206)(도 2 참조)는 애플리케이션 컨테이너(300)를 생성하고, 컨테이너(300)에서 무선 컴포넌트 애플리케이션(600)을 실행하며, 이 컨테이너는 프레임워크 서비스(304)에 대한 액세스를 무선 컴포넌트 애플리케이션(600)에 제공한다.
- <77> 도 6 및 도 8을 참조하면, 무선 컴포넌트 애플리케이션(600)이 이동 통신 기기(100) 상에서 실행되고 있을 때, 본 방법은 단계 "608"로 진행하여 무선 컴포넌트 애플리케이션(600)이 무선 메시지(502)를 외부의 웹 서비스에 발송하기를 원하는지의 여부를 판정한다. 무선 컴포넌트 애플리케이션(600)은 메시지(502)를 웹 서비스에 발송하여, 웹 서비스로부터 데이터를 요청하거나, 웹 서비스에 데이터를 전송하거나, 또는 웹 서비스가 무선 컴포넌트 애플리케이션(600) 대신에 동작을 수행하도록 요청한다. 여기서, 무선 컴포넌트 애플리케이션(600)이 발송할 더 이상의 메시지(502)를 갖지 않는 것으로 판정되면, 본 방법은 단계 "610"에서 완료한다. 그렇지 않다면, 본 방법은 단계 "612"로 진행하여 발송하기 위한 메시지(502)를 구성한다. 메시지(502)는 메시지 컴포넌트(404)(도 3 참조) 중의 하나에서 정의된 포맷에 따라 통신 서비스에 의해 구성된다. 통신 서비스는 그 후 메시지(502)를 무선 네트워크(102)를 통해 전송한다.
- <78> 도 6 및 도 8을 참조하면, 메시지(502)는 무선 네트워크(102) 및 인터넷(104)을 통해 메시지-맵 서비스(108)에 의해 수신된다. 메시지-맵 서비스(108)는 그 후 메시지(502)를 단계 "604"에서 배치된 애플리케이션 메시지-맵(500)을 이용하여 SOAP 포맷으로 변환한다. 애플리케이션 메시지-맵(500)은 웹 서비스에 의해 요구된 SOAP 포맷 중의 어느 부분이 무선 컴포넌트 애플리케이션(600)으로부터의 메시지(502)의 각각의 부분에 대응하는지를 특정한다. 단계 "616"에서는 변환된 메시지(504)가 인터넷(104)을 통해 SOAP 프로토콜을 이용하여 웹 서비스에 전송된다. 또는, 단계 "614"에서는 메시지(502)가 이동 통신 기기(100) 상에 위치한 애플리케이션 메시지-맵(500)을 이용하여 SOAP 포맷으로 변환될 수도 있으며, 그 후 단계 "616"에서 무선 네트워크(102) 및 인터넷(104)을 통해 kSOAP 프로토콜을 이용하여 웹 서비스에 직접 전송될 수도 있을 것이다.
- <79> 다시 도 6 및 도 8을 참조하면, 메시지(504)가 웹 서비스에 의해 수신될 때, 본 방법은 단계 "618"로 진행하여 웹 서비스가 메시지(504)에 대한 응답을 구성한다. SOAP 포맷의 응답은 인터넷(104)을 통해 메시지-맵 서비스(108)에 전송되며, 단계 "620"에서는 응답이 무선 컴포넌트 애플리케이션(600)에 의해 요구된 무선 포맷(502)으로 변환된다. 요구된 무선 포맷은 무선 컴포넌트 애플리케이션(600)에 포함된 메시지 컴포넌트(404)(도 3 참조) 중의 하나에서 정의된다. 메시지-맵 서비스(108)는 그 후 인터넷(104) 및 무선 네트워크(102)를 통해 단계 "622"에서 무선 컴포넌트 애플리케이션(600)에 응답을 전송한다. 또는, 웹 서비스는 SOAP 응답 메시지(504)를 직접 이동 통신 기기(100)에 전송할 수도 있을 것이며, 이 메시지는 이동 통신 기기(100)에 상주하는 애플리케이션 메시지-맵(500)을 이용하여 단계 "620"에서의 요구된 포맷으로 변환된다. 변환된 메시지(502)는 그

후 단계 "622"에서 무선 컴포넌트 애플리케이션(600)에 제공된다. 본 방법은 그 후 단계 "608"을 지속하며, 이 단계에서는 전송할 메시지(502)가 더 있는지의 여부가 판정된다. 더 이상의 메시지(502)가 존재하지 않는다면, 본 방법은 단계 "610"에서 종료한다. 무선 컴포넌트 애플리케이션(600)을 채용하여 통신하는 방법은 도 8에 도 시된 것보다 더 적거나 더 많은 단계를 포함할 수도 있을 것이라는 점은 자명하다.

- <80> 도 9는 도 1 및 도 6의 기기(100)의 또다른 예인 듀얼-모드 이동 통신 기기(710)의 블록도이다. 듀얼-모드 이 동 통신 기기(710)는 송수신기(711), 마이크로프로세서(738), 디스플레이(722), 플래시 메모리(724), RAM 메모 리(726), 보조 입력/출력(I/O) 장치(728), 시리얼 포트(730), 키보드(732), 스피커(734), 마이크로폰(736), 단 거리 무선 통신 서브-시스템(740)을 포함하며, 또한 다른 기기 서브-시스템(742)을 포함할 수도 있을 것이다. 송수신기(711)는 송신 및 수신 안테나(716, 718), 수신기(712), 송신기(714), 하나 이상의 국부 발진기(713) 및 디지털 신호 처리장치(720)를 포함하는 것이 바람직하다. 플래시 메모리(724)내에서, 듀얼-모드 이동 통신 기 기(710)는 마이크로프로세서(738)(및/또는 DSP(720))에 의해 실행될 수 있는 복수의 소프트웨어 모듈(724A 내지 724N)을 포함하는 것이 바람직하며, 이 소프트웨어 모듈은 음성 통신 모듈(724A), 데이터 통신 모듈(724B) 및 복수의 다른 기능을 수행하기 위한 복수의 다른 동작 모듈(724N)을 포함한다.
- <81> 듀얼-모드 이동 통신 기기(710)는 음성 및 데이터 통신 성능을 갖는 양방향 통신 장치인 것이 바람직하다. 그 러므로, 예를 들어, 듀얼-모드 이동 통신 기기(710)는 아날로그 또는 디지털 셀룰러 네트워크 중의 임의의 것과 같은 음성 네트워크를 통해 통신할 것이며, 또한 데이터 네트워크를 통해 통신할 수도 있을 것이다. 음성 및 데이터 네트워크는 통신 타워(719)로 도 9에 도시되어 있다. 이들 음성 및 데이터 네트워크는 기지국, 네트워 크 컨트롤러 등과 같은 별도의 인프라스트럭처를 이용하는 별도의 통신 네트워크가 되거나, 또는 단일의 무선 네트워크로 통합될 수도 있을 것이다.
- <82> 통신 서브시스템(711)은 음성 및 데이터 네트워크(719)와 통신하기 위해 사용되며, 수신기(712), 송신기(714), 하나 이상의 국부 발진기(713)를 포함하며, 또한 DSP(720)를 포함할 수도 있을 것이다. DSP(720)는 신호를 송 신기(714)에 전송하고 신호를 수신기(712)로부터 수신하기 위해 사용되며, 또한 송신기(714)로부터 제어 정보를 수신하고 제어 정보를 수신기(712)에 제공하기 위해 이용된다. 음성 및 데이터 통신이 단일 주파수 또는 인접 이격된 세트의 주파수로 발생한다면, 송신기(714) 및 수신기(712)와 함께 단일 국부 발진기(713)가 이용될 수도 있을 것이다. 또는, 음성 통신과 데이터 통신을 위해 상이한 주파수가 이용된다면, 음성 및 데이터 네트워크 (719)에 대응하는 복수의 주파수를 생성하기 위해 복수의 국부 발진기(713)가 이용될 수 있다. 도 9에는 2개의 안테나(716, 718)가 도시되어 있지만, 듀얼-모드 이동 통신 기기(710)는 단일의 안테나 구조체와 함께 사용될 수 있다. 음성 및 데이터 정보 모듈을 포함하는 정보는 DSP(720)와 마이크로프로세서(738)간의 링크를 통해 통 신 모듈(711)과 통신된다. 주파수 대역, 컴포넌트 선택, 파워 레벨 등과 같은 통신 서브시스템(711)의 세부 설 계는 듀얼-모드 이동 통신 기기(710)가 동작할 통신 네트워크(719)에 좌우된다. 예를 들어, 북미 지역의 시장 에서 동작할 듀얼-모드 이동 통신 기기(710)는 Mobitex™ 또는 DataTAC™ 이동 데이터 통신 네트워크와 동작하 도록 설계되고, 또한 AMPS, TDMA, CDMA, PCS 등과 같은 임의의 다양한 음성 통신 네트워크와 동작하도록 설계된 통신 서브시스템(711)을 포함할 것인 반면, 유럽 지역에서 사용될 듀얼-모드 이동 통신 기기(710)는 General Packet Radio Service(GPRS) 데이터 통신 네트워크 및 GSM 음성 통신 네트워크와 동작하도록 구성될 것이다. 분리형 및 통합형의 다른 유형의 데이터 및 음성 네트워크 또한 듀얼-모드 이동 통신 기기(710)와 함께 이용될 수도 있을 것이다.
- <83> 네트워크 또는 네트워크들(719)의 유형에 따라 듀얼-모드 이동 통신 기기(710)에 대한 액세스 조건 또한 변화할 것이다. 예를 들어, Mobitex 및 DataTAC 데이터 네트워크에서, 이동 기기는 각각의 기기와 관련된 고유 식별자 번호를 이용하여 네트워크 상에 등록된다. 그러나, GPRS 데이터 네트워크에서, 네트워크 액세스는 이동 기기의 가입자 또는 사용자와 관련된다. GPRS 기기는 통상적으로 GPRS 네트워크 상의 듀얼-모드 이동 통신 기기를 동 작시키기 위해 요구되는 가입자 아이덴티티 모듈("SIM")을 요구한다. 로컬 또는 비네트워크 통신 기능(필요 시)은 SIM없이도 동작 가능할 것이지만, 듀얼-모드 통신 기기는 911 긴급 통화와 같은 어떠한 법적으로 요구된 조작 이외의 데이터 네트워크(719)를 통한 통신을 수반하는 임의의 기능을 수행할 수 없게 될 것이다.
- <84> 임의의 요구되는 네트워크 등록 또는 활성화 절차가 완료된 후, 듀얼-모드 이동 통신 기기(710)는 네트워크 (719)(또는 네트워크들)를 통해 음성 및 데이터 신호 양자를 포함하는 통신 신호를 전송하고 수신할 수 있을 것 이다. 안테나(716)에 의해 통신 네트워크(719)로부터 수신된 신호는 수신기(712)에 라우팅되며, 이 수신기는 신호 증폭, 주파수 하향 변환, 필터링, 채널 선택 등을 제공하고, 또한 아날로그-디지털 변환을 제공할 수도 있 을 것이다. 수신된 신호의 아날로그-디지털 변환은 DSP(720)를 이용하여 수행되는 디지털 복조 및 디코딩 등의 더욱 복잡한 통신 기능을 허용한다. 유사한 방식으로, 네트워크에 전송될 신호가 예를 들어 DSP(720)에 의해

변조 및 인코딩을 포함하여 처리되며, 그 후 디지털-아날로그 변환, 주파수 상향 변환, 증폭 및 안테나(718)를 통한 통신 네트워크(719)(또는 네트워크들)에의 송신을 위해 송신기(714)에 제공된다. 음성 및 데이터 양자의 통신을 위해 도 9에 신호 송수신기(711)가 도시되어 있지만, 듀얼-모드 이동 통신 기기(710)가 2개의 별개 송수신기와, 음성 신호를 송신 및 수신하기 위한 제1 송수신기와, 데이터 신호를 송신 및 수신하기 위한 제2 송수신기를 포함하는 것도 가능하다.

<85> DSP(720)는 통신 신호를 처리하는 것 외에도 수신기 및 송신기 제어를 제공한다. 예를 들어, 수신기(712) 및 송신기(714)에서 통신 신호에 가해진 이득 레벨은 DSP(720)에서 구현된 자동 이득 제어 알고리즘을 통해 적응식으로 제어될 수도 있을 것이다. 송수신기(711)의 더욱 복잡한 제어를 제공하기 위해 다른 송수신기 제어 알고리즘 또한 DSP(720)에서 구현될 수 있을 것이다.

<86> 마이크로프로세서(738)는 듀얼-모드 통신 기기(710)의 전체적인 동작을 관리 및 제어하는 것이 바람직하다. 여기에서는 다수 유형의 마이크로프로세서 또는 마이크로컨트롤러가 이용되거나, 또는 마이크로프로세서(738)의 기능을 수행하기 위해 단일의 DSP(720)가 이용될 수도 있다. 적어도 데이터 및 음성 통신을 포함하는 저레벨 통신 기능은 송수신기(711) 내의 DSP(720)를 통해 수행된다. 음성 통신 애플리케이션(724A) 및 데이터 통신 애플리케이션(724B) 등의 다른 고레벨 통신 애플리케이션은 마이크로프로세서(738)에 의해 실행을 위해 플래시 메모리(724)에 저장될 수도 있을 것이다. 예를 들어, 음성 통신 모듈(724A)은 네트워크(719)를 통한 듀얼-모드 이동 통신 기기(710)와 복수의 다른 음성 기기간의 음성 통화를 송신 및 수신하도록 작동할 수 있는 고레벨 사용자 인터페이스를 제공할 수도 있을 것이다. 유사하게, 데이터 통신 모듈(724B)은 네트워크(719)를 통해 듀얼-모드 이동 통신 기기(710)와 복수의 다른 데이터 장치간의 e-mail 메시지, 파일, 오거나이저 정보(organizer information), 단문 메시지 등과 같은 데이터를 발송 및 수신하기 위해 동작할 수 있는 고레벨 사용자 인터페이스를 제공할 수도 있을 것이다. 듀얼-모드 이동 통신 기기(710)에서, 전술된 컴포넌트 프레임워크(206) 또한 소프트웨어 모듈 또는 애플리케이션으로써 구현되거나 또는 소프트웨어 모듈(724A 내지 724B) 중의 하나에 통합될 수도 있을 것이다.

<87> 마이크로프로세서(738)는 또한 디스플레이(722), 플래시 메모리(724), 랜덤 액세스 메모리(RAM)(726), 보조 입력/출력(I/O) 서브시스템(728), 시리얼 포트(730), 키보드(732), 스피커(734), 마이크(736), 단거리 통신 서브시스템(740) 및 전반적으로 도면부호 "742"가 부여된 임의의 다른 듀얼-모드 이동 통신 기기 서브시스템 등의 다른 듀얼-모드 이동 통신 기기 서브시스템과 상호작용한다.

<88> 도 9에 도시된 서브시스템의 일부는 통신 관련 기능을 수행하는 반면, 다른 서브시스템은 상주 또는 내장 기능을 제공할 수도 있을 것이다. 명확히, 키보드(732) 및 디스플레이(722) 등의 일부 서브시스템은 데이터 통신 네트워크를 통한 전송을 위해 텍스트 메시지를 입력하는 등의 통신 관련 기능과 계산기 또는 업무 목록 혹은 다른 PDA 타입 기능 등의 장치 상주 기능 모두를 위해 사용될 수 있을 것이다.

<89> 마이크로프로세서(738)에 의해 사용된 운영체제 소프트웨어는 플래시 메모리(724) 등의 영구 저장장치에 저장되는 것이 바람직하다. 듀얼-모드 이동 통신 기기(710)의 저레벨 기능의 전부를 제어하는 운영체제 외에도, 플래시 메모리(724)는 복수의 고레벨 소프트웨어 애플리케이션 프로그램 또는 음성 통신 모듈(724A), 데이터 통신 모듈(724B), 오거나이저 모듈(도시 생략) 혹은 임의의 다른 유형의 소프트웨어 모듈(724N) 등의 모듈을 포함할 수도 있을 것이다. 플래시 메모리(724)는 또한 데이터를 저장하기 위한 파일 시스템을 포함할 수도 있을 것이다. 이들 모듈은 마이크로프로세서(738)에 의해 실행되며, 듀얼-모드 이동 통신 기기의 사용자와 이동 기기간의 고레벨 인터페이스를 제공한다. 이 인터페이스는 통상적으로 디스플레이(722)를 통해 제공된 그래픽 컴포넌트와, 보조 I/O(728), 키보드(732), 스피커(734) 및 마이크(736)를 통해 제공된 입력/출력 컴포넌트를 포함한다. 운영체제, 특정 듀얼-모드 이동 통신 기기 소프트웨어 애플리케이션이나 모듈 혹은 그들의 일부가 고속 동작을 위해 RAM(726) 등의 휘발성 저장장치에 일시적으로 로드될 수도 있을 것이다. 더욱이, 수신된 통신 신호는 그들을 영구 저장장치(724)에 위치한 파일 시스템에 영구적으로 기입하기 전에 RAM(726)에 일시적으로 저장될 수도 있을 것이다.

<90> 듀얼-모드 이동 통신 기기(710) 상에 로드될 수도 있는 보조 애플리케이션 모듈(724N)은 캘린더 이벤트, 약속 및 업무 항목 등의 PDA 기능성을 제공하는 개인 정보 관리자(PIM) 애플리케이션이다. 이 모듈(724N)은 또한 전화 통화, 음성 메일 등을 관리하기 위해 음성 통신 모듈(724A)과 상호작용할 수도 있을 것이며, 또한 e-mail 통신 및 다른 데이터 전송을 관리하기 위해 데이터 통신 모듈과도 상호작용할 수 있을 것이다. 또는, 음성 통신 모듈(724A) 및 데이터 통신 모듈(724B)의 기능성의 전부가 PIM 모듈에 통합될 수도 있을 것이다.

<91> 플래시 메모리(724)는 듀얼-모드 이동 통신 기기(710)에 관한 PIM 데이터 항목의 저장을 용이하게 하기 위해 과

일 시스템을 제공하는 것이 바람직하다. PIM 애플리케이션은 그 자체로 또는 음성 및 데이터 통신 모듈(724A, 724B)과 조정하여 무선 네트워크(719)를 통해 데이터 항목을 발송 및 수신하는 능력을 포함하는 것이 바람직하다. PIM 데이터 항목은 무선 네트워크(719)를 통해 단절없이(seamlessly) 일체화되고, 호스트 컴퓨터 시스템에 저장 또는 관련된 대응 세트의 데이터 항목에 동기되어 업데이트되는 것이 바람직하며, 이에 의해 특정 사용자와 관련된 데이터 항목에 대한 미러링 시스템을 구축한다.

- <92> 듀얼-모드 이동 통신 기기(710)는 또한 듀얼-모드 이동 통신 기기(710)의 시리얼 포트(730)를 호스트 시스템의 시리얼 포트에 연결하는 인터페이스 크래들에 듀얼-모드 이동 통신 기기(710)를 위치시킴으로써 호스트 시스템과 수동으로 동기될 수도 있을 것이다. 시리얼 포트(730)는 또한 사용자가 외부 기기 또는 소프트웨어 애플리케이션을 통해 선호사항을 설정하도록 하거나, 또는 설치를 위해 다른 애플리케이션 모듈(724N)을 다운로드하도록 하기 위해 사용될 수도 있을 것이다. 이 유선 다운로드 경로는 듀얼-모드 이동 통신 기기(710) 상에 암호화 키를 로드하기 위해 사용될 수도 있으며, 이것은 무선 네트워크(719)를 통해 암호화 정보를 교환하는 것보다 더욱 안전한 방법이다.
- <93> 또 다른 애플리케이션 모듈(724N)은 네트워크(719), 보조 I/O 서브시스템(728), 시리얼 포트(730), 단거리 통신 서브시스템(740) 또는 임의의 다른 적합한 서브시스템(742)을 통해 듀얼-모드 이동 통신 기기(710) 상에 로드되고, 사용자에 의해 플래시 메모리(724) 또는 RAM(726)에 설치될 것이다. 애플리케이션 설치에 있어서의 이러한 유연성은 듀얼-모드 이동 통신 기기(710)의 기능성을 증가시키며, 향상된 기기 내장 기능, 통신 관련 기능 또는 이 둘 모두를 제공할 것이다. 예를 들어, 보안 통신 애플리케이션은 듀얼-모드 이동 통신 기기(710)를 이용하여 수행될 전자 상거래 기능 및 다른 이러한 금융 거래를 가능하게 할 것이다.
- <94> 듀얼-모드 기기(710)가 데이터 통신 모드로 동작하고 있을 때, 텍스트 메시지 또는 웹 페이지 다운로드 등의 수신 신호는 송수신기(711)에 의해 처리되어 마이크로프로세서(738)에 제공될 것이며, 이 마이크로프로세서는 수신 신호를 디스플레이(722)에 출력하거나 또는 대안적으로 보조 I/O 장치(728)에 출력하기 위해 추가로 처리하는 것이 바람직할 것이다. 듀얼-모드 이동 통신 기기(710)의 사용자는 또한 키보드(732)를 이용하여 e-mail 메시지 등의 데이터 항목을 작성할 수도 있을 것이며, 이 키보드로는 기존의 DVORAK 스타일 등의 다른 스타일의 완전한 문자숫자 키보드도 이용될 수 있기는 하지만, QWERTY 스타일로 설계된 완전한 문자숫자 키보드가 바람직하다. 듀얼-모드 이동 통신 기기(710)에 대한 사용자 입력은 덤브휠 입력 장치, 터치패드, 다양한 스위치, 자물쇠 입력 스위치 등을 포함할 수도 있는 복수의 보조 I/O 장치(728)로 추가로 향상된다. 사용자에 의해 입력된 작성된 데이터 항목은 그 후 송수신기(711)를 경유하여 통신 네트워크(719)를 통해 송신될 수도 있을 것이다.
- <95> 듀얼-모드 이동 통신 기기(710)가 음성 통신 모드로 동작하고 있을 때, 듀얼-모드 이동 통신 기기(710)의 전체적인 동작은 수신된 신호가 스피커(734)에 출력되고 송신을 위한 음성 신호가 마이크(736)에 의해 생성된다는 점을 제외하고는 데이터 모드와 실질적으로 유사하다. 음성 메시지 레코딩 서브시스템 등의 다른 음성 또는 오디오 I/O 서브시스템이 듀얼-모드 이동 통신 기기(710) 상에 구현될 수도 있을 것이다. 음성 또는 오디오 신호 출력이 1차적으로는 스피커(734)를 통해 달성되는 것이 바람직하기는 하지만, 호출하는 측(calling party)의 식별 표시, 음성 통화의 지속시간 또는 다른 음성 통화 관련 정보를 제공하기 위해 디스플레이(722)가 사용될 수도 있을 것이다. 예를 들어, 마이크로프로세서(738)는 음성 통신 모듈 및 운영체제 소프트웨어로 조정되어 인입 음성 통화의 발신자 식별 정보를 검출하고, 이것을 디스플레이(722) 상에 디스플레이할 수도 있을 것이다.
- <96> 듀얼-모드 이동 통신 기기(710)에는 단거리 통신 서브시스템(740)도 포함된다. 예를 들어, 단거리 통신 서브시스템(740)은 유사하게 인에이블되는 시스템 및 장치와의 통신을 제공하기 위해 적외선 장치와 관련 회로 및 부품 또는 Bluetooth™ 모듈이나 802.11 모듈 등의 단거리 무선 통신 모듈을 포함할 수도 있을 것이다. 당업자는 "Bluetooth" 및 802.11 이 각각 무선 사용자 영역 네트워크 및 무선 LAN에 관련하여 전기전자 기술자 협회(IEEE)로부터 이용 가능한 사양의 세트를 지칭한다는 것을 이해할 것이다.
- <97> 상기의 설명은 하나 이상의 일례의 시스템 및 방법에 관하여 이루어졌다. 다양한 변형이 가능하다는 것은 본 기술분야의 당업자에게는 자명한 것이며, 이러한 변형은 본 특허출원의 기술사상 내에 있는 것이다. 예를 들어, 상기의 예에서 SOAP 프로토콜을 구현하는 웹 서비스를 설명하였지만, 웹 서비스는 이동 통신 기기에 액세스 가능한 임의의 정보 소스가 될 수도 있으며, 이 정보 소스는 상이한 통신 프로토콜을 구현할 수도 있을 것이다.
- <98> 본 명세서의 개시 내용이 하나 이상의 예시 시스템 및 방법에 대해 이루어졌지만, 첨부된 특허청구 범위에 의해 한정되는 본 발명의 정신 및 기술사상으로부터 벗어나지 않고 당업자에 의해 다양한 변형이 이루어질 수 있음은

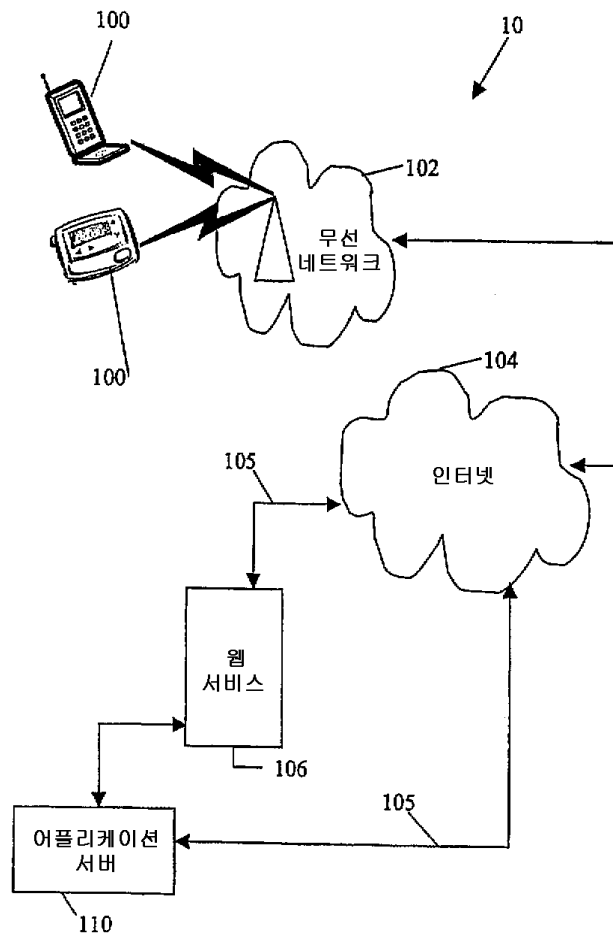
자명할 것이다.

도면의 간단한 설명

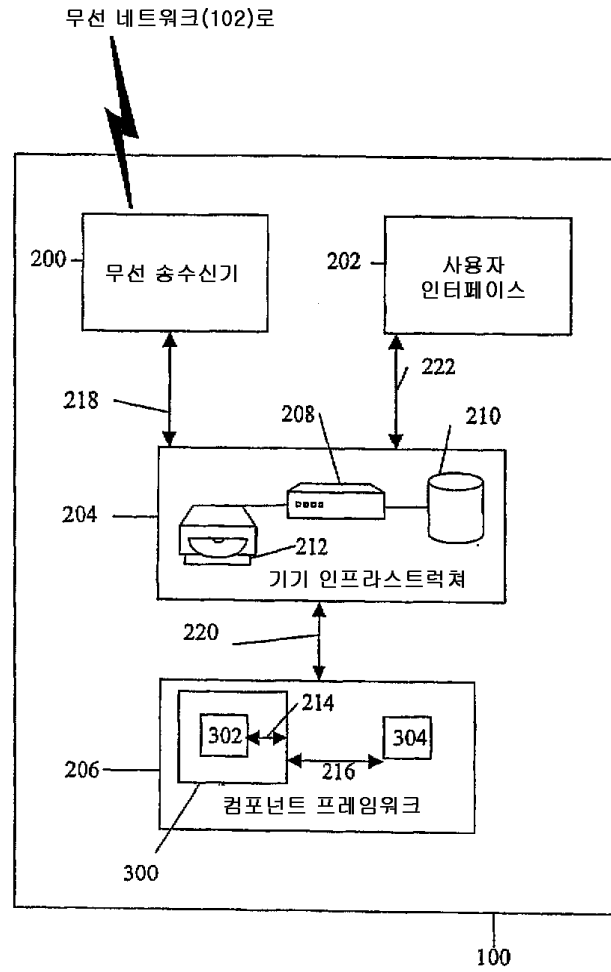
- <12> 도 1은 통신 시스템의 블록도이다.
- <13> 도 2는 도 1의 이동 통신 기기의 블록도이다.
- <14> 도 3은 도 2의 컴포넌트 애플리케이션 프로그램의 블록도이다.
- <15> 도 4는 도 3의 컴포넌트 프레임워크의 블록도이다.
- <16> 도 5는 도 2의 이동 기기의 일례의 동작의 흐름도이다.
- <17> 도 6은 도 1의 통신 시스템의 또 다른 예이다.
- <18> 도 7은 도 6의 메시지-맵 서비스를 이용하는 메시징 시스템의 블록도이다.
- <19> 도 8은 도 6의 통신 기기의 무선 컴포넌트 애플리케이션을 작동시키는 방법을 예시하는 흐름도이다.
- <20> 도 9는 도 2의 이동 통신 기기의 또다른 예의 블록도이다.

도면

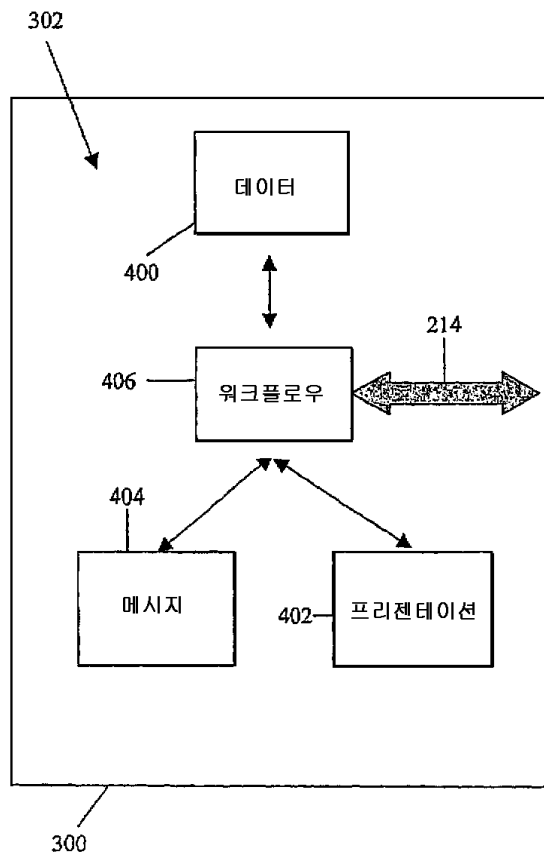
도면1



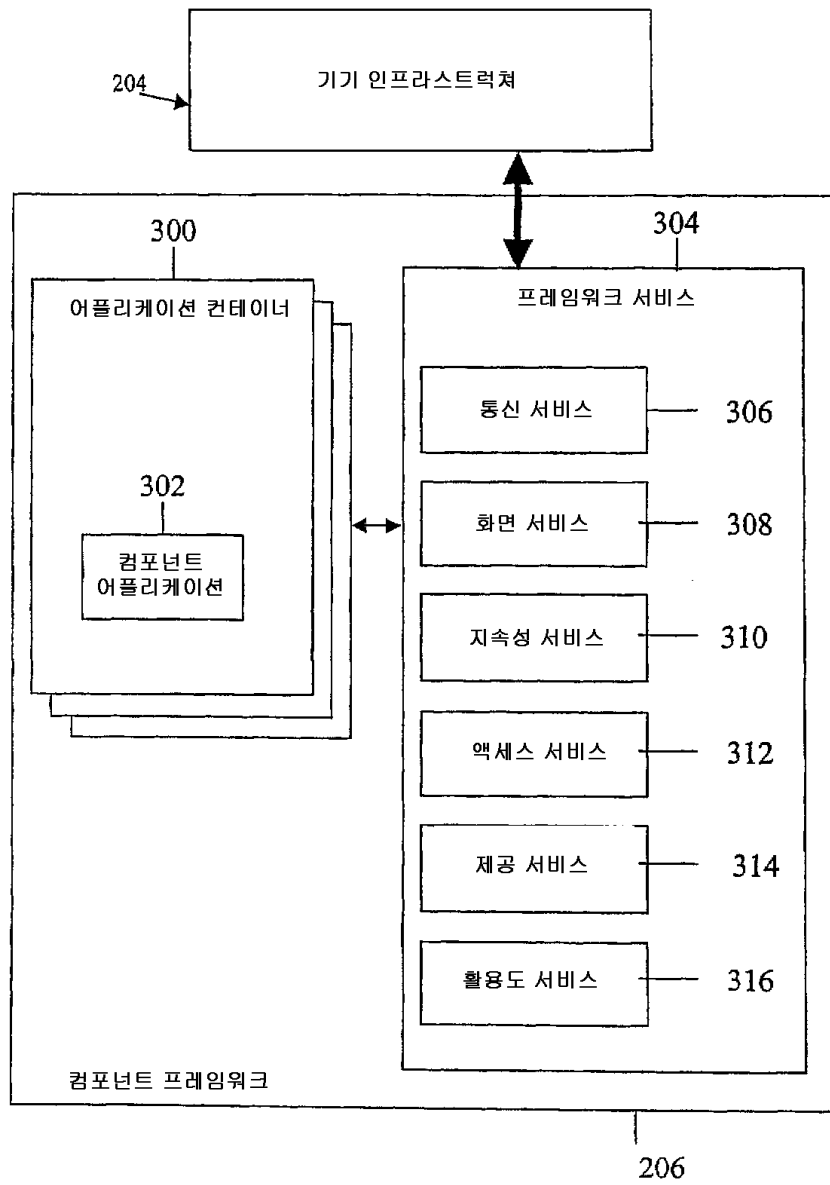
도면2



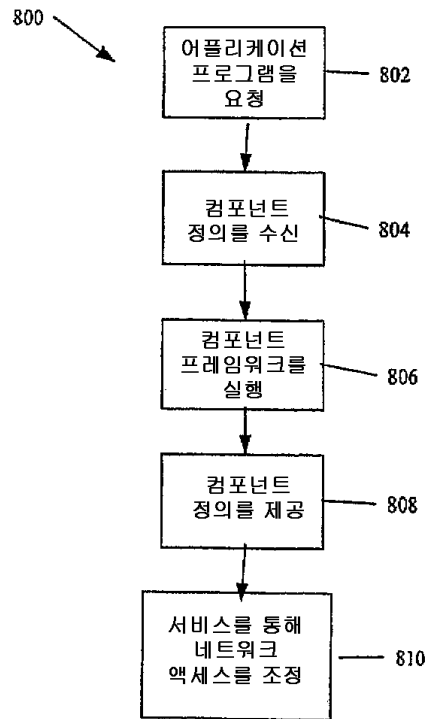
도면3



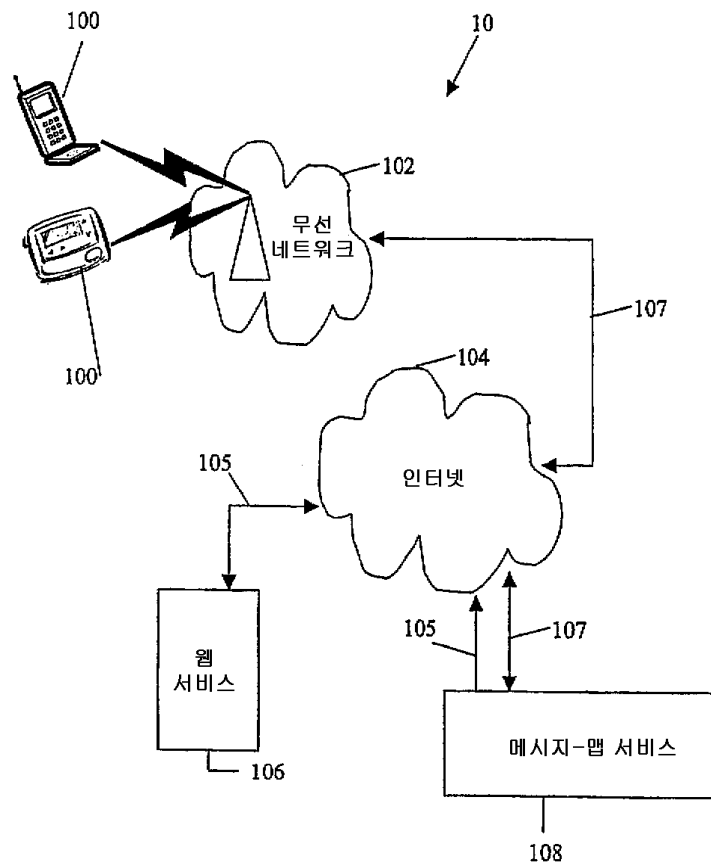
도면4



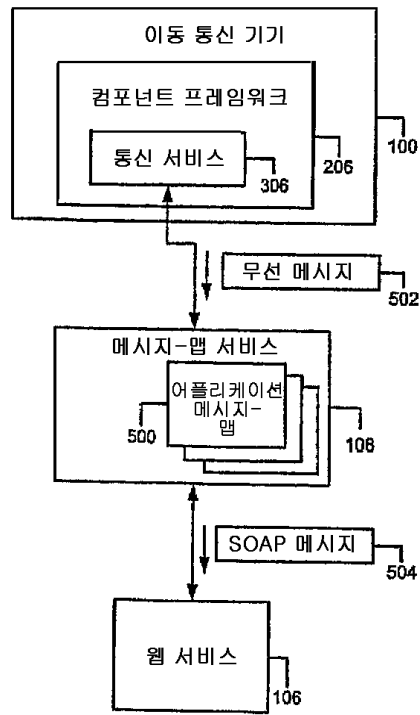
도면5



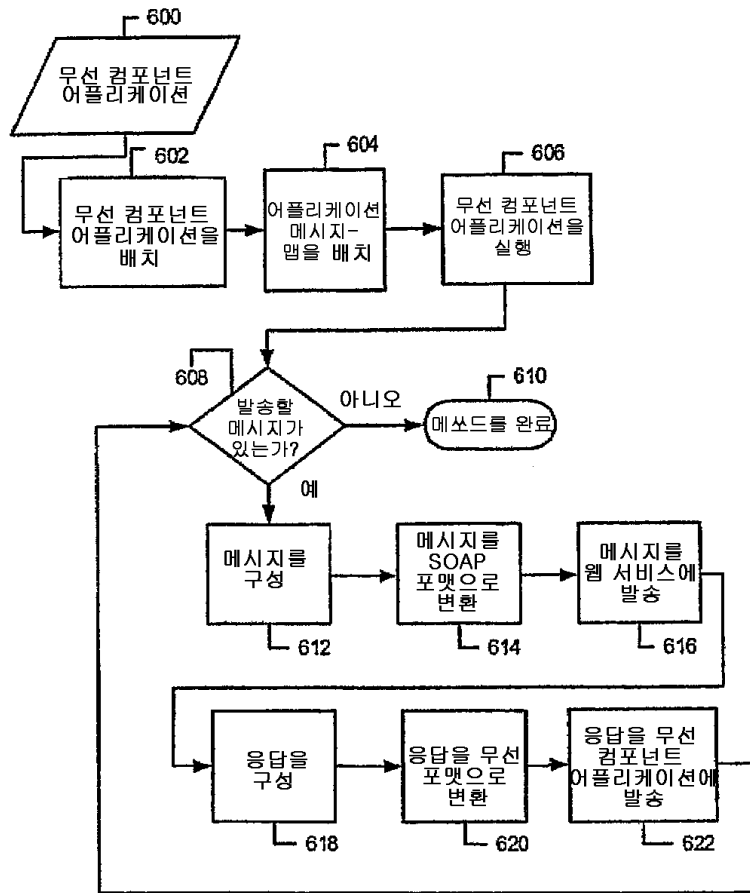
도면6



도면7



도면8



도면9

