



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0108499
 (43) 공개일자 2010년10월07일

- | | |
|---|---|
| <p>(51) Int. Cl.
 G06F 9/46 (2006.01) H04L 12/24 (2006.01)
 H04L 12/56 (2006.01) H04L 12/66 (2006.01)</p> <p>(21) 출원번호 10-2009-7003818</p> <p>(22) 출원일자(국제출원일자) 2009년12월31일
 심사청구일자 없음</p> <p>(85) 번역문제출일자 2009년02월24일</p> <p>(86) 국제출원번호 PCT/KR2008/007848</p> <p>(87) 국제공개번호 WO 2009/084911
 국제공개일자 2009년07월09일</p> <p>(30) 우선권주장
 3161/CHE/2007 2007년12월31일 인도(IN)</p> | <p>(71) 출원인
 삼성전자주식회사
 경기도 수원시 영통구 매탄동 416</p> <p>(72) 발명자
 브러샤벤드라파, 로히쓰
 인도 560004 방갈로르 브이.브이.푸람 디아고날
 로드 플랫 넘버 21, 카나라 बैंक 아파트먼트 #110
 란지트신, 바라 반라신
 인도 560017 방갈로르 무주레스팔라, 칼라파 레이
 아웃, 메인 3알디 #172</p> <p>(74) 대리인
 특허법인 신지</p> |
|---|---|

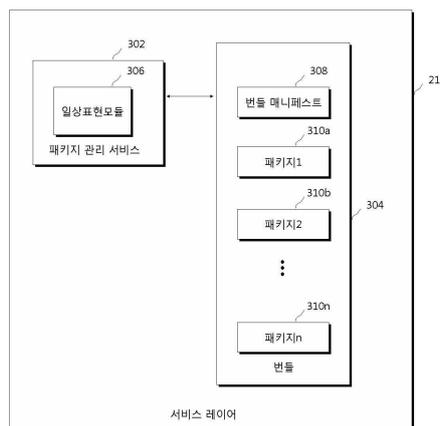
전체 청구항 수 : 총 15 항

(54) 프레임워크에서 패키지들의 공유를 위한 방법 및 시스템

(57) 요약

본 발명의 일양상에 따라, 번들에 의해 단일 엔트리로 공유될 필요가 있는 다수의 패키지들을 구체화하는 방법 및 시스템이 제공된다. 이 방법은 표현을 다수의 패키지들과 연관시키는 단계를 포함한다. 이 표현은 다수의 패키지들을 단일 엔트리로 기술한다. 또한 이 방법은 이 표현을 번들을 기술하는데 포함시키는 단계를 포함한다. 이 번들은 프레임워크내에서 다수의 패키지들의 전개를 가능하게 한다. 더욱이, 이 방법은 번들에 의해 다수의 패키지들을 프레임워크로 익스포트하는 단계를 포함한다. 이 프레임워크는 이 다수의 패키지들의 실행을 위한 환경을 제공한다. 이 디바이스는 표현을 다수의 패키지들과 연관시키도록 구성된 프로세서를 포함한다. 이 표현은 다수의 패키지들을 단일 엔트리로 기술한다. 또한 이 프로세서는 번들을 기술하는 표현을 포함하도록 구성된다. 이 번들은 프레임워크내에서 이 다수의 패키지들의 전개를 가능하게 한다. 이 디바이스는 또한 프레임워크내에서 다수의 패키지들을 익스포트하도록 구성된 패키지 관리 서비스 모듈을 포함한다. 이 프레임워크는 이 다수의 패키지들의 실행을 위한 환경을 제공한다.

대표도 - 도3



특허청구의 범위

청구항 1

프레임워크에서의 패키지들의 공유방법에 있어서,

다수의 패키지들의 단일 엔트리로의 표현을, 상기 다수의 패키지들과 연관시키는 단계;

상기 표현을, 상기 프레임워크에서의 상기 다수의 패키지들의 전개(deployment)를 가능하게 하는 번들의 표현에 포함시키는 단계; 및

상기 다수의 패키지들을, 상기 다수의 패키지들의 실행을 위한 환경을 제공하는 프레임워크로 익스포트하는 단계를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 표현을 분석하는 단계; 및

상기 표현을 다수의 엔트리들로 디코딩하되, 상기 다수의 엔트리들의 각각의 엔트리는 상기 다수의 패키지들에 유일하게 대응되는 단계를 더 포함하는 방법.

청구항 3

제1항에 있어서,

상기 프레임워크에 의해 수신되는 번들을 수신하는 단계; 및

상기 번들을 상기 프레임워크에 인스톨하는 단계를 더 포함하는 방법.

청구항 4

제1항에 있어서,

상기 다수의 패키지들을 상기 프레임워크에 임포트하는 단계를 더 포함하는 방법.

청구항 5

제1항에 있어서,

상기 표현은 일상표현(regular expression)인 방법.

청구항 6

제5항에 있어서,

상기 일상표현은 *, +, [], ? 중 적어도 어느 하나인 것인 방법.

청구항 7

제1항에 있어서,

상기 프레임워크는 오픈 서비스 게이트웨이 이니셔티브 프레임워크인 방법.

청구항 8

다수의 패키지들의 단일 엔트리로의 표현을, 상기 다수의 패키지들과 연관시키고, 상기 표현을, 상기 프레임워크에서의 상기 다수의 패키지들의 전개(deployment)를 가능하게 하는 번들의 표현에 포함시키도록 구성된 프로세서; 및

상기 다수의 패키지들을, 상기 다수의 패키지들의 실행을 위한 환경을 제공하는 프레임워크로 익스포트하도록 구성된 패키지 관리 서비스 모듈을 포함하는 디바이스.

청구항 9

제8항에 있어서, 상기 프로세서는

상기 표현을 분석하고, 상기 표현을 다수의 엔트리들로 디코딩하되, 상기 다수의 엔트리들의 각각의 엔트리는 상기 다수의 패키지들에 유일하게 대응되도록 하는 디바이스.

청구항 10

제8항에 있어서,

상기 표현은 일상표현(regular expression)인 디바이스.

청구항 11

제10항에 있어서,

상기 일상표현은 *, +, [], ? 중 적어도 어느 하나인 것인 디바이스.

청구항 12

제8항에 있어서,

상기 프레임워크는 오픈 서비스 게이트웨이 이니셔티브 프레임워크인 디바이스.

청구항 13

제8항에 있어서,

상기 디바이스는 오픈 서비스 게이트웨이 이니셔티브 디바이스인 디바이스.

청구항 14

도면을 참조하여 기술되고 설명된 바와 같은, 프레임워크에서의 패키지 공유 방법.

청구항 15

도면을 참조하여 실질적으로 기술되고 설명된 바와 같은 디바이스.

명세서

기술분야

[0001] 프레임워크(framework)에서 패키지들(packages)의 공유에 관련된다. 보다 구체적으로는 프레임워크에서 패키지들의 공유를 위한 방법 및 시스템에 관련된다.

배경기술

[0002] 오픈 서비스 게이트웨이 이니셔티브(Open Service Gateway Initiative, OSGi) 구조는 네트워크 디바이스(networked device)들에 대해 표준화되고(standardized), 컴포넌트에 기초한(component-oriented), 계산환경을 제공한다. OSGi 구조는, 예를 들어 모바일 디바이스, 홈 오토메이션, 산업 자동화(industrial automation), 차량(vehicle), 텔레메틱스(telematics), 임베디드 장치(embedded appliance) 등과 같이 다양한 범위에서 넓은 애플리케이션을 가지고 있다.

[0003] OSGi 구조에서의 핵심 컴포넌트(core component)는 OSGi 프레임워크로, 이 OSGi 프레임워크는 OSGi 번들(bundle)에 의해 사용될 수 있는 실행환경을 제공한다. 네트워크 디바이스에 전개되는(deployed) OSGi 서비스 플랫폼(framework)은 네트워크 디바이스에서의 다양한 소프트웨어 컴포넌트들의 라이프사이클(lifecycle)을 조화로운 방법(coordinated manner)에 따라 전개시키고(deploy) 다룰(manage) 수 있도록 해준다.

[0004] 이 OSGi 프레임워크는 번들이라 불리는 다운로드 가능하고 확장 가능한 컴포넌트들의 전개(deployment)를 지원한다. 각각의 번들은 네트워크 디바이스의 엔드 유저들 및 타 번들들에게 서비스들을 제공하는 애플리케이션 단위이다. OSGi 서비스는 얻을 수 있는 하나의 번들로부터 타 번들들로의 오브젝트이다. 다시 말하면 서비스

는, 번들이 타 번들들에게 기능성(functionality)을 제공하는 것을 가능하게 하는 메카니즘이다.

- [0005] 전형적으로 번들은, 자바 아카이브(Jave Archive, JAR) 파일 포맷으로 저장되며, 이 자바 아카이브 파일 포맷은 널리 사용되는 Zip 파일 포맷과 연동된다. OSGi 연동 디바이스는 OSGi 번들들을 다운로드하고 인스톨할 수 있으며, OSGi 번들들이 더 이상 필요하지 않을때는 이를 언인스톨할 수 있다.
- [0006] 이 프레임워크는, 번들들을 동적으로 인스톨하고(installing), 시작시키고(starting), 정지시키고(stoping), 업데이트시키고(updating), 언인스톨하는(uninstalling) 능력을 개발자들에게 제공함으로써, 번들들의 라이프사이클을 다루는데 있어 개발자들에게 유연성을 제공한다. 이 프레임워크는 또한, (조건적인) 허가 관리(permission admin), 이벤트 관리(event admin), 전개 관리(deployment admin) 등과 같은 서로 다른 선택적인 서비스들을 제공하여 이들 번들들을 다루는 것을 도와준다.
- [0007] 번들의 주 내용은 번들의 실행부분인 클래스 파일들이다. 클래스 파일들의 집합은 패키지내에서 그룹화될 수 있으며, 이는 OSGi 프레임워크에서 서로 다른 번들들간에 공유될 수 있다. OSGi 프레임워크는 패키지 관리 서비스(package admin service)라고 알려져 있으며, 이는 번들들간에 패키지-공유를 관리하기 위한 것이다.
- [0008] 이와 같이 번들은 패키지를 타 번들들로 익스포트(export)할 수 있다. 이와 유사하게 번들은 타 번들들로부터 패키지들을 임포트(import)할 수 있다. 번들의 개발자는, 번들이 프레임워크내에서 타 번들들로 익스포트할 수 있는 패키지들을 구체화할 필요가 있다. 개발자는 이들 패키지들을 번들의 매니페스트 파일(mainfest file)내에서 구체화할 수 있다. 이 매니페스트 파일은 번들의 내용을 설명하고 있고, 대응 번들을 인스톨하고 활성화하기 위해 프레임워크에 의해 필요로 되는 정보를 포함한다. 이 매니페스트 파일은 번들의 종속성(dependency)와 같은 다른 정보를 포함할 수 있다.
- [0009] 현재 기술에서, 번들의 개발자는 번들의 매니페스트 파일에서 익스포트될 수 있는 각각의 패키지에 대한 완전한 자격있는(qualified) 이름을 제공할 필요가 있다. 이와 유사하게, 임포트될 필요가 있는 모든 패키지들의 완전한 자격있는 이름들은 매니페스트 파일내에서 구체화될 필요가 있다. 이렇게 구체화하는 것은 비록 적은 수의 패키지들에 대해서는 쉽게 수행될 수 있어도, 번들에 의해 많은 수의 패키지들이 익스포트되거나 임포트될 필요가 있는 경우에는 지루한 일이다.
- [0010] 전문한 관점에서, 번들의 매니페스트 파일에서, 다수의 익스포트/임포트 패키지들을 단일 엔트리(entry)로써 포함할 수 있는 재능(facility)을 제공할 필요가 있다. 개발자는 다수의 엔트리들 대신 단일 엔트리를 사용함으로써 번들에 의해 익스포트되거나 임포트될 필요가 있는 패키지들을 특정할 수 있어야 한다.

발명의 상세한 설명

- [0011] 본 발명의 일양상에 따르면, 본 발명은 OSGi 프레임워크에서 번들들간에 패키지들을 효율적으로 공유하기 위한 기능을 제공하는 것이다.
- [0012] 본 발명의 다른 양상에 따르면, 본 발명은 번들의 매니페스트 파일에서 단일 엔트리를 통해 번들에 의해 익스포트될 필요가 있는 다수의 패키지들을 구체화하는 프로세스를 촉진하는 것이다.
- [0013] 본 발명의 또 다른 양상에 따르면, 본 발명은 번들의 매니페스트 파일에서 단일 엔트리를 통해 번들에 의해 임포트될 필요가 있는 다수의 패키지들을 구체화하는 프로세스를 촉진하는 것이다.
- [0014] 이에 따라 본 발명의 일실시예에서, 프레임워크에서 패키지들을 공유하는 방법이 제공된다. 이 방법은 표현(expression)을 복수의 패키지들과 연관시키는(associating) 단계를 포함한다. 이때 이 표현은 이들 다수의 패키지들을 단일 엔트리로써 기술한다. 또한 본 방법은, 번들의 기술(description) 내에서 이 표현을 포함시키는 단계를 포함한다. 이 번들은 프레임워크내에서 복수의 패키지들의 전개(deployment)를 가능하도록 한다.
- [0015] 더 나아가, 이 방법은 이 번들에 의해 다수의 패키지들을 익스포트 하는 단계를 포함한다. 이 프레임워크는 이 다수의 패키지들의 실행을 위한 환경을 제공한다. 본 실시예에서 이 방법은, 이 표현을 해석하고 이 표현을 다수의 엔트리들로 디코딩하는 단계를 포함한다. 다수의 엔트리들의 각 엔트리는 다수의 패키지들의 패키지에 유일하게(unique) 대응된다.
- [0016] 다른 실시예에서 장치가 제공된다. 이 장치는 표현(expression)을 다수의 패키지들에 연관시키도록 구성된 프로세서를 포함한다. 이 표현은 이들 다수의 패키지들을 단일 엔트리로써 기술한다. 그리고 이 프로세서는, 번들의 기술(description) 내에서 이 표현을 포함하도록 구성된다. 이 번들은 프레임워크내에서 복수의 패키지들의 전개(deployment)를 가능하게 한다.

[0017] 이 장치는 또한, 프레임워크내에서 다수의 패키지들을 익스포트하도록 구성된 패키지-관리-서비스 모듈(package-admin-service module)을 포함한다. 이 프레임워크는 다수의 패키지들의 실행을 위한 환경을 제공한다. 일실시예에서, 이 프로세서는 또한 이 표현을 분석하여 이를 다수의 엔트리들로 디코딩하도록 구성된다. 다수의 엔트리들의 각 엔트리는 다수의 패키지들의 패키지에 유일하게(unique) 대응된다.

[0018] 본 발명의 다양한 구현에는 이하 수반되는 도면과 관련하여 기술될 것이며, 설명하기 위해 제공될 것이며, 본 발명에 한정되는 것은 아니도록 설명하기 위해 제공될 것이다. 여기서 유사한 지정(designation)은 유사한 엘레먼트를 의미한다.

실시예

[0025] 본 발명의 다양한 실시예는, 공유될 필요가 있는 복수의 패키지들을 번들의 매니페스트 파일에서 단일 엔트리의 형태로 구체화하는 프로세스를 촉진한다. 프레임워크에서 다른 번들들과 공유될 필요가 있는 패키지들의 셋을 구체화하기 위해, 일상의 표현(regular expression)이 매니페스트 파일에서 사용된다. 이 일상의 표현은 분석되고, 공유될 필요가 있는 대응 패키지들이 결정된다. 이에 따라, 대응 패키지들이 익스포트 및/또는 임포트된다.

[0026] 도 1은 본 발명의 다양한 실시예가 실행될 수 있는 오픈 서비스 게이트웨이 이니셔티브(OSGi) 네트워크의 블록도이다.

[0027] OSGi 네트워크(110)는 OSGi 관리 시스템(102)과, 이하에서 OSGi 디바이스들(104)로 인용되는 다수의 OSGi 디바이스들(104a, 104b, 104c)을 포함한다. OSGi 관리 시스템(102)은 OSGi 디바이스들(104)을 원격으로 관리한다.

[0028] OSGi 디바이스들(104)는 모두 OSGi 연동의 네트워크 디바이스들일 수 있다. OSGi 디바이스들(104)의 일례로 모바일 폰, PDA(Personal Digital Assistant), 랩톱(loptop), 데스크탑 컴퓨터, 임베디드 장비(embedded appliance), 주거 게이트웨이(residential gateway), 산업용 컴퓨터(industrial computer), 고사양 서버(high-end server), 소비자 가전(consumer electronic device)을 포함하나, 이에 한정되는 것은 아니다.

[0029] OSGi 디바이스들(104)은 개발자 또는 공급자(vendor)로부터 번들을 다운로드 받을 수 있다. OSGi 관리 시스템(102)은, 인스톨되거나, 시작되거나, 업데이트되거나, 정지되거나 또는 언인스톨되는 번들의 라이프사이클을 관리한다. 따라서, 디바이스의 다양한 기능들과 특징들이 관리된다. OSGi 관리 시스템(102)은 적절한 원격 관리 프로토콜을 사용함으로써 이 디바이스들을 관리한다. 원격 관리 프로토콜의 일례로, SNMP(Simple Network Management Protocol), CMISE(Common Management Information Service Element), 텔넷(Telnet), CIM(Common Information Model), OMADM(Open Mobile Alliance Device Management) 등을 포함한다.

[0030] 비록, 전술한 실시예가 OSGi 네트워크를 가지고 설명되었으나, 본 발명에 따라 다른 적절한 네트워크를 가지고도 실행되거나 설명될 수 있음은 당업자에게 자명하다.

[0031] 도 2는 본 발명의 일실시예에 따른 오픈 서비스 게이트웨이 이니셔티브(OSGi) 구조(200)를 묘사하는 블록도이다.

[0032] OSGi 구조(200)는 운영체제(Operation System, OS) 및 하드웨어 레이어(202), 자바 가상 머신(Java Virtual Machine, JVM) 레이어(204), 클래스-로딩 레이어(class-loading layer)(206), 라이프사이클 레이어(life cycle layer)(208), 서비스 레지스트리 레이어(service registry layer)(210), 서비스 레이어(service layer)(212), 애플리케이션 레이어(214) 및 보안 레이어(security layer)(216)를 포함한다.

[0033] OS 및 하드웨어 레이어(202)는 가장 하부 레이어이며, OSGi 구조에서 실행환경(JVM을 위한)을 제공한다. 운영체제(OS)는 예를 들어, 유닉스(Unix)나 윈도우즈(Windows)일 수 있다. 운영체제는 하드웨어 플랫폼 위에서 돌아간다. JVM 레이어(204)는 OSGi 프레임워크의 동작을 위한 런타임(runtime) 환경을 제공한다. JVM은 Java 프로그래밍 언어를 지원하는 런타임 엔진이며, 이는 캘리포니아 산타클라라의 썬마이크로시스템즈(SUN Microsystems)의 제품이다. 일반적인 방법에서, 단일 자바 애플리케이션은 하나의 JVM 상에서 수행된다. 그러나 OSGi 프레임워크가, 번들이 OSGi 규격에 따라 협업적으로 행동하는 협업환경을 제공하기 때문에, OSGi 프레임워크에서 몇개의 번들이 동일한 JVM 상에서 수행될 수 있다.

[0034] 클래스-로딩 레이어(206)는 번들에 대한 클래스-로딩 정책을 정의한다. 번들의 주 내용은 그 클래스 파일들이다. 이들 클래스 파일들은 번들의 실행가능부분이다. 클래스들의 셋은 패키지에서 그룹화될 수 있다. 게다가, 이 패키지들은 OSGi 프레임워크에서 서로 다른 번들들간에 공유될 수 있다. 클래스-로딩 레이어(206)의 상

부에는 라이프사이클 레이어(208)라고 불리우는 레이어가 있다.

- [0035] 라이프사이클 레이어(208)는 OSGi 프레임워크에서 번들들의 라이프사이클을 관리하기 위한 애플리케이션 프로그램 인터페이스(Application Program Interface, API)를 제공한다. 이 번들들은 동적으로 인스톨, 시작, 정지, 업데이트 및 언인스톨 될 수 있다. 라이프사이클 동작은 악의적인 번들들에 의한 공격을 방지하기 위해 보안구조를 가지고 보호된다. 라이프사이클 레이어 다음에는 서비스 레지스트리 레이어(210)라고 불리우는 레이어가 있다.
- [0036] 서비스 레지스트리 레이어(210)는 서비스들을 번들들간 공유하는 것을 가능하게 한다. OSGi 프레임워크에서, 번들이 인스톨될 때, 번들내에서의 하나 또는 그 이상의 서비스들이 서비스 레지스트리에 등록되어, 다른 번들들이 하나 또는 그 이상의 서비스들을 사용할 수 있다. OSGi 서비스는 하나의 번들로부터 다른 번들로도 가용할 수 있도록 만들어진 오브젝트이다. 이와 같이, 서비스 레지스트리의 도움으로, 번들들은 오브젝트들을 등록할 수 있고, 오브젝트들을 서치할 수 있으며, 새로운 서비스들이 등록되거나 기존의 서비스들이 미등록되어 있을 때 통지(notification)를 수신하다. 서비스 레지스트리 레이어(210) 다음에는 서비스 레이어(212)라고 불리우는 레이어가 있다.
- [0037] 서비스 레이어(212)는 OSGi 프레임워크에서 허락관리 서비스(permission admin service), 조건부 허락관리 서비스(conditional permission admin service), 패키지 관리 서비스(package admin service), 전개관리 서비스(deployment admin service) 등과 같은 다양한 표준화된 선택적인 서비스들을 제공한다. 이러한 서비스들은 OSGi 프레임워크에서, 예를 들어 번들들의 인스톨레이션, 이 번들들과 연관된 허락관리, 번들들의 종속성(dependency) 계산 등과 같은 다양한 기능들을 가능하게 한다.
- [0038] 서비스 레이어(212) 다음에는 애플리케이션 레이어(214)라고 불리우는 레이어가 있다. 애플리케이션 레이어(214)는 다른 번들들에게 서비스를 제공하며, 이는 OSGi 프레임워크에서 하나 또는 그 이상의 서비스들을 구현할 수 있다. 추가적으로 OSGi 연동 번들들은 매니페스트 파일을 포함할 수 있으며, 이 매니페스트 파일은 번들들의 내용을 기술하며, 프레임워크에 의해 인스톨되고 대응번들을 시작시킬 필요가 있는 정보를 포함한다. 애플리케이션 레이어(214) 다음에는 보안 레이어(216)라고 불리우는 레이어가 있다.
- [0039] 보안 레이어(216)는 서로 다른 소스들로부터의 애플리케이션들이 아무 문제없이 OSGi 프레임워크에서 돌아갈 수 있도록 하는 완전한 보안모델을 제공한다. 각각의 번들은 허락의 셋(set of permission)을 가지고 있으며, 이는 동적으로 변화될 수 있다. 일단 설정되면 새로운 허락(permission)은 즉각적으로 유효하게 된다. 번들들간의 패키지 공유를 통한 악의적인 번들들에 의한 공격을 방지하기 위하여, OSGi 규격은 다른 번들로의 익스포트와 임포트를 제한하는 패키지 허락을 담고 있다. 더 나아가 서비스 허락(service permission)은 특정 서비스들이 오직 권한있는 번들(authorized bundle)들에 의해 사용될 수 있음을 확인하는데 사용된다.
- [0040] 도 3은 본 발명의 일실시예에 따라, OSGi 프레임워크의 서비스 레이어(212)에서 전개되는 패키지 관리 서비스(package admin service)(302)를 묘사하는 블록도이다.
- [0041] 서비스 레이어(212)는 패키지 관리 서비스(302)와 번들(304)을 포함한다. 패키지 관리 서비스(302)는 일상 표현 모듈(regular expression module)(306)을 포함한다. 도 2와 관련하여 설명된 바와 같이, 서비스 레이어(212)는 허락 관리 서비스, 이벤트 관리 서비스 등등(도면에서 도시하지 않음)과 같은 다른 서비스들을 포함한다.
- [0042] 번들(304)은 번들 매니페스트(308)와, '패키지1(310a)', '패키지2(310b)', '패키지n(310n)' 와 같은 다수의 패키지들을 포함한다. 번들 매니페스트(308)은 번들(304)와 연관된 매니페스트 파일이다.
- [0043] OSGi 프레임워크에서, 번들에 포함되는 하나 또는 그 이상의 서비스들은 익스포트-임포트 메카니즘을 통해 다른 번들들끼리 공유될 수 있다. 제1번들은 패키지에 담긴 클래스들을 제2번들이 액세스할 수 있게 하기 위해, 패키지를 익스포트한다. 다음에는 제2번들은, 필요할 때마다, 패키지에 담긴 클래스들을 액세스하기 위해 패키지를 임포트한다.
- [0044] 번들 매니페스트(308)는, 번들(304)을 인스톨하고 시작시키기 위해 OSGi 프레임워크에서 필요로 하는 정보뿐만 아니라 번들의 내용과 관련된 정보를 포함한다. 번들 매니페스트(308)는 또한 패키지의 익스포트와 임포트와 관련된 정보를 포함한다. 본 발명의 실시예에 따라 번들 매니페스트(308)는, 일상적인 표현의 구문(regular expression syntax)을 갖는 하나 또는 그 이상의 익스포트 명령문(export statement)과 하나 또는 그 이상의 임포트 명령문(import statement)을 포함한다. 더 나아가 일상적인 표현은 보통(ordinary) 문자 그리고 특수문자(special character)를 모두 포함한다. 특수문자는 *, ?, [], + 등을 포함한다. 이러한 특수문자는 번들

의 매니페스트 파일에서, 패키지 이름과 관련하여 사용될 수 있다. 각각의 특수문자는 미리 정해진 규칙에 따라 결정되어, 일상표현(regular expression)은 스트링의 셋(set of string)을 나타낸다.

- [0045] 패키지 관리 서비스(package admin service)(302)는 번들들의 실제 패키지-공유 상태에 대한 정보를 제공한다. 번들들간의 패키지 공유는 번들들간의 종속성 셋(set of dependency)을 생성한다. 패키지 관리 서비스(2302)는 OSGi 프레임워크에 의해 제공되는 동적 환경에서, 번들들의 종속성을 관리한다. 번들의 상태가 변할때마다, 즉 번들이 인스톨되거나 업데이트되거나 언인스톨되거나 할때마다, 패키지 관리 서비스(302)는 번들들간의 종속성을 재계산하고, 번들들간의 공유를 확립한다(establish).
- [0046] 도 3을 참조하면, 번들(304)이 그 상태를 변화할 때, 예를 들어 번들(304)이 인스톨되거나 업데이트되거나 언인스톨될때, 패키지 관리 서비스(302)는 OSGi 프레임워크에서 생성되는 통지를 수신한다. 패키지 관리 서비스(302)는 번들 매니페스트(308)로부터의 패키지 공유와 관련된 정보를 탐색한다(retrieve). 패키지 관리 서비스(302)는 매니페스트에 포함된 하나 또는 그 이상의 импорт 명령문(import statement)과 하나 또는 그 이상의 익스포트 명령문(export statement)을 실행한다. 패키지 관리 서비스(302)는 번들 매니페스트(308)로부터 수신된 일상표현(regular expression)을 일상표현모듈(regular expression module)(306)로 제공한다.
- [0047] 일실시예에서, 일상표현모듈(306)은 일상표현에 의해 구체화된 다수의 패키지들을 결정하기 위하여 일상표현을 분석하고 디코딩한다. 그리고 나서 이 일상표현과 매치되는 패키지들은 이에 따라 익스포트되거나 импорт될 수 있다. 하나 또는 그 이상의 특수문자를 포함하는 일상표현은, 미리 정해진 규칙에 따라 다수의 패키지들로 분해될(resolve) 수 있다. 다수의 패키지들을 구체화하기 위한 일상표현의 사용이, 후술하는 실시예와 연관지어 설명된다.
- [0048] 일예에 따르면, 번들의 개발자는 익스포트될 수 있는 com.abc.package1, com.abc.package2 및 com.abc.package3 라는 세개의 패키지들을 제공한다. 이 경우 세개의 패키지들에 대한 개별 익스포트 명령문을 제공하거나, 각각의 패키지의 완전 자격있는(fully qualified) 이름을 특정하는 대신, 개발자는 단일 익스포트 명령문 "com.abc.package*"를 번들의 매니페스트 파일에 포함시킬 수 있다. 패키지 관리 서비스(302)가 번들의 매니페스트 파일에서 이 일상표현을 발견하면, 번들은 이 일상표현을 일상표현모듈(306)로 전달한다. 일상표현모듈(306)은 이 일상표현 "com.abc.package*" 를 세개의 패키지, 즉 com.abc.package1, com.abc.package2, com.abc.package3로 분해한다. 이들 세개의 패키지들은 프레임워크에서의 다른 번들들로 익스포트된다.
- [0049] 사용될 수 있는 일상표현의 일예는, "com.abc.package*", "com.", "*" 등을 포함하나 이에 한정되는 것은 아니다. "com.abc.package*" 표현은 패키지 이름, 예를 들어 com.abc.package1, com.abc.package2 및 com.abc.package3.example 등과 같은 패키지 이름과 함께 시작되는 번들에 포함된 모든 패키지들을 지칭할 수 있다. "com.*" 표현은 com 이하에 순환적으로(recursively) 포함되는 모든 패키지들을 지칭할 수 있으며, "*" 표현은 번들에 포함되는 모든 패키지들을 지칭할 수 있다.
- [0050] 다른 일예에 따르면, 개발자는 예를 들어 번들에서의 com.abc.package1, com.abc.package2, com.abc.package3, com.abc.package4, com.abc.package.example1, com.abc.package.example2 및 com.abc.package.example3 와 같은 다수의 패키지들을 제공한다. 일실시예에서, 만일 개발자가 com.abc.package라는 이름으로 시작되는 모든 패키지들을 익스포트하기를 원한다면, 개발자는 단일 익스포트 명령문 "com.abc.package"를 번들의 매니페스트 파일에 포함시킬 수 있다.
- [0051] 이 경우, 위에서 언급된 모든 패키지들은 익스포트될 것이다. 다른 일실시예에서, 만일 개발자가 오직 com.abc.package1, com.abc.package2, com.abc.package3 및 com.abc.package4 라는 패키지들만을 익스포트하기를 원한다면, 단일 익스포트 명령문 "com.abc.package?"를 포함시킬 수 있다. 또 다른 일실시예에서, 만일 개발자가 오직 com.abc.package1 및 com.abc.package2 만을 익스포트하기를 원한다면, 단일 익스포트 명령문 "com.abc.package[1,2]"를 포함시킬 수 있다.
- [0052] 유사하게, 번들에 의해 임포트될 필요가 있는 패키지들을 구체화하기 위해, 단일 импорт 명령문도 번들의 매니페스트 파일에 포함될 수 있다. 결과적으로, 각 개개의 패키지에 대한 완전한 이름을 제공하는 대신, 다수의 패키지들은 패키지들의 셋으로부터 선택적으로 구체화될 수 있다. 다른 적절한 일상표현이, 번들에 의해 익스포트되거나 임포트될 필요가 있는 패키지들을 구체화하기 위해 사용될 수 있다.
- [0053] 도 4는 본 발명의 일실시예에 따라, 프레임워크에서 패키지들을 공유하는 것을 묘사하는 플로우차트이다.
- [0054] 프레임워크의 예들로 모바일 정보 프로파일(Mobile Information Profile, MIDP), 프레임워크 및 Xlets 프레임워크를 포함하지만 이에 한정되는 것은 아니다. 패키지들을 공유하는 방법은 도 4에 묘사된 바와 같은 얼마간

의 패키지들을 포함한다.

- [0055] 단계 402에서, 일상표현이 번들에 의해 제공되는 다수의 패키지들과 연관된다. 여기에서 사용된 바와 같이 패키지는, OSGi 프레임워크에서 다른 번들에 의해 사용되는 번들에 의해 익스포트 될 수 있는 패키지들을 지칭한다. 일반적으로 일상표현은 표현과 매치되는 스트링의 셋을 구체화한다. 더 나아가 일상표현은, 보통문자 및 특수문자 둘 다 사용한다. 특수문자는 *, ?, [], + 등을 포함한다. 이러한 특수문자는 번들의 매니페스트 파일에서의 패키지 이름들과 연관되어 사용될 수 있다. 각각의 특수문자는 미리 정해진 규칙에 따라 분해될 수 있으며, 결과적으로 일상표현을 스트링의 셋을 구체화한다. 따라서, 하나 또는 그 이상의 특수문자를 포함하는 일상표현은, 미리 정해진 규칙에 따라 다수의 패키지들로 분해될 수 있다.
- [0056] 단계 404에서 이 표현은, 번들의 매니페스트 파일에 포함된다. 다양한 구현예에 따라, 하나 또는 그 이상의 익스포트 명령문들과 하나 또는 그 이상의 임포트 명령문들은 일상적인 표현구문에 따라 구체화된다. 단계 406에서, 이 일상표현과 매치되는 패키지들은 다른 번들과 공유된다. 예를 들어, 일상표현과 매치되는 패키지들은 익스포트되거나 임포트될 수 있다.
- [0057] 도 5는 본 발명의 다른 일실시예에 따라, 프레임워크에서 패키지들을 공유하는 것을 묘사하는 플로우차트이다.
- [0058] 단계 502에서, 번들의 상태 변화에 관한 통지가 수신된다. 일례에 있어서, 번들은 이 통지를 수신할 수 있다. 번들은 OSGi 프레임워크에서 인스톨되거나 업데이트되거나 언인스톨될 수 있다. 이러한 이벤트들 각각은 OSGi 프레임워크에서 통지를 생성한다.
- [0059] 번들은, 번들을 인스톨하고 시작시키기 위한 OSGi 프레임워크에 의해 필요로 하는 정보뿐만 아니라, 번들의 내용과 관련된 정보를 포함하는 번들 매니페스트를 포함한다. 번들 매니페스트는 또한, 패키지들의 익스포트 및 임포트와 관련된 정보를 포함한다. 본 발명의 일실시예에 따라, 번들 매니페스트는 일상표현구문에 따른 하나 또는 그 이상의 익스포트 명령문을 포함한다. 일반적으로, 일상표현은 그 표현과 매치되는 스트링 셋을 구체화한다. 더 나아가 이 일상표현은, 보통문자 또는 특수문자들을 포함한다. 특수문자는 *, ?, [], + 등이 될 수 있다. 이들 특수문자들은 번들의 매니페스트 파일에서 패키지의 이름과 연관되어 사용될 수 있다. 각각의 특수문자는 미리 정해진 규칙에 따라 정해질 수 있고, 결과적으로 일상표현은 스트링의 셋을 구체화한다.
- [0060] 단계 504에서 번들의 번들 매니페스트가 탐색된다. 패키지들의 공유와 관련된 정보가 번들 매니페스트로부터 탐색된다. 단계 506에서, 매니페스트에 포함된 하나 또는 그 이상의 임포트 명령문들과 하나 또는 그 이상의 익스포트 명령문들이 실행된다. 하나 또는 그 이상의 임포트 명령문들과 하나 또는 그 이상의 익스포트 명령문들은 일상표현구문으로 될 수 있다. 번들 매니페스트로부터 수신된 일상표현은, 일상표현에 의해 구체화된 다수의 패키지들을 결정하기 위하여 분석되고 디코딩된다. 그리고 나서 일상표현과 매치되는 패키지들은 이에 따라 익스포트되고 임포트된다. 하나 또는 그 이상의 특수문자들을 포함하는 일상표현은, 미리 정해진 규칙에 따라 다수의 패키지들로 분해된다. 단계 508에서, 단계 506에서 결정된 패키지들은 OSGi 프레임워크에서 공유된다.
- [0061] 도 6은 본 발명의 일실시예에 따른 디바이스를 묘사하는 블록도이다.
- [0062] 디바이스(600)는 프로세서(602) 및 패키지 관리 서비스 모듈(package-admin-srevice module)(604)을 포함한다. 디바이스(600)의 예로, 오픈 서비스 게이트웨이 이니셔티브(Open Service Gateway Initiative, OSGi) 디바이스를 포함하지만, 이에 한정되는 것은 아니다. 프로세서(602)는 표현을 다수의 패키지들과 연관시키도록 구성된다. 이 표현은 다수의 패키지들을 단일 엔트리로 기술한다. 이 표현의 일례로, 일상표현을 포함하지만 이에 한정되는 것은 아니다.
- [0063] 일상표현은 '*', '+', '[' alc '?'를 포함하나 이에 한정되는 것은 아니다. 프로세서(602)는 또한 번들을 기술하는데 있어서 이 표현을 포함하도록 구성된다. 프레임워크의 일례로, 오픈 서비스 게이트웨이 이니셔티브(OSGi) 프레임워크를 포함하나 이에 한정되는 것은 아니다. 이 번들은 프레임워크에서 다수의 패키지들의 전개(deployment)를 가능하도록 한다. 패키지 관리 서비스 모듈(604)은 프레임워크에서 다수의 패키지들을 익스포트하도록 구성된다. 프레임워크는 다수의 패키지들의 실행을 위한 환경을 제공한다. 일례에 있어서, 프로세서(602)는 또한 이 표현을 분석하고 다수의 엔트리들로 디코딩하도록 구성된다. 다수의 엔트리들의 각각의 엔트리는 다수의 패키지들의 패키지에 유일하게 대응된다.
- [0064] 비록 본 발명의 다양한 구현예가 일상표현과 OSGi 프레임워크의 도움을 받아 설명되었으나, 다른 적절한 표현 및/또는 프레임워크를 사용하여 수행되거나 설명될 수 있음은 당업자에게 자명하다.

[0065] 전술한 본 발명의 다양한 구현에는 다음과 같은 장점이 있다. 본 발명은 임포트되거나 익스포트될 필요가 있는 각각의 패키지 이름에 대해 완전한 이름을 주어야 할 필요가 없어진다. 이에 따라 일상표현이 다수의 익스포트/임포트 패키지들을 포함하도록 사용될 수 있기 때문에, 개발자의 작업이 촉진될 수 있다. 더 나아가 번들의 프로세싱이 더욱 효과적으로 된다.

[0066] 이제까지 본 발명의 다양한 실시예가 기술되고 설명되었으나, 이들 실시예에 제한되지 않음은 당업자에게 자명할 것이다. 본 발명의 참뜻과 범위는 별론으로 하고, 본 발명의 다수의 개량(modification), 변경(change), 변이(variation), 대체(substitution)와 균등물(equivalents)은, 클레임에서 기술된 바와 같이 당업자에게 자명할 것이다.

산업상 이용 가능성

[0067] 본 발명은 오픈 서비스 게이트웨이 이니셔티브(OSGi) 규격에 관련되어 있는 소프트웨어에 이용될 수 있다.

도면의 간단한 설명

[0019] 도 1은 본 발명의 다양한 실시예가 실행될 수 있는 오픈 서비스 게이트웨이 이니셔티브(OSGi) 네트워크의 블록도,

[0020] 도 2는 본 발명의 일실시예에 따른 오픈 서비스 게이트웨이 이니셔티브(OSGi) 구조를 묘사하는 블록도,

[0021] 도 3은 본 발명의 일실시예에 따라, OSGi 프레임워크의 서비스 레이어에서 전개되는 패키지 관리 서비스(package admin service)를 묘사하는 블록도,

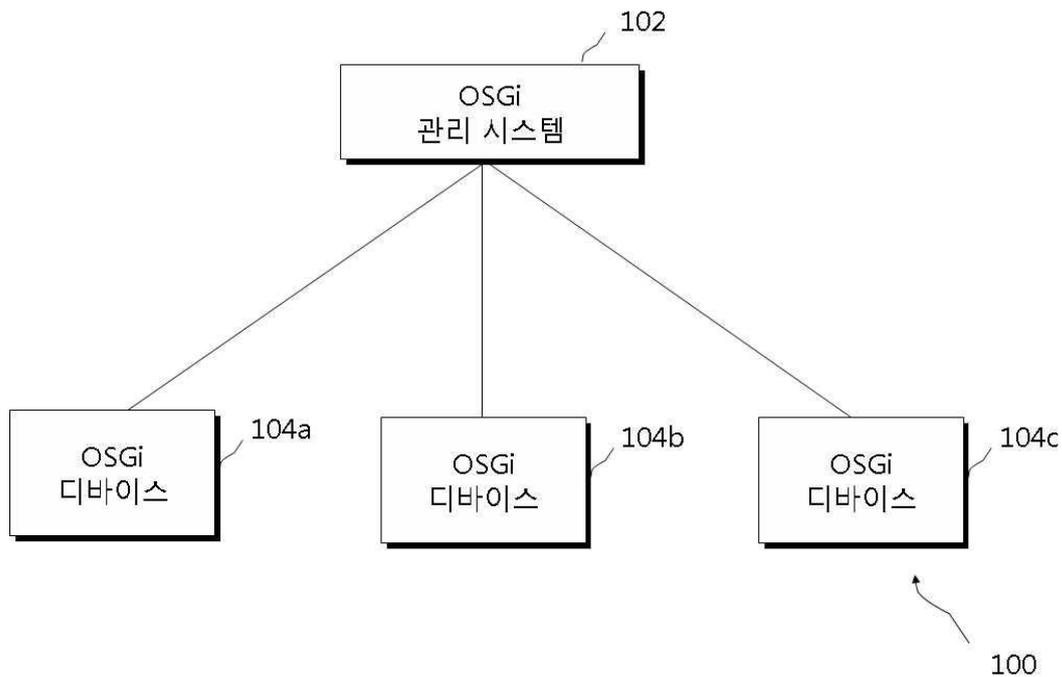
[0022] 도 4는 본 발명의 일실시예에 따라, 프레임워크에서 패키지들을 공유하는 것을 묘사하는 플로우차트,

[0023] 도 5는 본 발명의 다른 일실시예에 따라, 프레임워크에서 패키지들을 공유하는 것을 묘사하는 플로우차트,

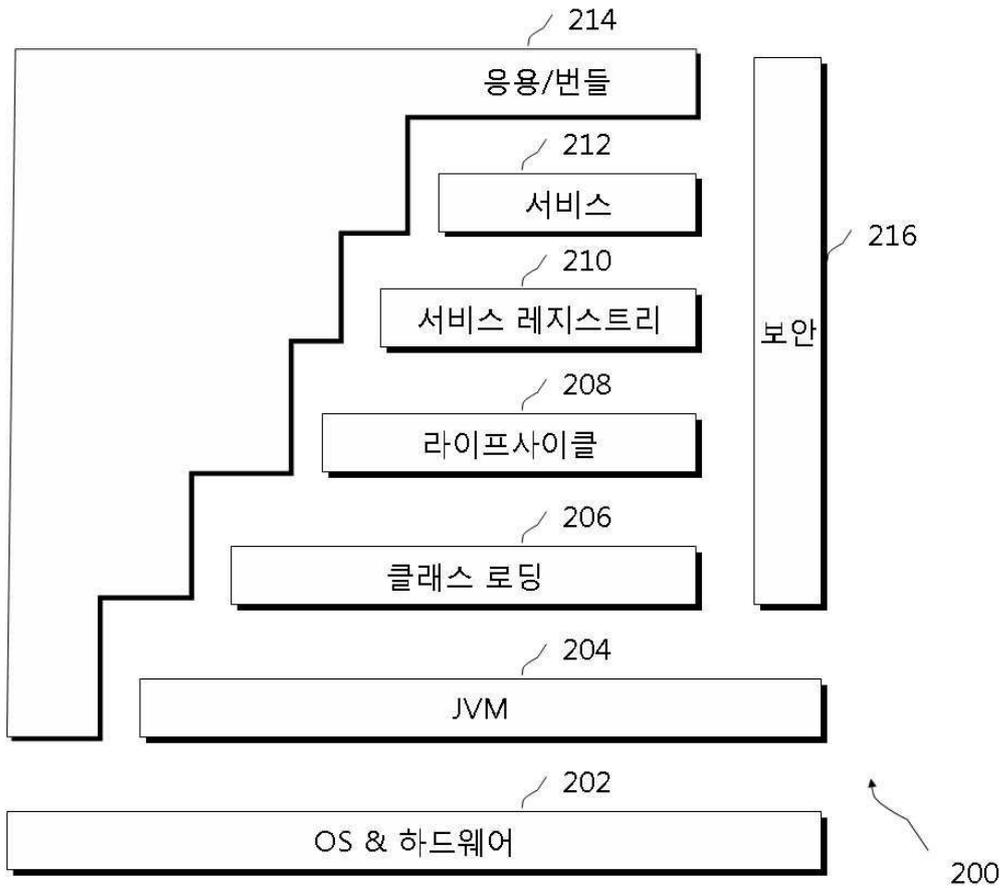
[0024] 도 6은 본 발명의 일실시예에 따른 디바이스를 묘사하는 블록도이다.

도면

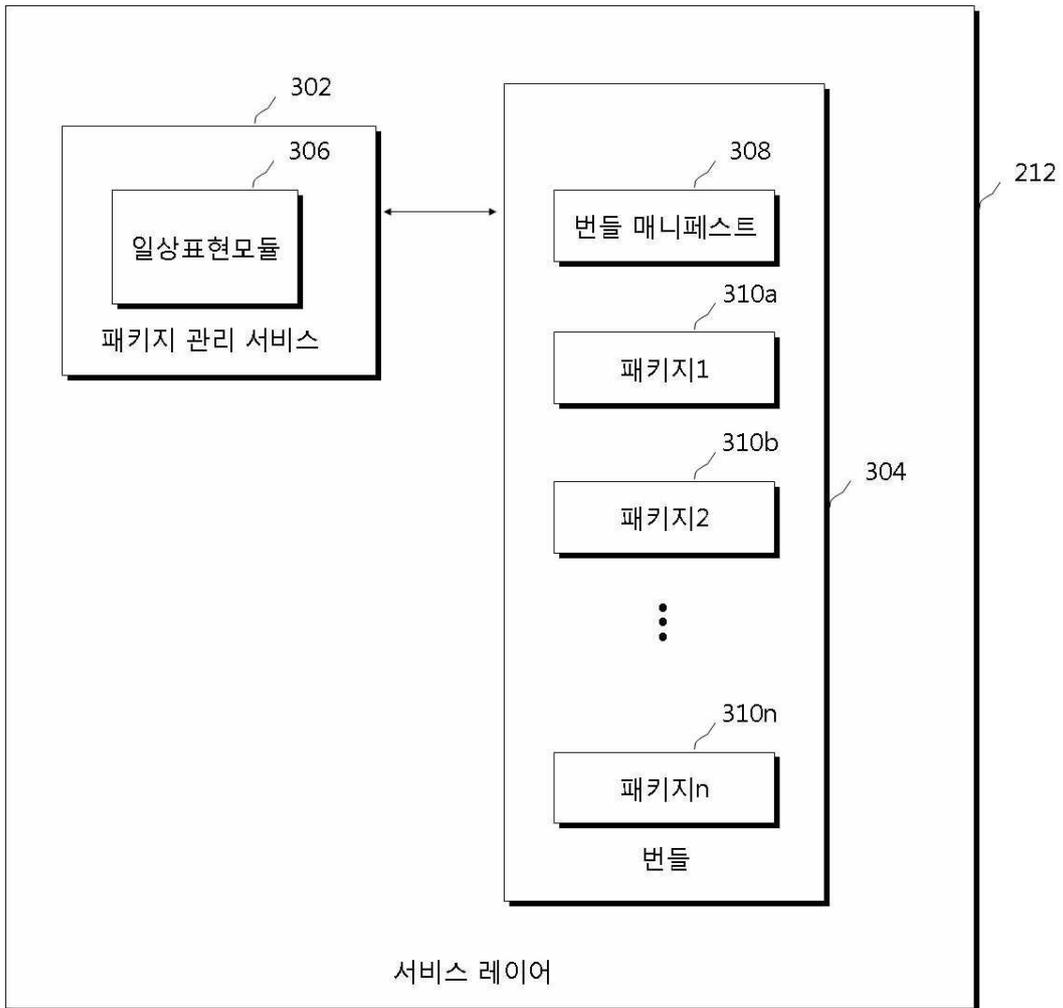
도면1



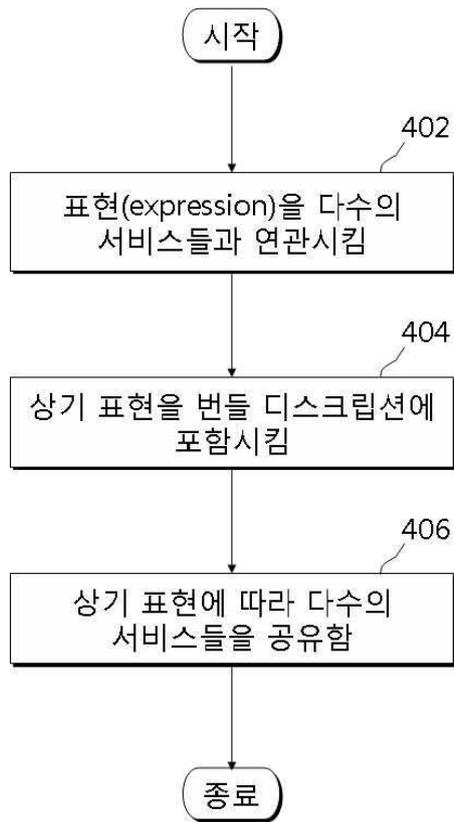
도면2



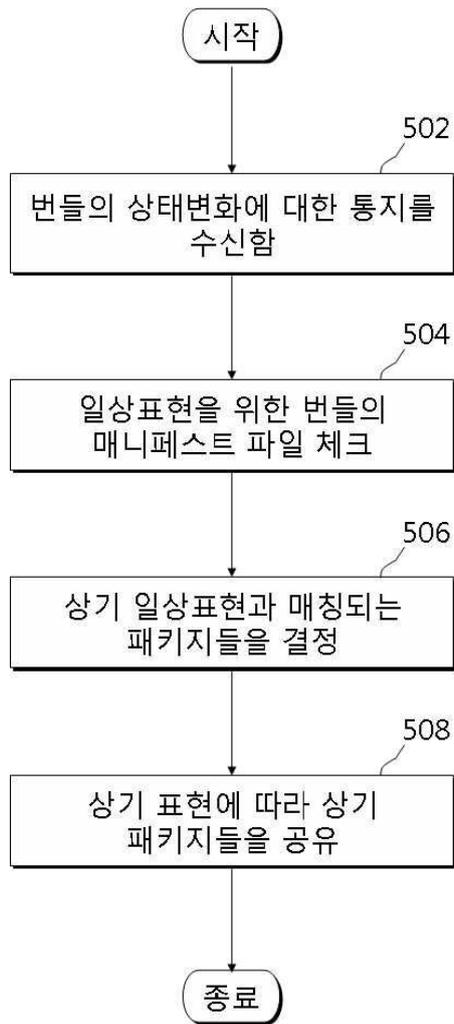
도면3



도면4



도면5



도면6

