



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2016년03월10일

(11) 등록번호 10-1602100

(24) 등록일자 2016년03월03일

(51) 국제특허분류(Int. Cl.)

H04L 12/24 (2006.01)

(52) CPC특허분류

H04L 41/02 (2013.01)

H04L 51/36 (2013.01)

(21) 출원번호 10-2015-0016511

(22) 출원일자 2015년02월03일

심사청구일자 2015년02월03일

(56) 선행기술조사문헌

KR101157039 B1

(73) 특허권자

충남대학교산학협력단

대전광역시 유성구 대학로 99 (궁동, 충남대학교)

(72) 발명자

이규철

대전 유성구 배울2로 133, 205동 203호 (용산동, 강남아너스빌2단지)

임형준

대전광역시 대덕구 계족산로 136, 502동904호(송촌동, 선비마을5단지아파트)

(뒷면에 계속)

(74) 대리인

특허법인태동

전체 청구항 수 : 총 10 항

심사관 : 김종권

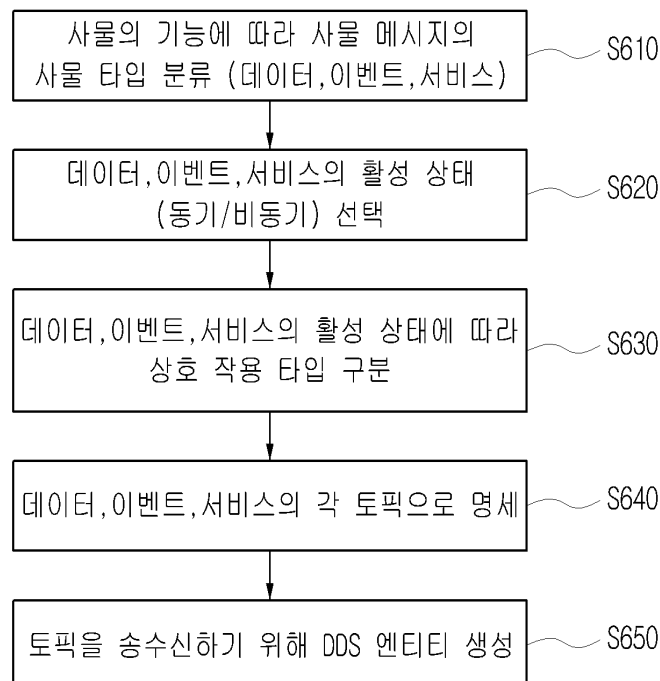
(54) 발명의 명칭 **사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법**

(57) 요약

본 발명은 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법에 관한 것으로서, 발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS) 시스템에서 분산 환경의 이질적인 사물의 메시지들을 통합하는 방법에 있어서, 상기 DDS 기반으로 송수신되는 사물의 메시지를 사물의 기능에 따라 데이터, 이벤트 또는

(뒷면에 계속)

대표도 - 도6



서비스로 사물 타입을 분류하는 단계; 상기 분류한 사물 타입에 따라 데이터, 이벤트 또는 서비스를 동기/비동기 방식의 활성화 방식을 선택하며, 상기 선택한 활성 상태에 따라 상호작용(Interaction) 타입을 구분하여 DDS 기반의 각 토픽으로 명세하여 통합하는 단계; 및 상기 데이터, 이벤트 또는 서비스의 각 토픽을 실시간 발간/구독 하도록 해당 토픽을 포함하는 DDS 엔티티를 동적으로 생성하는 단계를 포함한다. 따라서, 본 발명은 콘텐츠 중심으로 통합 방식은 네트워크 관점의 통합 문제를 해결하여 유무선 통신 프로토콜에 종속적이지 않으면서 사물 인터넷의 수많은 사물들이 유기적으로 통신하고, 일관적인 인터페이스를 통해 접근이 가능하며, 키를 통해 다수의 인스턴스를 갖는 하나의 토픽을 생성하여 DDS 엔티티의 수를 줄일 수 있다.

(72) 발명자

백문기

충북 제천시 용두대로7길 25, 107동 602호 (하소동, 그린코아루아파트)

김인아

경기 안산시 상록구 호동로1길 15, 102호 (일동, 아산빌라)

이 발명을 지원한 국가연구개발사업

과제고유번호 2012R1A1A4A01015696

부처명 교육부

연구관리전문기관 한국연구재단

연구사업명 일반연구자지원사업 지역대학우수과학자

연구과제명 사물 인터넷의 이질적 데이터/이벤트/서비스 통합을 위한 연동 및 조합 기법 개발

기 여 율 1/1

주관기관 충남대학교 산학협력단

연구기간 2012.05.01 ~ 2015.04.30

공지예외적용 : 있음

명세서

청구범위

청구항 1

발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS) 시스템에서 분산 환경의 이질적인 사물의 메시지들을 통합하는 방법에 있어서,

상기 DDS 기반으로 송수신되는 사물의 메시지를 사물의 기능에 따라 데이터, 이벤트 또는 서비스로 사물 타입을 분류하는 단계;

상기 분류한 사물 타입에 따라 데이터, 이벤트 또는 서비스를 동기/비동기 방식의 활성화 방식을 선택하며, 상기 선택한 활성화 상태에 따라 상호작용(Interaction) 타입을 구분하여 DDS 기반의 각 토픽으로 명세하여 통합하는 단계; 및

상기 데이터, 이벤트 또는 서비스의 각 토픽을 실시간 발간/구독하도록 해당 토픽을 포함하는 DDS 엔티티를 동적으로 생성하는 단계를 포함하는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 2

제1항에 있어서,

상기 상호작용 타입은,

상기 데이터를 동기 방식의 수집(Collection) 및 비동기 방식의 로깅(Logging)으로 구분하고, 상기 이벤트를 동기 방식의 전파(Prppagation) 및 비동기 방식의 분배(Distribution)로 구분하여, 상기 서비스를 동기/비동기 방식의 응답 형태의 호출(Invocation)과 무응답 형태의 호출(Evocation)로 구분되는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 3

제2항에 있어서,

상기 서비스의 응답 형태의 호출은 서비스 요청(Request) 정보와 응답(Response) 정보로 구분되고, 상기 무응답 형태의 호출은 요청 정보로 이루어진 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 4

제1항에 있어서,

상기 상호작용 타입은 해당 토픽의 키(Key)로 이용되고, 상기 키를 통해 인스턴스를 식별하는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 5

제4항에 있어서,

상기 토픽은 ID와 인스턴스를 식별 규칙으로 하고, 데이터 중심의 배포 서비스를 위한 네트워크 계층 기반의 데이터 전송 프로토콜인 RTPS(Real-Time Publish-Subscribe)를 이용하는 주소 체계를 가지는 것을 특징으로 하는

사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 6

제1항에 있어서,

상기 데이터 토픽과 이벤트 토픽은 사물의 발간자 ID, 상호 작용 타입, 전송인자(Parameter), 값(Value)을 포함하여 구성되는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 7

제1항에 있어서,

상기 서비스 토픽은 동기 방식의 경우에 서비스 요청 토픽과 서비스 응답 토픽으로 구분되고, 비동기 방식의 경우에 서비스 요청 토픽으로 이루어지며,

상기 서비스 요청 토픽은 서비스 요청자 ID, 서비스 ID, 상호 작용 타입, 전송인자(Parameter), 값(Value)으로 구성되고, 상기 서비스 응답 토픽은 서비스 요청자 ID, 상호 작용 타입, 전송인자, 값을 포함하여 구성되는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 8

제6항 또는 제7항에 있어서,

상기 전송인자와 값은 쌍으로 구성하여 샘플을 구성하는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 9

제1항에 있어서,

상기 분류한 사물 타입에 따라 데이터, 이벤트 또는 서비스를 동기/비동기 방식에 따라 상호작용(Interaction) 타입을 구분하여 DDS 기반의 각 토픽으로 명세하여 통합하는 단계는,

사물 타입, 상호작용 타입, 활성 상태, DDS의 발간/구독, 토픽의 수, 일대일/일대다/다대일을 포함한 연결 형태 또는 콘텐츠 필터링 여부를 나타내는 각 항목들을 적어도 하나 이상을 포함하여 분류체계를 구성하고, 상기 분류 체계를 이용하여 상기 데이터, 이벤트 또는 서비스를 각각의 토픽으로 변환하는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

청구항 10

제9항에 있어서,

상기 토픽은 상기 분류체계에 의해 사물타입, 상호작용 타입, 활성화 방식, 해당 사물이 제공하는 토픽의 이름을 추가하여 URI 형태로 식별규칙을 표현하는 것을 특징으로 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법.

발명의 설명

기술 분야

[0001]

본 발명은 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법에 관한 것으로, 보다 상세하게는 사물

인터넷에서 사물의 메시지를 사물의 기능에 따라 데이터, 이벤트 및 서비스로 분류하고, 이렇게 분류한 데이터, 이벤트 및 서비스를 동기/비동기 방식에 따라 상호 작용 타입을 구분하여 DDS 기반의 토픽으로 명세하여 통합하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법에 관한 것이다.

배경 기술

- [0002] 사물인터넷(Internet of Things, IoT)은 사물(thing)들이 유선 또는 무선 네트워크로 연결되어 유기적으로 정보를 주고받으면서 상호 작용하는 것을 의미한다. MIT의 Auto-ID 센터는 자동화와 효율 증대의 주 기술인 RFID(Radio Frequency Identification)를 통해 센서, 액추에이터(actuator) 등의 사물을 인터넷과 연결하는 개념으로 사물인터넷을 정의한다. 국제전기통신연합인 ITU-T(International Telecommunication Union)는 미래 인터넷은 사람뿐만 아니라 사물들이 정보를 주고받기 위해 언제, 어디서, 누구에게나 무엇이든 연결을 제공한다고 정의하며, 유럽의 통신 위원회는 사물이 사회, 환경, 사용자 콘텍스트를 통해 서로 연결하고 통신하기 위해 지능적인 인터페이스를 이용하여 식별하고 가상화하여 동작하는 것이라 정의한다.
- [0003] 사물인터넷에서 사물은 물리적으로 센서, 액추에이터, 디바이스 등을 의미하는데 정보를 읽거나 기록할 수 있고, 주변 환경의 정보를 측정 하거나 감지하여 통신하는 기능을 포함한다. 또한 사물은 다양한 정보를 수집하여 추가적인 처리를 위해 통신망으로 전달하는 기능을 수행하며, 통신망으로부터 전달 받은 명령을 실행한다.
- [0004] 도 1은 일반적인 사물 인터넷에서의 사물을 설명하는 개념도이다.
- [0005] 도 1에 도시된 바와 같이, 사물의 통신은 게이트웨이를 이용한 네트워크 연결(a), 직접 네트워크 연결(b), 네트워크를 통하지 않고 사물에 직접 연결(c)하는 경우가 있다. 정보 관점에서는 하나의 물리적인 사물을 하나 이상의 가상 사물(virtual thing)로 표현하여 정보를 저장, 처리, 접근할 수 있고, 물리적인 사물과 관계없이 가상의 사물로 존재할 수 있다.
- [0006] 사물인터넷에서 가상의 사물이 센서, 액추에이터, 디바이스 등의 물리적인 사물과 통신하기 위해서는 표준기반의 식별규칙(naming), 주소체계(addressing), 통신 방법(communication) 등이 필요하다. 이에 따라 통신 프로토콜을 통해 통신 범위, 데이터 형식, 데이터 전송 속도, 장치 비용, 보안 등에 따라 직접 통신 또는 게이트웨이를 통해 서로 통신한다. 대표적인 유무선 통신 프로토콜은 RFID, NFC(Near Field Communication), 지그비(Zigbee), Wi-Fi, 블루투스(Bluetooth), 6LowPAN, M2M(Machine to Machine) 등이 존재한다.
- [0007] 다양한 유무선 통신 프로토콜은 디바이스, 네트워크, 서비스에 따라 식별규칙, 데이터 형식, 입출력 파라미터, 전송 방법, 통신규약, 프로그래밍 언어 등이 상이하기 때문에 수많은 사물을 접근하고 정보를 송수신하기 위한 일관적인 방법을 제공해야 한다.
- [0008] 이를 위해, 비특허 문헌 1, 2 및 3에서는 미들웨어 기반의 네트워크 인터페이스를 통해 통신 프로토콜을 추상화하며, 상이한 사물들의 데이터를 송수신하고 서비스 요청/응답의 메시지를 중재하여 통합하는 내용이 기재되어 있다.
- [0009] 네트워크 관점의 통합은 하나의 미들웨어가 다양한 유무선 통신 프로토콜과 저수준의 네트워크 인터페이스를 모두 포함하지 못하며, 사물 특성을 반영하지 않고 메시지만 전달하는 문제점이 있다. 이러한 문제점을 해결하기 위해서는 메시지의 특성을 반영한 콘텐츠 중심(content-centric)의 통합이 필요하다.
- [0010] 사물 인터넷은 수많은 사물들이 유기적으로 통신하고, 상호 작용하기 위해 확장성(scalability), 실시간성(real-time), 신뢰성(reliability), 상호 운용성(interoperability) 등의 특성을 지원해야 한다. 네트워크로 연결된 수많은 콘텐츠, 서비스와 사물은 대량으로 발생하는 정보를 효율적으로 관리하고 결합할 수 있어야 하며, 현재 구성된 시스템에 영향을 주지 않고 새로운 사물들을 동적으로 네트워크에 적용시켜 바로 통신 또는 서비스로 제공해야 한다. 또한, 대부분이 무선 환경의 사물이기 때문에 네트워크에 추가/변경/삭제가 동적으로 가능해야 하고, 신뢰성 및 실시간성 등의 메시지 송수신에 필요한 서비스 품질(QoS; Quality of Services)을 보장해야 한다.
- [0011] 사물 인터넷의 분산 환경의 특징 중 하나는 이질성을 가지고 있다는 것이다. 모바일 디바이스, 스마트 디바이스, PC, 센서, 액추에이터 등 다양한 유형의 디바이스들이 사물 인터넷의 분산 환경에 참여하고 있다. 이러한 디바이스들은 같은 목적을 위해 만들어졌다고 하더라도 서로 다른 방법으로 목적을 수행하게 되고, 사물 인터넷의 디바이스는 항상 인터넷에 연결되어 있다는 특징도 함께 가지고 있다. 이러한 요인들 때문에 사물 인

터넷의 이질적인 디바이스들로부터 송수신되는 사물의 메시지를 일관적이고 안정적으로 관리하는게 어렵기 때문에 사물 인터넷의 일지적인 문제를 해결해야 한다.

선행기술문헌

비특허문헌

[0012]

(비특허문헌 0001) 1. Vlacheas, P. and et al. "Enabling smart cities through a cognitive management framework for the internet of things", IEEE Communications Magazine, Vol. 51, Issue 6, pp. 102~111, June, 2013.

(비특허문헌 0002) 2. Mandler, B. and et al. "COMPOSE - A Journey from the Internet of Things to the Internet of Services", In proceeding of the 27th International Conference on Advanced Information Networking and Applications Workshops, pp. 1217~1222, March, 2013.

(비특허문헌 0003) 3. Wu Yuexin and et al. "Service-Oriented Middleware for Heterogeneous Environment in Internet of Things", Network Technology and Application, China Communication, pp. 41~51, September, 2012.

발명의 내용

해결하려는 과제

[0013]

본 발명은 사물 인터넷의 이질적인 사물의 메시지들을 사물의 기능에 따라 데이터, 이벤트, 서비스로 사물 타입을 분류하고, 분류한 데이터, 이벤트 및 서비스가 구독자에 의해 동기/비동기 방식이 선택되며, 동기/비동기 방식에 따라 상호 작용 타입을 구분하여 DDS 기반의 토픽으로 명세하여 통합함으로써 사물 인터넷의 수많은 사물들이 유기적으로 통신하고, 상호 작용하도록 하는 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법을 제공한다.

과제의 해결 수단

[0014]

실시예들 중에서, 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법은, 발간/구독 모델의 데이터 분산 서비스(Data Distribution Service, DDS) 시스템에서 분산 환경의 이질적인 사물의 메시지들을 통합하는 방법에 있어서, 상기 DDS 기반으로 송수신되는 사물의 메시지를 사물의 기능에 따라 데이터, 이벤트 또는 서비스로 사물 타입을 분류하는 단계; 상기 분류한 사물 타입에 따라 데이터, 이벤트 또는 서비스를 동기/비동기 방식의 활성 상태를 선택하며, 상기 선택한 활성 상태에 따라 상호작용(Interaction) 타입을 구분하여 DDS 기반의 각 토픽으로 명세하여 통합하는 단계; 및 상기 데이터, 이벤트 또는 서비스의 각 토픽을 실시간 발간/구독하도록 해당 토픽을 포함하는 DDS 엔티티를 동적으로 생성하는 단계를 포함하는 것을 특징으로 한다.

[0015]

상기 상호작용 타입은, 상기 데이터를 동기 방식의 수집(Collection) 및 비동기 방식의 로깅(Logging)으로 구분하고, 상기 이벤트를 동기 방식의 전파(Propagation) 및 비동기 방식의 분배(Distribution)로 구분하여, 상기 서비스를 동기/비동기 방식의 응답 형태의 호출(Invocation)과 무응답 형태의 호출(Evocation)로 구분되는 것을 특징으로 한다.

[0016]

이때, 상기 서비스의 응답 형태의 호출은 서비스 요청(Request) 정보와 응답(Response) 정보로 구분되고, 상기 무응답 형태의 호출은 요청 정보로 이루어진 것을 특징으로 한다.

[0017]

상기 상호작용 타입은 해당 토픽의 키(Key)로 이용되고, 상기 키를 통해 인스턴스를 식별하는 것을 특징으로 한다.

[0018]

따라서, 상기 토픽은 ID와 인스턴스를 식별 규칙으로 하고, 데이터 중심의 배포 서비스를 위한 네트워크 계층 기반의 데이터 전송 프로토콜인 RTPS(Real-Time Publish-Subscribe)를 이용하는 주소 체계를 가지는 것을 특징으로 한다.

- [0019] 상기 데이터 토픽과 이벤트 토픽은 사물의 발간자 ID, 상호 작용 타입, 전송인자(Parameter), 값(Value)을 포함하여 구성되는 것을 특징으로 한다.
- [0020] 상기 서비스 토픽은 동기 방식의 경우에 서비스 요청 토픽과 서비스 응답 토픽으로 구분되고, 비동기 방식의 경우에 서비스 요청 토픽으로 이루어지며, 상기 서비스 요청 토픽은 서비스 요청자 ID, 서비스 ID, 상호 작용 타입, 전송인자(Parameter), 값(Value)으로 구성되고, 상기 서비스 응답 토픽은 서비스 요청자 ID, 상호 작용 타입, 전송인자, 값을 포함하여 구성되는 것을 특징으로 한다.
- [0021] 상기 전송인자와 값은 쌍으로 구성하여 샘플을 구성하는 것을 특징으로 한다.
- [0022] 상기 분류한 사물 타입에 따라 데이터, 이벤트 또는 서비스를 동기/비동기 방식에 따라 상호작용(Interaction) 타입을 구분하여 DDS 기반의 각 토픽으로 명세하여 통합하는 단계는, 상기 사물 타입, 상호작용 타입, 활성 상태, DDS의 발간/구독, 토픽의 수, 연결 형태 또는 콘텐츠 필터링 여부를 나타내는 각 항목들을 적어도 하나 이상을 포함하여 분류체계를 구성하고, 상기 분류 체계를 이용하여 상기 데이터, 이벤트 또는 서비스를 각각의 토픽으로 변환하는 것을 특징으로 한다.
- [0023] 상기 토픽은 상기 분류체계에 의해 사물타입, 상호작용 타입, 활성화 방식, 해당 사물이 제공하는 토픽의 이름을 추가하여 URI 형태로 식별규칙을 표현하는 것을 특징으로 한다.

발명의 효과

- [0024] 본 발명의 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법은, 사물 인터넷의 이질적인 사물의 메시지들을 사물의 기능에 따라 데이터, 이벤트, 서비스로 사물 타입을 분류하고, 데이터, 이벤트 및 서비스를 동기/비동기 방식에 따라 상호 작용 타입을 구분하여 토픽으로 명세할 수 있고, 이러한 콘텐츠 중심으로 통합 방식은 네트워크 관점의 통합 문제를 해결하여 유무선 통신 프로토콜에 종속적이지 않으면서 사물 인터넷의 수많은 사물들이 유기적으로 통신하고, 일관적인 인터페이스를 통해 접근이 가능하며, 키를 통해 다수의 인스턴스를 갖는 하나의 토픽을 생성하여 DDS 엔티티의 수를 줄일 수 있는 효과가 있다.

도면의 간단한 설명

- [0025] 도 1은 일반적인 사물 인터넷에서의 사물을 설명하는 개념도이다.
- 도 2는 본 발명의 일 실시예에 따른 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법을 구현하는 DDS 시스템을 설명하는 도면이다.
- 도 3은 도 2에서 명세된 토픽의 상호 작용을 설명하는 도면이다.
- 도 4는 도 2의 DDS 시스템을 설명하는 기본 개념도이다.
- 도 5는 도 2의 토픽에 전송 과정을 설명하는 도면이다.
- 도 6은 본 발명의 일 실시예에 따른 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법을 설명하는 순서도이다.
- 도 7은 사물 인터넷에서 데이터의 전송 과정을 설명하는 도면이다.
- 도 8은 사물 인터넷에서 이벤트의 전송 과정을 설명하는 도면이다.
- 도 9는 사물 인터넷에서 서비스의 호출 과정을 설명하는 도면이다.
- 도 10은 사물 인터넷에서 데이터와 이벤트에서 사용하는 동기 방식의 웨이트셋 방식을 설명하는 도면이다.
- 도 11은 사물 인터넷의 데이터 토픽에 대한 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이다.
- 도 12는 사물 인터넷의 이벤트 토픽에 대한 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이다.
- 도 13은 사물 인터넷의 서비스 요청 토픽의 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이다.
- 도 14는 사물 인터넷의 서비스 응답 토픽의 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이다.

도 15는 사물 인터넷의 서비스 응답 토픽에 대한 콘텐츠 필터링을 이용한 샘플 송수신 과정을 설명하는 도면이다.

도 16은 데이터 토픽의 수집에 대한 IDL 예제를 설명하는 도면이다.

도 17은 데이터 토픽의 수집에 대한 샘플 코드를 설명하는 도면이다.

도 18은 이벤트 토픽의 분배에 대한 IDL 예제를 설명하는 도면이다.

도 19는 이벤트 토픽의 분배에 대한 콘텐츠 필터링 샘플 코드를 설명하는 도면이다.

도 20은 데이터 토픽과 이벤트 토픽을 송수신하기 위한 DDS 엔티티의 구성을 설명하는 도면이다.

도 21은 서비스 토픽을 송수신하기 위한 DDS 엔티티의 구성을 설명하는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0026] 본 발명에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 본 발명의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 본 발명의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다. 또한, 본 발명에서 제시된 목적 또는 효과는 특정 실시예가 이를 전부 포함하여야 한다거나 그러한 효과만을 포함하여야 한다는 의미는 아니므로, 본 발명의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.
- [0027] 한편, 본 발명에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.
- [0028] "제1", "제2" 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로, 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.
- [0029] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다고 언급된 때에는 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.
- [0030] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함하다" 또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0031] 각 단계들에 있어 식별부호(예를 들어, a, b, c 등)는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0032] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 본 발명이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 발명에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.
- [0033] 도 2는 본 발명의 일 실시예에 따른 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법을 구현하는 DDS 시스템을 설명하는 도면이다.
- [0034] 도 2를 참고하면, 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법은, 발간/구독(Publish/Subscribe) 모델의 실시간 미들웨어 표준인 DDS(Data Distribution Service) 시스템(100)에 의해 DDS 도메인에 속한 참가자들간에 데이터 객체인 토픽을 실시간 발간/구독하도록 DDS 엔티티를 동적으로 생성함으로써 해당 토픽을 송수신한다.

- [0035] DDS 시스템(100)은 송수신하는 토픽을 사물의 특성 또는 기능에 따라 데이터, 이벤트 및 서비스로 사물 타입을 분류하고, 분류한 데이터, 이벤트 및 서비스를 통합하기 위해 식별규칙과 주소체계를 갖는 DDS 기반의 토픽으로 명세한다.
- [0036] DDS 기반의 토픽의 식별 체계는 인스턴스를 통해 사물을 데이터, 이벤트 및 서비스로 식별하고, 주소체계는 데이터 중심의 배포 서비스를 위한 네트워크 계층 기반의 데이터 전송 프로토콜인 RTPS를 이용한다. 즉, 사물 인터넷의 토픽은 DDS를 통해 자동적으로 주소체계를 관리하기 때문에 DCPS를 통해 인스턴스만 식별하면 RTPS를 알지 못하더라도 데이터, 이벤트 및 서비스에 대한 각 토픽의 샘플을 송수신할 수 있다.
- [0037] DDS 시스템(100)은 DDS의 응용 프로그램(Application) 사이의 데이터 전송 프로토콜인 RTPS(Real-Time Publish-Subscribe) 계층과, 데이터의 발간/구독을 위한 인터페이스인 DCPS(Data-Centric Publish-Subscribe) 계층을 포함한다.
- [0038] 하나의 사물(Thing)은 사물의 기능에 따라 데이터, 이벤트 또는 서비스를 적어도 하나 또는 모두 제공할 수 있다. 따라서, DDS 시스템(100)은 하나의 사물이 제공하는 데이터, 이벤트 또는 서비스를 각각 토픽으로 명세하여 사물 인터넷의 응용 프로그램과 사물의 다대다 통신을 위한 인터페이스를 제공한다.
- [0039] DDS의 응용 프로그램은 데이터에 대한 발간 및 구독을 요청하는 디바이스로서, 통신 기능이 탑재된 다양한 센서들 또는 사용자 단말 등이 해당되고, 도메인 내에서 데이터의 발간/구독 서비스 실행을 위한 미들웨어가 설치된다.
- [0040] 도 3은 도 2에서 명세된 토픽의 상호 작용을 설명하는 도면이다.
- [0041] 도 3을 참고하면, DDS 시스템(100)은 OMG(Object Management Group)에서 제정한 데이터 중심의 실시간 미들웨어 표준으로서, 분산 환경을 지원하며 도메인에 참여하는 다수의 발간자(110)와 구독자(120) 사이에 약결합(Decoupling)으로 동기 방식 또는 비동기 방식 통신을 수행한다.
- [0042] 이러한 DDS 시스템(100)은 이질적인 사물들의 데이터, 이벤트 및 서비스에 대한 각각의 토픽을 동기/비동기 방식, 약결합 기반의 상호 작용(Interaction) 방식과 일관적인 접근 방법을 제공한다.
- [0043] 도 4는 도 2의 DDS 시스템을 설명하는 기본 개념도이다.
- [0044] 도 4를 참고하면, DDS 시스템은 DDS 도메인에 속한 참가자들 간에 데이터 객체(Data-Object)인 토픽 이름과 데이터 타입을 정의하고, 실시간으로 토픽을 발간/구독하도록 DDS 엔티티(Entities)를 동적으로 생성한다.
- [0045] 이때, DDS 엔티티는 도메인(Domain), 도메인참가자(DomainParticipant), 발간자(Publisher), 구독자(Subscriber), 데이터라이터(DataWriter), 데이터리더(DataReader), 토픽(Topic) 등을 포함한다.
- [0046] DDS 시스템에서는 참가자들이 실시간으로 발간/구독하는 토픽을 가상 공간의 개념인 글로벌 데이터 스페이스(Global Data Space)로 관리하는데, 이는 토픽과 DDS 엔티티에 명시된 QoS에 따라 발간된 토픽을 추후 참가한 구독자가 수신할 수 있도록 한다.
- [0047] 데이터라이터(130)에서 데이터 값들을 발간자(110)에게 제공하면, 발간자(110)는 다수의 구독자(120)에서 데이터 값들을 보급하고, 데이터리더(140)는 구독자(120)로부터 데이터 값들을 읽어들이어 데이터 객체인 토픽으로 정의한다.
- [0048] 토픽은 토픽 이름, 키(Key)를 포함한 데이터 타입, QoS로 구성되며 IDL(Interface Description Language)로 표현된다. 토픽은 도메인 내에서 발간/구독하기 위해 고유한 이름으로 식별하며, 타입서포트(TypeSupport) 인터페이스를 통해 데이터 타입에 접근한다. 데이터 타입은 송수신하는 자료형을 IDL, XML, 자바 등의 프로그래밍 언어에 따라 종속적으로 정의할 수 있으며 확장 가능하다. QoS는 신뢰성, 우선 순위 등의 비기능적 특성의 집합으로 토픽, DDS 엔티티에 적용된다.
- [0049] 도 5는 도 2의 토픽에 전송 과정을 설명하는 도면이다.
- [0050] 도 5를 참고하면, 토픽(500)은 키로 구분되는 인스턴스(510)와 전송 단위인 샘플(520)을 통해 데이터를 송수신

한다. 하나의 토픽은 하나의 데이터 타입만 참조할 수 있지만, 다수의 토픽이 동일한 데이터 타입을 참조할 수 있다. 이를 통해, 동일한 데이터 타입을 참조하는 다수의 토픽을 만들어 데이터를 송수신할 수 있고, 키를 통해 다수의 인스턴스(510)를 갖는 하나의 토픽(500)을 생성하면 DDS 엔티티의 수를 줄일 수 있다.

[0051] 인스턴스(510)의 전송 단위인 샘플(520)은 직렬화(Serialization)되어 네트워크 상에서 전송되고, 네트워크를 통해 수신한 샘플의 정보를 추출하기 위해 역직렬화(Deserialization)된다.

[0052] 따라서, DDS 시스템은 사물의 물리적인 특성에 관계없이 사물의 메시지를 이질적인 데이터, 이벤트 및 서비스로 분류하여 DDS 기반의 토픽으로 명세하고, 키를 이용하여 다수의 인스턴스를 식별할 수 있다. 따라서, 사물 인터넷의 모든 사물을 데이터, 이벤트 및 서비스로 추상화하면 DDS의 응용 프로그램은 일관적인 인터페이스를 통해 사물들을 접근하고 통신할 수 있다.

[0053] 사물 인터넷에서 사물을 접근하고 사용하기 위해서는 식별되어야 하고, 통신할 수 있어야 하며, 다른 사물과 네트워크를 구성하여 상호작용하거나 사용자가 접근할 수 있어야 한다. 하나의 사물은 적어도 하나 이상의 식별 규칙과 주소체계를 보유해야 한다. 식별 규칙은 사물마다 고유한 형태로 인식할 수 있는 것을 나타내며, 사람이 해석(human-readable)할 수 있도록 명세해야 하고, 주소 체계는 기계가 인식(machine-readable)할 수 있도록 하여 통신이 가능하도록 한다.

[0054] 따라서, 본 발명의 일 실시예에 따른 DDS 시스템은 식별 규칙을 토픽의 인스턴스를 통해 사물을 데이터, 이벤트 및 서비스로 식별하며, 주소체계를 데이터 중심의 배포 서비스를 위한 네트워크 계층 기반의 데이터 전송 프로토콜인 RTPS를 이용한다. 즉, DDS 시스템이 자동적으로 주소체계를 관리하기 때문에 DCPS를 통해 인스턴스만 식별하면 RTPS를 알지 못하더라도 데이터, 이벤트 및 서비스의 각 토픽을 송수신할 수 있다.

[0055] 도 6은 본 발명의 일 실시예에 따른 사물 인터넷에서 DDS 기반의 이질적인 사물 메시지 통합 방법을 설명하는 순서도이다.

[0056] 도 6을 참고하면, DDS 시스템은 사물 인터넷에서 수많은 사물의 기능에 따라 사물의 메시지를 데이터, 이벤트 또는 서비스로 사물 타입을 분류하고, 이렇게 분류한 사물 타입의 데이터, 이벤트 또는 서비스에 대한 동기/비동기 방식의 활성 상태를 선택한다.(S610, S620)

[0057] 이렇게 선택한 동기/비동기 방식에 따라 데이터, 이벤트 및 서비스의 상호 작용 타입을 구분하여 데이터 토픽, 이벤트 토픽 또는 서비스 토픽으로 명세한다.(S630, S640) 이때, 상호 작용 타입은 해당 토픽의 키로 활용될 수 있고, 토픽은 키를 통해 인스턴스를 식별할 수 있으므로, 결국 상호 작용 타입은 토픽의 ID와 함께 식별 규칙이 될 수 있다.

[0058] 분산 환경에 적합한 발간/구독 기반의 실시간 미들웨어인 DDS 시스템에서 사용하기 위해, 사물 인터넷의 데이터, 이벤트 및 서비스의 특성 또는 기능에 따라 메시지의 기능을 정의하고 분류한 분류 체계는 다음 표 1과 같고, 표 1을 통해 사물 인터넷의 데이터, 이벤트 및 서비스가 새롭게 정의될 수 있다.

표 1

활성(Activation)	사물타입 (Things Type)	상호작용 타입 (Interaction (Connection) Type)
동기(Synchronous) (Delegation)	데이터(Data)	Collection(N:1)
	이벤트(Event)	Propagation(1:N)
	서비스(Service)	Invocation(1:1)
비동기(Asynchronous) (Notification)	데이터(Data)	Logging(N:1)
	이벤트(Event)	Distribution(1:N)
	서비스(Service)	Evocation(1:1)

[0059]

[0060]

일반적으로, 정보 교환은 데이터, 이벤트 및 서비스에서 동기/비동기 방식으로 구분되고, 동기 방식은 클라이언트-서버 형태의 통신으로 클라이언트는 서버에 요청한 결과를 받을때까지 대기 상태를 유지하고, 비동기 방식은 발간자와 상관없이 구독자가 원하는 작업을 처리한다.

[0061]

표 2는 사물 인터넷의 데이터, 이벤트 및 서비스를 토픽으로 변환하기 위한 분류표로서, DDS의 발간/구독, 토픽의 수, 연결 형태, 샘플 수신 조건(content filtering)을 포함한다.

표 2

사물 타입 (Things Type)	상호작용 타입 (Interaction Type)		패러다임 (Paradigm)	토픽 (Topic)	연결 형태 (Connection Type)	샘플 수신 조건 (Content Filtering)
Data	Collection		Publish/Subscribe	1	Mary to One	Yes
	Logging				Mary to One	No
Event	Propagation			1	One to Mary	Yes
	Distribution				One to Mary	Yes
Service	Invocation	Response	Client/Server	1	One to One	No
		Request		1	One to One	No
	Evocation (Request/Non-Response)				One to One	No

[0062]

[0063]

데이터와 이벤트는 발간/구독 모델로 동작하고, 서비스는 1:1로 서비스 요청/응답하는 서버-클라이언트 형태이지만 실제로 DDS 엔티티를 이용하여 토픽을 발간/구독한다.

[0064]

데이터, 이벤트 및 서비스로 통합하기 위해 사용하는 토픽의 수는 데이터(D)/이벤트(E)/서비스 요청(S_{req})/서비

스 응답(S_{res})의 토픽으로 총 4개이다. 연결 형태는 다대다 통신이 가능하지만, 데이터는 구독자가 원하는 시점에 발간하고, 이벤트는 조건에 해당하는 경우에 발간하기 때문에 다대일 또는 일대다로 표현한다.

- [0065] 데이터와 이벤트는 동기 방식과 비동기 방식에 관계없이 동일한 토픽을 사용하기 때문에 각각 1개로 구성된다. DDS 시스템에서 토픽은 도메인 내에 유일한 이름으로 인식하여 발간자와 구독자가 전송 단위인 샘플을 통해 토픽을 송수신한다. 토픽은 송수신하는 샘플을 구분하는 이름으로만 사용하기 때문에 동일한 이름을 갖는 토픽은 키를 통해 인스턴스로 식별하여 샘플을 송수신할 수 있다.
- [0066] 인스턴스는 동일한 토픽을 송수신하더라도 키로 구분되기 때문에 사물에 제공하는 데이터/이벤트/서비스에 따라 각각 해당하는 토픽으로 표현되고, 인스턴스를 통해 사물을 식별하여 사물 인터넷의 이질적인 문제를 해결할 수 있다. 즉, DDS 시스템은 데이터/이벤트/서비스 요청/서비스 응답의 토픽 4개만 생성하면 식별 규칙을 키로 사용하여 인스턴스를 통해 샘플을 송수신할 수 있다.
- [0067] 표 3은 사물 인터넷의 데이터, 이벤트 및 서비스의 각 토픽을 IDL로 명세한 것으로, 데이터와 이벤트의 토픽은 사물의 발간자 ID, 상호작용 타입, 전송인자, 값으로 구성된다. 이벤트는 단일 이벤트의 경우에 하나의 값만 전송하게 되어 전송인자가 필요하지 않지만, 추후 복합 이벤트의 전송을 위해 전송인자를 추가한다.
- [0068] 비동기 방식의 서비스는 요청/응답의 토픽으로 구분되고, 비동기 방식의 서비스는 요청 토픽만 사용한다. 서비스의 요청 토픽은 요청자 ID, 서비스 ID, 상호작용 타입, 전송인자, 값으로 구성되고, 서비스의 응답 토픽은 요청자 ID, 상호작용 타입, 전송인자, 값으로 구성한다.

표 3

Things Type	Interaction Type		Interface Definition(IDL)
Data	Collection		struct dataTopic { string id; string name; string parameters; string values; }; #pragma keylist dataTopic id name // two keys
	Logging		
Event	Propagation		struct eventTopic { string id; string name; string parameters; string values; }; #pragma keylist eventTopic id name // two keys
	Distribution		
Service	Invocation (Request/Response)	Response	struct ServiceResponseTopic { string requestID; string name; string parameters; string values; }; #pragma keylist ServiceResponseTopic requestID name // two keys
	Evocation (Request)	Request	struct ServiceRequestTopic { string requestID; string serviceID; string name; string parameters; string values; }; #pragma keylist ServiceRequestTopic serviceID name // two keys

- [0069]
- [0070] 상호 작용 타입은 식별 규칙을 의미하며, 토픽의 키로 활용한다. 식별 규칙은 표 1 및 표 2에서 정의한 데이터, 이벤트 및 서비스의 분류 체계에 따라 다음과 같이 URI 형태로 표현한다.

- [0071] 식별 규칙의 URI 형태의 표현 : Things Type/Interaction Type/Activation/(Request or Response)/Name
- [0072] 사물 타입(ThingsType)은 사물이 발간하는 토픽으로 데이터, 이벤트 및 서비스 중 해당하는 토픽을 선택한다. 상호작용 타입(InteractionType)은 상호 작용하는 방법을 의미하며, 사물 타입에 따라 데이터의 수집 또는 로깅, 이벤트의 전파 또는 분배, 서비스의 경우에 동기 방식의 서비스 요청/응답, 비동기 방식의 경우에 서비스 요청으로 하위 분류를 나타낸다.
- [0073] 활성화(Activation)은 동기 방식과 비동기 방식에 따라 구분하고, 서비스일 경우에 서비스 요청과 응답 정보를 추가한다. 마지막으로 사물이 제공하는 데이터/이벤트/서비스 이름을 추가하여 ID와 함께 인스턴스를 식별하기 위한 키로 활용한다.
- [0074] 표 4는 식별 체계의 분류한 것이다.

표 4

Interaction Type	Naming
collection	data/collection/sync/
logging	data/logging/async/
propagation	event/distribution/sync/
distribution	event/propagation/async/
invocation (response)	service/invocation/sync/response/
invocation (request)	service/invocation/sync/request/
evocation (request)	service/evocation/async/request/

- [0075]
- [0076] 예를 들어, 동기 방식의 온도 데이터 수집은 “data/collection/sync/temperature” 와 같이 식별 체계를 표현하여 ID와 함께 인스턴스를 식별하기 때문에 “temperature” 의 이름을 갖는 다른 사물과 구분할 수 있다.
- [0077] 전송 인자와 값은 다수의 형태를 지원하기 위해 JSON 형태로 구성하여 전송인자와 값이 순서대로 쌍을 이루도록 한다. 예를 들어, 온도와 습도를 함께 발간하는 데이터는 전송 인자를 “temperature# humidity” 로, 값은 “30#60” 으로 샘플을 구성하고, 데이터라이터(DataWriter)를 통해 직렬화하여 네트워크에 전송한다.
- [0078] 구독자는 데이터리더(DataReader)가 수신한 샘플을 응용 프로그램에 전달하기 위해 “temperature = 30, humidity = 60” 으로 역직렬화한다.
- [0079] 샘플의 수신 조건인 콘텐츠 필터링은 구독자가 SQL(Structured Query Language)과 유사한 형태로 정의한 조건에 의해 데이터리더(DataReader)가 수신하는 샘플을 의미한다. 조건은 수신하는 샘플 중 전송인자와 값을 쌍으로 구성하여 비교연산자(=, <, ≤, >, ≥)를 통해 정의한다.
- [0080] 예를 들어, 도 6에서 발간된 온도 샘플이 {31, 25, 33, 29, 26, 28, 32, 29, 28, 31, 32, 24, 25, 36, 27}일 경우, 구독자가 조건을 “temperature ≥ 30” 과 같이 정의하면 데이터리더(DataReader)가 필터링하여 {31, 33, 32, 31, 32, 36}의 샘플만 수신한다. 또한, 전송인자와 값의 쌍은 and/or 연산자를 이용하여 복수 형태의 조건을 지원한다.
- [0081] 콘텐츠 필터링은 구독자가 주체가 되어 원하는 샘플만 수신하는 기능이다. 데이터의 수집, 이벤트의 전파와 분배는 구독자가 콘텐츠 필터링을 이용하여 원하는 샘플을 구독할 수 있지만, 데이터의 로깅은 모든 샘플의 수신 이 목적이기 때문에 사용하지 않는다. 서비스는 인스턴스로 식별되기 때문에 서비스 요청/응답 토픽은 사용하지

않는다.

- [0082] DDS 시스템은 사물 인터넷의 데이터, 이벤트 및 서비스를 통합하기 위해 DDS 기반의 토픽으로 명세한 후, 이 토픽들을 송수신하기 위해 DDS 엔티티를 동적으로 생성한다.(S650)
- [0083] 도 7은 사물 인터넷에서 데이터의 전송 과정을 설명하는 도면이며, 도 8은 사물 인터넷에서 이벤트의 전송 과정을 설명하는 도면이고, 도 9는 사물 인터넷에서 서비스의 호출 과정을 설명하는 도면이다.
- [0084] 도 7 내지 도 9를 참조하여 사물 인터넷에서 데이터, 이벤트 및 서비스를 정의하면 다음과 같다.
- [0085] 사물 인터넷에서는 사물의 기능에 따라 수동적 객체, 응답 객체 및 자동적 객체로 구분하고, 사물들이 전송하는 메시지에 포함된 정보에 따라 다수의 메시지로 구분한다. 사물이 표현하는 메시지의 형태와 포함하는 정보에 따라 지능 수준을 나타내지만, 사물의 특성은 다양하기 때문에 범주에 명확하게 포함되지 않거나 일관적으로 표현하기 어렵다.
- [0086] 사물 인터넷에서 수많은 사물이 각기 다른 형태의 이질적인 메시지를 생성하기 때문에 데이터, 이벤트 및 서비스를 구분하여 추상화하면 데이터 중심의 일관적인 방법으로 메시지 통합이 가능하다.
- [0087] 응용 프로그램이나 엔터프라이즈 시스템에서 데이터는 변수, 정수의 값 등의 속성 값을 의미하며, 이벤트는 특정한 상태나 조건에 해당하는 경우이고, 서비스는 미리 정의한 인터페이스를 통해 특정 기능을 수행하도록 한다. 그러나, 사물 인터넷에서는 수많은 사물의 특징을 구분하여 기존의 데이터, 이벤트 및 서비스를 재정의하고, 이를 기반으로 통합할 수 있는 명세 방식이 필요하다.
- [0088] 먼저, 사물 인터넷에서 데이터는 도 7에 도시된 바와 같이, 센서에서 온도, 습도 등의 상태를 감지하거나 측정하는 값으로 일회성과 연속성(Stream) 특성을 지닌다. 사물의 데이터는 물리적 현상을 센싱하여 지속적으로 발생시키거나 정해진 데이터 타입에 별개의 값을 발생시키는 범주로 나뉜다. 센서가 측정하는 값은 지속적으로 데이터를 발생하는 동시에 사라지기 때문에 일회성을 띠며, 일련의 값의 집합으로 데이터는 연속적인 흐름이 된다.
- [0089] 데이터가 온도 데이터인 경우에 센서 ID, 토픽 이름, 값(Value)을 포함하고, 사용자별로 원하는 시작 시점과 종료 시점에 데이터의 연속적인 흐름인 데이터 스트림을 선택하여 수집, 로깅, 모니터링할 수 있다.
- [0090] 도 8에 도시된 바와 같이, 사물 인터넷에서 이벤트는 사물이 특정한 조건에 해당하거나 상태가 변하면 알리는 정보로 비정기적이며, 단일 이벤트 또는 스트림 이벤트가 가능하다. 또한, 데이터나 이벤트를 수집하여 CEP를 통해 새로운 이벤트 생성도 가능하다. 이벤트는 데이터와 같이 일회성인 특징이 있지만 중요한 이벤트는 전송시 손실되면 심각한 오류를 발생시킬 수 있기 때문에 신뢰성 기반의 전송이 필요하다.
- [0091] 이러한 이벤트는 발간 이벤트의 조건을 "Value≥30"으로 할 경우에, 사용자가 원하는 시작 시점과 종료 시점에 따라 사물이 측정한 온도에서 "Value≥30"의 조건에 해당하는 이벤트, 31, 33, 32, 31, 32, 36℃만 발간한다.
- [0092] 도 9에 도시된 바와 같이, 사물 인터넷에서 서비스는 사물을 동작하거나 제어하는 기능으로 서비스 사용자와 서비스 제공자 사이에 동기/비동기 방식의 서비스 호출이 가능하며, 요청/응답과 요청/무응답의 서비스 형태가 존재한다.
- [0093] 동기 방식의 서비스 호출은 사용자(Client)가 서비스(Service)를 요청(request)한 후 응답(response) 메시지를 수신할 때까지 대기 상태로 기다리는 경우(a)이고, 비동기 방식의 서비스 호출은 사용자가 서비스를 요청한 동시에 응답 메시지와 상관없이 사용자가 원하는 작업을 처리한다.(b)
- [0094] 도 10은 사물 인터넷에서 데이터와 이벤트에서 사용하는 동기 방식의 웨이트셋 방식을 설명하는 도면이다.
- [0095] 도 10을 참고하면, 사물 인터넷에서 서비스의 동기/비동기 방식은 요청에 따라 응답의 유무로 결정된다. 데이터와 이벤트의 경우에는 구독자가 미리 정한 데이터리더의 수신 방식에 따라 동기는 웨이트셋(WaitSet) 방식, 비동기 방식은 리스너(Listener) 방식을 사용한다.
- [0096] 웨이트셋 방식은 사용자에게 의해 정의된 조건을 사용하여 DDS의 응용 프로그램(Application)을 제어하고, 리스너 방식은 DDS 시스템의 동작 중에 발생하는 상태 정보 이벤트를 비동기적으로 응용 프로그램에 제공한다.

- [0097] 사물 인터넷에서 데이터와 이벤트의 경우에 웨이트셋 방식은 블록 상태(Blocked)와 비블록 상태(Unblocked)로 각각 변경되면서 쓰레드를 이용하여 샘플을 수신하기 때문에 비블록 상태에서는 샘플을 수신하는 동안 다른 작업을 진행할 수 없다.
- [0098] 그러나, 리스너 방식은 데이터리더가 콜백(callback) 함수를 통해 샘플을 수신하면 구독자의 응용 프로그램에서 read() 함수 또는 take() 함수를 이용하여 샘플을 전달하기 때문에 데이터, 이벤트 및 서비스 요청에 대한 응답에 상관없이 독립적으로 수행한다.
- [0099] 상기한 표 1에 나타난 분류 체계는 모두 토픽으로 명세하는 식별 규칙에 표기하기 때문에 구독자는 동기/비동기 방식을 선택할 수 있다. 동기 방식의 경우에 웨이트셋 방식은 리스너 방식에 비해 지연 시간이 오래 걸리지만 처리량이 높고 CPU 사용량이 적기 때문에 안전하게 샘플을 수신할 경우에 적합하다. 비동기 방식의 경우에 리스너 방식은 웨이트셋 방식에 비해 처리량이 낮고 CPU 사용량이 높지만 샘플을 읽는 속도가 빠르기 때문에 지연 시간이 적은 응용 프로그램에 적합하다.
- [0100] 데이터는 구독자의 선택에 따라 동기/비동기 방식의 수집(collection)과 로깅(logging)으로 구분한다. 원칙으로 다수의 사물을 설정하거나 정보를 전달할 수 있고, 사물이 측정한 값을 주기적으로 수집하고, 연결된 수많은 사물의 상태를 확인하거나 진단하는 경우가 있다. 데이터는 전송 우선 순위와 도착 시간에 민감하지 않게 전송이 가능하며, 구독자의 목적에 따라 수신한 샘플은 조건을 두어 원하는 데이터만 응용 프로그램에 전달할 수 있다.
- [0101] 이벤트는 구독자의 선택에 따라 동기/비동기 방식의 전파(propagation)와 분배(distribution)로 구분한다. 사물에 특정한 조건에 해당하는 경우나 사물의 상태 변화에 따른 알람을 전달하기 때문에 도착 시간에 민감하며, 송수신 중 누락된 이벤트가 있는지 확인하는 신뢰성 기반의 전송이 필요하다.
- [0102] 서비스는 서비스 제공자가 미리 정한 응답/무응답 형태에 따라 동기/비동기 방식이 결정된다. 동기 방식의 서비스 호출(invocation)은 서비스 제공자와 서비스 요청자간에 1:1로 인터페이스를 통해 원격의 사물을 동작하기 위한 과정으로 서비스 제공자는 결과에 대한 응답을 보낸다. 즉, "A invokes B"는 A가 인터페이스를 통해 B의 기능을 수행하고, 그 결과를 B가 A에게 전달하는 것을 의미한다.
- [0103] 비동기 방식의 서비스 호출(evocation)은 동기 방식의 서비스 호출과 의미는 동일하지만, 서비스를 요청한 결과에 대한 응답이 없는 경우이다. 이 경우에, 보통 중요하지 않은 서비스를 요청하기 때문에 전송 우선순위를 낮게 설정한다.
- [0104] 이와 같이, DDS 시스템이 동기/비동기 방식으로 구분하여 데이터, 이벤트 및 서비스를 송수신하기 때문에 네트워크 관점의 통합 문제점이 해결될 수 있고, 콘텐츠 중심으로 통합되어 사물이 갖는 특성에 따라 데이터, 이벤트 및 서비스로 명세할 수 있다. 이를 통해, 유무선 통신 프로토콜에 종속적이지 않고, 수많은 사물의 이질적인 메시지를 통합할 수 있으며, 일관적인 인터페이스를 통해 접근이 가능하다.
- [0105] 도 11은 사물 인터넷의 데이터 토픽에 대한 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이고, 도 12는 사물 인터넷의 이벤트 토픽에 대한 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이며, 도 13은 사물 인터넷의 서비스 요청 토픽의 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이고, 도 14는 사물 인터넷의 서비스 응답 토픽의 동기/비동기 방식의 샘플 송수신 과정을 설명하는 도면이며, 도 15는 사물 인터넷의 서비스 응답 토픽에 대한 콘텐츠 필터링을 이용한 샘플 송수신 과정을 설명하는 도면이다.
- [0106] 도 11 내지 도 15를 참고하면, 동기 방식과 비동기 방식의 데이터, 이벤트, 서비스 요청 및 서비스 응답의 각 토픽은 인스턴스를 통해 샘플을 송수신하고 있고, 발간자와 구독자는 키를 통해 인스턴스를 식별할 수 있다.
- [0107] 도 11에 도시된 바와 같이, 동기 방식과 비동기 방식의 데이터 토픽은 사물의 ID(A, B, C)와 식별체계인 상호작용 타입(interactionType)을 키로 인스턴스가 식별된다.
- [0108] 사물 A, B, C는 데이터 토픽의 동일한 타입을 사용하기 때문에 하나의 데이터라이터(DataWriter)를 사용하지만, 경우에 따라 다수의 데이터라이터를 사용할 수 있다. 즉, 다수의 사물은 하나의 데이터 토픽으로 표현하여 통합하고, 키를 통한 인스턴스를 식별하여 샘플을 받게 된다.
- [0109] 사물은 측정한 값을 계속 발간하기 때문에 구독자는 원하는 시작 시점과 종료 시점에 따라 샘플(D_A , D_B , D_C)을 수신할 수 있다. 데이터를 발간하는 사물은 데이터 토픽을 사용하며 구독자가 동기 방식 또는 비동기 방식을 선택한다. 이에 따라 데이터리더(DataReader)는 웨이트셋(WaitSet) 방식과 리스너(Listener) 방식을 선택하여 샘플

플을 수신한다.

- [0110] 도 12에 도시된 바와 같이, 동기 방식과 비동기 방식의 이벤트 토픽은 상기한 데이터 토픽과 동일하게 동작한다.
- [0111] 도 13에 도시된 바와 같이, 서비스 요청 토픽은 응용 프로그램에 의해 발간되고, 데이터라이터를 사용하여 샘플(S_g, S_h, S_i)을 송신하고, 구독자는 데이터리더를 통해 샘플을 수신한다. 이때, 사물은 샘플 수신시 서비스 요청 토픽으로 표현하여 통합된 샘플을 수신하고, 키를 통한 인스턴스를 식별한다. 도 14에 도시된 바와 같이, 서비스 응답 토픽은 키를 이용한 인스턴스로 식별되어 사물과 응용 프로그램 간에 샘플(S_j, S_k, S_l)이 송수신된다.
- [0112] 이와 같이, 서비스 요청 토픽과 서비스 응답 토픽은 키를 이용하여 인스턴스를 식별하고, 서비스 응답 토픽의 인스턴스는 요청자 ID를 키로 활용하기 때문에 서비스 요청자는 서비스 응답 샘플을 수신할 수 있다.
- [0113] 이와 달리, 콘텐츠 필터링을 통한 서비스 제공자의 식별은, 도 15에 도시된 바와 같이, 구독자가 서비스 응답 토픽에 대한 모든 샘플을 조건에 의해 필터링하여 수신하기 때문에 전송과 조건을 검사하는 과부하가 생긴다. 서비스 요청 토픽도 동일하게 모든 샘플을 조건에 의해 필터링하여 수신하는 방식이 적용된다.
- [0114] 즉, 키를 이용한 인스턴스 식별은 서비스 제공자까지 전송하는 경로를 제시하지만, 콘텐츠 필터링을 통한 서비스 제공자의 식별은 한 곳에서 샘플을 검사하여 분배한다. 따라서, DDS 시스템은 키를 이용한 인스턴스 식별로 서비스 요청 토픽과 응답 토픽의 샘플을 송수신한다.
- [0115] 한편, DDS 시스템은 토픽의 주소체계를 별도로 제시하지 않는다. DDS 기반의 토픽의 식별 체계는 인스턴스를 통해 사물을 데이터, 이벤트 및 서비스로 식별하고, 주소체계는 데이터 중심의 배포 서비스를 위한 네트워크 계층 기반의 데이터 전송 프로토콜인 RTPS를 이용한다. 즉, 사물 인터넷의 토픽은 DDS를 통해 자동적으로 주소체계를 관리하기 때문에 DCPS를 통해 인스턴스만 식별하면 RTPS를 알지 못하더라도 데이터, 이벤트 및 서비스에 대한 각 토픽의 샘플을 송수신할 수 있다.
- [0116] 도 16은 데이터 토픽의 수집에 대한 IDL 예제를 설명하는 도면이고, 도 17은 데이터 토픽의 수집에 대한 샘플 코드를 설명하는 도면이며, 도 18은 이벤트 토픽의 분배에 대한 IDL 예제를 설명하는 도면이고, 도 19는 이벤트 토픽의 분배에 대한 콘텐츠 필터링 샘플 코드를 설명하는 도면이다.
- [0117] 도 16 및 도 18은 데이터 토픽의 수집과 이벤트 토픽의 분배를 IDL로 명세한 예제이고, 도 17 및 도 19는 데이터 토픽과 이벤트 토픽을 발간/구독을 담당하는 샘플 코드이다. 샘플 코드에는 사물의 ID, 상호작용 타입(interactionType), 전송인자, 값을 각각 임의로 입력한다.
- [0118] 도 17 및 도 19에 도시된 바와 같이, 데이터 토픽과 이벤트 토픽은 IDL을 통해 발간 부분과 구독 부분으로 나뉘며, DDS 엔티티를 생성하기 위해 도메인참가자(DomainParticipant)로 토픽, 발간자(Publisher), 데이터라이터(DataWriter)를 등록한 후 write() 함수를 이용하여 샘플을 발간한다. 구독 부분은 토픽, 구독자(Subscriber) 및 데이터리더(DataReader)를 생성하고, read() 또는 take() 함수를 통해 샘플을 수신한다. 조건에 해당하는 샘플만 구독하기 위한 샘플 수신 조건인 콘텐츠 필터링은 도 18에 도시된 바와 같이, 구독 부분이 콘텐츠필터토픽(ContentFilteredTopic)으로 토픽을 명시하고, 조건을 추가한다.
- [0119] 도 20은 데이터 토픽과 이벤트 토픽을 송수신하기 위한 DDS 엔티티의 구성을 설명하는 도면이고, 도 21은 서비스 토픽을 송수신하기 위한 DDS 엔티티의 구성을 설명하는 도면이다.
- [0120] 도 20 및 도 21을 참고하면, 데이터 토픽과 이벤트 토픽을 발간/구독하기 위해 생성되는 DDS 엔티티는 도메인참가자, 발간자, 데이터라이터, 데이터리더, 구독자로 구성되며, 서비스 토픽을 송수신하기 위한 DDS 엔티티 구성도 도메인참가자, 발간자, 데이터라이터, 데이터리더 및 구독자로 구성되지만 서비스 요청 토픽과 서비스 응답 토픽에 해당하는 DDS 엔티티를 각각 생성한다.
- [0121] 사물인터넷(IoT; Internet of Things)은 사물들이 유무선 네트워크로 연결되어 유기적으로 정보를 주고받으면서 상호작용하는 개념으로 수많은 사물을 접근하고 정보를 송수신하기 위한 클라우드 기반의 사물인터넷(CoT; Cloud of Things) 구축이 필요하다. 클라우드 기반의 사물인터넷은 네트워크로 연결된 수많은 콘텐츠, 서비스와 사물은 대량으로 발생하는 정보를 효율적으로 관리하고 결합할 수 있어야 하며, 현재 구성된 시스템에 영향을

주지 않고 새로운 사물들을 동적으로 네트워크에 적용시켜 바로 통신 또는 서비스로 제공해야 한다. 또한, 대부분이 무선 환경의 사물이기 때문에 네트워크에 추가/변경/삭제가 동적으로 가능해야 하고, 신뢰성 및 실시간성 등의 메시지 송수신에 필요한 서비스 품질(QoS; Quality of Services)을 보장해야 한다.

[0122]

상기에서는 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

부호의 설명

[0123]

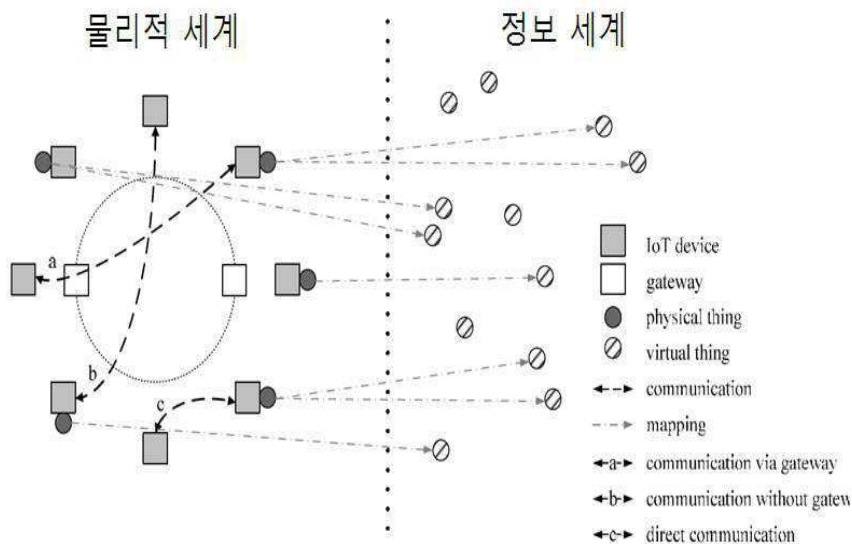
100 : DDS 시스템

110 : 발간자

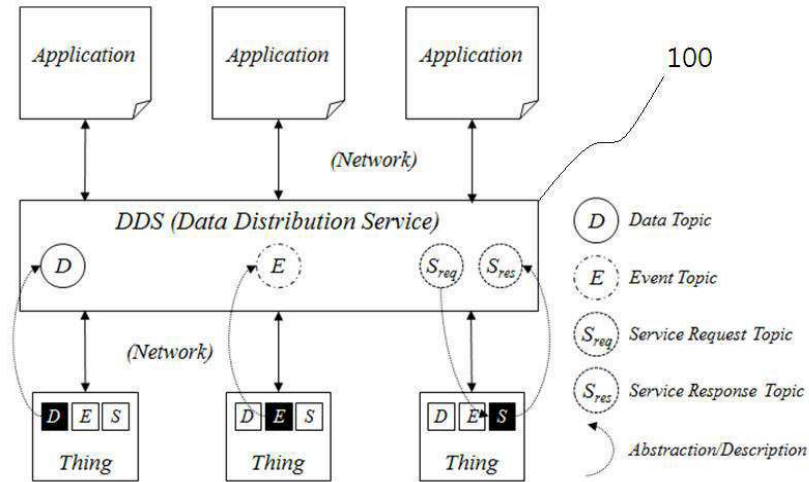
120 : 구독자

도면

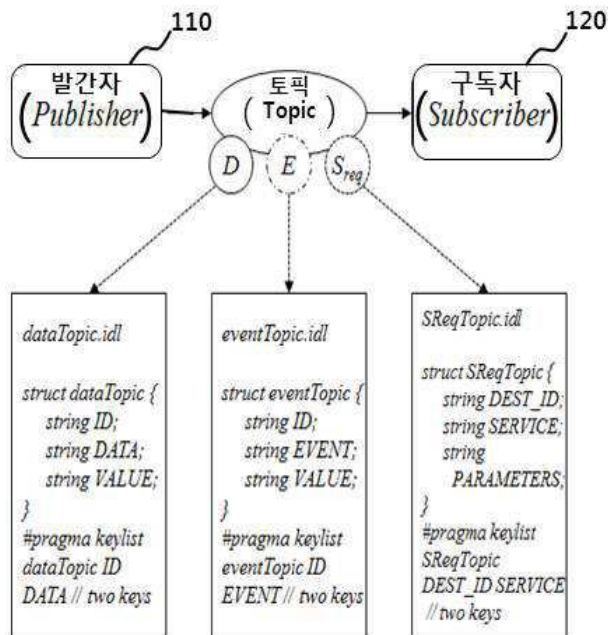
도면1



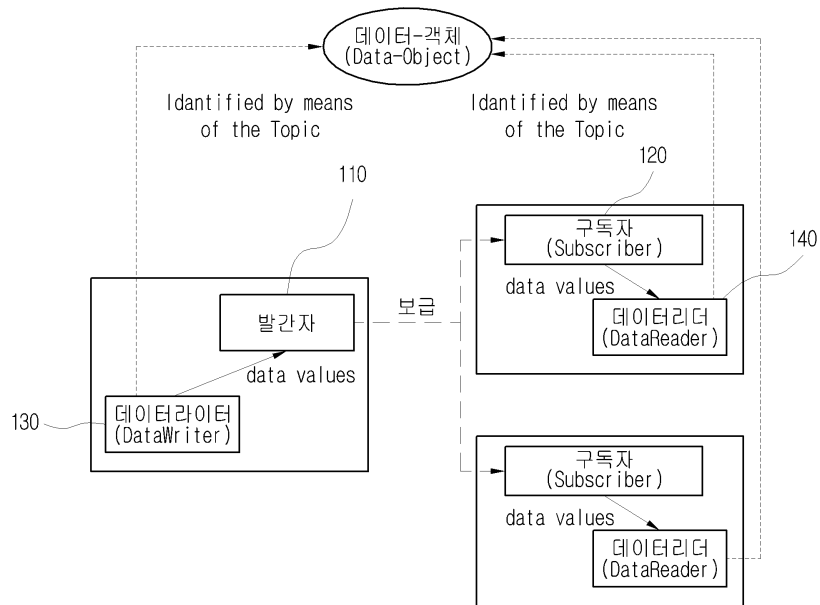
도면2



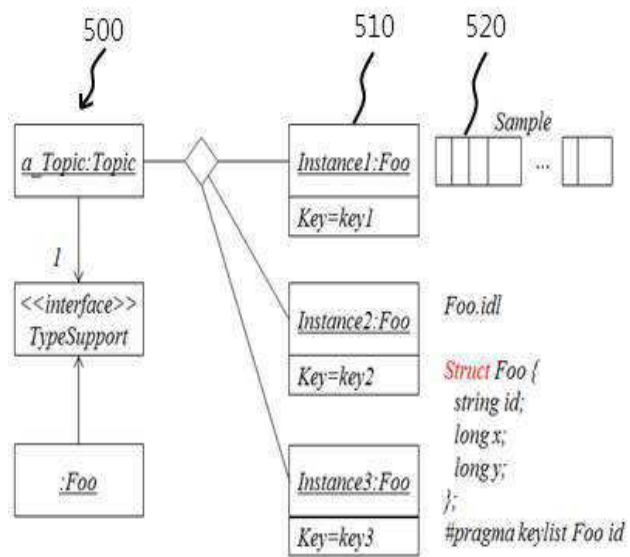
도면3



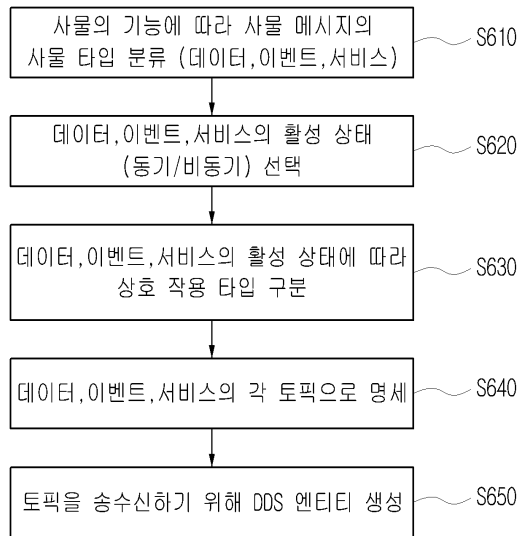
도면4



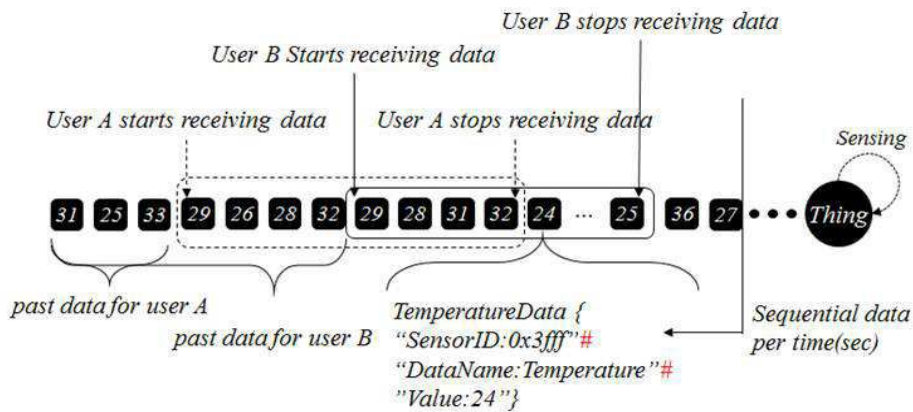
도면5



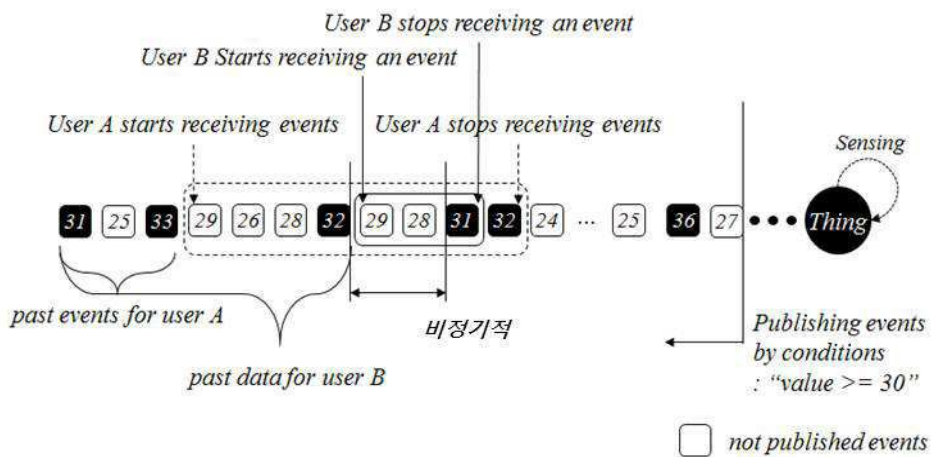
도면6



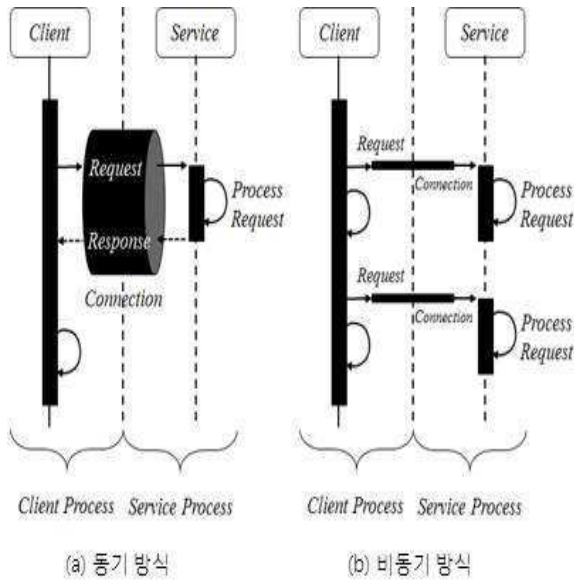
도면7



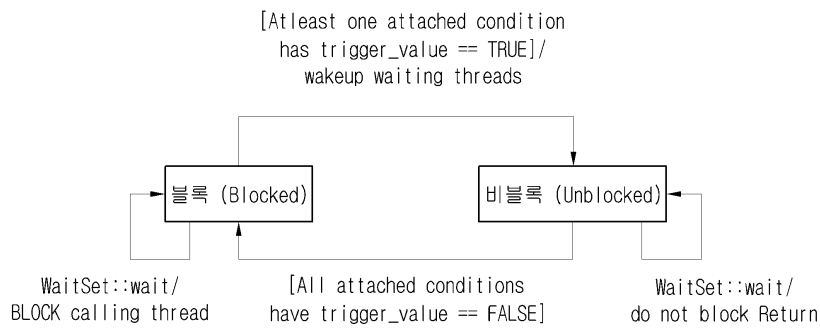
도면8



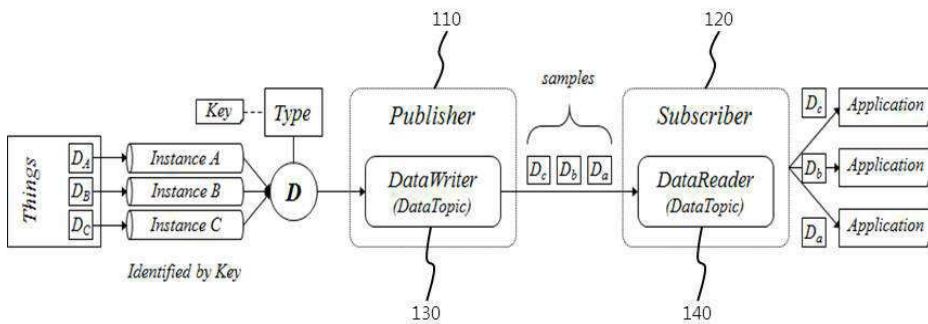
도면9



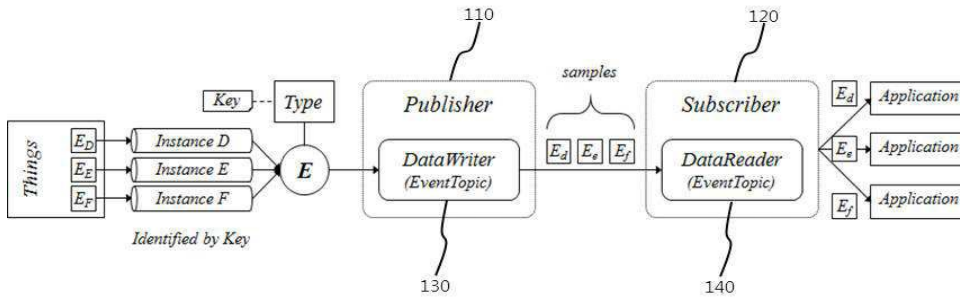
도면10



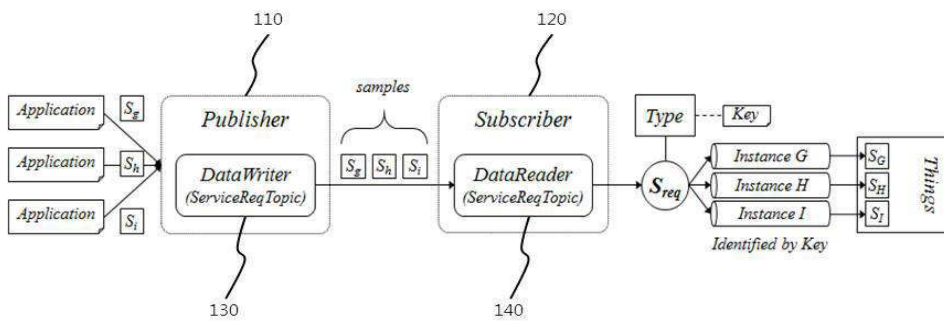
도면11



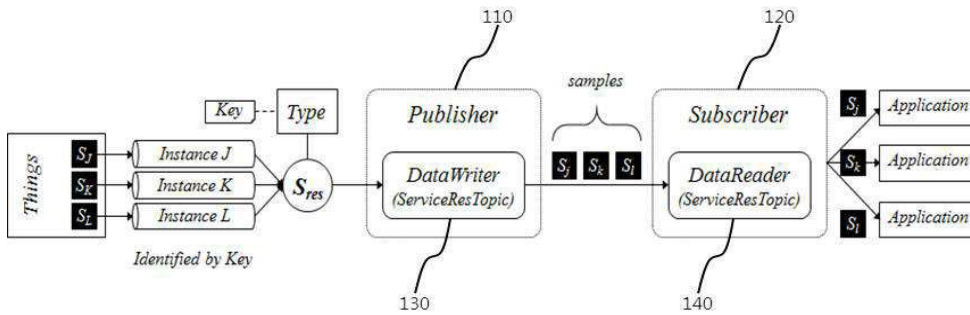
도면12



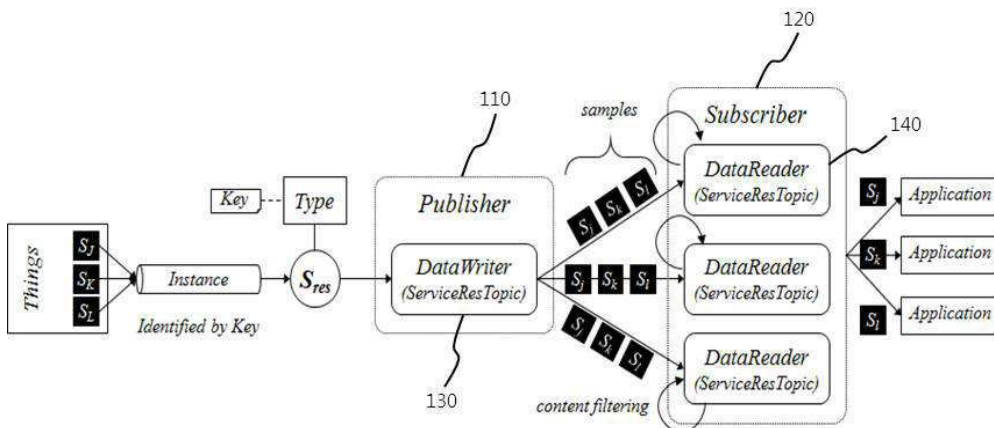
도면13



도면14



도면15



도면16

```

struct dataTopic {
    string id = "0x3fff";
    string name =
        "data/collection/sync/temperature";
    string parameters = "measurement°C";
    string values = "30";
} #pragma keylist dataTopic
    id name // two keys

```

도면17

```

struct eventTopic {
    string id = "0x5fff";
    string name =
        "event/distribution/async/humidity";
    string parameters = "measurement%";
    string values = "50";
} #pragma keylist eventTopic
    id name // two keys

```

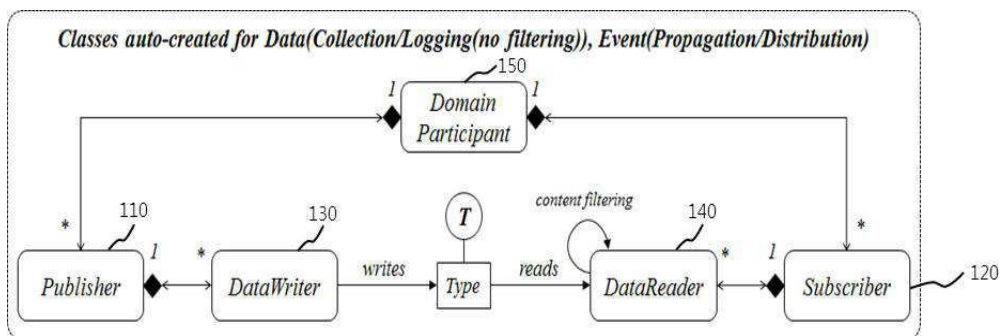
도면18

Side	Sample Code
	<pre> int domain_id = 1; auto dp = DomainParticipant(domain_id); auto topic = Topic<dataType> (dp, "dataType"); </pre>
Pub	
Side	<pre> auto pub = Publisher(dp); auto dw = DataWriter<dataType> (pub, topic); dw.write(dataType("0x3fff", "data/collection/sync/temperature", "measurement°C", "30")); </pre>
	<pre> auto sub = Subscriber(dp); </pre>
Sub	<pre> auto dr = DataReader<dataType> (sub, topic); </pre>
Side	<pre> auto data = dr.read(); </pre>

도면19

Side	Sample Code
	<pre>int domain_id = 1; auto dp = DomainParticipant(domain_id); auto topic = Topic<eventType> (dp, "eventType");</pre>
Pub	
Side	<pre>auto pub = Publisher(dp); auto dw = DataWriter<eventType> (pub, topic); dw.write(eventType("0x5fff", "event/distribution/async/humidity", "measurement%", "50"));</pre>
	<pre>auto sub = Subscriber(dp); auto filter = Filter("measurement% > 40");</pre>
Sub	
Side	<pre>ContentFilteredTopic<eventType> ("CFeventTopic", topic, filter); auto dr = DataReader<dataType> (sub, cftopic); auto data = dr.read();</pre>

도면20



도면21

