



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년09월01일
 (11) 등록번호 10-1549002
 (24) 등록일자 2015년08월26일

(51) 국제특허분류(Int. Cl.)
 G06F 9/50 (2006.01) G06F 11/30 (2006.01)
 (21) 출원번호 10-2013-0121233
 (22) 출원일자 2013년10월11일
 심사청구일자 2013년10월11일
 (65) 공개번호 10-2015-0042502
 (43) 공개일자 2015년04월21일
 (56) 선행기술조사문헌
 KR1020110138547 A*
 JP2002287987 A*
 KR1020120110693 A*
 KR101034426 B1*
 *는 심사관에 의하여 인용된 문헌

(73) 특허권자
 한국과학기술원
 대전광역시 유성구 대학로 291(구성동)
 (72) 발명자
 송준화
 대전광역시 유성구 대학로 291, 한국과학기술원
 전산학과
 주영현
 대전광역시 유성구 대학로 291, 한국과학기술원
 전산학과
 (뒷면에 계속)
 (74) 대리인
 박영우

전체 청구항 수 : 총 19 항

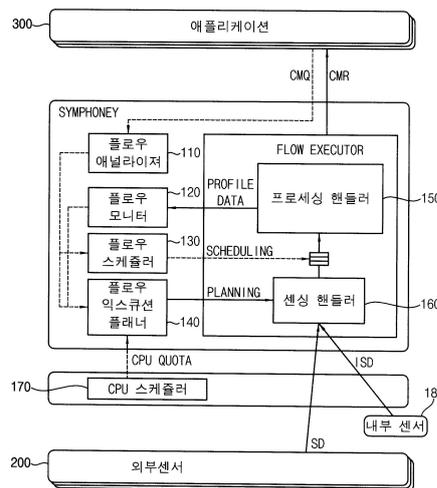
심사관 : 유진태

(54) 발명의 명칭 **모바일 상황 모니터링을 위한 센싱 플로우를 처리하는 모바일 장치, 이를 이용하는 센싱 플로우 처리 방법, 이를 이용하는 상황 모니터링 방법 및 이를 포함하는 상황 모니터링 시스템**

(57) 요약

모바일 장치는 센싱 핸들러 및 프로세싱 핸들러를 포함한다. 센싱 핸들러는 상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 복수의 센싱 오퍼레이터를 포함한다. 프로세싱 핸들러는 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 센싱 오퍼레이터의 센싱 데이터를 처리하는 복수의 프로세싱 오퍼레이터를 포함한다. 이에 따라, 동시에 수행되는 애플리케이션들의 자원 사용을 조율하여 애플리케이션의 유틸리티를 향상시킬 수 있다.

대표도 - 도2



(72) 발명자

이영기

대전광역시 유성구 대학로 291, 한국과학기술원 전
산학과

유지현

대전광역시 유성구 대학로 291, 한국과학기술원 전
산학과

민철홍

대전광역시 유성구 대학로 291, 한국과학기술원 전
산학과

신인식

대전광역시 유성구 대학로 291, 한국과학기술원 전
산학과

이 발명을 지원한 국가연구개발사업

과제고유번호 10041313
부처명 지식경제부
연구관리전문기관 한국산업기술평가관리원
연구사업명 지식경제기술혁신사업
연구과제명 UX 지향 모바일 SW 플랫폼 개발
기 여 율 1/1
주관기관 한국과학기술원
연구기간 2011.12.01 ~ 2012.11.30

특허청구의 범위

청구항 1

상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 복수의 센싱 오퍼레이터를 포함하는 센싱 핸들러; 및 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하는 복수의 프로세싱 오퍼레이터를 포함하는 프로세싱 핸들러를 포함하는 모바일 장치.

청구항 2

제1항에 있어서, 상기 C-FRAME은 복수의 상기 F-FRAME을 포함하는 것을 특징으로 하는 모바일 장치.

청구항 3

제1항에 있어서, 애플리케이션으로부터 센싱 플로우 정보를 입력 받아 상기 센싱 플로우의 상기 C-FRAME의 크기 및 상기 F-FRAME의 크기를 판단하는 플로우 애널라이저를 더 포함하는 모바일 장치.

청구항 4

제3항에 있어서, 상기 플로우 애널라이저는 상기 애플리케이션으로부터 상기 애플리케이션이 사용자의 상황을 얼마나 자주 모니터링 해야 하는 지를 나타내는 필요 모니터링 인터벌 및 마지막 F-FRAME이 도착한 시점부터 유의미한 상기 상황 모니터링 결과를 얻을 수 있는 시간을 나타내는 상기 모니터링 딜레이를 수신하는 것을 특징으로 하는 모바일 장치.

청구항 5

제4항에 있어서, 상기 C-FRAME의 크기, 상기 F-FRAME의 크기 및 상기 필요 모니터링 인터벌을 기초로 상기 C-FRAME의 모니터링 인터벌을 결정하여 상기 센싱 핸들러에 출력하는 플로우 익스큐션 플래너를 더 포함하고, 상기 센싱 플로우의 모니터링 인터벌은 상기 센싱 타임 및 상기 스킵 타임의 합과 일치하는 것을 특징으로 하는 모바일 장치.

청구항 6

제5항에 있어서, 상기 모바일 장치의 CPU 이용률을 판단하여 상기 플로우 익스큐션 플래너에 출력하는 리소스 모니터를 더 포함하고, 상기 플로우 익스큐션 플래너는 상기 C-FRAME의 크기, 상기 F-FRAME의 크기, 상기 필요 모니터링 인터벌 및 상기 CPU 이용률을 기초로 상기 센싱 플로우의 모니터링 인터벌을 결정하는 것을 특징으로 하는 모바일 장치.

청구항 7

제5항에 있어서, 상기 필요 모니터링 인터벌은 상기 모니터링 인터벌에 따른 상기 애플리케이션의 유틸리티 값으로 정의되는 유틸리티 함수 형태를 갖는 것을 특징으로 하는 모바일 장치.

청구항 8

제7항에 있어서, 상기 플로우 익스큐션 플래너는 복수의 센싱 플로우의 상기 모니터링 인터벌이 π_i , 상기 센싱 플로우들의 상기 유틸리티 함수가 u_i 일 때, $maximize \sum u_i(\pi_i)$ 로 상기 모니터링 인터벌을 결정하는 것을 특징으로 하는 모바일 장치.

청구항 9

제8항에 있어서, 상기 플로우 익스큐션 플래너는 ct_i 가 플로우 $FLOW_i$ 의 C-FRAME을 처리하기 위해 필요한 CPU 타

임이고, 1-CPU가 상기 애플리케이션들에게 주어진 CPU availability일 때, $\sum eci/pi \leq 1-CPUj$ 를 만족하는 상기 모니터링 인터벌을 결정하는 것을 특징으로 하는 모바일 장치.

청구항 10

제8항에 있어서, 상기 플로우 익스큐션 플래너는 eci가 플로우 FLOWi의 C-FRAME을 처리하기 위해 필요한 에너지이고, Elimit가 상기 애플리케이션들에게 주어진 에너지 availability일 때, $\sum eci/pi \leq Elimit$ 를 만족하는 상기 모니터링 인터벌을 결정하는 것을 특징으로 하는 모바일 장치.

청구항 11

제5항에 있어서, 복수의 센싱 플로우들은 각각의 F-FRAME 큐(Queue)를 포함하고, 상기 복수의 F-FRAME 큐 중 어느 하나를 선택하여 상기 F-FRAME 처리 순서를 결정하는 플로우 스케줄러를 더 포함하는 것을 특징으로 하는 모바일 장치.

청구항 12

제11항에 있어서, 상기 플로우 스케줄러는 상기 모니터링 딜레이를 기초로 상기 F-FRAME 처리 순서를 결정하는 것을 특징으로 하는 모바일 장치.

청구항 13

제12항에 있어서, 상기 플로우 스케줄러는 satisfy(ci)함수가 상기 F-FRAME 큐 내의 상기 F-FRAME을 수행하여 상기 모니터링 딜레이 내에 i번째 플로우(FLOWi)의 상황 모니터링 결과를 얻을 수 있다면, 1을 나타내고, 그렇지

않은 경우, 0을 나타낸다고 할 때, $\text{maximize } \sum \text{satisfy}(ci)$ 로 상기 F-FRAME 처리 순서를 결정하는 것을 특징으로 하는 모바일 장치.

청구항 14

제13항에 있어서, F-FRAMEi,j가 i번째 C-FRAME(C-FRAMEi)의 j번째 F-FRAME이고, di는 FLOWi의 최대 수용 가능 딜레이이며, ri,j는 상기 C-FRAMEi의 남아있는 모든 F-FRAME들을 모으는 데에 걸리는 시간이고, epti,j는 상기 C-FRAMEi 내에서 수행되지 않은 F-FRAME들을 수행하는 데에 걸리는 시간이라고 할 때, 상기 F-FRAMEi,j의 슬랙 타임은 $\text{slacktime}(F-FRAMEi,j) = di + ri,j - epti,j$ 이고,

상기 플로우 스케줄러는 상기 슬랙 타임이 최소인 F-FRAME을 선택하는 것을 특징으로 하는 모바일 장치.

청구항 15

제5항에 있어서, 상기 플로우 익스큐션 플래너에 CPU 한도를 출력하는 CPU 스케줄러를 더 포함하는 것을 특징으로 하는 모바일 장치.

청구항 16

센싱 핸들러에서, 복수의 센싱 오퍼레이터를 이용하여 상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 단계; 및

프로세싱 핸들러에서, 복수의 프로세싱 오퍼레이터를 이용하여 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하는 단계를 포함하는 센싱 플로우 처리 방법.

청구항 17

제16항에 있어서, 상기 C-FRAME은 복수의 상기 F-FRAME을 포함하는 것을 특징으로 하는 센싱 플로우 처리 방법.

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

모바일 장치의 플로우 애널라이저에서, 애플리케이션으로부터 상황 모니터링 리퀘스트를 수신하는 단계;

상기 모바일 장치의 센싱 핸들러에서, 상기 상황 모니터링 리퀘스트를 기초로, 복수의 센싱 오퍼레이터를 이용하여, 상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 단계;

상기 모바일 장치의 프로세싱 핸들러에서, 상기 상황 모니터링 리퀘스트를 기초로, 복수의 프로세싱 오퍼레이터를 이용하여, 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하여 상황 모니터링 결과를 생성하는 단계; 및

상기 모바일 장치에서, 상기 상황 모니터링 결과를 상기 애플리케이션에 출력하는 단계를 포함하는 상황 모니터링 방법.

청구항 26

센싱 데이터를 생성하는 센서;

상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 복수의 센싱 오퍼레이터를 포함하는 센싱 핸들러; 및

특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하여 상황 모니터링 결과를 생성하는 복수의 프로세싱 오퍼레이터를 포함하는 프로세싱 핸들러를 포함하는 모바일 장치; 및

상기 모바일 장치로부터 상기 상황 모니터링 결과를 수신하는 애플리케이션을 포함하는 상황 모니터링 시스템.

명세서

기술분야

본 발명은 모바일 장치에 관한 것으로, 보다 상세하게는 동시에 수행되는 애플리케이션들의 자원 사용을 조율하여 애플리케이션의 유틸리티를 향상시킬 수 있는 모바일 장치, 이를 이용하는 센싱 플로우 처리 방법, 이를 이

[0001]

용하는 상황 모니터링 방법 및 이를 포함하는 상황 모니터링 방법에 관한 것이다.

배경 기술

- [0002] 최근의 정보통신 패러다임은 유비쿼터스 컴퓨팅(Ubiquitous Computing), 유비쿼터스 네트워크(Ubiquitous Network), 퍼베이시브 컴퓨팅(Pervasive Computing) 등이라 할 수 있다. 이는 모두 동일한 개념으로서, 유비쿼터스가 의미하는 것은 사용자가 언제, 어디서라도 원하는 정보를 손쉽게 얻을 수 있음을 말한다. 향후 도래할 유비쿼터스 시대에는 컴퓨팅 및 커뮤니케이션 능력을 가진 스마트 객체들이 동적인 환경 변화를 인식하고 이에 적응할 수 있는 특성, 즉 상황 인식 특성을 갖게 될 것이다.
- [0003] 개인 영역 네트워크(Personal Area Network; 이하 PAN이라 칭함)는 상황 인식 특성을 갖게 될 유비쿼터스 네트워크를 구현하는 핵심기술이다. PAN은 개인을 중심으로 근거리 이내에서 사용할 수 있는 네트워크 기술을 가리키는데, 이는 네트워크를 사용할 개인을 중심으로 10여 미터 정도까지 네트워크로 다양한 장치들을 연결할 수 있는 기술을 의미한다.
- [0004] 개인이 중심이 된 PAN은 사용자의 행동, 몸 상태, 환경 등과 관련된 상황들에 맞춰 적절한 서비스를 제공하는 상황 모니터링 애플리케이션(Context Monitoring Application)들에 적합하다. PAN 환경에서는 개인을 중심으로 네트워크가 구성되므로, 휴대가 가능한 모바일 장치는 다양한 센서들로부터 데이터를 받아 상황 모니터링 애플리케이션들에게 사용자 상황 정보를 제공하는 핵심 플랫폼이 될 수 있다. 예를 들어, 이동통신 단말기 내에서 상황 정보 수집 및 교환을 통해 사용자가 처한 상황을 모니터링하고, 해석 및 추론과 같은 처리 과정을 거쳐 상황 모니터링 애플리케이션들에게 정보를 제공하고, 상황 모니터링 애플리케이션들은 사용자에게 상황에 적합한 서비스를 제공할 수 있다. 이때에, 다수의 상황 모니터링 애플리케이션을 동시에 효율적으로 지원할 수 있는 모바일 장치가 필요하다.
- [0005] 성장하고 있는 지속적인 모바일 센싱 애플리케이션은 모바일 장치의 워크 로드를 크게 변화시킨다. 또한, 모바일 장치에서 웹 브라우저나 게임과 같은 유저 인터랙티브 애플리케이션도 꾸준히 수행될 수 있다. 이와 같은 동시적인 워크 로드와 의해 자원이 한정적인 모바일 장치에서는 애플리케이션 간의 자원 이용 경쟁이 심각한 문제가 될 수 있다. 또한, 각 센싱 애플리케이션들은 지속적으로 많은 CPU를 사용하게 되어, 이와 같은 자원 이용 경쟁을 심화시키기도 한다. 게다가 상기 센싱 애플리케이션의 서비스를 정상적으로 제공하기 위해서는 상기 센싱 및 프로세싱 워크 로드가 거의 실시간으로 처리되어야 한다. 또한, 사용자는 이와 같은 배경 애플리케이션을 위해 CPU나 배터리를 모두 소모할 수는 없으므로, 모바일 장치의 이용가능한 자원은 더욱 제한될 수밖에 없다. 이와 같은 경쟁적인 상황에서, 자원을 무분별하게 사용하는 애플리케이션은 다른 애플리케이션의 서비스의 퀄리티를 심히 떨어뜨리게 된다. 또한, 이와 같은 상황이 매일 매일 반복될 경우, 사용자의 모바일 장치의 성능 자체를 감소시킬 수 있다.

발명의 내용

해결하려는 과제

- [0006] 상기와 같은 문제점을 해결하기 위한 본 발명의 목적은 동시에 수행되는 애플리케이션들의 자원 사용을 조율하여 애플리케이션의 유틸리티를 향상시킬 수 있는 상황 모니터링 모바일 장치를 제공하는 것이다.
- [0007] 본 발명의 다른 목적은 상기 모바일 장치를 이용하는 센싱 플로우 처리 방법을 제공하는 것이다.
- [0008] 본 발명의 다른 목적은 상기 모바일 장치를 이용하는 상황 모니터링 방법을 제공하는 것이다.
- [0009] 본 발명의 또 다른 목적은 상기 모바일 장치를 포함하는 상황 모니터링 시스템을 제공하는 것이다.

과제의 해결 수단

- [0010] 상기한 본 발명의 목적을 달성하기 위한 일 실시예에 따른 모바일 장치는 센싱 핸들러 및 프로세싱 핸들러를 포함한다. 상기 센싱 핸들러는 상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 복수의 센싱 오퍼레이터를 포함한다. 상기 프로세싱 핸들러는 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하는 복수의 프로세싱 오퍼레이터를 포함한다.
- [0011] 본 발명의 일 실시예에 있어서, 상기 C-FRAME은 복수의 상기 F-FRAME을 포함할 수 있다.

- [0012] 본 발명의 일 실시예에 있어서, 상기 모바일 장치는 애플리케이션으로부터 센싱 플로우 정보를 입력 받아 상기 센싱 플로우의 상기 C-FRAME의 크기 및 상기 F-FRAME의 크기를 판단하는 플로우 애널라이저를 더 포함할 수 있다.
- [0013] 본 발명의 일 실시예에 있어서, 상기 플로우 애널라이저는 상기 애플리케이션으로부터 상기 애플리케이션이 사용자의 상황을 얼마나 자주 모니터링 해야 하는 지를 나타내는 필요 모니터링 인터벌 및 마지막 F-FRAME이 도착한 시점부터 유의미한 상기 상황 모니터링 결과를 얻을 수 있는 시간을 나타내는 상기 모니터링 딜레이를 수신할 수 있다.
- [0014] 본 발명의 일 실시예에 있어서, 상기 모바일 장치는 상기 C-FRAME의 크기, 상기 F-FRAME의 크기 및 상기 필요 모니터링 인터벌을 기초로 상기 C-FRAME의 모니터링 인터벌을 결정하여 상기 센싱 핸들러에 출력하는 플로우 익스큐션 플래너를 더 포함할 수 있다. 상기 센싱 플로우의 모니터링 인터벌은 상기 센싱 타임 및 상기 스킵 타임의 합과 일치할 수 있다.
- [0015] 본 발명의 일 실시예에 있어서, 상기 모바일 장치는 상기 모바일 장치의 CPU 이용률을 판단하여 상기 플로우 익스큐션 플래너에 출력하는 리소스 모니터를 더 포함할 수 있다. 상기 플로우 익스큐션 플래너는 상기 C-FRAME의 크기, 상기 F-FRAME의 크기, 상기 필요 모니터링 인터벌 및 상기 CPU 이용률을 기초로 상기 센싱 플로우의 모니터링 인터벌을 결정할 수 있다.
- [0016] 본 발명의 일 실시예에 있어서, 상기 필요 모니터링 인터벌은 상기 모니터링 인터벌에 따른 상기 애플리케이션의 유틸리티 값으로 정의되는 유틸리티 함수 형태를 가질 수 있다.
- [0017] 본 발명의 일 실시예에 있어서, 상기 플로우 익스큐션 플래너는 복수의 센싱 플로우의 상기 모니터링 인터벌이 pi , 상기 센싱 플로우들의 상기 유틸리티 함수가 ui 일 때, $maximize \sum ui(pi)$ 로 상기 모니터링 인터벌을 결정할 수 있다.
- [0018] 본 발명의 일 실시예에 있어서, 상기 플로우 익스큐션 플래너는 cti 가 플로우 FLOW i 의 C-FRAME을 처리하기 위해 필요한 CPU 타임이고, $1-CPU_f$ 가 상기 애플리케이션들에게 주어진 CPU availability일 때, $\sum cti/pi \leq 1-CPU_f$ 를 만족하는 상기 모니터링 인터벌을 결정할 수 있다.
- [0019] 본 발명의 일 실시예에 있어서, 상기 플로우 익스큐션 플래너는 eci 가 플로우 FLOW i 의 C-FRAME을 처리하기 위해 필요한 에너지이고, $Elimit$ 가 상기 애플리케이션들에게 주어진 에너지 availability일 때, $\sum eci/pi \leq Elimit$ 를 만족하는 상기 모니터링 인터벌을 결정할 수 있다.
- [0020] 본 발명의 일 실시예에 있어서, 상기 모바일 장치는 복수의 센싱 플로우들은 각각의 F-FRAME 큐(Queue)를 포함하고, 상기 복수의 F-FRAME 큐 중 어느 하나를 선택하여 상기 F-FRAME 처리 순서를 결정하는 플로우 스케줄러를 더 포함할 수 있다.
- [0021] 본 발명의 일 실시예에 있어서, 상기 플로우 스케줄러는 상기 모니터링 딜레이를 기초로 상기 F-FRAME 처리 순서를 결정할 수 있다.
- [0022] 본 발명의 일 실시예에 있어서, 상기 플로우 스케줄러는 $satisfy(ci)$ 함수가 상기 F-FRAME 큐 내의 상기 F-FRAME을 수행하여 상기 모니터링 딜레이 내에 i 번째 플로우(FLOW i)의 상황 모니터링 결과를 얻을 수 있다면, 1을 나타내고, 그렇지 않은 경우, 0을 나타낸다고 할 때, $maximize \sum satisfy(ci)$ 로 상기 F-FRAME 처리 순서를 결정할 수 있다.
- [0023] 본 발명의 일 실시예에 있어서, F-FRAME i,j 가 i 번째 C-FRAME(C-FRAME i)의 j 번째 F-FRAME이고, d_i 는 FLOW i 의 최대 수용 가능 딜레이이며, $r_{i,j}$ 는 상기 C-FRAME i 의 남아있는 모든 F-FRAME들을 모으는 데에 걸리는 시간이고, $e_{pti,j}$ 는 상기 C-FRAME i 내에서 수행되지 않은 F-FRAME들을 수행하는 데에 걸리는 시간이라고 할 때, 상기 F-FRAME i,j 의 슬랙 타임은 $slacktime(F-FRAME_{i,j})=d_i+r_{i,j}-e_{pti,j}$ 일 수 있다. 상기 플로우 스케줄러는 상기 슬랙 타임이 최소인 F-FRAME을 선택할 수 있다.
- [0024] 본 발명의 일 실시예에 있어서, 상기 모바일 장치는 상기 플로우 익스큐션 플래너에 CPU 한도를 출력하는 CPU

스케줄러를 더 포함할 수 있다.

- [0025] 상기한 본 발명의 다른 목적을 달성하기 위한 일 실시예에 따른 센싱 플로우 처리 방법은 복수의 센싱 오퍼레이터를 이용하여, 상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 단계 및 복수의 프로세싱 오퍼레이터를 이용하여, 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하는 단계를 포함한다.
- [0026] 본 발명의 일 실시예에 있어서, 상기 C-FRAME은 복수의 상기 F-FRAME을 포함할 수 있다.
- [0027] 본 발명의 일 실시예에 있어서, 애플리케이션으로부터 센싱 플로우 정보를 입력 받아 상기 센싱 플로우의 상기 C-FRAME의 크기 및 상기 F-FRAME의 크기를 판단하는 단계를 더 포함할 수 있다.
- [0028] 본 발명의 일 실시예에 있어서, 상기 애플리케이션으로부터 상기 애플리케이션이 사용자의 상황을 얼마나 자주 모니터링 해야 하는 지를 나타내는 필요 모니터링 인터벌 및 마지막 F-FRAME이 도착한 시점부터 유의미한 상기 상황 모니터링 결과를 얻을 수 있는 시간을 나타내는 상기 모니터링 딜레이를 수신하는 단계를 더 포함할 수 있다.
- [0029] 본 발명의 일 실시예에 있어서, 상기 C-FRAME의 크기, 상기 F-FRAME의 크기 및 상기 필요 모니터링 인터벌을 기초로 상기 C-FRAME의 모니터링 인터벌을 결정하는 단계를 더 포함할 수 있다. 상기 센싱 플로우의 모니터링 인터벌은 상기 센싱 타임 및 상기 스킵 타임의 합과 일치할 수 있다.
- [0030] 본 발명의 일 실시예에 있어서, 모바일 장치의 CPU 이용률을 판단하는 단계를 더 포함할 수 있다. 상기 C-FRAME의 모니터링 인터벌을 결정하는 단계는 상기 C-FRAME의 크기, 상기 F-FRAME의 크기, 상기 필요 모니터링 인터벌 및 상기 CPU 이용률을 기초로 상기 센싱 플로우의 모니터링 인터벌을 결정할 수 있다.
- [0031] 본 발명의 일 실시예에 있어서, 상기 필요 모니터링 인터벌은 상기 모니터링 인터벌에 따른 상기 애플리케이션의 유틸리티 값으로 정의되는 유틸리티 함수 형태를 가질 수 있다.
- [0032] 본 발명의 일 실시예에 있어서, 복수의 센싱 플로우들은 각각의 F-FRAME 큐(QUEUE)를 포함하고, 상기 복수의 F-FRAME 큐 중 어느 하나를 선택하여 상기 F-FRAME 처리 순서를 결정하는 단계를 더 포함할 수 있다.
- [0033] 본 발명의 일 실시예에 있어서, 상기 F-FRAME 처리 순서를 결정하는 단계는 상기 모니터링 딜레이를 기초로 상기 F-FRAME 처리 순서를 결정할 수 있다.
- [0034] 상기한 본 발명의 또 다른 목적을 달성하기 위한 일 실시예에 따른 상황 모니터링 방법은 애플리케이션으로부터 상황 모니터링 리퀘스트를 수신하는 단계, 상기 상황 모니터링 리퀘스트를 기초로, 복수의 센싱 오퍼레이터를 이용하여, 상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 단계, 상기 상황 모니터링 리퀘스트를 기초로, 복수의 프로세싱 오퍼레이터를 이용하여, 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하여 상황 모니터링 결과를 생성하는 단계 및 상기 상황 모니터링 결과를 상기 애플리케이션에 출력하는 단계를 포함한다.
- [0035] 상기한 본 발명의 또 다른 목적을 달성하기 위한 일 실시예에 따른 상황 모니터링 시스템은 센싱 데이터를 생성하는 센서, 상황 모니터링 결과를 생성하기 위해 필요한 센싱 데이터의 시퀀스인 C-FRAME의 크기에 대응하는 센싱 타임 동안 센싱을 수행하고, 스킵 타임 동안 센싱을 중단하는 복수의 센싱 오퍼레이터를 포함하는 센싱 핸들러 및 특성을 추출하기 위해 필요한 센싱 데이터의 시퀀스인 F-FRAME의 단위로 상기 센싱 오퍼레이터의 센싱 데이터를 처리하여 상황 모니터링 결과를 생성하는 복수의 프로세싱 오퍼레이터를 포함하는 프로세싱 핸들러를 포함하는 모바일 장치 및 상기 모바일 장치로부터 상기 상황 모니터링 결과를 수신하는 애플리케이션을 포함한다.

발명의 효과

- [0036] 상기와 같은 본 발명의 실시예에 따른 모바일 장치, 데이터 처리 방법, 상황 모니터링 방법 및 상황 모니터링 시스템은 동시에 수행되는 애플리케이션들의 자원 경쟁적인 워크 로드를 전반적으로 조율하여 애플리케이션의 서비스 품질의 불균형의 문제를 효율적으로 해결할 수 있다. 또한, 개발자는 동시 수행되는 애플리케이션들에 의해 발생할 수 있는 자원 경쟁을 고려하지 않고 쉽게 센싱 애플리케이션을 개발할 수 있다. 따라서, 자원 경쟁적인 애플리케이션의 자원 사용을 조율하여 애플리케이션의 유틸리티를 극대화할 수 있다.
- [0037] 또한, 애플리케이션들의 자원 사용의 변화에 적절히 대응하여 인터랙티브 애플리케이션의 서비스 품질의 감소

를 최소화할 수 있다.

도면의 간단한 설명

- [0038] 도 1은 본 발명의 일 실시예에 따른 상황 모니터링 시스템을 나타내는 블록도이다.
- 도 2는 도 1의 상황 모니터링 시스템을 나타내는 상세 블록도이다.
- 도 3a는 도 1의 모바일 장치에 의해 수행되는 제1 상황 모니터링 애플리케이션의 센싱 플로우를 나타내는 개념도이다.
- 도 3b는 도 1의 모바일 장치에 의해 수행되는 제2 상황 모니터링 애플리케이션의 센싱 플로우를 나타내는 개념도이다.
- 도 3c는 도 1의 모바일 장치에 의해 수행되는 제3 상황 모니터링 애플리케이션의 센싱 플로우를 나타내는 개념도이다.
- 도 4는 도 1의 모바일 장치에 의해 수행되는 센싱 데이터 스트림의 프레임 표면화를 나타내는 개념도이다.
- 도 5a 및 도 5b는 도 1의 모바일 장치에 의해 수행되는 C-FRAME 기반의 자원 할당을 나타내는 개념도이다.
- 도 6은 도 1의 모바일 장치에 의해 수행되는 C-FRAME 기반의 플로우 조율을 나타내는 개념도이다.
- 도 7은 도 2의 플로우 익스큐터의 동작을 나타내는 개념도이다.
- 도 8은 도 2의 플로우 스케줄러에 의해 수행되는 F-FRAME의 슬랙 타임 예측을 나타내는 개념도이다.

발명을 실시하기 위한 구체적인 내용

- [0039] 본문에 개시되어 있는 본 발명의 실시예들에 대해서, 특정한 구조적 내지 기능적 설명들은 단지 본 발명의 실시예를 설명하기 위한 목적으로 예시된 것으로, 본 발명의 실시예들은 다양한 형태로 실시될 수 있으며 본문에 설명된 실시예들에 한정되는 것으로 해석되어서는 아니 된다.
- [0040] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 본문에 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 개시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 구성요소에 대해 사용하였다.
- [0041] 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위로부터 이탈되지 않은 채 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.
- [0042] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.
- [0043] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0044] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적

인 의미로 해석되지 않는다.

- [0045] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 도면 상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.
- [0046] 도 1은 본 발명의 일 실시예에 따른 상황 모니터링 시스템을 나타내는 블록도이다.
- [0047] 도 1을 참조하면, 상황 모니터링 시스템은 모바일 장치(100), 외부 센서(200) 및 상황 모니터링 애플리케이션(300)을 포함한다.
- [0048] 상기 모바일 장치(100)는 상황 모니터링 기능을 기반으로 동작하는 상기 상황 모니터링 애플리케이션(300)으로부터 상황 모니터링 리퀘스트(Context Monitoring Request, CMQ)를 수신하고, 상기 외부 센서(200)로부터 센싱 데이터(SD)를 수신하여 상황 모니터링의 결과(CMR)를 판단한다. 상기 모바일 장치(100)는 상기 상황 모니터링 결과(CMR)를 상기 애플리케이션(300)으로 출력한다. 상기 모바일 장치(100)의 구성에 대해서는 도 2를 참조하여 상세히 설명한다.
- [0049] 상기 외부 센서(200)는 상기 센싱 데이터(SD)를 상기 모바일 장치(100)에 제공한다. 이와는 달리, 상기 외부 센서(200)는 상기 센싱 데이터(SD)로부터 도출된 특성(Feature)을 상기 모바일 장치(100)에 전달할 수 있다.
- [0050] 예를 들어, 상기 외부 센서(200)는 복수 개의 외부 센서들(200)일 수 있다. 상기 외부 센서(200)는 빛 센서, 온도 센서, 위치 센서, 먼지 센서, 자외선 센서, 온도계, 습도계, 이산화탄소 측정기, 주변 소리 측정기, 가속도 측정기 등이 될 수 있고, 각각의 외부 센서(200)들이 센싱하는 상황 요소는 빛, 온도, 위치, 먼지량, 자외선량, 온도, 습도, 이산화탄소량, 주변 소리, 가속도 등이 될 수 있다. 상기 외부 센서(200)는 상기 모바일 장치(100)의 사용자의 신체에 착용하는 웨어러블(Wearable) 센서일 수 있다.
- [0051] 상기 외부 센서(200)에서 측정된 상기 센싱 데이터(SD)는 상기 모바일 장치(100)에 제공되어 상기 상황 인식 애플리케이션(300)이 요구한 상기 상황(CONTEXT)의 만족 여부를 모니터링하는 데 활용된다.
- [0052] 상기 상황 모니터링 애플리케이션(300)은 프로그램 목적에 따른 상기 상황 모니터링 리퀘스트를 상기 모바일 장치(100)에 출력한다. 상기 상황 모니터링 애플리케이션(300)은 상기 모바일 장치(100)로부터 상기 상황 모니터링 결과(CMR)를 수신한다.
- [0053] 도 2는 도 1의 상황 모니터링 시스템을 나타내는 상세 블록도이다.
- [0054] 도 1 및 도 2를 참조하면, 상기 모바일 장치(100)는 플로우 애널라이저(110), 리소스 모니터(120), 플로우 스케줄러(130), 플로우 익스큐션 플래너(140), 플로우 익스큐터 및 CPU 스케줄러(170)를 포함한다. 상기 플로우 익스큐터는 프로세싱 핸들러(150) 및 센싱 핸들러(160)를 포함한다. 상기 모바일 장치(100)는 내부센서(180)를 더 포함할 수 있다.
- [0055] 상기 플로우 애널라이저(110)는 상기 상황 모니터링 애플리케이션(300)으로부터 상황 모니터링 리퀘스트(CMQ)를 입력 받는다. 상기 상황 모니터링 리퀘스트(CMQ)는 센싱 플로우 정보 및 익스큐션 리콰이어먼트를 포함할 수 있다.
- [0056] 상기 센싱 플로우 정보는 상기 상황 모니터링 애플리케이션(300)이 필요로 하는 센싱 정보의 플로우를 나타낸다. 상기 센싱 플로우 정보는 도 3a 내지 3c에 도시된 데이터 플로우 그래프에 대응된다.
- [0057] 예를 들어, 상기 센싱 플로우 정보는 XML 인터페이스의 데이터플로우 프로그래밍 모델일 수 있다. 상기 XML 인터페이스의 상기 센싱 플로우 정보는 오퍼레이터 및 상기 오퍼레이터 사이의 에지를 나타낼 수 있다. 상기 오퍼레이터는 계산 또는 입력력의 수행 단위이다. 상기 에지는 상기 오퍼레이터 간의 종속성을 나타낸다.
- [0058] 상기 익스큐션 리콰이어먼트는 필요 모니터링 인터벌 및 모니터링 딜레이를 포함한다.
- [0059] 상기 필요 모니터링 인터벌은 사용자의 상황을 얼마나 자주 모니터링 해야 하는지를 나타낸다. 예를 들어, 칼로리몬이라는 애플리케이션은 사용자의 칼로리 소모량을 계산하기 위해 사용자의 물리적 활동을 수초마다 모니터링 해야 한다. 상기 모니터링 인터벌이 짧을수록 상기 상황 모니터링 애플리케이션(300)의 유틸리티는 증가하게 된다. 상기 필요 모니터링 인터벌 정보는 상기 모니터링 인터벌에 따른 상기 상황 모니터링 애플리케이션(300)의 유틸리티를 나타내는 유틸리티 함수로 표현될 수 있다. 상기 모바일 장치(100)는 상기 상황 모니터링 애플리케이션(300)의 유틸리티를 최대화하는 방향으로 자원 사용을 조절할 수 있다.
- [0060] 상기 모니터링 딜레이는 상기 상황 모니터링 애플리케이션(300)의 최대 수용 가능 딜레이를 의미한다. 상기 딜

레이는 센싱 순간으로부터 최종 상황 모니터링 결과(CMR)를 출력하기까지의 시간을 나타낸다.

- [0061] 또한, 상기 익스큐션 리콰이어먼트는 모니터링 결과의 정확성(accuracy)을 더 포함할 수 있다.
- [0062] 상기 플로우 애널라이저(110)는 상기 센싱 플로우를 기초로 프레임 구조를 판단한다. 상기 플로우 애널라이저(110)는 상기 센싱 플로우를 기초로 C-FRAME의 크기 및 F-FRAME의 크기를 판단할 수 있다. 상기 플로우 애널라이저(110)는 상기 C-FRAME의 크기 및 F-FRAME의 크기를 상기 플로우 익스큐션 플래너(140)에 출력한다.
- [0063] 일반적으로 센싱 데이터 스트림은 데이터의 의미와는 관계없는 반복적인 데이터 시퀀스에 불과하다. 그러나 상기 애플리케이션(300)에서는 상기 데이터 스트림으로부터 특정한 상황을 추출 및 판단하게 된다. 자원 경쟁적인 상황에서 복수의 애플리케이션들이 동시에 수행되어야 할 때, 센싱 데이터 스트림의 프레임 구조는 애플리케이션들에게 자원을 분배하기 위한 중요한 힌트가 될 수 있다.
- [0064] 애플리케이션 A가 센싱 데이터 스트림 S를 입력받겠다고 할 때, 상기 애플리케이션 A의 플로우를 판단함으로써 상기 센싱 데이터 스트림 S의 가상 구조를 추출할 수 있다. 상기 애플리케이션 A에 대한 상기 센싱 데이터 스트림 S의 가상 구조를 추출하는 것을 프레임링(framing) 또는 프레임 표면화(frame externalization)라고 한다. 상기 프레임 표면화를 거친 상기 센싱 데이터 스트림 S의 개체를 프레임이라고 할 수 있다. 상기 센싱 데이터 스트림은 애플리케이션(300)에 따라 서로 다르게 프레임링될 수 있다. 그뿐만 아니라, 상기 센싱 데이터 스트림은 동일 애플리케이션(300) 내에서도 서로 다르게 프레임링될 수 있다.
- [0065] 상기 C-FRAME은 context-frame의 약자로, 상황 모니터링 결과(CMR)를 생성하기 위해 필요한 센싱 데이터의 시퀀스를 의미한다. 상기 모바일 장치(100)는 상기 C-FRAME을 데이터 프로세싱 및 자원 할당의 기본 단위로 사용한다.
- [0066] 상기 F-FRAME은 feature-frame의 약자로, 특성 추출 오퍼레이션을 수행하기 위해 필요한 센싱 데이터의 시퀀스를 의미한다. 상기 C-FRAME은 복수의 F-FRAME들을 포함한다.
- [0067] 상기 센싱 데이터 스트림의 프레임링에 대해서는 도 4를 참조하여 상세히 설명한다.
- [0068] 상기 플로우 애널라이저(110)는 상기 익스큐션 리콰이어먼트 중 상기 모니터링 딜레이를 상기 플로우 스케줄러(130)에 출력한다. 상기 플로우 애널라이저(110)는 상기 익스큐션 리콰이어먼트 중 상기 모니터링 인터벌을 상기 플로우 익스큐션 플래너(140)에 출력한다. 예를 들어, 상기 모니터링 인터벌은 상기 모니터링 인터벌에 따른 유틸리티 값으로 표현되는 유틸리티 함수의 형태를 가질 수 있다.
- [0069] 상기 리소스 모니터(120)는 상기 플로우 익스큐터로부터 프로파일 데이터를 수신한다. 상기 프로파일 데이터는 상기 모바일 장치(100)의 CPU 이용률 및 에너지 이용률을 포함할 수 있다. 상기 CPU 이용률 및 상기 에너지 이용률은 다른 애플리케이션의 영향에 따라 실시간으로 변화할 수 있다. 상기 프로파일 데이터는 상기 프로세싱 핸들러(150) 및 상기 센싱 핸들러(160)의 CPU 이용률 및 에너지 이용률을 포함할 수 있다.
- [0070] 상기 리소스 모니터(120)는 상기 CPU 이용률 및 상기 에너지 이용률을 상기 플로우 익스큐션 플래너(140)에 출력한다.
- [0071] 예를 들어, 상기 CPU 이용률은 C-FRAME 수행의 시작과 종료 사이의 시간인 C-FRAME 수행의 경과 시간(elapsed time)을 이용하여 판단될 수 있다.
- [0072] 예를 들어, 상기 에너지 이용률은 오프라인 프로파일링 방법을 이용하여 판단될 수 있다. 상기 오프라인 프로파일링 방법에서는 센싱 애플리케이션에 의해 사용되는 주요 하드웨어 컴포넌트에 해당하는 센서, CPU 및 네트워크 인터페이스의 에너지 프로파일을 미리 계산하여 놓고, 상기 에너지 프로파일에 C-FRAME을 수집 및 처리하는 오퍼레이션을 수행하는 데에 필요한 에너지를 합산하여 상기 C-FRAME의 에너지 이용률을 판단한다.
- [0073] 상기 플로우 스케줄러(130)는 상기 플로우 애널라이저(110)로부터 상기 모니터링 딜레이를 입력받는다. 상기 플로우 스케줄러(130)는 상기 모니터링 딜레이를 기초로 상기 센싱 핸들러의 F-FRAME의 실행 순서를 결정한다.
- [0074] 상기 플로우 스케줄러(130)가 상기 F-FRAME의 실행 순서를 결정하는 방법에 대해서는 도 8을 참조하여 상세히 설명한다.
- [0075] 상기 플로우 익스큐션 플래너(140)는 상기 플로우 애널라이저(110)로부터 상기 C-FRAME의 크기, 상기 F-FRAME의 크기 및 상기 모니터링 인터벌을 수신한다. 상기 플로우 익스큐션 플래너(140)는 상기 리소스 모니터(120)로부터 상기 CPU 이용률 및 상기 에너지 이용률을 수신한다. 상기 플로우 익스큐션 플래너(140)는 상기 CPU 스케줄

러(170)로부터 상기 센싱 애플리케이션의 수행에 허용되는 CPU 한도를 더 수신할 수 있다.

[0076] 상기 플로우 익스큐션 플래너(140)는 상기 C-FRAME의 크기, 상기 F-FRAME의 크기 및 상기 필요 모니터링 인터벌을 이용하여 상기 센싱 핸들러(160)의 센싱 플로우를 제어한다.

[0077] 상기 플로우 익스큐션 플래너(140)는 상기 C-FRAME의 크기, 상기 F-FRAME의 크기 및 상기 필요 모니터링 인터벌을 이용하여 상기 센싱 핸들러(160)의 센싱 오퍼레이터의 모니터링 인터벌을 결정한다. 상기 C-FRAME의 크기 및 상기 모니터링 인터벌에 의해 skip 타임이 결정될 수 있다. 상기 센싱 핸들러(160)는 상기 C-FRAME의 크기에 대응하여 센싱을 수행하고, 상기 skip 타임만큼 센싱을 중단한다. 상기 센싱 핸들러(160)는 상기 센싱의 수행 및 중단을 상기 모니터링 인터벌을 주기로 반복한다.

[0078] 상기 플로우 익스큐션 플래너(140)는 상기 센싱 오퍼레이터가 복수 개일때, 각 센싱 오퍼레이터의 상기 모니터링 인터벌을 결정할 수 있다.

[0079] 상기 플로우 익스큐션 플래너(140)는 상기 C-FRAME의 크기, 상기 F-FRAME의 크기, 상기 필요 모니터링 인터벌, 상기 CPU 이용률 및 상기 에너지 이용률을 이용하여 상기 센싱 핸들러(160)의 센싱 플로우를 제어할 수 있다. 상기 플로우 익스큐션 플래너(140)는 상기 CPU 한도를 더 고려하여 상기 센싱 핸들러(160)의 센싱 플로우를 제어할 수 있다. 상기 CPU 한도는 상기 상황 모니터링 애플리케이션(300)에 주어진 CPU 이용률의 한도를 의미하며, 운영 체제(Operating system)에 의해 설정되거나, 사용자에게 의해 설정될 수 있다.

[0080] 동시에 주어진 복수의 센싱 플로우는 FLOWi라하고, 상기 C-FRAME 모니터링 인터벌을 pi라하면, 상기 플로우 익스큐션 플래너(140)는 아래 수식 1 내지 3을 기초로 상기 C-FRAME 모니터링 인터벌(pi)을 결정할 수 있다.

[0081] [수식 1]

$$\text{maximize } \sum u_i(p_i)$$

[0082]

[0083] [수식 2]

$$\sum c_{ti}/p_i \leq 1 - CPU_f$$

[0084]

[0085] [수식 3]

$$\sum e_{ci}/p_i \leq E_{limit}$$

[0086]

[0087] 수식 1에서, u_i 는 상기 FLOWi의 유틸리티 함수를 나타낸다. 상기 유틸리티 함수는 모니터링 인터벌(pi)에 대한 애플리케이션의 유틸리티 수치로 표현된다. 상기 플로우 익스큐션 플래너(140)는 복수의 센싱 플로우에 대해 유틸리티를 최대화할 수 있도록 상기 모니터링 인터벌(pi)을 결정한다.

[0088] 수식 2에서, c_{ti} 는 FLOWi의 C-FRAME을 처리하기 위해 필요한 CPU 타임을 의미하고, CPU_f 는 다른 애플리케이션들에 의해 점유된 CPU를 의미한다. $1 - CPU_f$ 는 센싱 애플리케이션들에게 주어진 CPU availability라고 할 수 있다. 물론, 상기 CPU availability는 다른 애플리케이션들에 의해 점유된 CPU인 CPU_f 와 무관하게 사용자에게 의해 설정될 수 있다. 상기 플로우 익스큐션 플래너(140)는 수식 2의 CPU 제약 조건 내에서 상기 모니터링 인터벌(pi)을 결정한다.

[0089] 수식 3에서, e_{ci} 는 FLOWi의 C-FRAME을 처리하기 위해 필요한 에너지량을 의미하고, E_{limit} 은 센싱 애플리케이션들에게 주어진 에너지 availability라고 할 수 있다. 상기 플로우 익스큐션 플래너(140)는 수식 3의 에너지 제약 조건 내에서 상기 모니터링 인터벌(pi)을 결정한다.

[0090] 상기 CPU 이용률은 C-FRAME 수행의 시작과 종료 사이의 시간인 C-FRAME 수행의 경과 시간(e_{ti})을 이용하여 판단될 수 있다. 상기 경과 시간을 이용할 경우, 수식 2의 c_{ti} 는 $e_{ti} * (1 - CPU_f)$ 로 근사될 수 있다. 이 때, 상기 수식 2는 아래 수식 4로 표현될 수 있다.

[0091] [수식 4]

$$\sum e_{ti}/p_i \leq 1$$

[0092]

- [0093] 상기 플로우 익스큐션 플래너(140)의 일부 동작에 대해서는 도 5 및 도 6을 참조하여 상세히 설명한다.
- [0094] 상기 프로세싱 핸들러(150)는 복수의 프로세싱 오퍼레이터를 포함한다. 상기 프로세싱 핸들러(150)는 상기 프로세싱 오퍼레이터들의 동작을 제어한다. 상기 프로세싱 핸들러(150)는 상기 플로우 스케줄러(130)의 스케줄링에 따라 상기 프로세싱 오퍼레이터들의 동작을 제어한다. 상기 프로세싱 오퍼레이터는 F-FRAME의 단위로 동작할 수 있다.
- [0095] 상기 센싱 핸들러(160)는 복수의 센싱 오퍼레이터를 포함한다. 상기 센싱 핸들러(160)는 상기 플로우 익스큐션 플래너(140)의 플래닝에 따라 상기 센싱 오퍼레이터들의 동작을 제어한다. 상기 센싱 오퍼레이터들은 상기 플로우 익스큐션 플래너(140)의 모니터링 인터벌 정보를 수신하여 상기 C-FRAME 단위로 센싱을 진행하고, 상기 skip 타입에 대응하여 센싱을 일시 정지한다.
- [0096] 상기 프로세싱 핸들러(150) 및 상기 센싱 핸들러(160)의 동작에 대해서는 도 7을 참조하여 상세히 설명한다.
- [0097] 상기 CPU 스케줄러(170)는 상기 CPU 한도를 상기 플로우 익스큐션 플래너(140)에 출력한다. 상기 CPU 스케줄러(170)는 운영 체제 내에 포함될 수 있다.
- [0098] 상기 내부 센서(180)는 내부 센싱 데이터(ISD)를 상기 센싱 핸들러(160)의 센싱 오퍼레이터에 전달한다. 이와는 달리, 상기 내부 센서(180)는 상기 내부 센싱 데이터(ISD)로부터 도출된 특성(Feature)을 상기 센싱 오퍼레이터에 전달할 수 있다.
- [0099] 예를 들어, 상기 내부 센서(180)는 복수 개의 내부 센서들(180)일 수 있다. 상기 내부 센서(180)는 빛 센서, 온도 센서, 위치 센서, 먼지 센서, 자외선 센서, 온도계, 습도계, 이산화탄소 측정기, 주변 소리 측정기, 가속도 측정기 등이 될 수 있고, 각각의 내부 센서(180)들이 센싱하는 상황 요소는 빛, 온도, 위치, 먼지량, 자외선량, 온도, 습도, 이산화탄소량, 주변 소리, 가속도 등이 될 수 있다.
- [0100] 상기 내부 센서(180)에서 측정된 상기 내부 센싱 데이터(ISD)는 상기 센싱 핸들러(160)에 제공되어 상기 상황 인식 애플리케이션(300)이 요구한 상기 상황(CONTEXT)의 만족 여부를 모니터링하는 데 활용된다.
- [0101] 도 3a는 도 1의 모바일 장치(100)에 의해 수행되는 제1 상황 모니터링 애플리케이션의 센싱 플로우를 나타내는 개념도이다. 도 3b는 도 1의 모바일 장치(100)에 의해 수행되는 제2 상황 모니터링 애플리케이션의 센싱 플로우를 나타내는 개념도이다. 도 3c는 도 1의 모바일 장치(100)에 의해 수행되는 제3 상황 모니터링 애플리케이션의 센싱 플로우를 나타내는 개념도이다.
- [0102] 도 3a 내지 도 3c에서, 최하단의 사각형 박스는 외부 센서 또는 내부 센서를 나타내고, 모따기된 사각형 박스는 센싱 오퍼레이터 및 프로세싱 오퍼레이터를 나타내며, 상기 오퍼레이터 사이에 표시된 숫자는 센싱 데이터 스트림의 윈도우 사이즈를 나타낸다.
- [0103] 도 3a 내지 도 3c를 참조하면, 도 3a는 ChildMon이라는 상황 모니터링 애플리케이션의 센싱 플로우를 나타낸다. 상기 ChildMon은 근무 중인 부모가 수업 중이거나 야외 활동 중인 아동의 활동을 인지할 수 있도록 해준다. ChildMon은 아동의 책방에 부착된 모바일 장치를 통해 아동들의 대화, 놀이와 같은 활동을 인지하게 해준다. 상기 아동의 활동을 파악하기 위해 소리를 센싱하게 된다.
- [0104] 도 3a에서 센싱 오퍼레이터인 SOUND는 8kHz의 주파수로 오디오 데이터를 샘플링한다. 상기 오디오 데이터는 상기 SOUND에 연결된 두 개의 프로세싱 오퍼레이터인 RMS 및 FFT에게 전달된다. 상기 RMS 및 상기 FFT의 처리 결과는 일련의 프로세스를 거쳐 GMM(Gaussian Mixture Model) 및 SMOOTHING 오퍼레이터로 전달되고, 상기 SMOOTHING에 의해 아동의 활동이 인지된다.
- [0105] 도 3b는 IndoorNavi라는 상황 모니터링 애플리케이션의 센싱 플로우를 나타낸다. 상기 IndoorNavi는 대규모 빌딩 내에서 모바일 장치 사용자의 길을 인도해 준다. 길 안내를 위해, 애플리케이션은 모바일 장치의 위치를 꾸준히 로컬라이즈한다.
- [0106] 도 3b에서 센싱 오퍼레이터인 SOUND는 44.1kHz의 주파수로 계속하여 사운드 데이터를 생성하여, 다음 프로세싱 원시 오퍼레이터인 WINDOW FUNCTION에 전달한다. 상기 WINDOW FUNCTION의 처리 결과는 일련의 프로세스를 거쳐 NEAREST-NEIGHBOR에 전달된다.
- [0107] 도 3c는 CalorieMon이라는 상황 모니터링 애플리케이션의 센싱 플로우를 나타낸다. 상기 CalorieMon은 사용자의 하루 동안의 육체적 활동을 감지하여 사용자의 칼로리 소모를 실시간으로 계산한다. 본 애플리케이션은 사용자

의 몸체에 부착된 복수의 가속도계를 이용한다.

- [0108] 도 3c에서 센싱 오퍼레이터인 ACCEL_X, ACCEL_Y 등은 50Hz의 주파수로 가속도 데이터를 샘플링한다. 상기 가속도 데이터는 상기 ACCEL_X, ACCEL_Y 등에 연결된 FFT, CORRELATION 등의 프로세싱 오퍼레이터에게 전달된다. 상기 FFT, CORRELATION의 처리 결과는 일련의 프로세스를 거쳐 DECISION TREE 오퍼레이터로 전달되고, 상기 DECISION TREE에 의해 사용자의 칼로리 소모량이 계산된다.
- [0109] 도 4는 도 1의 모바일 장치에 의해 수행되는 센싱 데이터 스트림의 프레임 표면화를 나타내는 개념도이다.
- [0110] 도 4는 도 3a의 ChildMon에서 사운드 스트림을 C-FRAME 및 F-FRAME으로 프레임화하는 경우를 도시한다.
- [0111] 도 1 내지 도 4를 참조하면, 상기 플로우 애널리저(110)는 상기 ChildMon의 센싱 플로우를 기초로 C-FRAME의 크기 및 F-FRAME의 크기를 판단할 수 있다.
- [0112] 도 4를 보면, 3개의 레이어에서 8kHz의 사운드 스트림이 ChildMon에 의해 프레임화된다. 제1 레이어에서 특성 추출을 위해 512의 연속된 샘플의 시퀀스가 프레임화된다. 제2 레이어에서 클래시피케이션을 위해 20개의 제1 레이어 프레임들이 결합된다. 제3 레이어에서 최종 상황 모니터링 결과를 생성하기 위해 3개의 제2 레이어 프레임들이 다시 결합된다. 결과적으로 512*20*3개의 샘플들이 최종 상황 모니터링 결과를 생성한다. 여기서, 상기 512개의 샘플이 F-FRAME의 크기가 된다. 512*20*3개의 샘플이 C-FRAME의 크기가 된다.
- [0113] 도 5a 및 도 5b는 도 1의 모바일 장치(100)에 의해 수행되는 C-FRAME 기반의 자원 할당을 나타내는 개념도이다. 도 6은 도 1의 모바일 장치(100)에 의해 수행되는 C-FRAME 기반의 플로우 조율을 나타내는 개념도이다.
- [0114] 도 1 내지 도 6을 참조하면, 상기 C-FRAME은 데이터 프로세싱 및 자원 할당의 기본 단위가 된다. 정확한 상황 모니터링 결과를 얻기 위해서는 상기 C-FRAME의 수집 및 그에 따른 오퍼레이션의 동작을 중간에 중단하여서는 안된다. 반면, C-FRAME의 처리가 완전히 끝난 이후에는 상기 오퍼레이션의 일시 중단을 하더라도 큰 문제가 발생하지는 않는다.
- [0115] 따라서, C-FRAME의 수집 및 처리는 자원 할당의 기본 단위가 되어야 한다. 각각의 C-FRAME에는 상기 C-FRAME의 수집 및 처리를 위해 충분한 자원이 확보된다. 상기 플로우는 C-FRAME의 인터벌을 설정함으로써 조율된다. 상기 C-FRAME의 인터벌은 상기 C-FRAME의 인터벌내에서 상기 C-FRAME의 처리를 담보할 수 있는 충분한 자원이 확보되도록 설정될 수 있다.
- [0116] 도 5a에서, 현재 모바일 장치(100)에서 이용 가능한 CPU는 40%인 경우를 가정한다. 상기 C-FRAME의 길이는 2sec이고, 이를 처리하기 위한 CPU time은 500ms이다. 따라서, 상기 C-FRAME의 최대 CPU 이용률은 500ms/2sec, 25%에 해당한다.
- [0117] 상기 C-FRAME의 최대 CPU 이용률에 비해 상기 모바일 장치(100)의 이용 가능한 CPU가 크므로, 상기 C-FRAME은 최대 CPU를 이용하도록 할 수 있다. 따라서, 상기 C-FRAME의 인터벌은 2sec로 설정될 수 있다. 도 5a에서, 센싱 오퍼레이터가 센싱을 일시 중단하는 스킵 타임은 없다. 따라서, 상기 센싱 오퍼레이터는 센싱을 계속하여 진행한다.
- [0118] 도 5b에서, 현재 모바일 장치(100)에서 이용 가능한 CPU는 10%인 경우를 가정한다. 상기 C-FRAME의 길이는 2sec이고, 이를 처리하기 위한 CPU time은 500ms이다. 따라서, 상기 C-FRAME의 최대 CPU 이용률은 25%에 해당한다.
- [0119] 상기 C-FRAME의 최대 CPU 이용률에 비해 상기 모바일 장치(100)의 이용 가능한 CPU가 작으므로, 상기 C-FRAME은 최대 CPU를 이용하도록 할 수 없으며, 상기 C-FRAME은 10% 이하의 CPU를 이용하여야 한다. 따라서, 상기 C-FRAME의 인터벌은 5sec로 설정될 수 있고, 이 때의 C-FRAME의 CPU 이용률은 500ms/5sec, 10%에 해당한다.
- [0120] 도 5b에서, 센싱 오퍼레이터가 센싱을 진행하는 시간은 2sec이고, 센싱을 일시 중단하는 스킵 타임은 3sec이다. 따라서, 상기 센싱 오퍼레이터는 센싱의 진행 및 중단을 5sec 주기로 반복한다.
- [0121] 복수의 플로우가 공존하는 경우, 서로 다른 플로우에 대해 서로 다른 C-FRAME 인터벌을 갖도록 상기 플로우의 조율이 수행될 수 있다. 이 때, 주어진 자원 상황에서 상기 애플리케이션들이 최대의 유틸리티 합을 갖도록 상기 플로우들이 조율될 수 있다.
- [0122] 도 6에서, 이용 가능한 CPU가 40%를 나타낼 때, C-FRAME의 길이가 2sec인 플로우 f0의 센싱 스트림은 2sec의 인터벌로 센싱되며, 이 때, 플로우 f0의 CPU 이용률은 500ms/2sec, 25%이다. C-FRAME의 길이가 3sec인 플로우 f1의 센싱 스트림은 3sec의 인터벌로 센싱되며, 이 때, 플로우 f1의 CPU 이용률은 200ms/3sec, 6.67%이다. 상기

플로우 f0의 CPU 이용률 및 상기 플로우 f1의 CPU 이용률을 합하더라도 상기 이용 가능한 CPU 40%보다 작으므로, 상기 플로우 f0 및 상기 플로우 f1은 스킵 타임 없이 최대로 CPU를 이용할 수 있다.

[0123] 반면, 이용 가능한 CPU가 40%에서 10%로 변경되면, 상기 플로우 f0 및 상기 플로우 f1의 CPU 이용률이 조정되어야 한다. 이 때, 상기 플로우 f0 및 상기 플로우 f1의 CPU 이용률을 결정함에 있어, 상기 플로우 f0의 유틸리티 함수 및 상기 플로우 f1의 유틸리티 함수를 고려할 수 있다. 상기 플로우 익스큐션 플래너(140)는 주어진 자원의 제약 속에서 상기 플로우 f0 및 상기 플로우 f1의 유틸리티가 최대가 되도록 상기 플로우 f0의 C-FRAME 인터벌 및 상기 플로우 f1의 C-FRAME의 인터벌을 결정할 수 있다. 예를 들어, 상기 플로우 f0의 C-FRAME의 인터벌은 7.1sec로 결정될 수 있다. 이 때, 상기 플로우 f0의 스킵 타임은 5.1sec일 수 있다. 상기 플로우 f0의 CPU 이용률은 500ms/7.1sec, 약 7.0%이다. 예를 들어, 상기 플로우 f1의 C-FRAME의 인터벌은 6.7sec로 결정될 수 있다. 이 때, 상기 플로우 f1의 스킵 타임은 4.7sec일 수 있다. 상기 플로우 f1의 CPU 이용률은 500ms/7.1sec, 약 3.0%이다.

[0124] 상기한 바와 같이, 본 발명의 모바일 장치(100)에서 자원의 할당은 C-FRAME 단위로 수행되는 반면, 실제 플로우의 처리는 F-FRAME 단위로 수행된다. 따라서, 플로우 처리의 스케줄링에 유연성이 확보된다. 예를 들어, 하나의 C-FRAME은 상대적으로 큰 사이즈를 갖는다.(IndoorNavi에서 약 4MB에 해당) 따라서, 전체 C-FRAME을 하나의 단위로 처리하기 위해서는 갑작스럽게 높은 CPU 이용률을 필요로 하게 되고 긴 딜레이를 유발하게 된다. 그러나, 본 실시예에서는 C-FRAME보다 훨씬 작은 사이즈를 갖는 F-FRAME 단위로 플로우가 처리되므로 자원 경쟁적인 환경에서 딜레이 리콰이어먼트를 충족시킬 수 있도록 애플리케이션의 스케줄링을 유연하게 처리할 수 있다.

[0125] 도 7은 도 2의 플로우 익스큐터의 동작을 나타내는 개념도이다.

[0126] 도 1 내지 도 7을 참조하면, 상기 모바일 시스템은 가속도계(Accel), 자이로스코프(Gyro) 및 마이크(Mic.) 센서를 포함한다.

[0127] 상기 센싱 핸들러(160)는 상기 가속도계(Accel)의 센싱 스트림을 처리하는 제1 센싱 오퍼레이터(S01), 상기 자이로스코프(Gyro)의 센싱 스트림을 처리하는 제2 센싱 오퍼레이터(S02) 및 상기 마이크(Mic.) 센서의 센싱 스트림을 처리하는 제3 센싱 오퍼레이터(S03)를 포함한다.

[0128] 상기 제1 내지 제3 센싱 오퍼레이터(S01, S02, S03)는 각각의 C-FRAME의 길이에 대응하는 센싱 타임동안 계속하여 센싱을 수행하고, 모니터링 인터벌에서 센싱 타임을 뺀 스킵 타임동안 센싱을 일시 중단한다. 상기 제1 내지 제3 센싱 오퍼레이터(S01, S02, S03)는 서로 다른 C-FRAME 사이즈를 가질 수 있다. 상기 제1 내지 제3 센싱 오퍼레이터(S01, S02, S03)는 서로 다른 모니터링 인터벌을 가질 수 있다.

[0129] 상기 센싱 핸들러(160)는 상기 센싱 오퍼레이터들에 대해 별도의 스레드를 관리한다. 상기 제1 센싱 오퍼레이터(S01)는 제1 F-FRAME 큐(QUEUE)에 가속도 센싱 스트림을 F-FRAME 단위로 저장한다. 상기 제2 센싱 오퍼레이터(S02)는 제2 F-FRAME 큐에 자이로 센싱 스트림을 F-FRAME 단위로 저장한다. 상기 제2 센싱 오퍼레이터(S02)는 제2 F-FRAME 큐에 자이로 센싱 스트림을 F-FRAME 단위로 저장한다. 상기 제3 센싱 오퍼레이터(S03)는 제3 F-FRAME 큐에 사운드 센싱 스트림을 F-FRAME 단위로 저장한다. 상기 제1 내지 제3 센싱 오퍼레이터(S01, S02, S03)는 서로 다른 F-FRAME 사이즈를 가질 수 있다.

[0130] 상기 프로세싱 핸들러(150)는 연속적인 프로세싱 핸들러의 동작을 제어한다. 상기 플로우 스케줄러(130)가 하나의 플로우를 선택하면, 상기 프로세싱 핸들러(150)는 상기 플로우의 F-FRAME 큐에서 하나의 F-FRAME을 취하여 상기 프로세싱 오퍼레이터들을 이용하여 F-FRAME을 처리한다. 예를 들어, 상기 플로우 스케줄러(130)가 제1 플로우를 선택한 경우, 상기 제1 F-FRAME 큐에서 가속도 F-FRAME을 취하여 상기 프로세싱 오퍼레이터들(P01 내지 P05)을 이용하여 상기 가속도 F-FRAME을 처리한다. 예를 들어, 상기 플로우 스케줄러(130)가 제2 플로우를 선택한 경우, 상기 제2 F-FRAME 큐에서 자이로 F-FRAME을 취하여 상기 프로세싱 오퍼레이터들(P06 내지 P08)을 이용하여 상기 자이로 F-FRAME을 처리한다. 예를 들어, 상기 플로우 스케줄러(130)가 제3 플로우를 선택한 경우, 상기 제3 F-FRAME 큐에서 사운드 F-FRAME을 취하여 상기 프로세싱 오퍼레이터들(P09 내지 P14)을 이용하여 상기 사운드 F-FRAME을 처리한다.

[0131] 도 8은 도 2의 플로우 스케줄러(130)에 의해 수행되는 F-FRAME의 슬랙 타임 예측을 나타내는 개념도이다.

[0132] 도 1 내지 도 8을 참조하면, 상기 플로우 스케줄러(130)는 상기 F-FRAME 큐에 누적된 F-FRAME들의 수행 순서를 결정한다. 예를 들어, 상기 플로우 스케줄러(130)는 F-FRAME들의 수행 순서를 결정할 때, 아래 수식 5를 이용할 수 있다.

[0133] [수식 5]

$$\text{maximize } \sum \text{satisfy}(ci)$$

[0134]

[0135] 상기 satisfy(ci)함수는 상기 F-FRAME 큐 내의 F-FRAME을 수행하여 최대 수용 가능한 딜레이 내에 i번째 플로우(FLowi)의 상황 모니터링 결과를 얻을 수 있다면, 1을 나타내고, 그렇지 않은 경우, 0을 나타낸다. 상기 최대 수용 가능한 딜레이는 C-FRAME 내의 마지막 F-FRAME이 도착한 시점부터 유의미한 상황 모니터링 결과를 얻을 수 있는 시간을 의미한다.

[0136] 즉, 상기 플로우 스케줄러(130)는 상기 F-FRAME을 수행함으로써 인해 상기 수용 가능한 딜레이 내에서 가장 많은 상황 모니터링 결과를 얻을 수 있도록 상기 F-FRAME들의 수행 순서를 결정한다.

[0137] 본 발명의 실시예에서, 상기 플로우 스케줄러(130)는 최소 슬랙 타임(least slack time) 방식을 이용할 수 있다. 상기 플로우 스케줄러(130)는 상기 F-FRAME 큐 내에 저장된 모든 F-FRAME에 대해 슬랙 타임을 계산한다. 상기 플로우 스케줄러(130)는 상기 F-FRAME 중 최소 슬랙 타임을 갖는 F-FRAME을 수행하도록 상기 프로세싱 핸들러(150)를 제어한다.

[0138] F-FRAME_{i,j}는 i번째 C-FRAME(C-FRAME_i)의 j번째 F-FRAME을 의미한다. 상기 F-FRAME_{i,j}의 슬랙 타임은 도착 데드라인으로부터 C-FRAME_i의 남아있는 F-FRAME들의 수행 시간을 뺀셈하여 계산할 수 있다. 상기 F-FRAME_{i,j}의 슬랙 타임은 아래 수식 6으로 표현될 수 있다.

[0139] [수식 6]

$$\text{slacktime}(F\text{-FRAME}_{i,j}) = d_i + r_{i,j} - e_{pti,j}$$

[0140]

[0141] d_i는 FLOW_i의 허용 가능한 딜레이이고, r_{i,j}는 C-FRAME_i의 남아있는 모든 F-FRAME들을 모으는 데에 걸리는 시간이고, e_{pti,j}는 C-FRAME_i 내에서 수행되지 않은 F-FRAME들을 수행하는 데에 걸리는 시간이다. 상기 수행되지 않은 F-FRAME들이란 F-FRAME_{i,j}로부터 상기 C-FRAME_i 내의 마지막 F-FRAME을 의미한다.

[0142] 상기 플로우 스케줄러(130)는 C-FRAME_i 내에서 F-FRAME_{i,j}의 위치, 남아 있는 F-FRAME의 개수 및 샘플링 레이트를 이용하여 r_{i,j}를 판단할 수 있다. 상기 플로우 스케줄러(130)는 F-FRAME의 처리 시간을 프로파일하여, 이전 C-FRAME의 F-FRAME 처리 시간에 대한 프로파일 결과를 기초로 e_{pti,j}를 예측할 수 있다.

산업상 이용가능성

[0143] 본 발명에 따르면, 상기 센싱 오퍼레이터(160)는 C-FRAME 단위로 상기 센싱 데이터를 수집하며, 상기 프로세싱 오퍼레이터(150)는 F-FRAME 단위로 센싱 플로우를 처리하므로 동시에 수행되는 애플리케이션들의 자원 사용을 조절하여 애플리케이션의 유틸리티를 향상시킬 수 있다.

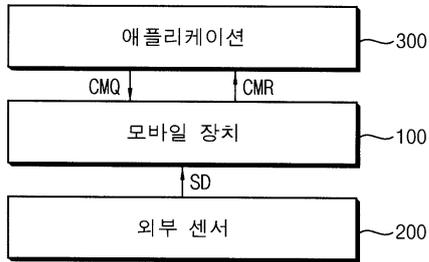
[0144] 상기 플로우 익스큐션 플래너(140)는 상기 센싱 핸들러(160)의 센싱 오퍼레이터의 모니터링 인터벌을 조절하여 상기 센싱 오퍼레이터 간의 센싱 데이터 수집을 적절히 제어할 수 있다.

[0145] 상기 플로우 스케줄러(130)는 복수의 플로우의 상기 F-FRAME의 처리 순서를 적절히 조절할 수 있다.

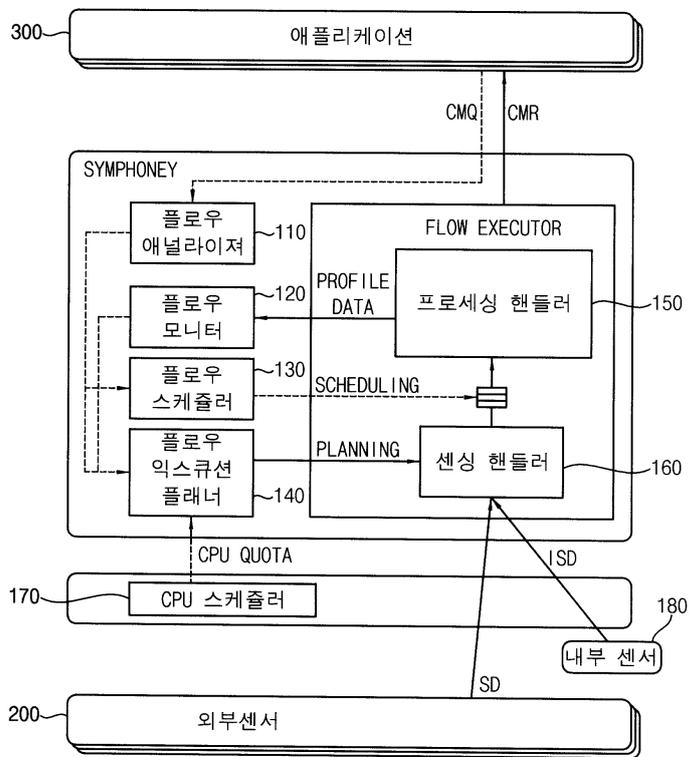
[0146] 상술한 바와 같이, 본 발명의 바람직한 실시예를 참조하여 설명하였지만 해당 기술 분야에서 통상의 지식을 가진 자라면 하기의 특허청구범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

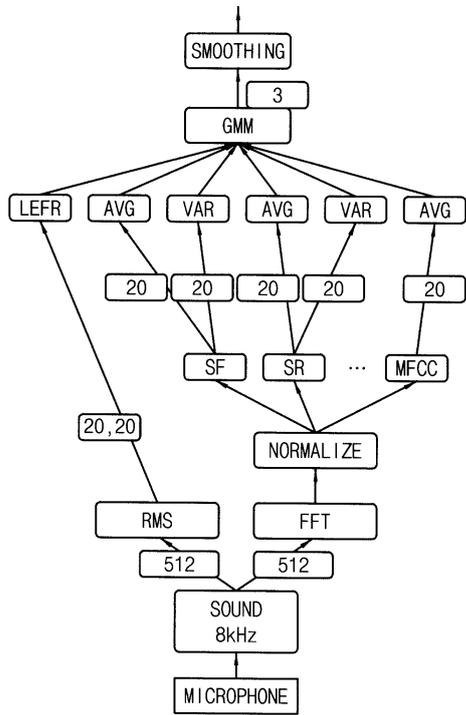
도면1



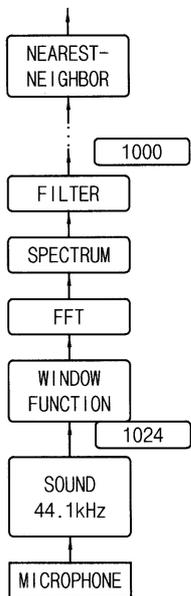
도면2



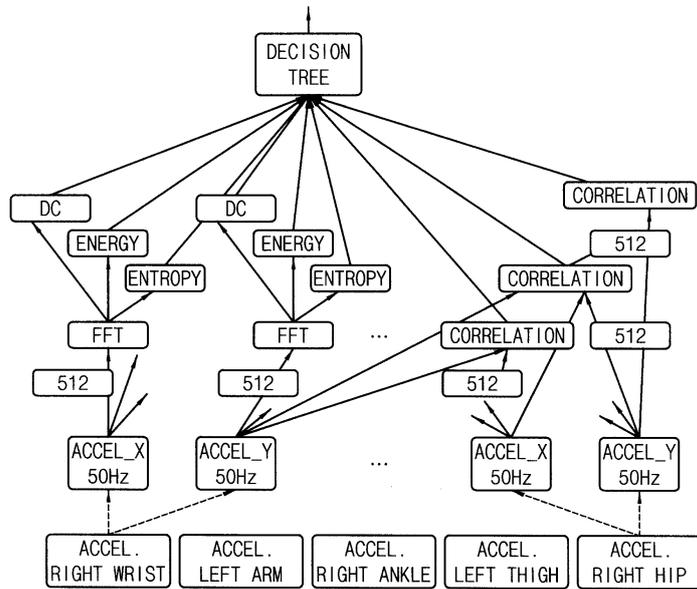
도면3a



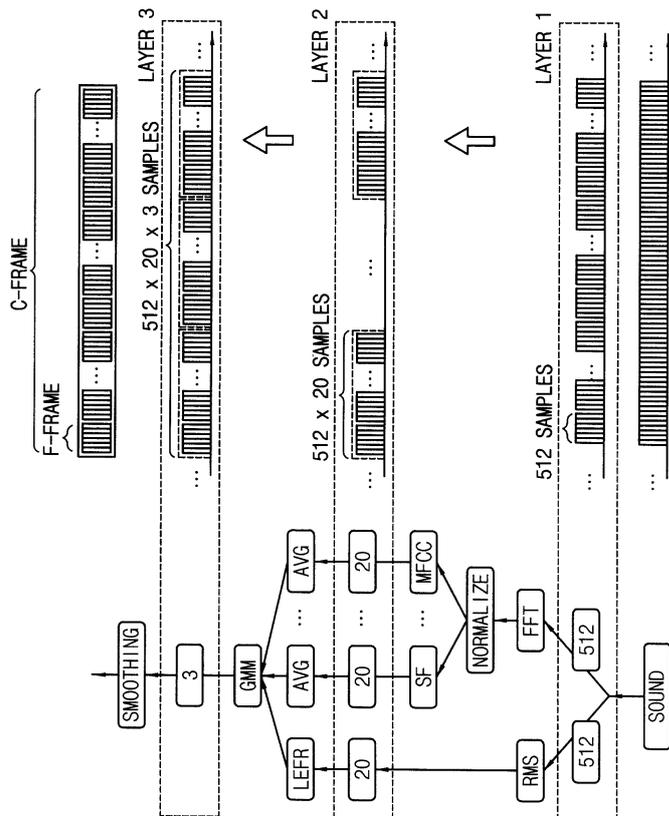
도면3b



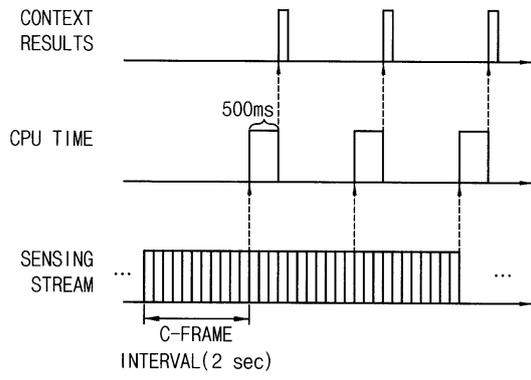
도면3c



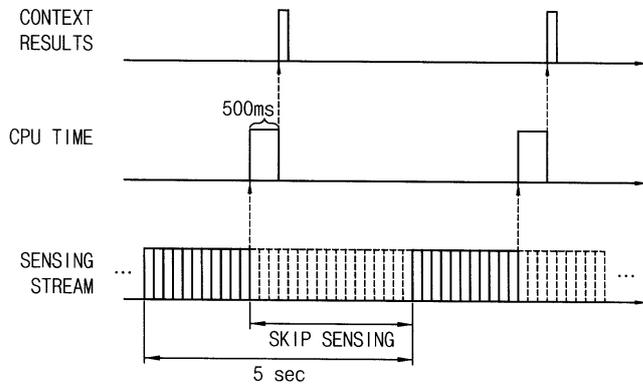
도면4



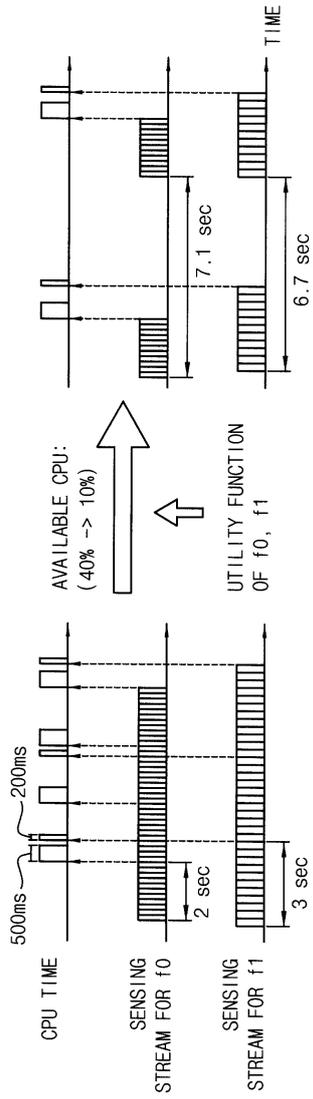
도면5a



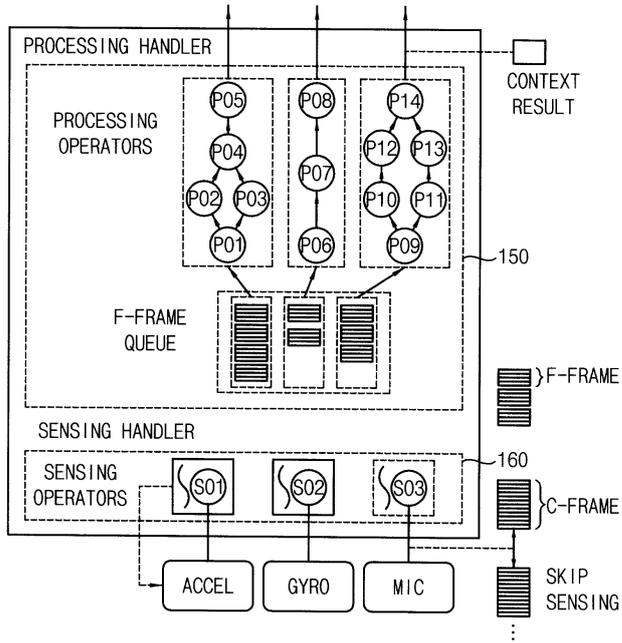
도면5b



도면6



도면7



도면8

