
센서네트워크상에서 능동규칙 이동에이전트의 역 방향 이주

이연식*

Backward Migration of an Active Rule Mobile Agent on the Sensor Network

Yonsik Lee*

이 논문은 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(No. C0028654)의
지원으로 수행한 연구임

요 약

센서네트워크의 능동성과 자율성을 위한 이동에이전트의 효율적 이주 방법과 일관된 네이밍 서비스는 센서네트워크 미들웨어의 주요 요소이다. 따라서 본 논문에서는 센서네트워크 구성 성분들에 대한 다양한 정보를 새로운 메타데이터에 저장하여 RMI 기반의 네이밍 기법을 통하여 능동규칙 이동에이전트의 역 방향 이주와 탑재한 능동규칙 실행을 실험한다. 본 연구는 기존 정 방향 이주[12]의 확장을 기반으로 이루어지며, 실험 결과는 센서네트워크 환경에서의 이동에이전트 미들웨어의 유효성과 효율적인 센서네트워크 응용환경 구축 가능성을 제시하며, 이는 센서네트워크 응용 개발에 있어서 동적인 환경 변화에 대한 적응성을 향상시킬 수 있다.

ABSTRACT

For the activeness and autonomy of a sensor network, the efficient migration method of a mobile agent and the consistent naming services are the required components of a sensor network. Accordingly, this paper presented the implementation of backward migration of an active rule mobile agent applying the naming method based on RMI that used the meta_table including the informations about the components of a sensor network. This study implemented based on the extension of the forward migration[12], and the results of the various experiments present the efficacy of mobile agent middleware system and the possibility of constructing efficient sensor network application environment. And, the results of this study are able to enhance the adaptability on dynamic changes of environment in sensor network application development.

키워드

센서네트워크 미들웨어, 능동규칙 이동에이전트, 메타데이터, 역 방향 이주

Key words

Sensor Network Middleware, Active Rule Mobile Agent, Meta_data, Backward Migration

* 정회원 : 군산대학교(yslee@kunsan.ac.kr)

접수일자 : 2012. 09. 18

심사완료일자 : 2012. 10. 16

I. 서 론

센서네트워크 응용 환경에서 센서네트워크 미들웨어의 추상화 개념을 적용하여 미들웨어의 역할과 기능을 확장시키고자 하는 연구가 필요하다[1,2,5]. 본 논문에서는 이동 에이전트를 이용하여 원격에서 자동으로 센서노드들이 동작할 수 있도록 하며, 제한된 자원을 조절하여 활용할 수 있는 알고리즘을 포함하는 능동규칙의 탑재와 실행을 통하여 이벤트 중심의 능동적 이동 에이전트 미들웨어를 설계 및 구현한다. 이를 위하여, 기존 연구[3,11,12]에서 구현한 능동규칙 이동 에이전트 시스템, 센서네트워크 구성요소들의 메타데이터 저장 및 관리 방법 및 이동 에이전트의 정 방향 이주 방법들을 기반으로 이동 에이전트의 역 방향 이주와 인접 노드를 제어하는 능동규칙의 실행을 실험을 통하여 보임으로써 전체적으로 구현된 센서네트워크 미들웨어시스템의 기능 확장성과 응용 가능성을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서 이동 에이전트의 역 방향 이주 방법 구현을 위한 주요 요소인 메타데이터, 이동 에이전트 구조 및 정 방향 이주에 관하여 설명하고, 3장에서는 이동 에이전트의 역 방향 이주 알고리즘과 이주 후 센서노드에서의 동작 과정을 설명한다. 4장은 역 방향 이주를 위한 실험 환경과 이주를 위한 서버리스트 생성 및 역 방향 이주 실험 결과를 보이고, 마지막으로 5장에서 결론 및 향후 연구 방향을 제시한다.

II. 기존 연구의 주요사항

2.1. 메타데이터

센서네트워크 환경에서의 에이전트 기반 미들웨어 시스템에서도 다양한 구성 요소들의 정보를 수집하여 통합된 네이밍 서비스를 제공하는 기능이 요구되는데, 이를 위하여 센서데이터 서버를 포함한 싱크 및 센서노드의 식별이 필요함에 따라 USN 노드들의 주소 및 네임 체계, 네임 리졸브(resolve) 및 비동기적 데이터처리 체계가 필요하다[4,6,10]. [11,12]에서는 주소체계를 인터넷망과 USN망으로 된 2계층으로 구분하고, 데이터 서버와 연동된 USN은 싱크노드와 센서노드 계층으로 구성하여 각 계층의 노드에 일련번호를 부여하고 번호를 조

합한 주소를 사용하여 센서네트워크상의 싱크 및 센서노드까지 검색이 가능하고, 또한 노드의 수에 따라 확장이 가능한 형태의 메타데이터를 제안하였으며, 본 논문에서는 이를 이용한 네이밍 서비스를 기반으로 이동 에이전트의 이주 및 처리 관련 구현을 수행한다.

2.2. 이동 에이전트 구조

이동 에이전트는 이주대상 목록인 서버리스트를 가지고 센서노드들 사이를 이주하며 센서데이터를 획득, 전송하며 센서데이터를 수집하고 탑재한 능동규칙을 실행하는 역할을 하며[2,4], 역 방향 이주 시에는 센서데이터 서버나 싱크노드에서 사용자가 응용에 따라 다양한 서버리스트를 구성할 수 있도록 한다. 이주와 규칙 실행을 위한 정보를 포함하는 이동 에이전트 구조의 각 요소들의 기능 및 역할에 대한 설명은 [6,11]을 참고한다. 본 이동 에이전트는 인접한 2개 노드의 센서데이터를 가지고 이동하며, 해당 센서데이터를 기반으로 중복 데이터 처리를 한다. Threshold는 센서별 중복 데이터 처리의 임계값이며, 본 논문에서는 조도의 임계값을 설정하여 사용한다.

2.3. 이동 에이전트의 정 방향 이주

이동 에이전트의 정 방향 이주는 에이전트가 센서노드에서 지정된 처리를 수행하면 다음노드로 이주하는 방식이다. 이동 에이전트의 노드 이주는 에이전트 이주 관리자가 Mig_List를 참조하여 Dest_Node를 선택하고 에이전트 통신 모듈의 에이전트 송신기를 통해 수행한다[11,12]. 이주는 다음 목적지 노드인 Dest_Node를 Mig_List에 있는 노드들 중에서 선택한 후에 이루어지며, 다음 노드의 선택은 이동 에이전트의 이동 거리, 정확도, 중요도 및 경로 회피방법 등 다양한 라우팅 방법을 적용하여 결정될 수 있다[2,8,9].

이동 에이전트의 이주는 message_t 구조체로 표현되는 TinyOS의 액티브 메시지 구조를 통해 이루어진다. [11]과 [12]는 이러한 액티브 메시지 구조를 이용하여 이동 에이전트가 정 방향 이주를 하며 탑재된 능동규칙을 실행하는 방법과 그 과정을 실험을 통하여 보임으로써, 응용의 환경과 특성에 적합하도록 능동규칙 이동 에이전트의 역할과 기능을 최적화할 수 있음을 제시하였다.

III. 이동에이전트의 역 방향 이주

3.1. 이동에이전트의 역 방향 이주알고리즘

역 방향 이주방식의 이동에이전트는 싱크노드에서 센서노드로부터 센서데이터 값을 전송 받는다. 그 후 센서데이터 값에 따라 센서데이터 서버나 싱크노드에서 응용에 적합하도록 사용자에게 의해서 설정(동적 생성 가능 또는 규칙베이스와 연동 가능)된 이주를 위한 서버리스트를 이용하여 이주한 뒤 정의된 능동규칙을 실행한다. 싱크노드는 Mig_List상의 첫 번째 노드로 Mig_List를 보내고, 센서노드에서는 수신된 Mig_List를 가지고 Dest_Node를 선택하여 에이전트 통신 모듈의 에이전트 송신기를 통해 보내진다. 다음 그림 1은 센서네트워크상의 이동에이전트 역 방향 이주알고리즘을 나타낸다.

```

INPUT : MA_Info, Mig_List, reading
OUTPUT:
PROCESS:
IF(nodeId == Sink_ID) THEN
IF(Mig_List[MA_Info.count] == local.id) THEN
call Leds.led0On();
MA_Info.count++;
IF(MA_Infocount <= Mig_List.Length()) THEN
IF(!sendbusy && sizeof ma_data <=
call AMSend.maxPayloadLength()) THEN
memcpy(call AMSend.getPayload(&send_data),
&local, sizeof ma_data);
call AMSend.send(Mig_List[MA_Info.count],
&send_data, sizeof ma_data) == SUCCESS
    
```

그림 1. 이동에이전트의 역 방향 이주알고리즘
Fig. 1 Backward Migration Algorithm of Mobile Agent

역 방향 이주방식의 이동에이전트 이주는 정 방향 이주[12]와 마찬가지로 라디오 통신과 TinyOS의 액티브 메시지 구조를 통해 이루어지며, TinyOS는 라디오 스택의 추상화 과정을 통해 IEEE 802.15.4의 패킷 및 프레임 구조에 대응하는 액티브 메시지 구조를 정의해 사용자에게 제공한다. 이러한 기능들을 이용하여 이동에이전트인 mobileAgent_t 구조체를 액티브 메시지인 message_t에 포함시켜 전송하는 방법으로 이주를 실행한다.

3.2. 역 방향 이주 후 센서노드에서의 동작

이동에이전트의 역 방향 이주는 각 센서노드에서 센싱된 값이 THRESHOLD보다 적을 경우에만 싱크노드에 데이터를 전송하고, 싱크노드에서는 수신된 노드의 ID에 따른 메타데이터의 정보와 사용자 정의 규칙을 이용하여 서버리스트를 생성하고 그에 따라 이동에이전트를 역 방향으로 이주시키는 방식이다. 싱크노드에서 전송된 이동에이전트는 서버리스트를 따라 순회하면서 탑재된 능동규칙을 실행한다. 센서노드에서는 자신에게 전달된 능동규칙 탑재 이동에이전트가 있을 경우에만 해당 규칙을 실행한다. 본 논문에서는 서버리스트에 따라 센서노드들을 방문하며 연결된 전등을 제어하는 규칙을 실행하도록 한다. 임계값에 따라 규칙을 수행해야 할 첫 번째 노드로 이동에이전트를 전송하면, 이를 수신한 센서노드에서는 역 방향으로 전송된 이동에이전트인지 판별하고 해당 규칙을 실행한다. 또한, 규칙 실행을 위한 대상 노드가 추가로 있을 경우(사용자 정의 규칙에 따라) 이주 및 규칙 실행을 지속적으로 수행하며 완료된 경우에는 자동적으로 제거되도록 한다. 이 때 정 방향 이주를 위한 서버리스트에 이주할 노드가 남아 있을 경우 정 방향 이주는 지속되며, 메타데이터에 의한 이동에이전트의 역 방향 이주과정 중 센서노드에서의 에이전트 수신과 센서데이터 획득 및 송신 과정은 정 방향 이주 시와 동일하다[12]. 다음 그림 2는 각 센서노드에서 이동에이전트 수신 후 처리되는 과정을 구현한 알고리즘이다.

```

INPUT : MA_Info, Mig_List, reading
OUTPUT:
PROCESS:
IF(nodeId == Sink_ID) THEN
IF(Mig_List[MA_Info.count] == local.id) THEN
call Leds.led0On();
MA_Info.count++;
IF(MA_Infocount <= Mig_List.Length()) THEN
IF(!sendbusy && sizeof ma_data <=
call AMSend.maxPayloadLength()) THEN
memcpy(call AMSend.getPayload(&send_data),
&local, sizeof ma_data);
call AMSend.send(Mig_List[MA_Info.count],
&send_data, sizeof ma_data) == SUCCESS
ELSE
    
```

```

IF(nodeId == local.id - 1 || modiId == local.id + 1)
THEN call Leds.led0On();
MA_Info.count++;
IF(MA_Infocount <= Mig_List.Length()) THEN
IF(!sendbusy && sizeof ma_data <=
call AMSend.maxPayloadLength()) THEN
memcpy(call AMSend.getPayload(&send_data),
&local, sizeof ma_data);
call AMSend.send(Mig_List[MA_Info.count],
&send_data, sizeof ma_data) == SUCCESS
    
```

그림 2. 이동 에이전트 수신 후 센서노드의 처리 알고리즘
 Fig. 2 Processing Algorithm of Sensor Node after Receiving Mobile Agent

IV. 실험 및 결과

4.1. 실험 환경과 데이터 설정

본 실험은 능동규칙 이동 에이전트의 역 방향 이주에 대한 실험이며, 실험 센서데이터는 조도 데이터이며 싱크노드를 포함하여 총 7개의 센서노드를 사용하였다. 실험 환경은 기존 연구[12]에서 실시한 정 방향 이주를 위한 실험의 환경과 동일하다.

이동 에이전트의 이동과 데이터 전송 여부는 각 센서노드에 부착된 Led의 점등 여부로 확인할 수 있도록 하였으며, 실험의 명확성을 위하여 각 센서노드에서의 데이터를 다음 표 1과 같이 설정하여 실험하였다. 설정된 센서데이터들은 실제계의 센서노드들의 특징을 기반으로 각 인접한 센서들의 조도 값을 인접할수록 유사한 값을 가지도록 하였다.

표 1. 실험용 센서데이터 설정 값
 Table. 1 Sensor Data for Experiments

	센서1	센서2	센서3	센서4	센서5	센서6
조도(lux)	70	65	55	30	55	40

실험은 일정 주기로 6개의 센서노드에서 획득한 센서 데이터를 조건에 따라 싱크노드로 전송하면 싱크노드에서는 수신된 정보(센서노드 번호, 센서데이터)에 따라 사용자에게 의해서 설정(동적 생성 가능 또는 규칙베이스

와 연동 가능)된 이주를 위한 서버리스트를 이용하여 싱크노드에서 능동규칙을 탑재한 이동 에이전트를 역 방향으로 이주시켜 해당 규칙을 수행하는 결과를 보이도록 한다. 조도의 THRESHOLD는 50으로 설정하였으며, 임의의 센서노드에서의 센싱된 조도 값이 THRESHOLD 값 미만이라는 조건을 만족하면 역 방향 이주를 수행하도록 하기 위함이다. 실험을 위한 메타테이블의 정보는 다음 표 2와 같다.

표 2. 메타테이블 정보
 Table. 2 Meta_table Information

host Name	host URL	keyWord	pointers to SubMetaTablea	
Data Server	"210.118.84.37"	"Data"	a1	
싱크노드 ID	sink Number	sinkInfo	sinkURL	
a1	0	0	"210.118.84.37:0"	
센서노드 ID	node Number	nodeInfo		nodeAddr
		use Sensor Number	use Area	
1	1	2	8	"210.118.84.12:0.1"
2	2	2	2	"210.118.84.12:0.2"
3	3	1	3	"210.118.84.12:0.3"
4	4	2	4	"210.118.84.12:0.4"
5	5	1	5	"210.118.84.12:0.5"
6	6	1	7	"210.118.84.12:0.6"

메타테이블에서 싱크노드는 0번 하나임을 알 수 있고, SubMetaData(싱크 및 센서노드 관련 데이터 저장)에서 실험에 사용된 총 6개의 센서노드에 대한 정보를 알 수 있다. 센서노드의 ID와 센서노드에 할당된 번호는 본 실험에서는 동일하게 입력하였다. 또한 nodeInfo의 useSensorNumber는 현재 센서노드에서 사용되고 있는 센서의 숫자를 나타내며, useArea는 센서노드가 위치한 지역의 번호를 나타낸다. 지역의 번호는 임의로 입력하였으며, 인접한 지역은 숫자의 간격이 가까움을 나타낸다.

4.2. 역 방향 이주를 위한 서버리스트 생성 및 이주 실험

THRESHOLD에 따른 능동규칙을 실행하기 위한 역 방향 이주대상 서버리스트 생성조건은 다음 표 3과 같다.

표 3. 서버리스트 테이블 생성 조건
Table. 3 Generating Condition of Server List Table

수신된 노드 ID	서버리스트 테이블 생성 조건
1, 3	자신을 포함한 다음 노드로 이주
2, 6	자신을 포함한 인접 노드로 이주
4	센서를 1개만 사용하는 노드로 이주
5	자신(5번 노드)으로 재 이주

수신된 노드의 ID를 메타데이터의 SubMetaData에 등록된 싱크노드 및 센서노드의 정보와 비교하여, 1번 또는 3번의 경우 수신된 노드를 포함하여 그 다음 노드들로 이주하며, 2번 또는 6번의 경우 수신된 노드를 포함하여 수신 노드 지역의 인접지역의 노드들로 이주하게 된다. 또한 ID가 4번인 경우 센서노드들 중 센서를 1개만 사용하는 노드들로 이주하게 되고, 마지막으로 5번의 경우 자신으로 재 이주하도록 서버리스트를 생성하게 된다. 표 2의 메타데이터 정보와 표 3의 서버리스트 테이블 생성 조건을 적용하여 서버리스트를 생성하면 실험 결과 창에 다음 표 4와 같은 서버리스트 테이블이 생성된다.

표 4. 생성된 서버리스트 테이블
Table. 4 The Generated Server List Table

서버리스트 테이블	
Sensor node #	Migration List
1	1 -> 2
2	2 -> 3 -> 1
3	3 -> 4
4	3 -> 5 -> 6
5	5
6	6 -> 1 -> 5

생성된 서버리스트 테이블은 센서노드 1번의 데이터 값에 의한 이동에이전트의 서버리스트는 [표 3]에 정의된 내용에 따라 1->2가 되며, 센서노드 4번의 데이터 값에 따른 서버리스트는 센서를 1개만 사용하는 노드들로만 이주하게 되므로 [표 2]의 MetaTable 정보에서 nodeInfo의 useSensorNumber가 1인 센서노드들로만 구성되어 3->5->6이 된다. 이와 같이 서버리스트 테이블은 사용자에게 의해 응용에 적합하게 구성할 수 있다. 본 실험의 결과인 다음 그림 3은 정 방향 이주에 따라 각 센서노드에서 획득한 조도 데이터가 THRESHOLD 미만일 경

우 단순히 싱크노드로 전송한 결과를 나타낸다. 이는 역 방향 이주방법을 적용하지 않았기 때문에 센서노드에서 아무런 규칙을 수행하지 않는다.

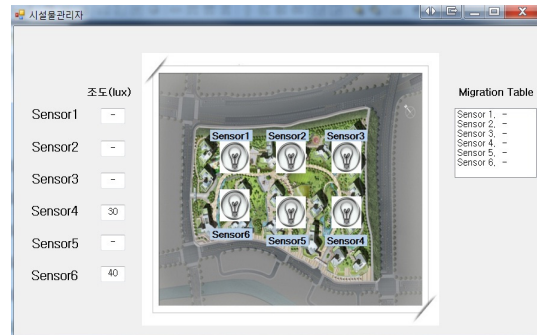


그림 3. 서버리스트를 적용하지 않은 이동에이전트의 역 방향 이주 결과

Fig. 3 The Result of the Backward Migration of Mobile Agent without Server List

다음 그림 4는 THRESHOLD보다 작은 조도 데이터 30을 센서노드 4번으로부터 수신한 뒤 서버리스트(3 -> 5 -> 6)를 생성하여 해당 센서노드로 능동규칙을 탑재한 이동에이전트를 이주시켜 실행한 결과이고, 그림 5는 센서노드 6번으로부터 조도 값을 수신 받았을 경우의 서버리스트(6 -> 1 -> 5)에 따른 규칙 실행 결과이다.

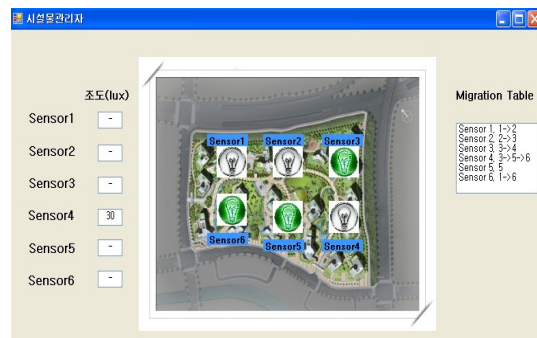


그림 4. 센서노드 4번의 조도 값에 따른 역 방향 이주 후 규칙 실행 결과

Fig. 4 The Results of Active Rule Execution on Node #4 after Backward Migration

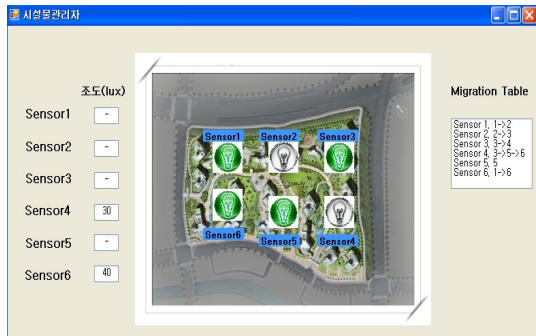


그림 5. 센서노드 6번의 조도 값에 따른 역 방향 이주 후 규칙 실행 결과
 Fig. 5 The Results of Active Rule Execution on Node #6 after Backward Migration

센서노드에서는 주기적으로 데이터를 센싱 하다가 이동에이전트를 수신하였을 경우, 자신의 id와 서버리스트의 id를 비교하고 이동에이전트에 탑재된 능동규칙을 비교하여 이주 방향을 결정하여 다음 노드로 이동에이전트를 전송하며, 더 이상 이주할 노드가 없을 경우에는 종료한다.

V. 결 론

본 논문에서는 기존 Java RMI 기반의 환경으로 구성된 센서네트워크를 연동한 멀티 에이전트 시스템의 네이밍 에이전트의 메타데이터에 센서데이터 서버, 싱크 및 센서노드의 위치정보와 다양한 특성 등의 정보를 표현하는 서브 메타데이터를 추가하여 센서네트워크 연동 능동규칙 에이전트시스템에 적합한 새로운 메타데이터[11,12]를 적용하여, 센서네트워크 환경에서 센서노드까지 능동규칙을 탑재한 이동에이전트가 이주하며 센서데이터를 획득하고, 사용자나 관리자의 특별한 조작 없이 센서노드들을 이주하며 센서데이터에 따라 2가지 이상의 서로 다른 능동규칙을 실행하는 능동규칙 탑재 이동에이전트시스템을 구현하였다. 이동에이전트의 정 방향 이주방법을 기반으로 이동에이전트의 역 방향 이주방법을 구현하였으며, Hmote2420 센서노드들을 사용하여 역 방향 이주 후 센서데이터 수집, 중복 데이터

제거 및 인접 노드 제어 모듈 등의 사용자 정의 능동규칙을 탑재한 이동에이전트의 규칙 실행과정과 결과를 실험을 통하여 보임으로써 센서네트워크 환경에서의 이동에이전트 미들웨어의 유효성과 응용 가능성을 제시하였다.

본 연구의 결과는 센서네트워크 응용 개발에 있어서 동적인 환경 변화에 빠르게 적응 할 수 있으며, 응용에 적합한 다양한 능동규칙의 탑재 및 실행과 이동에이전트의 다양한 이주방법을 제공함으로써 네트워크 부하경감, 수명 연장 및 개발 적응성을 향상시킬 수 있다.

향후 연구로는 이동에이전트의 이주경로의 안전성을 확보 관련 연구와 응용 환경에 적합한 규칙 생성 및 관리를 위한 센서_규칙 온톨로지 관련 연구 등이 필요하다.

감사의 글

이 논문은 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(No. C0028654)의 지원으로 수행한 연구임

참고문헌

- [1] P. K. Biswas, Hairong Qi, Yingyue Xu, "A Mobile Agent Based Collaborative Framework for Sensor Network Applications," Proc. of Int'l Conference on Mobile ad hoc and Sensor Systems, pp. 650 - 655, Oct. 2006.
- [2] Konstantopoulos C. et al., "Effective Determination of Mobile Agent Itineraries for Data Aggregation on Sensor Networks," IEEE Transactions on Knowledge and Data Engineering, Vol.22, pp. 1679-1693, 2010.
- [3] 이연식, 이정수, "센서데이터 획득을 위한 이동에이전트 설계," 정보처리학회 춘계학술발표대회논문집, 제17권, 제1호, pp. 1070-1073, 2010.
- [4] J. W. Yoon, et al., "Agent Based Sensor Network Middleware using Reputation Mechanism over Heterogeneous Network Environments," Proc. of Int'l

- Conference on ICCAE, pp. 373-376, 2010.
- [5] Heimfarth T. et al., "Experimental Analysis of a Wireless Sensor Network Setup Strategy Provided by an Agent-oriented Middleware," Proc. of Int'l Conference on AINA, pp. 820-826, 2010.
 - [6] 정의현, "무선 센서네트워크를 위한 속성 기반 네이밍 구조," 한국컴퓨터정보학회논문지, 제12권, 제4호, pp. 95-102, 2007.
 - [7] 정재윤, "능동형 규칙 기반 유비쿼터스 프로세스 설계의 워크플로우 패턴 분석," 한국전자거래학회지, Vol. 14, No. 1, 2009.
 - [8] 신문선, 이명진, "트리거를 이용한 네트워크 관리 프로그램 자동생성 지원 능동적 네트워크 관리시스템," 한국인터넷정보학회논문지, Vol.10, No.1, pp.78-84, 2009.
 - [9] C. Hermann and W. Dargie, "Senseive: A Middleware for a Wireless Sensor Network," Proc. of Int'l Conference on AINA, pp.612-619, 2008.
 - [10] D. Ballari, M. Wachowicz, and M. A. M. Callejo, "Metadata behind the Interoperability of Wireless Sensor Networks," Sensors, Vol.9, No.5, pp.3635-3651, 2009.
 - [11] 이연식, 이정수, "이동 에이전트 미들웨어를 이용한 중복 센서데이터 제거", 한국인터넷정보학회논문지, 제12권, 제3호, pp.27-36, 2011.6
 - [12] 이연식, 이준호, "메타데이터를 이용한 능동규칙 이동 에이전트의 정 방향 이주," 한국정보통신학회 논문지, 제16권, 제7호, pp.1567-1574, 2012.7

저자소개



이연식(Yonsik Lee)

1982년 전남대학교
전자계산학과(이학사)
1984년 전남대학교 대학원
전자계산학과(이학석사)

1994년 : 전북대학교 대학원 전산응용공학전공
(공학박사)

1997년~1998년 : University of Missouri 교환교수

2004년~2005년 : Ohio State University 교환교수

1986년~현재 : 군산대학교 컴퓨터정보공학과 교수

※관심분야 : 객체지향 시스템, 능동규칙 시스템,
센서네트워크 에이전트 미들웨어, USN 응용