

재난 구조 로봇을 위한

목적 주도의 정황 인지 미들웨어

신연균[○] 이동현 인 호
고려대학교 컴퓨터, 전파통신공학과

Goal-driven, Situation-aware middleware for Disaster Relief Robot

Youn-Kyun Shin[○] Dong-hyun Lee Hoh Peter In
Department of Computer Science & Communication Engineering, Korea University

요 약

본 논문에서는 재난 구조 로봇의 목적을 수행하기 위한 Situation-aware Middleware 에 대해서 설명한다. 재난 구조 로봇은 정확한 정황인지와 빠른 정보 처리 능력이 필요하다. 하지만 실시간으로 수집되는 센서의 정보들은 재난 구조 로봇과 같은 인명 구조나 재해 수습을 위한 특정 목적을 수행하기 위해 설계된 지능형 로봇은 센서 정보의 시간과 무분별한 정보는 선별하여 수집해야 긴급 상황에서 효율 적으로 대처 할 수 있다. 따라서 이 논문에서는 Q-MAR(QoS-Mission-Action-Resource)모델을 참조 하여, 특정 목적에 맞는 정황 정보들을 선별 수집·융합 하는 새로운 Goal-driven Situation-Aware Middleware(GDSAM)를 제안하여 이 문제를 해결 하고자 한다.

1. 서 론

지능형로봇이란, 일반적으로 외부환경의 변화를 인식하고 스스로 상황을 판단하여, 자율적으로 동작하거나 인간과 상호작용을 하는 로봇을 의미한다. 로봇 기술(Robot Technology)을 이루는 요소 기술로는 ‘지각(perception),’ ‘인지(cognition),’ ‘동작(motion)’ 등이 있고, 특히 위치·속도·힘 등의 정보를 수집하는 지각 센서 기술과 시각·청각·촉각·미각·후각 등의 오감 센서 기술과 밀접한 관련이 있다. 로봇의 정확한 제어를 위해서는 초음파나 레이저 등을 이용한 거리 센서가 필요하고, 가속도와 각속도 센서 같은 위치 추정 센서 등도 장착되어야 한다. 이처럼 하나의 로봇을 만드는 데는 수많은 센서 기술이 결합되어야 한다. 전문서비스용 로봇은 구조화되지 않은 작업환경(Unstructured environment)에서 사용되므로 인공지능과 자율 기능이 크게 요구되며 극한환경(Hostile environment)에서 사용될 수 있는 내환경성, 안전성, 안정성이 요구된다.[1] 이런 지능형 로봇의 핵심 기술은 Perception(지각, 센싱), Cognition(인지, 지능)등이 있으며 시스템 통합(System Integration) 역시 중요하다.

로봇 지능은 다양하게 해석 될 수 있으며 해석에 따라 다른 요소 기술들을 포함하게 되는데 재난 구조 로봇은 화재, 지진 등과 같은 재난상황에 사용되는 특정 목적을

가지고 있는 로봇이기 때문에 본 논문에서는 로봇의 기능을 ‘특정 목적에 따른 주어진 정황을 선별 하고 수집·이해하여 수행할 임무를 완수하기 위한 일련의 작업 들을 하는 계획’ 이라고 설명 한다.

로봇이 정보를 습득하는 과정은 대체로 센서를 통한 정보 획득 이다. 재난 구조 로봇의 경우 주변 상황을 정확히 판단하기 위하여 미들웨어는 실시간으로 정보를 받아 와야 하고 각 센서로부터 오는 상황들을 발생 순서에 따라 신속히 전송해야 한다. 또한 필요한 정보만을 수집할 수 있는 기능이 필요하다. 그러나 기존의 연구되어진 미들웨어는 상황의 발생순서에 대한 고려가 부족하고 모호하고 불확실한 상황 데이터의 처리에 대한 연구는 부족한 실정이다.

Situation Aware Interface Definition Language (SA-IDL)는 다중의 Context 이력을 검사 할 수 있으며, 수행했던 Action의 이력도 검사하여 상황인지 서비스를 가능하게 해준다. 또한, 실행할 Action의 시간을 지정할 수 있어, 보다 정확한 상황포착에 능한 장점이 있다.[2] 이러한 SA-IDL을 기반으로 유비쿼터스 환경에 적합하도록 제안된 Q-MAR(Quality-Mission-Action-Resource) 모델은 SA-CSL(Situation-Aware Contract Specification Language)[4]를 이용하여 Application과 middleware 사이의 communication을 정의 하였다. 본 논문에서는 Q-MAR모델을 개선시켜 모호하고 불확실한 상황 데이터

의 처리에 대한 문제를 해결 하고자 한다. 이를 위하여 2장에선 관련연구에 대해 설명하고, 3장에서는 재난 구조 로봇과 같은 특수한 상황에서 사용될 수 있는 목적기반의 정황인지 미들웨어를 제안 한다. 4장에서는 결론에 대해 서술한다.

2. 관련 연구

2장에서는 Situation-Awareness 관련 연구에 대해서 언급한다. 2.1장에서는 본 논문의 기반이 되는 Q-MAR 모델 [3]을 2.2장에서는 정황인지(Situation-awareness) [5]에 대하여 설명한다.

2.1 Q-MAR 모델

Q-MAR 모델은 본 논문의 기반이 되는 모델이다. <그림1>에서 알 수 있듯이 Q-MAR모델의 주요 구성요소는 Situation-aware Manager, QoS Management Agent, Resource Management 이다. Situation aware manager는 센서로부터 들어오는 상황 정보를 수집하여 분석·종합 한다. QoS Management Agent는 요청된 QoS 요구사항들이 보증될 수 있도록 Resource Manager를 통해 resource들을 컨트롤 한다. RM은 높은 우선순위의 mission을 해결하기 위하여, resource를 재분배한다.

Q-MAR 모델에서는 mission들을 달성하기 위해서 무엇이, 얼마나 많은 자원이 요청 되었는지를 설명한다. Q-MAR 모델 정의 언어는 몇 개의 명세화된 컴포넌트 (mission, action, resource, situation, 그리고 QoS제약)들로 구성되어 있고, 하나의 QoS 제약은 하나 이상의 action에 주어지거나 여러 개의 QoS 제약이 하나 이상의 action에 주어 질 수 있다.

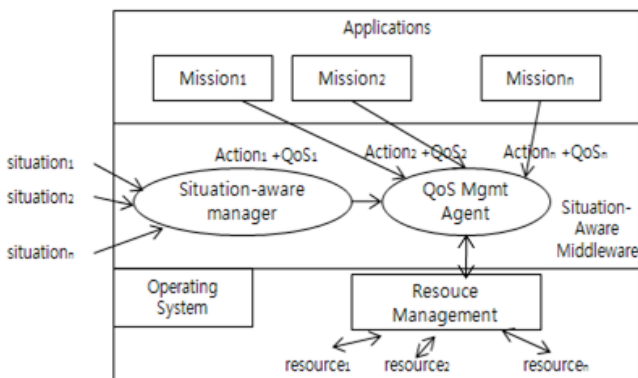


그림1. Overview Q-MAR Model

2.2 정황인지(Situation-awareness)

정황인지(SA, Situation-Awareness)[5]란 시간, 장소,

또는 사용자에 종속되어 동작하는 것이 아닌 인지된 상황을 추론한 결과를 바탕으로 동작이 이루어진다는 것을 의미 한다. 다시 말하면 프로세스가 정해진 시간간격을 두고 이루어지는 것이 아니라 상황의 변화에 따라 이루어진다는 것이다. 행위(Actions)와 다중 문맥(Multiple Contexts)간의 관계성을 시간의 변화에 따라 분석하고 이를 이해 할 수 있는 특성. 즉, 주변 상황(위치, 시간)에 따라 디바이스는 자동적으로 적시에 행동을 취할 수 있어야 한다는 개념이다. 이는 Context-Awareness와 비교하여 확연히 구별 할 수 있다.

- Context-Awareness: 현재 문맥과 작동 중인 장치의 변화를 문맥상 전후 관계의 데이터 비교를 통해서 발견하는 기능
- Situation-Awareness: 여러 개의 문맥 관계와 그에 해당하는 시간에 따른 실행 간의 관계를 분석하고 감지하여 데이터를 발견하는 기능

3. 목적 기반의 정황 인지 미들웨어 모델

3장에서는 2장에서 제안한 Q-MAR모델을 기반으로 하여 Goal-driven, Situation-aware middleware를 제안한다. 3.1장에서는 Goal-driven, situation-aware middleware(G-DSAM)의 설계에 대하여 설명하고 3.2장에서는 GDSAM 모델의 정의 언어에 대하여 설명한다.

3.1 목적 기반의 정황 인지 미들웨어 설계

본 논문에서 제안하는 Goal-driven, Situation-Aware Middleware(GDSAM) 모델의 목적은 재난 구조라는 특수한 상황에서 필요한 정보를 선별적으로 수집 할 수 있게 하고, 각 센서들에 의해 수집된 정보를 융합 또는 연합하여 현재 상황을 파악하거나 미래 상황을 예측 할 수 있도록 지원해 주는 것이다. Q-MAR 모델 역시 비슷한 기능을 제공해 줄 수 있다. 하지만 재난 구조 로봇의 경우에는 상황에 대한 처리가 실시간으로 요구되고 빠르게 행동을 취한다는 것에서 차이가 있다. 따라서 재난 구조 로봇과 같이 특별한 목적을 수행하는 상황에서는 사용자의 goal에 단 하나의 QoS를 접목 시켜 보다 빠르게 목적을 달성해야 한다. 또한 Action은 QoS만족하지 못한다면, Second Action을 취하여 새로운 QoS를 설정하고, 이를 수행 했을 경우 이전의 Action을 다시 수행 하도록 한다.

또한 데이터를 수집할 때에는 모호하고 불확실한 정보는 정보가 수집되어 처리되기 전에 미리 차단하여 실시간으로 전송되는 정보의 질을 높여 재난 구조 로봇이 빠르고 정확한 행동을 취할 수 있도록 신뢰도를 높여야 한다.

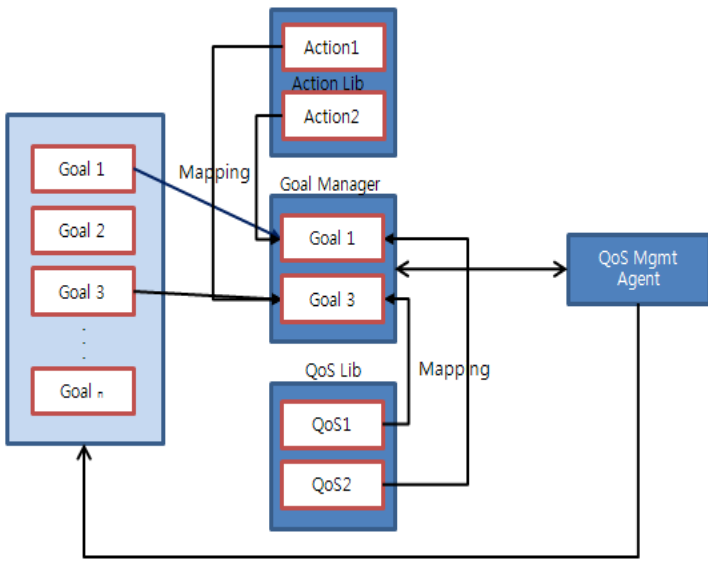


그림2. GDSAM의 개념 모델

<그림2>에서는 GDSAM의 개념 모델을 보여주고 있다. 각 Goal은 Goal Manager에서 Action과 QoS를 설정해 준다. 그리고 QoS Mgmt Agent는 각 Action에 대한 QoS를 만족했는지 판단하고 만족하지 못하였으면 새로운 Action과 QoS를 설정하여 다시 행동을 취한다. QoS를 만족하였을 경우에는 다음 Goal을 설정하여 임무를 수행 하도록 한다. 이와 같은 내용의 미들웨어 구조는 다음 <그림3>에서 보여준다.

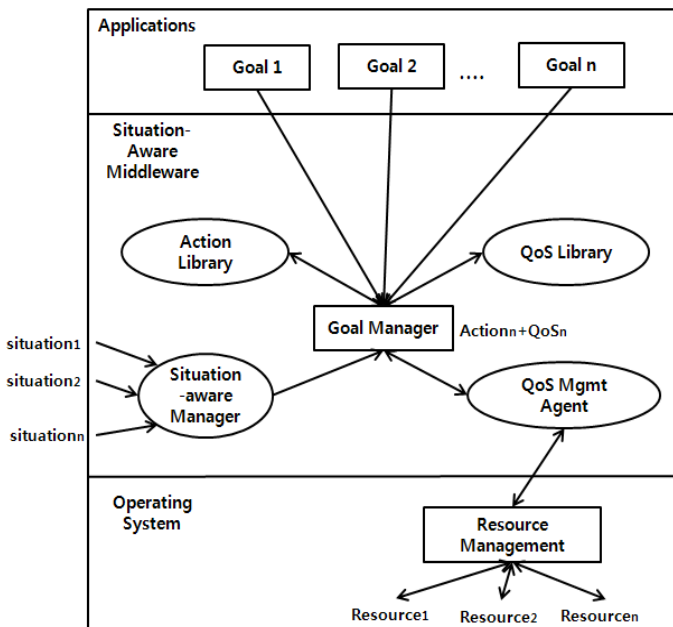


그림3. 제안된 GDSAM 모델의 구조

(1) Goal

Goal은 재난 구조 로봇이 달성해야 하는 목적 이다. Goal은 재난 구조 로봇의 특성에 맞게 화재발생시 인명 구조나 재난 상황의 파악 등으로 설정 될 수 있다. 따라서 이와 같은 몇 가지 Goal을 정하여 사용자가 필요에 맞게 설정 하도록 한다. Goal에는 생존자 위치를 파악 한다거나, 생존자를 구해오는것 등의 구체적인 목적이 설정되어야 한다.

(2) Situation-aware Manager

목적 관리자는 Application에서 정의한 조건에 따라 주변 센서로부터 들어오는 정보들을 선별하여 Goal Manager로 전송하여 준다. 즉, 센서로부터 들어오는 애매모호한 정보나 불확실한 정보들은 이 목적 관리자를 통해 제거된다. 따라서 실시간으로 들어오는 정보의 과부하를 막을 수 있고 더 빠른 정보 처리를 할 수 있게 된다.

(3) Action Library

Action Library는 Goal에 따라 취해질 Action들이 정의 되어 있다. Goal이 설정되면 Goal Manager 에서 Mission에 맞는 Action들을 가져와 Goal마다 실행시킬 Action을 정하게 된다. 예를 들면 로봇이 지하로 이동하기 위해 이동하는 것, 센서를 통해 정보를 수집하는 것 등이 설정 된다.

(4) QoS Library

QoS Library는 Action이 취해질 때 만족해야할 조건들이 정의 되어 있다. Goal Manager에서 Action과 함께 Goal 수행의 조건에 포함된다. QoS는 Action이 수행되고 나서 만족해야할 조건으로 로봇이 지하1층으로 제대로 이동하였는지, 센서로 수집된 CO2나 온도 센서의 정보가 올바른지의 조건이 설정된다.

(5) Goal Manager

Goal Manager는 재난 구조 로봇이 미션을 수행하기 위한 각 정보들을 수집하여 목적을 수행하기 위한 일련의 과정을 스케줄링 하는 것이 목적이다. 미션을 수행하기 위해 관리자부터 선별된 정보를 취합하고 각 Goal에 대한 Action과 QoS를 정해 준다. 이런 정보를 QMA에게 넘겨주고 QMA에서 fail을 리턴 받을 경우 다음 우선순위의 Action과 QoS를 다시 설정하여 QMA에게 다시 전송한다.

(6) QoS Management Agent(QMA)

QoS 관리 중계자는 Goal Manager로부터 받은 임무를 수행한다. RM을 컨트롤 하여 재난 구조 로봇이 적절한 행동을 취하게 하고 행동에 따른 QoS를 만족 하였는지

판단한다. 만약 QoS를 만족 하지 못하였다면, 만족할 때까지 행동을 취하거나 Goal Manager에서 새로 설정된 Action과 QoS를 설정하여 새로운 행동을 취하게 한다.

(7) Resource Management(RM)

자원 관리자는 QoS Management Agent로 부터 오는 명령에 따라 재난 구조 로봇이 행동을 할 수 있게 자원을 할당 한다. 재난구조 로봇에서의 Resource란 모터를 움직여 로봇이 이동하게 하거나 센서 나 카메라와 같은 로봇의 이동하고 정보를 취득하는데 필요한 재난구조 로봇에 속해있는 모든 하드웨어를 말한다. 행동을 취한 후 QoS를 만족하였는지 QoS Management Agent가 판단하여 다시 자원을 할당할 것인지 결정한다.

3.2 GDSAM 모델의 정의 언어

GDSAM 모델에서의 정의 언어는 Goal, action, QoS로 구성되어 있다. 이 컴포넌트들의 설명은 다음과 같다.

- Goal : 목표를 수행하기 위해 필요한 사용자에게 의 해 정의된 목적이다. 사용자가 지정하는 것으로 Goal은 Action이란 기능을 포함하고 있는 함수와 QoS로 구성된다.
- Action : 하나의 동작을 하는 함수 이다. Action에 는 하나의 QoS만 주어진다. Action은 Goal을 달성하기 위한 동작, 수단이 된다.
- QoS : 하나의 goal에 하나의 QoS가 주어진다. 하나의 goal에 하나의 QoS만 주어지기 때문에 신속 도를 높인다. 따라서 한 가지 목적에 대한 행동에 대해서 만 평가 한다.

```
Goal Mission1 [object1.Action1,object2.Action2,...,objectn.Actionn]
  Situation - Aware - Object {
    [activate at situation1]Action1
    RequireAction [Resource1[amount1],...,Resourcep[amountp]]
    WithQoSConstraint [QoSconstraint - list1]
    .....
  }object1;
  ...
  Situation - Aware - Object {
  ...
  }objectn;
```

그림4. SA-SCL 정의

Goal을 정의 하는 언어로는 <그림4>와 같이 Q-MAR 모델과 같은 SA-CSL(Situation-aware contract specification language) [4]을 사용한다. mission1은 Goal의 이름이 되고 object1.action1... objectn.action,

이라는 Action을 가지고 있다. 또한 각각의 situation에 의해 활동하게 되는 function은 각자 필요한 만큼의 action을 취한다.

RequireAction의 Resource는 로봇이 행동을 취하기 위해서 팔을 움직이거나 모터를 동작시키는 각 모듈을 의미 하고 WithQoSconstraint는 QoS library에서 가져오는 QoS 이다.

다음은 mission의 SA-CSL의 사용 예제와 Goal의 설정 사용 예제 이다. <그림5>는 SA-SCL 정의 언어를 이용하여 본 논문에서 제안한 Goal의 수행과정을 예를 들어 정의 한 것이다. SavePeople는 Goal이 되고 그에 따른 행동 과 QoS, Resource들이 정의 되어 있다.

```
Goal SavePeople (Robot.findway, Robot.move, Robot.findheart)
  Situation-aware-object {
    [activate at situation1] findway()
    RequiredResources ( FindWay(), MoveMotor(), UseSensor() )
    WithQoSConstraint ( QoS1);
    ...
    ...
  } Robot;
```

그림5. Mission의 SA-CSL사용 예제

여기서 SavePeople은 Goal의 이름이고 그 파라미터로 재난 구조 로봇이 취하여야 할 Action이 나열된다. 첫 번째 상황에서는 길을 찾아서 목표로 움직이기 위한 내용이 있고, 순차적으로 Action에 따라 재난 구조 로봇을 진행 시킨다.

4. 시나리오

찜질방 지하1층에서 화재가 발생 했다. 구조대가 들어가기엔 불길이 너무 강하고 연기가 심하여 들어 갈 수 없는 상황이다. 따라서 재난구조 로봇을 투입하기로 결정 했다.

재난구조 로봇의 첫 번째 Goal은 지하 1층까지의 이동이다. 따라서 로봇은 지하1층으로 내려가기 위해 모터를 동작 시키고 카메라를 동작시켜 영상을 확인한다. 이동 중에 연기가 너무 자욱하여 영상획득이 어려워지자 적외선 센서를 작동시켜 장애물을 피해 내려간다. 영상이 식별이 가능한가의 QoS를 만족 시키지 못했기 때문에 로봇은 적외선 센서를 동작 시켰고 이 때문에 장애물을 피해 내려갈 수 있었다.

목적지에 도착 후에는 주변 상황 파악 이라는 새로운 Goal이 설정된다. 따라서 재난 구조 로봇은 센서를 사용하여 CO2, 온도 등을 측정하여, 생존자가 있을만한 상황인지를 판단한다. 생존자가 있을 만한 상황이라고 판단되면, 로봇은 생존자를 찾아내는 Goal을 부여 받는다. 생존자를 감지하기 위해선 음성이나 움직임을 인식 하거나 카메라를 통하여 내부 상황을 파악한다. 만약 생존자를 발견 하였을 경

우 로봇은 다시 새로운 Goal을 부여 받아 그 임무를 수행하게 된다. Goal의 내용은 다음과 설정 된다.

- Goal1 : 지하 1층까지의 이동
- Goal2 : 주변 상황의 분석
- Goal3 : 생존자 확인

처음에 지하 1층까지의 QoS를 만족 시키지 못했기 때문에 장애물을 제거 하는 새로운 Goal이 interrupt된다. 이 interrupt를 해결하고 나면 다시 원래의 Goal인 지하1층까지의 이동을 수행하고, 그다음 Goal2를 진행시킨다. Action은 다음과 같은 내용으로 설정된다.

- Action1 : 모터를 동작
- Action2 : 센서를 동작
- Action3 : 카메라의 동작

QoS는 다음과 같은 내용으로 설정 된다.

- QoS1 : 원하는 만큼 모터가 동작 하였는가?
- QoS2 : 센서의 동작은 이상 없이 이루어지는가?
- QoS3 : 카메라의 동작은 이상 없이 이루어지는가?

5. 결론 및 향후연구

본 논문에서는 재난 로봇과 같이 특정한 목적을 가지고 임무를 수행하는 로봇에 적합한 상황인식 미들웨어를 제시 하고, 설계 하였다. 설계한 미들웨어는 사용자가 정의한 목적에 맞는 정보만을 수집하며, 수집된 정보를 연합하여 현재 상황을 파악한다. 따라서 정확한 상황파악으로 주어진 임무를 수행 할 수 있게 된다. 본 논문에서 제안한 목적주도의 상황인지 미들웨어는 재난 구조라는 특별한 상황에서 목적을 수행하기 위한 미들웨어이다. 정확한 정보에 따른 빠른 응답을 필요로 하는 목적으로 설계 되었다. 따라서 필요에 따라 정보를 선별 수집하기 때문에 정보의 양에 따른 과부하가 적어 실시간으로 응답 할 수 있을 것이다. 또한 각 QoS를 만족 시켜야만 다음 Goal을 수행하기 때문에 신뢰도 역시 향상될 것이다. 앞으로 이 미들웨어 모델을 실제로 구현하여 데이터를 검증하고 실제 생활에 활용 할 수 있게 하는 것이 연구 과제 이다. 유비쿼터스 사회에는 한 가지 목적을 위한 디바이스들의 출현이 많아 질 것이다. 따라서 재난 구조 로봇이나 특정한 목적을 수행하기 위해 만들어진 지능형 로봇, 또는 다양한 다른 디바이스 등에 응용 될 수 있다.

참고문헌

- [1] 조동일 ITFIND 주간기술 동향 2005.2.16 권호 1183 “지능형 로봇 센서”
- [2] S. Yau, Y. Wang and F. Karim, "Developing Situation-Awareness in Middleware for Ubicomp Environments" Proc. 26th Int'l Computer Software and Application Conf.(COMPSAC2002),pp.233-238
- [3] Hoh Peter In, ChangHwa Kim, and Stephen S. Yau, "Q-MAR: An Adaptive QoS Management Model for Situation-Aware Middleware" L.T. Yang et al(Eds.): EUC2004, LNCS 3207, pp. 972-981,2004
- [4] Hoh p. In, Stephen S. Yau, Yu Wang, and Dazhi Huang "Situation-Aware Contract Specification Language for Middleware for Ubiquitous Computing" Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)
- [5] 이동현(Dong-hyun Lee), 채희서(Heeseo Chae), 김현우(Hyunwoo Kim), 인호(Hoh Peter In), 한정현(JungHyun Han), 한국정보과학회, 한국정보과학회 학술 발표논문집 한국정보과학회 2005 한국컴퓨터종합학술대회 논문집(A), 2005. 7, pp. 445 ~ 447 (3pages) 공학