

중대 핵심 시스템 상에서 상황인지 적용

박지용^o 김상수 인호
고려대학교 컴퓨터학과
{jayyp^o, sookim, hoh_in}@korea.ac.kr

Supporting Situation Awareness in Mission Critical Systems

Jiyong Park^o
Sangsoo Kim, Hoh In
Department of Computer Science and Engineering, Korea University

요 약

중대 핵심 시스템은 변화하는 상황에 대해서 정보를 빠르게 획득하고, 대처하는 것이 매우 중요하다. 임무 수행 중에 시스템의 환경이나 예상하지 못한 상황이 발생하였을 경우, 임무를 효율적으로 처리하는 데 문제가 발생한다. 이를 해결하기 위해 상황 인지를 중대 핵심 시스템에 적용하여 환경의 변화와 상황에 대해서 유연하게 대응할 수 있게 한다. 이는 시스템 주변 환경과 상황을 인지하여서 적절한 판단을 내릴 수 있도록 하여 임무에 도움을 준다. 본 논문에서는 상황 인지를 적용한 시스템의 아키텍처를 제안하고 이를 검증하기 위해서 공중 방어 시스템 시뮬레이션에 적용한 실험 결과를 통해서 유효성을 검증한다.

1. 서 론

중대 핵심 시스템(Mission critical system)은 특수한 목적에 맞게 특화되어 있는 것으로, 사용자에게 높은 반응성과 유효성을 제공한다[1]. 이 시스템에서 주어진 임무를 수행하기 위해서 상황에 대한 정보를 빠르고 정확하게 습득하고, 대처하는 것은 매우 중요하다. 그러나 변화된 주변 환경과 시스템의 정황을 적절하게 판단하고 이를 대처할 수 있는 방법은 제시해주고 있지 않다. 따라서 중대 핵심 시스템은 지식 기반 데이터베이스 시스템(Knowledge based database system)을 동시에 운영하여 비슷한 상황에 대한 자료를 토대로 환경을 분석할 수 있도록 하고, 주요 예측 시나리오들을 사전에 준비하여 초기 정보를 가지고 일어날 상황을 예측하고 대비할 수 있게 한다.

하지만 급작스런 임무의 변경이나, 임무환경의 변화가 생겼을 때는 유연하게 대처하지 못하고 임무를 실패할 위험성이 있다. 전혀 새로운 상황이나 정보가 들어왔을 경우, 지식 기반 데이터베이스 시스템은 대처가 힘들다는 단점도 있다. 상황 인지(Situation awareness) 능력은 정보 획득 시스템끼리 실시간으로 협동하는 것을 가능하도록 하고, 목적을 먼저 인지하고, 처리할 수 있게 한다. 주변 환경의 각종 변화를 자동으로 인식하여 평가하여, 환경의 변화와 예측하지 못한 상황에 대해 유연하게 대응할 수 있게 한다.

본 연구에서는 상황 인지 기술을 적용한 중대 핵심 시스템의 아키텍처를 제시한다. 제시된 상황 인지 아키텍처는 Situation Awareness Unit과 Guidance & Control Unit으로 구성되어 있다. 또한, 제시된 아키텍처의 유효성을 검증하기 위하여, 실 체계와 가장 유사하게 개발된 실시간 분산처리 시뮬레이션을 개발 적용하여 실험한 결

과를 제시한다. 논문의 구성은 2장에서 본연구의 배경에 대하여 설명하고, 3장에서는 중대 핵심 시스템에서의 상황 인지 아키텍처를 제안한다. 4장에서는, 제시된 아키텍처의 유효성을 검증하기 위하여 구현된 시뮬레이션을 이용하여 실험한 결과를 제시하고, 마지막으로 5장에서는 결론과 향후 연구 방향에 대하여 기술한다.

2. 배 경

2.1 상황 인지

상황 인지(Situation awareness)는 일정한 주기에 발생한 정황(context)과 행동 간의 관계를 파악하는 것이다. 상황은 일반적으로 정황에 의해서 정의된다[2,4].

정황은 짧은 시간 동안, 시스템과 다른 시스템이나 주위 환경과의 상호작용의 특성을 나타낸다. 이러한 정황에 대한 정황 감도(context-sensitivity)는 정확한 자료에 대한 현재 정황과 변화에 대해서 탐지해내는 것을 의미한다. 상황이란, 일정 기간에 일어난 행동에 대해서 표현되는 것이고, 미래의 행동을 유발하는 동기가 될 수 있다. 상황 인지는 정황 인지를 포괄 할 수 있는 것으로서, 여기서 알아낸 관계를 통해서 새로운 정보를 찾아내거나, 상황에 맞는 최적의 반응을 알 수 있게 한다.

유비쿼터스 환경에서 상황 인지 미들웨어와 상황 인지 언어는 많이 연구되고 있다. [3]에서는 유비쿼터스 환경에서 재설정 가능한 정황 인식 미들웨어를 확장하여, 재사용성과 실시간 재구성을 할 수 있도록 연구하고 있다. 신뢰할 수 있는 유비쿼터스 어플리케이션을 위해서, [5]에서는 Adaptive fault tolerance 알고리즘을 적용한 시스템으로 소프트웨어 에러를 잡고, 재구성 가능한 정황 인지 미들웨어를 구성하도록 제안한다.

상황 인지는 중대 핵심 시스템에 높은 성능을 부여할

수 있다. 일반적인 실시간 시스템이나 지식 기반 시스템은 한정된 자료에 의해 미처 예상하지 못한 상황에 대해 유연한 대처가 불가능하다. 이러한 돌발 상황에서 민첩하게 대응하기 위해 기존의 시스템은 중앙 정보 시스템과의 통신을 하고, 기존의 누적 자료를 많이 가져와야 한다. 상황 인지 능력을 첨부한 시스템은 이러한 문제점을 극복할 수 있다.

2.2 공중 방어 시스템

공중 방어 시스템(Air defense system)은 실시간 분산처리 시스템으로 구성된다. 그 이유는 매우 많은 양의 정보를 수집하고, 처리해야 하기 때문이다. 이에 발생하는 스케줄링, 부하 조절(Load balancing), 노드 간의 배치 등의 문제가 연구되어 지고 있다[6,10]. 공중 방어 시스템은 효율적으로 긴급 상황을 처리하기 위해서 지식 기반 시스템을 동시에 운영한다. 이를 통해 불확실성, 불완전성 등을 줄일 수 있고 예상 시나리오에 따라 임무를 수행하는 것을 도와준다. [7]에서 제안된 시스템은 War Control Center를 이용하여 임무 결정 시스템을 제공하고 [8]에서는 퍼지 논리(Fuzzy logic) 기술로 위험 상황을 측정하고 분석한다.

본 논문에서 실험에 이용한 공중 방어 시스템 시뮬레이션은 HLA/RTI로 작성한 실시간 분산처리 시뮬레이션이다. HLA/RTI는 무기 시스템 구현과 실험을 위해 실시간 환경에 적합하도록 구성되어 있다[9]. 그림 1에서 보듯 시뮬레이션은 7단계로 구성되어 있다. 이는 SMCC (Surface Missile Control Center), ATS (Air Target Simulator), MFR (Multi Function RADAR), ECS (Engagement Control Simulator), LAU (LAUncher), MS (MiSsiLe)와 RTI (RunTime Infrastructure)이다. SMCC는 시스템을 제어하고 모니터링하고 ATS는 목표물을 지정하고 이들의 정보를 전달하는 역할을 한다. MFR은 레이더로서 비행체와 미사일의 위치나 종류 등의 필요한 정보를 실시간으로 획득한다. ECS는 미사일이나 비행체의 위험도를 판단하고 이를 설정한다. 그리고 LAU는 발사대의 기능을 시뮬레이션하는 부분으로, ECS의 정보를 MSL로 전송한다. 또한, MSL은 실제 미사일을 구현하고 설정된 목표물로 미사일을 추적하고 임무를 완수하는 기능을 한다. 여섯 개의 컴포넌트들은 RTI에 의해서 상호 영향을 미치게 된다.

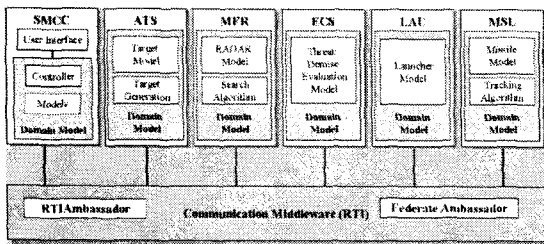


그림 1. 공중 방어 시스템 시뮬레이션 모델 [10]

3. 상황 인지 아키텍처

중대 핵심 시스템을 위한 상황 인지 아키텍처는 그림 2와 같이 Situation Awareness Unit(SAU)와 Guidance & Control Unit(GCU)로 크게 두 부분으로 구성된다.

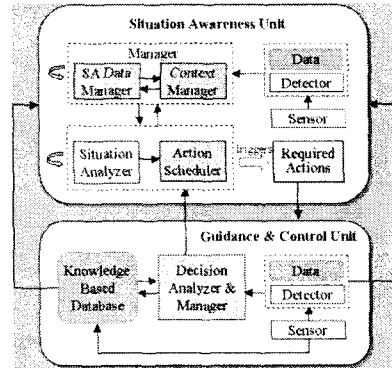


그림 2. 상황 인지 아키텍처

3.1 Situation Awareness Unit

기본적으로 상황 정보와 임무를 GCU에서 전달받는다. 하지만 정보는 자체 센서, 외부 정보를 통해서 실시간으로 수집하고 얻어진 정보는 전달된 임무와 부합되는지 실시간으로 분석한다. 만약 할당된 임무와 충돌이 상황이 발생된 경우, 상황 분석을 통해서 최적의 임무로 자체적으로 변동하고 GCU에게 전달하는 동시에, 수행하게 된다.

SA Data Manager: 상황 인지 정보와 임무 정보를 관리한다. 이 데이터는 자체 정보와 GCU에서 전달받은 정보 및 데이터베이스 자료로 구성된다. 상황 정보는 튜플 $\langle t_b, t_r, s_1, s_2, \dots, s_n \rangle$ 로 이루어진다. 여기서 t_b 와 t_r 는 각각 상황의 시작 시간과 종료 시간이다. s_n 은 상황에 대한 실제 정보를 담고 있는 것으로 논리 명제의 집합이다. $[\forall, \exists] t \text{ in } \langle \text{time range} \rangle [\text{context, derivative, action, situation}] \langle \text{compare} \rangle \langle \text{value} \rangle$ 와 같은 표현을 따른다[2].

Context Manager: SA Data Manager로부터 정보를 전달받고, 수집된 정보를 반대로 전달해 주기도 한다. 정황 정보(Context data)는 타임스탬프에 해당하는 시간과 상황 정보에서 전달된 정황으로 이루어진다.

Situation Analyzer: 상황 인지 정보를 바탕으로 상황을 분석하고, 임무를 수행하기 위한 최적의 선택을 하도록 한다. 튜플 $\langle ID, Type, Contents, Period, Deadline, Priority \rangle$ 로 GCU와 통신을 하게 된다. ID는 정보의 고유한 번호이고, Type은 정보의 종류를 나타낸다. 표1에서 보듯이 종류에 따라서 튜플에 들어가는 정보가 달라진다. Action Type일 경우, a_n은 할당된 행동을 의미하고,

r_n 은 행동을 취하기 위해서 필요한 자원을 나타낸다. 그리고 Period는 필요에 따라서 정보가 유효한 기간을 설정하는 기능을 한다. Deadline은 실시간 분산처리 방식에서 정보가 오가는 지연시간을 고려하는데 이를 위해 설정하는 것이다. 실시간 분산처리 시스템에서 정보를 서로 전달하는 데 걸리는 시간은 중요하다. 시간이 지체되거나 통신 장애로 정보가 전달되지 않는다면, 주어진 임무를 수행하는 데에 심각한 장애를 가져오거나, 정보의 불일치를 일으킬 수 있기 때문이다. 따라서 Deadline을 설정하여서 지정된 시간 동안 전달이 되지 않는지 확인하여야 한다. $a+d-c$ 로서 deadline을 고려하는데 a 는 정보가 도착한 시간을 나타내고, d 는 시스템에 맞게 설정되는 deadline 상수이다. 마지막으로 c 는 각 시스템간의 통신시간이다.

표 1. Contents of Tuple

Type	Contents
Situation	$\langle t_b, t_i, s_1, s_2, \dots, s_n \rangle$
Context	$\langle t_b, t_i, c_1, c_2, \dots, c_n \rangle$
Action	$\langle t_b, t_i, a_1, a_2, \dots, a_n, f_1, f_2, \dots, f_n \rangle$

Action Scheduler: 주어진 행동을 할당하고, 수행하도록 한다. 다수의 행동이 동시에 주어졌을 경우, 주어진 우선 순위에 따라서 스케줄링을 하도록 한다.

3.2 Guidance & Control Unit

레이더, GPS, Seeker 등 정보 수집이 가능한 기기를 통해서 정보를 수집하고, 이 정보와 데이터베이스의 이전 기록 및 예상 시나리오를 분석하여서 임무를 할당하게 된다. 이러한 상황은 데이터베이스에 저장되어서 차후 임무 수행을 위해서 사용되도록 한다.

Knowledge Based Database: 상황과 주변 환경에 대한 정보를 저장하고, 임무 완료 후 상태에 대해서도 데이터베이스에 저장한다. 이는 유사한 환경 하에서 임무를 수행할 경우, 판단을 내리는데 있어 참고할 수 있고 유사 시나리오를 유추할 수 있게 한다.

Decision Analyzer & Manager: 임무의 분석과 결정에 있어서는 퍼지 논리 기술 등을 적용하여 기반 정보를 바탕으로 행동을 결정할 수 있고, 직접 조작하여서 입력할 수 있다. 사람의 제어를 떠난 시스템들에게 행동을 전달하고 수집된 정보를 전달하고 받는 역할을 한다.

4. 실험 및 평가

중대 핵심 시스템의 대표적인 사례는 군사 임무와 관련된 시스템이다. 이는 상황에 맞도록 정확하고 빠르게 주어진 임무를 처리해야 한다. 군사 임무 시뮬레이션 중 실시간 분산 시스템으로 구성되는 공중 방어 시스템 시뮬레이션으로 본 논문에서 제안한 상황 인지를 적용하여

실험하여 본다.

그림 3은 공중 방어 상황에 대한 전체적인 시나리오를 보여준다. 중앙 본부는 지식 기반 데이터베이스를 구성하여 많은 정보를 저장해두고, 분석한 정보로부터 현재 상황에서 발생할 가능성이 있는 시나리오를 선택하게 된다. 실시간으로 들어오는 정보와 시나리오를 바탕으로 비상 상황에서 미사일을 발사하고 목표물을 결정한다. 발사된 미사일들은 주어진 목표물로 이동하면서 주위 정보를 장착된 센서를 통해 수집하고 본부와 통신한다. 주위의 환경과 상황을 빠르고 정확하게 알기 위해서 본부와 자신의 정보 뿐만이 아니라 발사된 미사일 간의 정보 공유를 한다.

원래 임무를 수행하기 힘든 상황이거나 새로운 목표물이 나타났을 경우, 본부에 정보를 보내고 결정을 기다리는 동시에 미사일 자체의 상황 인지 및 분석을 통해 목표물을 유동적으로 변동하고 주위 미사일들에게 전달한다.

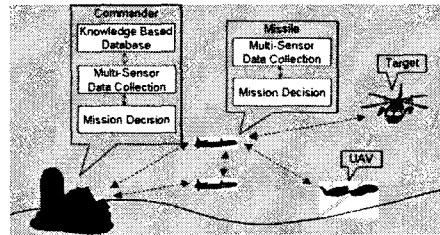


그림 3. 공중 방어 시나리오

그림 4는 영향 요소 변화에 따른 모의실험을 IDEFO 형태로 구체화하여 표현한 것이다. 성능 측정의 영향요소를 입력사항과 현재 실험환경 구축에 따라 결정된 실험 자산으로 다시 구분하였다. 따라서 성능 측정 간 가변적으로 적용할 수 있는 영향 요소는 입력사항으로 제한된다. 이러한 성능 측정 방법을 통해 얻고자 하는 결과는 미사일 임무 수행시간과 명중률이다. 이러한 측정 결과는 상황 인지를 적용한 시스템과 그렇지 않은 시스템을 비교하는 자료로 사용되어서 상황 인지를 적용한 시스템의 효용성을 평가할 수 있는 자료로 활용한다.

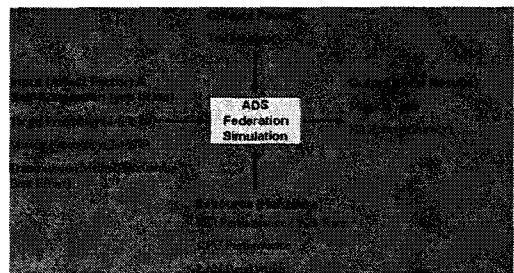


그림 4. 성능 측정 수행 계획 모델

시뮬레이션에서는 목표물의 개수를 변화시켜서 목표물을 격추시키기 위해 발사된 미사일의 격추 시간과 명중

를 기록하였다. 격추시간을 산출물로 포함한 이유는 상황 인지 기술에 따른 연산 과정에서 기존의 시스템에 비하여 지연될 수 있기 때문이다. 시뮬레이션에서 목표물이 격추되고 나면, 미사일의 총 비행시간, 저격시간, 발사시간, 본부 레이더와 미사일 간의 통신시간, 통신 횟수, 목표물 ID, 미사일 위치 등을 기록되고 데이터베이스에 저장한다.

표 2는 시뮬레이션 결과 측정된 결과치의 일부를 보여 준다. 분석결과, 상황 인지를 적용한 시스템이 적용하지 않은 기존의 시스템보다 미사일의 총 비행시간은 크게 차이가 나지 않았다. 측정 오차 10ms 이내의 결과 치에서 목표물의 수가 증가함에 따라서 차이는 조금 커졌으며 최대 500ms의 차이를 보였다. 임무 수행 결과를 보면, 중요도가 높게 판단되는 목표물을 우선적으로 처리하는 결과를 보였고, 또한 목표물의 명중률도 평균적으로 84%에서 88%로 향상되었다. 이를 통하여 상황 인지를 시스템에 적용하여 지연되는 시간은 크지 않았고, 지연되는 시간에 비하여 명중률은 매우 향상되었다. 중요 목표물이 포착될 경우, 미사일에서 우선적으로 상황을 인지하고 목표물을 수정한다. 본부로 전송된 정보를 토대로 본부에서 수정된 임무가 전송될 때까지 미사일 자체적으로 판단한 최적의 목표물로 변경하여서 추적하게 된다. 이는 본부와와의 통신이 지연되더라도 임무를 유동적으로 수행할 수 있고, 전송 시간 동안 정보를 기다리지 않고 임무를 빠르게 처리할 수 있다.

표 2. 실험 결과

목표물 수	상황 인지 미적용		상황 인지 적용	
	1st 미사일 격추 시간	2nd 미사일 격추시간	1st 미사일 격추 시간	2nd 미사일 격추시간
4	36777.02	38866.85	36776.03	38926.31
6	46384.25	49010.48	46589.68	49235.68
8	48293.83	51184.50	48582.40	51681.94
10	56700.14	59859.22	57053.07	60360.36

5. 결론 및 향후 연구

본 연구에서는 상황 인지를 적용한 중대 임무 시스템을 제안하였다. 일반적으로 센서를 통해 정보를 취득하고 지식 기반 데이터베이스와 함께 연동되는 아키텍처에 상황 인지 아키텍처를 추가하였다. 또한, 제시된 시스템의 유효성을 검증하기 위하여, 실시간 공중 방어 시뮬레이션을 개발하고 상황 인지 기술을 적용하여 비교 실험하였다. 실험 결과, 임무수행 중 상황 인지를 처리하기 위해 소요된 시간은 미적용 시스템에 비해 차이가 없는 것을 확인하였다. 이는 주어진 임무를 수행하기 위해 상황 인지 기술이 많은 부하를 주지 않는 것을 의미한다. 또한 임무 결과에서 이전 시스템에 비해 시스템 주변의 상황 변화에 따라 적절히 임무를 결정하여 처리하여 향상된 결과를 보였다. 실험환경은 무기 체계 구현 시뮬레이션을 위해 주로 사용되고 있는 HLA/RTI 기반으로 하였기 때문에 실제 상황에서도 실험 결과와 동일한 결과를 산출할 수 있다.

상황 인지를 적용하여 중대 임무 시스템을 구성 할 경

우, 임무를 수행하기 위한 성능을 향상 시킬 수 있고, 주변 환경 정보를 실시간으로 최대한 활용할 수 있다.

향후에는 실제 지식 기반 데이터베이스에서 도출된 시나리오와의 충돌에 따른 대처 알고리즘을 연구할 계획이다. 이를 토대로 시뮬레이션 상에서 그치지 않고, 실제 상황 속에서 로봇을 통해 실험하여 결과를 분석하고자 한다.

참고 문헌

- [1] C. M Krishna, K. G Shin, "Real-Time Systems" McGraw Hill, Newyork, 1999.
- [2] S. Yau, Y. Wang, and F. Karim, "Developing Situation-Awareness in Middleware for Ubicomp Environments," Proc. 26th Int'l Computer Software and Applications Conf. (COMPSAC 2002), pp. 233-238.
- [3] Yau S, Huang D, Gong H, Seth S, "Development and Runtime Support for Situation-Aware Application Software in Ubiquitous Computing Environments", COMPSAC -NEW YORK-, v.28, 2004, pp.452-457
- [4] Yau S, Huang D, Gong H, Yao Y, "Support for situation awareness in trustworthy ubiquitous computing application software", Software: practice & experience, v.36 no.9, 2006, pp.893-921
- [5] Ko, Eung-Nam, "An Adaptive Fault Tolerance for Situation-Aware Ubiquitous Computing", Software Engineering Research, Management and Applications, Third ACIS International Conference on, 2005, pp.144-151
- [6] "Distributed real-time problem statement: The Air Defense System ", Parallel and Distributed Real-Time Systems, 1997. Proceedings of the Joint Workshop on, 1997, pp.119-129
- [7] Lin C.E, Chen K.L, "Automated Air Defense System Using Knowledge-Based System", IEEE transactions on aerospace and electronic systems, v.27 no.1, 1991, pp.118-124
- [8] Hua Xing Qing, Jie Lei Ying, Xian Liu Fu, "Study on Knowledge Processing Techniques in Air Defense Operation Intelligent Aid Decision", Computational Intelligence and Multimedia Applications, ICCIMA 2003. Proceedings, Fifth International Conference on 2003, pp.114-119
- [9] U.S. Department of Defense (DMSO), "High Level Architecture Run-Time Infrastructure (RTI) Programmer's Guide Version 1.3," <http://hla.dmsomil>, 1998.
- [10] T. D. Lee, B. J. Jeon, and S. Y. Choi, "RISA: Object-oriented Modeling and Simulation of Real-time distributed Systems for Air defense" Lecture Notes in Computer Science, OOIS'03, September 2003.