상황인식 기반의 침입 감지 시스템에 관한 연구

윤이선*, 장규진*, 김규진*, 김문현*
*성균관대학교 컴퓨터공학과
e-mail:barisada84@naver.com

A Study for Intrusion Detection System Using Context-Awareness

Yi-Sun Yoon*, Gyu-Jin Jang*, Gyu-Jin Kim*, Moon-Hyun Kim*
*Dept of Computer Engineering, SungKyunkwan University

요 약

오늘날 사회가 발달하면서 기술 뿐 아니라 인적/물적 피해를 야기하는 여러 사건과 사고 또한 증가하고 있다. 상황인식 기반의 침입 감지 시스템의 효율적 개발이 가능하고 사건 및 사고를 사전에 방지하고 피해를 최소화 하는 수단이 될 것이다.

기존의 감시 시스템은 감시 관리자가 모니터링 장비 앞에서 상주하며 감시 장비가 전해오는 정보를 관리자가 직접 받으면서 스스로 판단해야만 했다. 이러한 방식은 인적 자원의 낭비를 초래하고 매우 비효율적이다. 또한 인적 자원에 의한 과실이 발생하기 쉽다는 문제점이 존재한다.

본 논문에서는 센서 등의 장비들을 통하여 정보를 수집하고 정의된 규칙에 따라 침입 상황을 감지하여 적절한 후속 조치를 취하는 상황 인식 기반의 침입 감지 시스템을 제안한다.

1. 서론

오늘날의 사회는 인적, 물적 피해를 야기하는 각종 사건 및 사고가 계속 증가하는 추세이며, 그에 따라 이러한 사건을 예방하기 위한 노력의 결과로 다양한 감시 시스템들이 등장하게 되었다. 특히 영상을 통해 유용한 정보를 많이 추출해 낼 수 있기 때문에 영상 기반 감시 시스템은 가장 널리 이용되어왔다. 그 중에서도 가장 일반적으로 사용되는 CCTV 방식은 비교적 저렴한 비용이 장점이지만, 감시업무를 맡은 담당자가 지속적으로 모니터링을 수행해야만 한다는 점과 직접 인력의 운용에 따라 발생할 수 있는 여러 문제들이 발생할 수 있다는 점에 있어 단점이 존재한다[1].

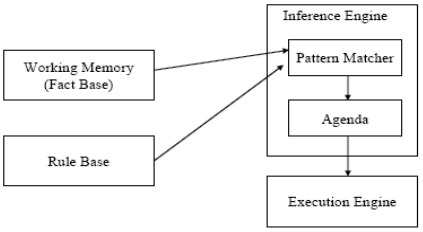
상황인식 기반의 감시 시스템은 이러한 단점을 극복할 수 있는 좋은 해결책이 될 수 있다. 본 논문에서는 감시 지역 내에 설치되어 있는 센서들로부터 받은 신호들을 기본으로 하여 상황을 인식하고 적절한 조치를 취하여, 사고 발생을 예방하거나 피해를 최소화하기 위한 상황인식 기반의 침입 감지 시스템을 제안한다. 본 논문은 2장에서 관련 연구를 소개하며, 3장에서 Jess를 이용한 침입 감지 시스템을 소개하고 4장에서 결론을 맺는다.

2. 관련 연구

상황 인식 컴퓨팅은 1994년 Schili와 Theimer에 의하여 최초로 논의되었다. 여기에서는 상황인식 컴퓨팅을 '사용 장소와 주변 사물, 물체의 집합에 따라 적응적이며 동시에 시간의 변화에 따르는 대상의 변화까지도 수용 가능한 소프트웨어'로 정의한다[2].

추론엔진은 이미 작성된 지식베이스나 사실들을 룰 기반으로 하여 새로운 사실이나 질의를 추론한다. 추론방법은 도메인의 특성이나 목적에 따라 정해진다[3].

Jess(Java Expert System Shell)는 1990년대 후반 캘리포니아 리버모어에 있는 국립 Sandia 연구소에서 개발된 Java 플랫폼을 위한 규칙 엔진 및 스크립트 언어이다. Jess는 동적이며 Java 중심언어이기 때문에 자동적으로 Java의 네트워킹, 그래픽, 데이터베이스 접근 등의 강력한 API를 호출 할 수 있다[4, 5].



(그림 1) Jess 시스템의 구조

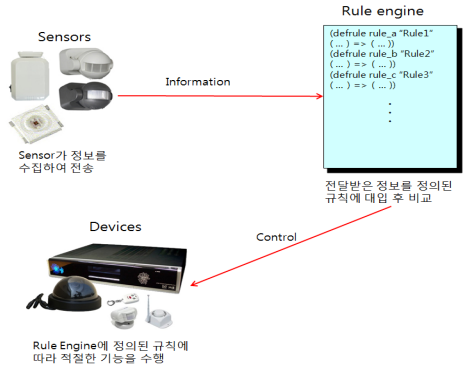
Jess의 시스템 구조는 그림 1과 같다. 규칙이 적용될 대상이 되는 데이터는 fact라고 하는 기본 원소로 구성되며 working memory 또는 fact base에 존재한다. Rule Base에는 지식을 나타내는 규칙의 집합이 존재하며, if-then 형태로 정의된다.

Working memory에 대해서 if 절이 만족되어 실행되어야 할 규칙이 존재하는지를 찾는 모듈이 Pattern Matcher이며 찾아진 규칙을 저장하고 실행할 순서를 결정하는 모듈이 Agenda이다. Excution Engine은 agenda에 존재하는 규칙들의 action 부분을 실제로 실행시키는 일을 한다.

3. 제안 시스템

본 논문에서는 감시 구역에서 발생할 수 있는 몇 개의 침입 상황을 가정하고 Jess를 사용하여 규칙을 설정하고,

설정된 규칙에 따라 반응하는 침입 감지 시스템을 구성하였다. 침입 감지 시스템의 구조는 (그림 4)와 같다.



(그림 2) 시스템 구성도

다음은 침입이 발생하는 등의 사고를 감지 시 정해진 규칙에 의하여 실행되는 함수를 정의하는 구문이다. 시스템에 정의된 규칙을 만족하는 경우 실행될 수 있는 함수로서는 경보를 발생하는 Alarm과 화면을 녹화하는 Record를 정의하였다.

```
(deffunction Alarm () (printout t "Alarm!" crlf))
(deffunction Record () (printout t "Recording!" crlf))
```

(그림 3) 함수 정의

다음은 감시 시스템의 객체가 되는 몇 가지 요소를 정의하는 구문이다. 인물을 나타내는 person은 객체의 요소로써 감시구역에 대한 접근 권한을 의미하는 access와 인증된 인물인지의 여부를 의미하는 auth를 정의하였다. sensor의 경우 움직임 탐지에 관련된 요소인 move와 음향 탐지 요소인 sound를 정의하였다. door의 경우 status를 통해 문의 열림/닫힘 상태를 나타내고, auth는 비밀번호 등을 통한 인증 여부, lock은 문의 잠김 상태를 표시한다. mode의 경우 status 항목을 통해 경비 상태인지, 일반 상태인지를 표시한다.

```
(deftemplate person (slot access) (slot auth))
(deftemplate sensor (slot move) (slot sound))
(deftemplate door (slot status)(slot auth)(slot lock))
(deftemplate mode (slot status))
```

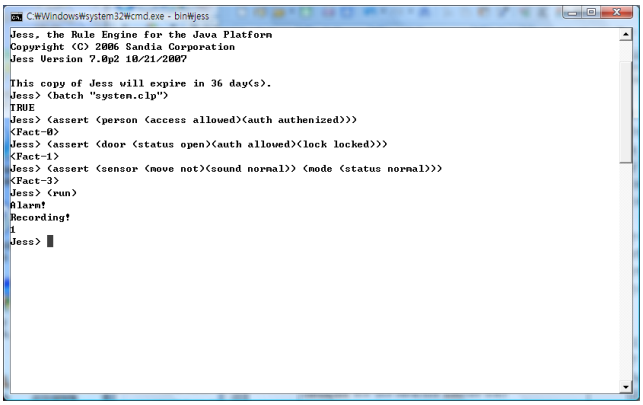
(그림 4) 템플릿 정의

다음은 감시 시스템에서 적용할 규칙을 정의하는 구문이다. Case 1은 “문의 비밀번호 등을 입력받지 않은 채 강제로 문이 열린 경우”를 가정하여 규칙을 정의 한 것이다. 비슷한 경우로 Case 2는 “문이 잠겨있는 상태에서 강제로 문이 열린 경우”를 가정하였다. Case 3는 “경비모드인 상태에서 감시구역 내부에서 센서 등에 의하여 움직임이 감지된 경우”를 가정하였다. Case 4는 “접근이 제한되거나 신원을 알 수 없는 자가 감시구역에서 감지된 경우”를 가정하였다. Case 5는 “감시구역 내에서 파열음이 들린 경우”로서, 강도 상황이나 폭발 사고와 같은 경우를 가정하였다.

```
(defrule invasion_a "Case 1"
(door {status == open && auth == not}) => (Alarm)(Record))
(defrule invasion_b "Case 2"
(door {status == open && lock == locked}) => (Alarm)(Record))
(defrule invasion_c "Case 3"
(sensor {move == moved})(mode {status == alert}) => (Alarm)(Record))
(defrule invasion_d "Case 4"
(person {access == not || auth == not}) => (Alarm)(Record))
(defrule invasion_e "Case 5" (sensor {sound == explosive}) => (Record))
```

(그림 5) 규칙(Rule) 정의

시스템을 시뮬레이션 하는데 있어 실제 디바이스와의 연동이 이루어지지 않은 상태였으나, 시뮬레이션의 결과는 규칙에 따라 모두 올바르게 동작하였다.



(그림 6) 실행 화면

4. 결론

본 논문에서는 Jess를 기반으로 침입 감지 시스템을 설계하고 테스트해 보았다. 상황인식 기반의 감시 시스템은 각종 사고에 대하여 기존의 감시 시스템의 결점에서 벗어나 효율적인 인력사용 및 보다 적절하고 신속한 조치를 취할 수 있도록 한다. Jess를 기반으로 하는 감시 시스템은 자바 플랫폼과 연동하여 사용할 수 있기 때문에 개발이 용이하고 기능 및 규칙의 수정이 가능한 이점이 있다.

앞으로 각 감시 지역에 적합한 구체적인 사고 상황을 분석하고 그에 따라 적용되어야 할 규칙 및 기능을 구체화할 필요가 있다고 생각된다. 또한 본 논문에서 수행하지 못한 실제 디바이스와의 연동에 관한 작업도 더 필요할 것이다.

참고문헌

[1] 정해원 외 2명, “지식 기반 지능형 감시 시스템에 관한 연구”, Proceedings of KIIS Fall Conference 2008, Vol .18, No.2, pp.343-346, 2008.

[2] 임신영 외 3명, “상황인식 컴퓨팅 기술 동향”, IITA 주간기술동향, 제1142호, pp.1-15, 2004.

[3] 김성태 외 2명, “추론엔진을 이용한 웹서비스 기반 추천 시스템”, 한국지능정보시스템학회, 10권, 3호, pp.59-72, 2004

[4] E.Friedman-hill, 'Jess in Action : Rule-Based Systems in Java', 1st ED., Manning, 2003.

[5] 최승훈, “JESS 규칙 기반 시스템을 이용한 특성 구성 검증”, 한국 인터넷 정보학회, 8권, 6호, pp.135-144, 2007.