

MADA Feedback Framework for Adaptive Inference Results of Situation Aware Services in Mobile Environments

Sooheye Shin[†] · Joonseok Park^{**} · Keunhyuk Yeom^{***}

ABSTRACT

The high portability and mobility of mobile devices have given rise to critical concerns with regard to mobile situation-aware services that utilize user situation information in the mobile domain. Mobile situation-aware services for mobile devices are provided by collecting and analyzing data, from which the user situation is inferred through an inference system. However, the high mobility of the mobile domain has made it difficult to infer the required results. In addition, previous studies have not identified a systematic approach to modifying and adapting. In this paper, we propose a systematic feedback process model based on software cybernetics in order to address the abovementioned problems. Further, we propose a MADA(Monitoring, Analysis, Determinating, Adaptation) framework for the feedback process model. Thus, the proposed approach supports the development of self-adaptive mobile situation-aware services that can infer appropriate results and manage the inferred results systematically.

Keywords : Mobile Situation Aware Services, Inference of Situation, Feedback, Software Cybernetics, Framework

모바일 상황인식 서비스의 추론 결과 적응을 지원하기 위한 MADA 피드백 프레임워크

신 수 혜[†] · 박 준 석^{**} · 엄 근 혁^{***}

요 약

최근 높은 휴대성과 이동성을 특징으로 하는 모바일 도메인에서 사용자의 상황정보를 이용한 모바일 상황인식 서비스에 대한 관심이 증대되고 있다. 모바일 상황인식 서비스들은 사용자의 상황정보를 모바일 기기로부터 수집 및 분석하여 추론 시스템을 통해 상황을 추론해야 한다. 그러나 모바일 도메인의 높은 이동성으로 인해 상황에 적합한 추론 결과를 얻기에 어려움이 있다. 또한, 잘못된 추론 결과를 수정 및 적용하기 위한 체계적인 방법에 대한 연구가 부족하다. 본 논문은 이러한 문제점을 개선하기 위해 소프트웨어 사이버네틱스 기반 피드백 개념을 적용한 체계적인 프로세스 모델을 제시한다. 또한 피드백 프로세스 모델을 적용, 설계 및 구현한 MADA(Monitoring, Analysis, Determinating, Adaptation) 프레임워크를 제시한다. 이를 통해 정확한 상황 추론이 가능한 적응적 모바일 상황인식 서비스 개발 환경을 구축할 수 있으며 추론 규칙의 체계적 관리가 가능하다.

키워드 : 모바일 상황인식 서비스, 상황 추론, 피드백, 소프트웨어 사이버네틱스, 프레임워크

1. 서 론

모바일 환경에서 사용자에게 정확한 상황인식 서비스를 제공하기 위해서는 잘못 추론한 상황 결과에 대해 원인을 분석하고 이를 수정 및 개선하여 정확한 상황을 추론할 수

있도록 지원하는 적응적(Adaptive) 상황인식 서비스 연구가 필요하다. 기존 적응적 상황인식 서비스를 위한 다양한 연구 중의 하나로 제어 이론과 피드백 기법을 포함한 소프트웨어 사이버네틱스 연구[1-3]가 있다. 이는 사용자와 서비스 간의 피드백을 일련의 체계적 프로세스를 통해 서비스의 변화를 제어할 수 있도록 연구하는 분야로 요구사항 습득 및 소프트웨어 테스트 등 소프트웨어 공학 전 과정에 걸쳐 적용되고 있다.

본 논문은 소프트웨어 사이버네틱스 개념을 적응적 모바일 상황인식 서비스 개발 연구에 적용하여 잘못 추론된 결과의 적응을 통해 정확한 상황 추론을 지원하는 상황인식 서비스 개발 방법을 제시한다. 이를 위해 모바일 상황 정보

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학IT연구센터육성 지원 사업의 연구결과로 수행되었음(NIPA-2013-(H0301-13-1012)).
† 비 회 원 : 부산대학교 컴퓨터공학과 석사
** 정 회 원 : 부산대학교 컴퓨터공학과 박사
*** 정 회 원 : 부산대학교 정보컴퓨터공학부 교수
논문접수: 2013년 1월 28일
수 정 일: 1차 2013년 3월 12일
심사완료: 2013년 3월 21일
* Corresponding Author: Keunhyuk Yeom(yeom@pusan.ac.kr)

를 모델링하고 잘못 추론된 결과의 수정 및 적응을 위한 피드백 프로세스 모델과 이를 지원 및 검증하기 위한 프레임워크를 제시한다.

2. 관련 연구

2.1 피드백

1) 소프트웨어 사이버네틱스 기반 피드백

소프트웨어 사이버네틱스는 달성하고자 하는 목표를 설정하고 피드백을 통해 목표 달성 방향으로 동작하도록 연구하는 학문이다. 소프트웨어 사이버네틱스 기반 시스템은 소프트웨어 및 서비스의 제어와 실행 문제를 피드백을 통해 해결하며, 자가 적응과 진화가 가능하도록 지원한다. 여기서 피드백은 “기계나 시스템, 프로세스에서 출력이 입력으로 회귀되는 상황”으로 정의된다.

소프트웨어 사이버네틱스 제어 이론에서 사용하는 피드백 제어[4]는 출력의 회귀 방향에 따라 개방 루프 제어와 폐쇄 루프 제어로 나뉜다. 일반적으로 폐쇄 루프 제어를 피드백 프로세스라 하며, 대표적인 피드백 프로세스 설계 모델로 MAPE(Monitoring, Analysis, Planning, Execution) 프로세스[5]와 PID(Proportional, Integral, Deviative) 컨트롤러[4]가 있다.

2) 피드백 프로세스 연구

소프트웨어 기존 피드백 프로세스 모델을 적용한 관련 연구[4-6]가 있다.

X. Peng은 런타임 시 변하는 품질 요구사항을 반영하기 위해 PID 컨트롤러 모델을 적용한 자가 튜닝(Self-tuning) 방법[4]을 제시하였다. 그러나 변화하는 요구사항의 상태 표현 방법이 부족하다.

C. Tudor는 분산적 실행 환경에서 변화를 감지하고 반응하기 위한 프로세스로 MAPE 피드백 프로세스[5]를 적용하였다. 이를 통하여 구체적이고 자동화된 피드백 처리를 기대할 수 있으나 상황에 대한 변화와 이를 감지하기 위한 대안이 부족하다.

또한, C. Chang은 상황인식 서비스에서 요구사항 분석 단계의 상황정보 분석과 목표 도출 단계를 통하여 피드백을 통한 제어를 위한 진화 프로세스[6]를 제안하였다. 설정된 목표를 기반으로 HMM (Hidden Markov Model) 기법을 사용하여 시스템적 제어를 판단 및 결정하는 순환적 프로세스를 제안하였다. 그러나 모바일이 아닌 일반적 상황인식 시스템을 도메인으로 하여 모바일의 특성이 반영되어 있지 않으며 결정된 변화에 대한 적응과 수정에 대한 방안이 제시되어 있지 않다.

2.2 온톨로지 기반 모바일 상황인식 서비스 개발 방법

Fig. 1과 같이 온톨로지 기반 모바일 상황인식 서비스 개발 방법은 온톨로지 및 OWL[7]을 이용하여 모바일 도메인

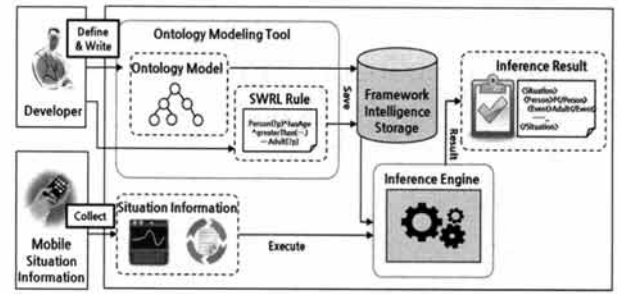


Fig. 1. A methodology to develop mobile situation-Aware Service based on ontology

의 상황 정보를 명세하고, SWRL[8] 기반 모바일 상황 규칙을 작성하여 추론 상황을 표현한다. 그리고 모바일 기기를 통해 상황정보를 획득하여 Jess[9] 등의 추론 엔진 질의를 통해 상황을 추론해 낸다.

3. MADA: 모바일 상황인식 서비스의 추론 결과 적응을 위한 피드백 프로세스 모델

본 논문은 기존 MAPE 피드백 프로세스를 적용하여 적응적 모바일 상황인식 서비스 개발을 위한 MADA (Monitoring, Analysis, Determinating, Adapting) 피드백 프로세스를 제안한다. MADA 피드백 프로세스는 모바일 상황 정보와 온톨로지, 추론 규칙을 사용하여 잘못 추론된 상황의 원인을 분석하고 정확한 상황을 추론하기 위한 체계화된 프로세스 모델이다. Fig. 2는 MADA 프로세스의 각 단계별 수행 내역과 그 산출물을 나타낸 것이다.

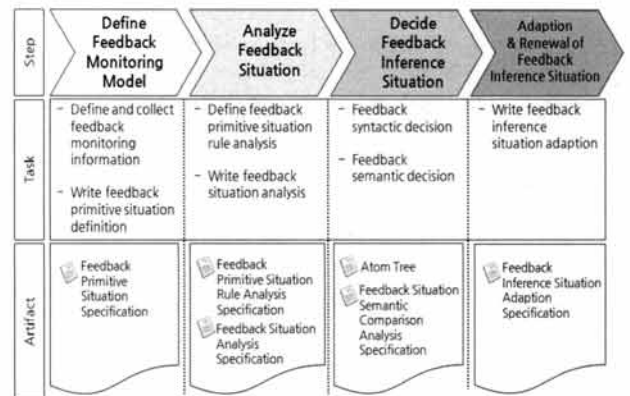


Fig. 2. Feedback Process Model based on Software Cybernetics

기존의 MAPE 피드백 프로세스의 계획 단계는 적절한 적응 동작을 선택 및 결정하는 단계라는 개념적 의미를 가진다. 그러나 MADA는 선택의 측면을 포함시키지 않고, 시스템으로부터 피드백을 결정하기 위한 구문적 결정과 의미적 결정에 초점을 두는 차이점이 존재한다. 따라서 계획 단계를 MADA에서는 결정단계로 변경하여 재정의하였다. 또한 MAPE 피드백 프로세스의 실행 단계는 계획 단계에서

선택 또는 결정된 적응 행위에 따라 시스템의 행위를 수정한다는 개념적 의미를 가진다. 그러나 MADA는 피드백을 통한 기존 상황정보의 수정 및 적응에 초점을 두고 있다는 차이점이 존재한다. 따라서 MADA에서는 적응 단계로 변경하여 재정의하였다.

3.1 피드백 모니터링 모델 정의

피드백 프로세스의 첫 단계는 추론하고자 하는 상황을 정의한 정보와 그에 따라 추론된 결과에 대한 정보가 필요하다. 따라서 적응적 모바일 상황인식 서비스를 위한 상황 및 추론에 대한 정보를 1차적으로 수집하는 피드백 모니터링 단계를 수행한다. 피드백 모니터링 단계는 적응을 위한 상황 및 추론 정보를 포함한 피드백 모니터링 정보를 정의하고 수집하는 단계와 상황을 기술하기 위해 상황을 단위적으로 표현한 피드백 기본 상황(Feedback Primitive Situation)을 정의한다.

1) 피드백 모니터링 정보 정의 및 수집 단계

피드백 모니터링 정보 정의 및 수집 단계에서는 온톨로지 모델에 대한 정보 정의와 상황에 대한 정보를 명세한다. 온톨로지 모델은 모바일 상황인식 서비스를 위한 상위 온톨로지 모델인 COM(Core Ontology for Mobile)[10]을 사용하여 표현한다. COM의 온톨로지 구조는 Fig. 3과 같다.

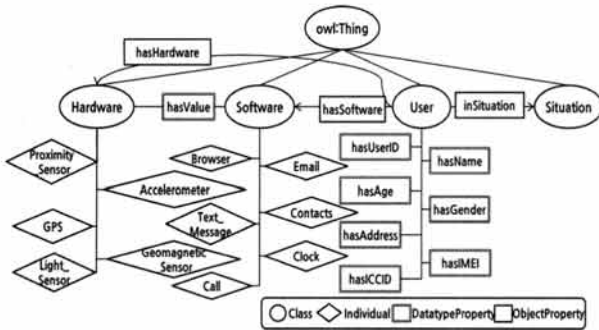


Fig. 3. COM Model[10]

COM은 모바일 기기나 센서를 표현한 하드웨어, 모바일 기기에 내장된 소프트웨어, 모바일 사용자인 사용자, 추론된 상황을 표현하는 상황 클래스로 구성된다. 또한, 하드웨어와 소프트웨어 클래스는 대부분의 모바일 기기가 가지는 하드웨어나 소프트웨어에 관한 개체들로 하드웨어는 GPS, Light_Sensor, Accelerometer, Geomagnetic_Sensor 등으로, 소프트웨어는 Clock, Browser, Call, Contacts 등의 개체를 가진다. 이와 같이 표현된 COM 모델을 사용하여 모바일 상황을 위한 사용자 및 하드웨어, 소프트웨어, 상황의 상황정보를 개체와 속성으로 표현할 수 있다.

2) 피드백 기본 상황 정의서 작성 단계

피드백 모니터링 정보는 COM 모델로 표현된 온톨로지 모델과 상황정보로 구성된다. 모바일 상황정보를 표현하기

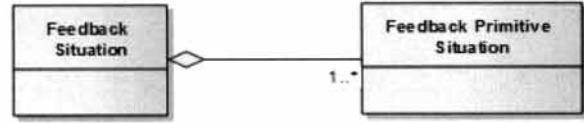


Fig. 4. Relationship between feedback situation and feedback primitive situation

위해서 피드백 상황 (Feedback Situation)과 피드백 기본 상황을 정의한다. 피드백 상황은 추론할 하나의 상황을 표현하기 위해 다수의 모바일 상황정보와 추론 규칙을 포함한다. 피드백 기본 상황은 상황정보의 기본 단위가 되는 COM 모델의 개체로 표현되며, 피드백 상황은 피드백 기본 상황의 집합으로 Fig. 4와 같이 관계를 표현할 수 있다.

피드백 기본 상황을 정의하기 위해 Table 1과 같이 피드백 기본 상황 정의서를 작성한다.

Table 1. Feedback primitive situation specification

ID	Primitive Situation	Individual	Base Value	User Information	Explanation

3.2 피드백 상황 분석

피드백 모니터링 모델 정의를 통해 정의된 피드백 기본 상황을 수집하여 피드백 상황을 분석한다. 피드백 상황 분석 단계는 피드백 기본 상황을 SWRL 규칙 기반으로 분석하는 단계와 이를 조합하여 하나의 피드백 대상 상황을 정의하는 단계로 구성된다.

1) 피드백 기본 상황 규칙 분석 단계

피드백 기본 상황 정의서를 분석하여 SWRL 규칙 기반 피드백 기본 상황 규칙 분석서로 확장 및 세분화하여 작성한다. 각 항목에 대한 작성 방법은 아래와 같다.

- 피드백 기본 상황명과 ID: 피드백 기본 상황 정의서의 기본 상황명과 ID를 기입한다.
- 조건부: SWRL 규칙은 조건들의 결합으로 구성된 조건부로 표현된다. 조건은 온톨로지 모델과 대응되는 클래스 연산(Class Atom), 데이터 값 속성 연산(Data-valued Property Atom), 비교 연산(Built-in Atom)으로 구성되며 이에 대한 표현법은 아래와 같다.
 - 클래스 연산: COM 모델의 클래스를 기입하며, 변수는 클래스 변수를 기입한다.
 - 데이터 값 속성 연산: 목표가 되는 상황과 현재 상황 값을 규칙으로 표현한다.
 - 비교 연산: 목표 상황과 현재 상황 비교를 위해 SWRL 규칙에서 제공하는 Built-in 함수를 사용하여 표현한다.
- 결론부: SWRL 규칙 추론결과 상황을 기술한다.
 - 개체 속성 연산: COM 모델의 클래스 변수와 추론 상황 개체를 기입한다.

Table 2. Feedback primitive situation rule specification - dark situation

Primitive Situation		Dark		ID	FPS-1
Antecedent					
Class Atom	Class	Class Variable		SWRL Rule	
	com:user	?user		com:user(?user)	
Data-valued Property Atom	Datatype Property	Individual	Data Variable	SWRL Rule	
	j.0:hasStandardLight_Dark	?user	?standVal	j.0:hasStandardLight_Dark(?user,?standVal)	
	
Built-in Atom	Built-in	Current Variable	Base Variable	SWRL Rule	
	swrlb:lessThan	?curVal	?standVal	swrlb:lessThan(?curVal,?standVal)	
Consequent					
Individual Property Atom	Object Property	Class Variable	Individual	SWRL Rule	
	com:inSituation	?user	j.0:Light_Low	com:inSituation(?user,j.0:Light_Low)	

Table 2는 '어두움' 이라는 기본 상황을 토대로 피드백 기본 상황 규칙 분석서를 작성한 예이다.

2) 피드백 상황 분석서 작성 단계

피드백 상황은 다수의 피드백 기본 상황의 조합으로 구성된다. 피드백 상황 분석은 피드백 기본 상황 정의서와 규칙 분석서를 바탕으로 나타낸다.

피드백 기본 상황은 기본 상황 각 항목에 기입하는데 ID와 상황명에는 피드백 기본 상황의 고유 ID와 상황 명을 기입하고 SWRL 규칙에는 피드백 기본 상황 규칙 분석서로부터 작성된 SWRL 규칙을 기입한다. 그리고 피드백 기본 상황마다 추론 성공률을 비율로 나타내는데, 이는 피드백 프로세스가 순환할 때 추론 결과에 따라 성공률을 계산하여 다음 피드백 결정을 위한 단서로 사용할 수 있다. 비율의 초기 값은 100%로 설정한다.

Table 3은 '입실'의 피드백 상황에 대한 분석서의 예이다.

3.3 피드백 추론 상황 결정

피드백 상황 분석을 통해 작성된 피드백 상황 SWRL 규칙을 추론 엔진에 질의하여 추론 결과를 출력한다. 잘못 추론된 결과에 대해 수정 및 적용이 필요하게 되는데, 본 논문은 피드백 결정 및 적용을 통하여 잘못된 추론 결과를 수정하도록 한다.

피드백 추론 상황 결정은 SWRL 규칙의 문법적 구조를 통한 구문적 결정과 SWRL 규칙의 상황정보 값의 의미적 비교 분석을 통한 의미적 결정으로 구성된다.

1) 피드백 구문적 결정 단계

SWRL 규칙은 다수의 연산들의 조합으로 구성되기 때문

Table 3. Feedback situation analysis specification - Entering the Room situation

Situation	Entering the Room			No.	1
Description	User entered the meeting room. The meeting room's level of illumination is dark. In addition, user's location is within the standard range of GPS coordinates				
Primitive Situation	ID	Name	SWRL Rule	Ratio	
	FPS-2	Dark	com:User(?user) ^ j.0:hasStandardLight_Dark(?user,?standVal) ^ com:hasValue(comm:Light_Sensor,?curVal) ^ swrlb:lessThan(?curVal,?standVal)	100%	
	
SWRL Rule	com:User(?user) ^ j.0:hasStandardLight_Dark(?user,?standVal) ^ ^ swrlb:lessThan(?curVal,?standVal) -> com:inSituation(?user, j.0:Enter_the_Room)				

에 SWRL 규칙을 해석하기 어려움이 있다. 이를 극복하기 위해 S. Hassanpour의 연구[11]에서 SWRL 규칙의 구조적 가시화, 의역화, 분류화 방법을 제시하였고, 특히 SWRL 규칙의 구조적 가시화 표현을 위한 연산 트리를 제시하였다. 이에 따라 SWRL 규칙 구문적 결정은 S. Hassanpour의 SWRL 규칙 가시화 방법을 활용하여 제시한다. SWRL 규칙은 다수의 연산의 결합으로 구성되므로 피드백 상황의 SWRL 규칙을 연산 종류에 따라 Fig. 5와 같이 연산 트리로 가시화 한다.

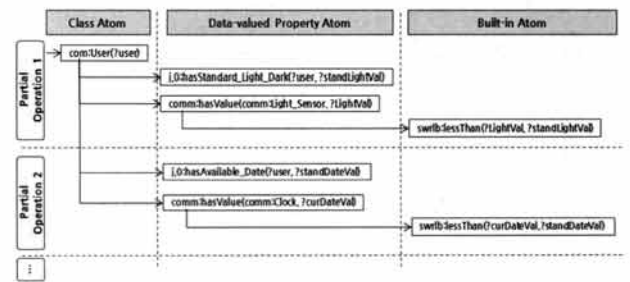


Fig. 5. Atom Tree

연산 트리의 횡적 측면은 연산 종류에 따른 분류이며 종적 측면은 피드백 기본 상황에 따른 분류로 구조적 가시화를 한다. 연산 트리로 가시화된 SWRL 규칙을 Table 4와 같이 EBNF 문법[12]에 따라 연산의 문법적 판단을 결정한다.

Table 4. SWRL rule operation representation based on EBNF grammar

<pre> atom ::= description '(i-object)' individualValuedPropertyID '(i-object i-object)' dataValuedPropertyID '(i-object d-object)' builtIn '(builtInID {d-object})' builtInID ::= URIReference i-object ::= i-variable individualID d-object ::= d-variable dataLiteral </pre>

각 연산의 변수가 EBNF 문법에 따라 정확하게 표현되었는지 확인하기 위해 피드백 상황의 각 피드백 기본 상황 별 구문적 결정을 수행한다.

2) 피드백 의미적 결정 단계

SWRL 규칙의 구문적 분석을 진행 한 후 구문적으로 이상이 없는 경우 의미적 결정을 수행한다. 의미적 결정은 구문적 분석을 통해 분석된 각 연산 변수 값의 실제 상황정보 값을 추출하여 피드백 기본 상황 정의서에 정의한 목표 설정 값과 비교 분석한다. 각 피드백 기본 상황에서 비교 대상인 변수를 비교 함수를 기준으로 비교 분석할 연산 변수 값을 추출한다. 피드백 기본 상황을 각 연산 별 EBNF로 표현하였을 때 데이터 값 속성 연산의 데이터 변수와 비교 연산의 현재 변수, 그리고 기준 변수의 대입되는 값을 비교한다.

피드백 상황 의미적 비교 분석서는 비교 분석할 변수와 피드백 모니터링 모델 정의에서 작성한 피드백 기본 상황의 목표 설정 값을 Table 5와 같이 비교 분석하여 정의한다.

Table 5. SWRL rule feedback situation semantic comparison specification - Entering the Room situation

Situation	Entering the Room			No.	1
Primitive Situation					
FPS-2-Dark	Feedback Primitive Situation Defined value	Individual	Base Value	User Info	
		Hardware :Light_Sensor	Lower than 70lux	Base bright range of user	
	Semantic Decision Value	i-variable	d-variable	d-variable	
		com :Light_Sensor	standLight Val	curLightVal	
...	...				

3.4 피드백 추론 상황 적응 및 갱신

잘못 추론된 SWRL 규칙의 수정을 위해 구문적, 의미적 결정을 피드백 추론 상황 결정에서 수행하였다. 이를 통해 구문적이거나 의미적으로 SWRL 규칙의 오류를 확인하고 실제 수정을 진행하는 피드백 추론 상황 적응 및 갱신 단계를 수행한다.

1) 피드백 추론 상황 적응서 작성 단계

피드백 구문적, 의미적 결정에 따라 오류의 원인을 파악할 수 있다. 본 논문은 피드백 모니터링 정보 모델 정의, 피드백 상황 분석에서 정형화된 형식의 SWRL 규칙을 생성하여 잘못 추론된 정보를 SWRL 규칙의 구문적, 의미적 결정에서 판단할 수 있다.

이처럼 판단된 오류의 원인을 파악하여 SWRL 규칙의 구문적, 의미적 수정을 진행한다. 구문적 수정은 SWRL 규칙의 문법적으로 수정하며, 의미적 수정은 피드백 기본 상황 정의서에 설정한 목표 값과 상황정보 값의 비교를 통해 설정 목표 값 및 상황정보 값의 변경을 수행한다. 위와 같

은 정보에 따라 SWRL 규칙의 적응 및 갱신을 수행한 뒤 수정된 내역에 대해 Table 6과 같은 피드백 추론 상황 적응서를 작성한다.

Table 6. Feedback inference situation adaptation-specification

Situation		Ver.		No.
Primitive Situation				
ID	Name	Syntactic Error	Semantic Error	Ratio

계속적인 적응을 나타내기 위한 피드백 상황 별 버전을 Ver에 기입한다. 피드백 기본 상황에는 고유 ID와 기본 상황 명을 기입하며, 각 피드백 기본 상황 별 잘못된 추론을 일으키는 상황에 구문적 오류인지 의미적 오류인지 동그라미로 표현한다. 또한 각 피드백 기본 상황 별 정확율을 비율로 나타낸다.

4. 피드백 프로세스 모델을 적용한 프레임워크 설계 및 구현

제시한 피드백 프로세스 모델을 적용한 프레임워크를 설계 및 구현하여 피드백 프로세스 모델을 검증한다. 기존 온톨로지 모델 기반 모바일 상황인식 서비스 개발 프레임워크 [10]를 확장하여 피드백 프로세스에 따라 추론 상황 적응이 가능하도록 확장 및 개발하였다. Fig. 6은 온톨로지 기반 모바일 상황인식 서비스 개발 지원 프레임워크 중 피드백 프로세스 적응을 위한 적응 관리자 확장 구조를 나타낸다.

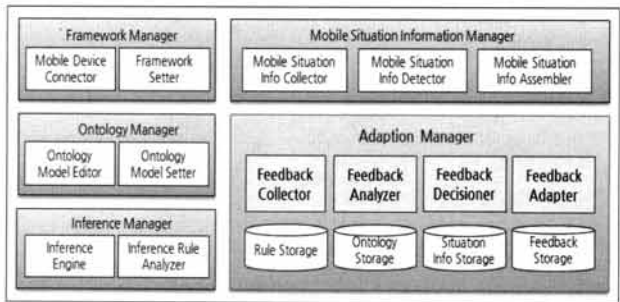


Fig. 6. Framework extension structure for applying feedback process

적용 관리자의 피드백 수집기, 피드백 분석기, 피드백 결정기, 피드백 적응기는 MADA 피드백 프로세스의 수집, 분석, 결정, 적응 단계를 지원한다. 각 단계별 모듈에 관한 입력 및 출력, 관계 흐름 등을 나타낸 추론 결과 적응을 위한 프레임워크 시나리오는 Fig. 7과 같다.

상황정보, 온톨로지, 추론 규칙을 피드백 수집기를 통해 수집하고 피드백 프로세스 각 단계별 수행 흐름에 따라 피드백 상황, 피드백 기본 상황을 정의하여 잘못 추론된 피드백 상황을 의미적, 구문적 결정을 통해 수정 및 적응을 수

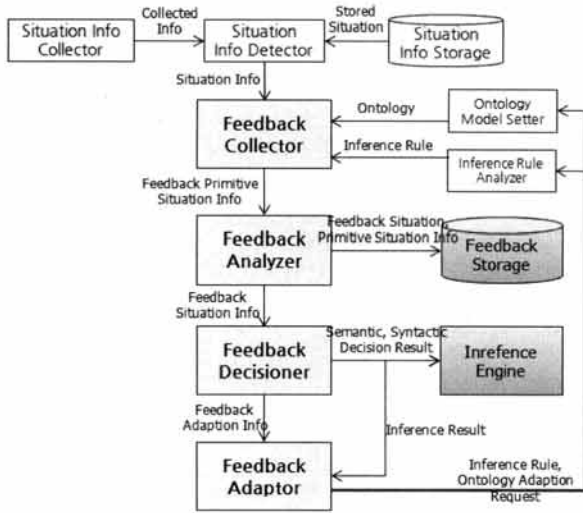


Fig. 7. Framework scenario for applying feedback process model

행하도록 프레임워크를 개발하였다. 프레임워크의 구현 언어는 자바이며, 개발 도구로 이클립스를 사용하였으며, OWL 편집 및 관리를 위한 도구로 Protégé API[12]를 사용하였다.

피드백 프로세스 단계 중 잘못 추론된 규칙의 피드백 상황 및 기본 상황을 판단하고 적용하기 위한 결정, 적용 단계를 반영한 프레임워크 실행 시퀀스 다이어그램은 Fig. 8 과 같다.

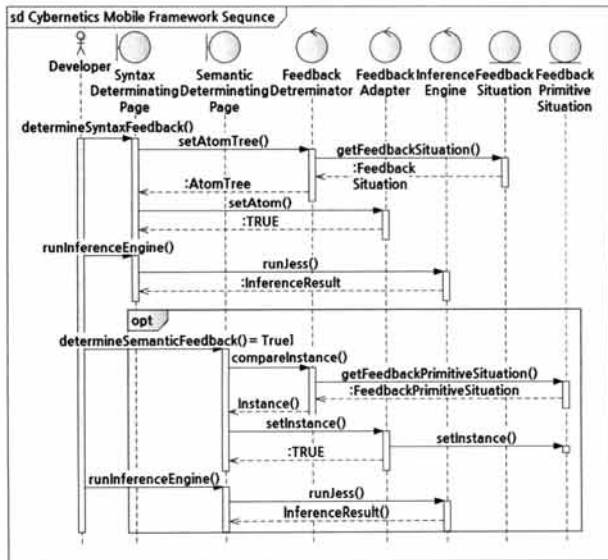


Fig. 8. Feedback decision, adaption execution sequence diagram

5. 사례 연구 및 검증

사례 연구를 위한 상황 시나리오는 모바일 기기를 통해 상황정보를 수집하여 해당 상황에 맞는 동작을 상황 추론을

통해 제어할 수 있는 스마트 오피스 컨트롤 시스템[2]이다.

스마트 오피스 내에서 추론 가능한 상황에는 입실, 프레젠테이션 준비 등이 있다. 프레젠테이션 준비 피드백 상황은 다섯 가지의 피드백 기본 상황으로 구성됨을 Fig. 9에서 나타내고 있다.



Fig. 9. Relationship between feedback situation and primitive situation - Ready for presentation situation

피드백 프로세스 중 첫 단계인 피드백 모니터링 정의 단계에서 작성하는 스마트 오피스 컨트롤 시스템의 피드백 기본 상황 정의서는 Fig. 10과 같다.

ID	Primitive Situation	Individual	Base Value	User Information	Description
FPS-1	Cold	Hardware :Temperature_Sensor	Below 18°	Based temperature range	Low Temperature
FPS-2	Dark	Hardware :Light_Sensor	Below 70lux	Based brightness range	Low Brightness
...

Fig. 10. Feedback primitive situation specification - smart office control system

두 번째 단계인 피드백 상황 분석 단계에서는 각 피드백 기본 상황의 추론 규칙을 작성하고 피드백 기본 상황의 조합을 통해 스마트 오피스 컨트롤 시스템에서 추론할 피드백 상황을 정의 및 분석한다. Fig. 11은 프레젠테이션 준비 피드백 상황에 대한 작성된 피드백 상황 분석서를 나타낸다.

Situation		Ready for Presentation		No.	4
Description The situation that user is ready for presentation. It should be available date for use conference room, and presentation file starts at slide 1. Room is dark. Temperature is lower than standard. Also user is placed within available GPS range. Distance is calculated by specific module.					
Primitive Situation	ID	Primitive Situation	SWRL Rules	Ratio	
	FPS-1	Cold	com:User(?user) ^ j:0:hasStandard_Temp_Winter(?standTempVal) ^ com:hasValue(com:Temperature_Sensor, ?curTempVal) ^ swrlb:greaterThan(?curTempVal, ?standTempVal)	100%	
	FPS-2	Dark	com:User(?user) ^ j:0:hasStandard_Light_Dark(?user, ?standLightVal) ^ com:hasValue(com:Light_Sensor, ?curLightVal) ^ swrlb:lessThan(?curLightVal, ?standLightVal)	100%	
SWRL Rules	com:User(?user) ^ j:0:hasStandard_Light_Dark(?user, ?standLightVal) ^ com:hasValue(com:Light_Sensor, ?LightVal) ^ swrlb:lessThan(?LightVal, ?standLightVal) ^ -- com:inSituation(?user, j:0:Ready_for_Presentation)				

Fig. 11. Feedback situation analysis specification - smart office control system

피드백 프로세스의 세 번째 단계인 피드백 추론 상황 결정 단계는 피드백 상황 분석을 통해 작성된 피드백 상황마다 추론 엔진에 질의하여 추론 상황의 잘못 유무를 판단한다. Fig. 12는 구문적 결정을 지원하기 위해 설계 및 구현된 프레임워크 화면이다. 해당 화면은 잘못 추론된 프레젠테이

션 준비 피드백 상황의 연산 트리를 자바의 jTree를 사용하여 가시화하고 EBNF 문법에 따라 비교하여 구문적 결정을 수행한다.

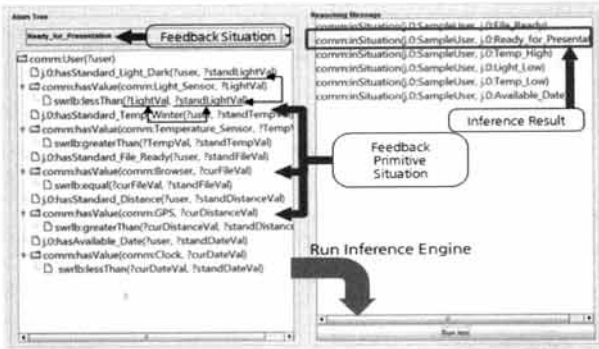


Fig. 12. User interface of feedback syntactic decision

피드백 구문적 결정을 통해 해당 피드백 상황에 오류가 없어 문제가 발생하지 않은 경우 Fig. 13과 같이 피드백 의미적 결정을 수행한다. 피드백 의미적 결정은 작성된 피드백 상황 의미적 비교 분석서에 따라 피드백 상황을 조합하는 각 피드백 기본 상황의 설정 값과 상황 값의 의미적 비교를 통해 수행된다.

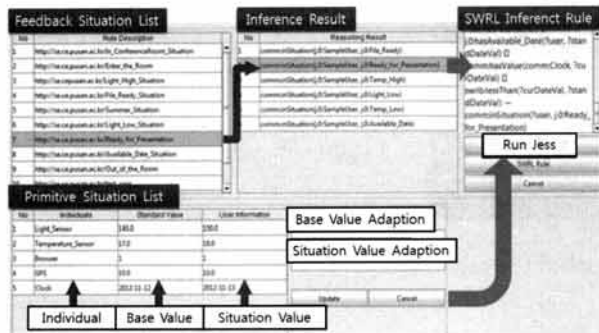


Fig. 13. User interface of feedback semantic decision

프로세스 마지막 단계는 피드백 적응 단계이다. Fig. 14와 같이 피드백 추론 상황 적용서를 작성하여 빈번한 오류를 일으키는 피드백 상황별 오류의 원인을 판별하고, 피드백 상황의 지속적인 추론 상태 관리를 가능하게 한다.

Situation	Enter the Room	Ver.	0.2	No.	1
Primitive Situation					
ID	Name	Syntax Error	Semantic Error	Ratio	
FPS-2	Light_Low	-	-	50%	
FPS-3	Available_Date	-	-	75%	
⋮					
Primitive Situation					
ID	Name	Syntax Error	Semantic Error	Ratio	
FPS-1	Temp_Low	○	○	20%	
FPS-2	Light_Low	-	-	50%	
FPS-3	Available_Date	-	-	75%	
FPS-4	File_Ready	-	-	90%	
FPS-5	Around_ConferenceRoom	-	-	100%	

Fig. 14. Feedback inference situation adaptation specification - smart office control system

스마트 오피스 컨트롤 시스템 사례에 기반하여 프레임워크 자체의 추론 결과 적응 정확도의 정량적 평가 결과를 제시한다. 사례 연구에서 추론한 상황들은 회의실에 대한 '입실', '퇴실', '프레젠테이션 준비', '프레젠테이션 종료', '질의 응답'의 총 5개의 상황이다. 구문적 추론상황 89개, 의미적 추론상황 150개의 총 239개의 추론 상황을 프레임워크 상에서 동작시켰다. Fig. 15는 의미적 추론의 각 상황별 추론 내역의 일부를 정리한 것이고, Fig. 16은 구문적 추론의 각 상황별 추론 내역의 일부를 보여준다.

No	Light_Sensor (lux)	GPS (m)	Projector (on/off)	Mic (on/off)	Clock (mm-dd)	Browser (on/off)	Inference	Situation
1	2000	5	1	0	11-10	0	Normal	Enter the Room
2	122	2	1	1	10-12	0	Abnormal	Enter the Room/Error:Light
3	148	5	0	0	07-17	1	Abnormal	Enter the Room/Error:Light
4	67	1	1	1	02-04	0	Abnormal	Enter the Room/Error:Light
5	11	2	0	0	08-14	1	Abnormal	Enter the Room/Error:Light

Fig. 15. Case for semantic inference

No	Rule	Inference	Situation
1	comm:User(?user) ^ j0:hasStandard_Light_Bright(?user, ?standVal) ^ comm:hasValue(comm:Light_Sensor, ?curVal) ^ ... ^ swrlb:lessThan(?curVal, ?standVal1) → comm:inSituation(?user, j0:Enter_the_Room)	Normal	Enter the Room
2	comm:User(?user) ^ ... ^ j1:hasStandard_Distance(?user, ?standDist) ^ comm:hasValue(comm:GPS, ?curGps) ^ swrlb:lessThan(?curGps, ?standDist) ^ ... ^ swrlb:lessThan(?curVal, ?standVal1) → comm:inSituation(?user, j0:Enter_the_Room)	Abnormal	Enter the Room

Fig. 16. Case for syntactic inference

구문적 추론상황 89개 중 5개는 정상적인 문법이었고 84개는 오류가 있는 문법이었다. 프레임워크는 모든 상황에 대해 추론 상황의 잘못 유무를 판단하였다. 의미적 추론상황 150개 중 10개는 정상적인 경우였고 140개는 오류가 있는 경우였다. 프레임워크는 정상적 상황 10개 중 6개의 경우를 의미적인 오류가 있는 상황이라고 판단하였다. 그리고 의미적인 오류가 있는 140개의 상황 중 93개는 오류로 판단하였으나, 47개는 오류를 판단하지 못하였다.

즉, 구문적 추론상황의 경우는 모든 추론 상황의 잘못 유무를 판단하여 100%의 성공률을 보였고, 의미적 추론상황의 경우는 150개의 추론 상황 중 97개의 상황만 잘못 유무를 판단하여 64.67%의 성공률을 보였다. 전체 추론 오류 판단률은 239개의 경우 중 186개의 경우를 판단하여 77.82% 이다.

의미적 추론상황의 경우 두 개 이상의 상황이 동시에 발생하거나 다수의 규칙들이 충돌하는 경우에 대한 대처를 수행하지 못하여 추론의 오류를 찾아내지 못하는 경우가 발생하였다. Fig. 17은 구문적, 의미적 추론 결과 정리 내역이다.

Type	Classification	Cases	Success	Failed	Success Ratio
Syntactic	Normal	5	5	0	89/89 (100%)
	Abnormal	84	84	0	
Semantic	Normal	10	4	6	97/150 (64.67%)
	Abnormal	140	93	47	
Total		239	186	53	186/239 (77.82%)

Fig. 17. Result of Inference

6. 결론 및 향후 연구

상황인식 서비스 개발 방법은 온톨로지 모델과 SWRL 규칙을 설계하여 추론 엔진을 통해 수집된 상황과 비교 매칭을 통해 상황의 옳고 그름만이 추론 가능하였다. 이러한 문제점을 개선하고자 소프트웨어 사이버네틱스 기반 피드백 개념을 적용하여 기존 MAPE 피드백 프로세스를 변경하여 수집, 분석, 결정, 적응 단계로 구성된 MADA 피드백 프로세스 모델을 제시하였다. 또한, 피드백 프로세스 모델의 각 단계에 대한 수행을 지원하는 프레임워크를 개발하여 프로세스 모델을 설계 및 검증하였다.

향후 연구로는 의미적 추론의 정확성을 높이기 위해 규칙의 충돌 처리 등 추론 룰에 대한 관리 기법과 자가 학습형 상황인식 서비스 개발 방법에 대해 연구할 것이다

참고 문헌

- [1] K. Cai et al., "An Overview of Software Cybernetics", in *Proceedings of the International Workshop on Software Technology and Engineering Practice*, pp.77-86, 2003.
- [2] L. Liu, Z. Jin and R. Lu, "Towards Controllable Requirements Engineering Processes based on Cybernetics", in *Proceedings of the International Conference on Computer Software and Applications*, Vol.2, pp.229-232, 2007.
- [3] J. Chen, Q. Zhang and S. D. Bruda, "Cybernetics in Software System Verification", in *Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics*, Vol.2, pp.247-277, 2009.
- [4] X. Peng et al., "Self-tuning of software systems through dynamic quality tradeoff and value-based feedback control loop", *International Journal on Systems and Software*, Vol.85, No.12, pp.2707-2719, 2012.
- [5] C. Tudor et al., "A self-adapting algorithm for context aware systems", in *Proceedings of the International Conference on Roedunet*, pp.374-379, 2010.
- [6] C. Chang et al., "Situ: A Situation-Theoretic Approach to Context-Aware Service Evolution", *International Journal on Transactions on Services Computing*, Vol.2, No.3, pp.261-275, 2009.
- [7] OWL (Web Ontology Language), W3C Recommendation, <http://www.w3.org/OWL>.
- [8] SWRL (A Semantic Web Rule language Combining OWL and RuleML), W3C Member Submission, <http://www.w3.org/Submission/SWRL>.
- [9] JESS (Java Expert System Shell), the Rule Engine for the Java Platform, <http://www.jessrules.com>.
- [10] O. Kweon, J. Park and K. Yeom, "Ontology Model Management Framework for Supporting Mobile Situation-aware Service Development", *The Journal of Korean Institute of Next Generation Computing*, Vol.7, No.6, pp.13-25, 2011.
- [11] S. Hassanpour, M. Oconnor and A. Das, "Exploration of SWRL Rule Bases through Visualization, Paraphrasing, and Categorization of Rules", in *Proceedings of International Conference on Rule Interchange and Applications*, pp. 246-261, 2009.
- [12] Protégé, <http://protege.stanford.edu>.



신 수 혜

e-mail : ssh5304@pusan.ac.kr

2011년 부산대학교 정보컴퓨터공학부(학사)

2013년 부산대학교 컴퓨터공학과(석사)

2013년~현 재 삼성전자 연구원

관심분야: Software Architecture, Software Reuse, Mobile Situation-Aware System, Software Cybernetics



박 준 석

e-mail : pjs50@pusan.ac.kr

1999년 부경대학교 컴퓨터공학과(학사)

2002년 부산대학교 컴퓨터공학과(석사)

2012년 부산대학교 컴퓨터공학과(박사)

2012년~현 재 부산대학교 물류혁신네트
워킹연구소 박사후연수연구원

관심분야: Software Reuse, Product Line, Service Oriented Architecture, Software Architecture, Ubiquitous Computing



염 근 혁

e-mail : yeom@pusan.ac.kr

1985년 서울대학교 계산통계학과(학사)

1992년 UNiv. of Florida 컴퓨터공학과(석사)

1995년 UNiv. of Florida 컴퓨터공학과(박사)

1996년~현 재 부산대학교 정보컴퓨터
공학부 교수

관심분야: Software Reuse, Product Line Engineering, Software Architecture, Situation-Awareness Middleware based on USN (Ubiquitous Sensor Network), Self-Adaptive Software Development