

| Oral Session IX : Big Data, Smart Energy ICT, Smart Information

좌장 : 신춘성 (전남대)

업종별 부채 예측 모델 개발 : 코로나 19 상황에서

김양석, 노미진, 김차미, 손승연, 조유진 (계명대학교) 114

녹조 발생 예측 AI모델 개발 연구

송수영, 송유선, 이유진, 홍경석, 김남호, 최광미 (호남대학교), 정희자(휴넷가이아) 116

Non-IID 환경에서 연합 학습 기반 전기 수요 예측

염성웅, Kolekar Shivani Sanjay, 조현준, 김경백 (전남대학교) 118

유사 서비스 함수를 위한 코드 모듈들의 구조 내 저전력 연구

윤예동, 문소영, 김영철 (홍익대학교) 120

복잡한 코드의 간결화를 통한 성능 및 저전력 개선

조재형, 문소영, 김영철 (홍익대학교) 123

CCTV 영상처리를 통한 화재감지기 오탐 개선에 관한 연구

황은호, 김남호 (호남대학교) 126

Knowledge Graph 확장을 위한 딥러닝 기반 관계 추출

최준호, 김형주 (조선대학교) 209

농경지 침수 분석을 위한 SWMM 모형의 적용성 검토

김규민, 원다윗, 양승원 (우석대학교) 211

공간정보 기반 농경지 침수피해의 선제적 대응을 위한 기초자료 구축

박석우, 양승원, 나인호 (군산대학교) 213

SWMM 해석 기반 공간분석 농경지 침수의 선제적 대응 연구

손성민, 김형진 (전북대학교) 215

색 추출 기법을 접목한 아트 플랫폼의 기대효과

유세빈, 황시준, 박남홍 (조선대학교) 217

알츠하이머병에 라지 스케일 네트워크의 연결 패턴 분석

라마라매쉬쿠마, 권구락 (조선대학교) 219

클라우드 컴퓨팅에서의 장애 허용 기법 분석

조만규, 이재환, 김찬수, 박상오 (중앙대학교) 222

기능점수 기반 정교한 비용 예측 추출을 위한 요구사항 스펙 구조화

문소영, 김영철 (홍익대학교) 224

신재생에너지 스마트팜 환경 기반 에너지 사용량 예측

임종현, 장경민, 오한별, 이명배, 신창선, 박장우, 조용윤 (순천대학교) 226

Firebase 클라우드 메시징을 활용한 스마트 헬스케어 플랫폼

남재경, 최민 (충북대학교), 김성준(중원대학교) 228

수경재배 양액관리를 위한 스마트 단말 모니터링 및 제어 시스템 구현

오한별, 이명배, 박장우, 조용윤, 신창선 (순천대학교) 230

데이터 분석 기반의 파프리카 온실 환경 예측에 대한 연구

장경민, 이명배, 조용윤, 신창선, 박장우 (순천대학교) 232

딥러닝 모델을 이용한 발전량 예측 방법

김지인, 이건우, 권구락 (조선대학교) 234

AMI 시스템에서 수집 시간 단축을 위한 기법 연구

나채훈, 김정인, 윤범식, 강향숙, 김판구 (조선대학교) 236

설계 메타모델링 기반 구현 언어별 가이드 코드 생성

정세준, 문소영, 김영철
 홍익대학교 소프트웨어공학연구실
 thom13579@g.hongik.ac.kr, whit2@hongik.ac.kr, bob@hongik.ac.kr

The Guide Code Generation for each Building Languages based on Design Metamodeling

Se jun jung, S. Y. Moon, R. Young Chul Kim
 SE Lab, Dept. of Software & Communications Engineering, Hongik University

요 약

소프트웨어의 비가시성으로 설계를 기반으로 구현을 진행하는 것에 어려움이 있다. 이를 해결하기 위해 Usecase 명세서와 설계(Use Case Scenarios Object Model등)를 통합한 ‘설계 메타 모델’을 제안하였다[1]. 또한 이를 기반으로 가이드 코드를 생성하여 설계에 따른 구현을 쉽게 만들 수 있다. 또한 이 가이드 코드를 구현하고자 하는 언어(Java, Cpp, Python, Lua, Sol)에 맞도록 설계 메타모델로부터 자동으로 생성하여 보다 더 쉬운 고품질의 코드를 개발할 수 있도록 한다.

1. 서 론

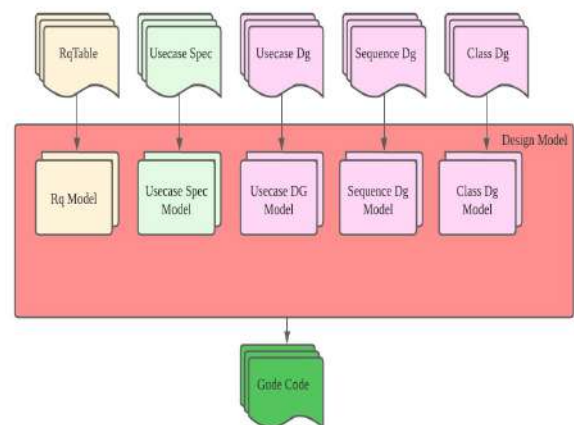
소프트웨어는 요구사항이 반영된 설계를 바탕으로 최종적으로 소스코드로 구현된다. 하지만 완성된 소스코드로부터 설계와 요구사항을 추적하는 것은 어려운 일이다. 이를 해결하고자 설계로부터 코드를 생성하려는 연구가 진행되고 있다. 특히 소프트웨어에 객체지향의 관점이 적용되면서 UML(Unified Modeling Language)은 소프트웨어의 대표적인 설계로 사용되고 있다. UML은 표준 모델로 다양한 다이어그램 모델(Use Case, Class, Sequence 등)이 존재한다. 이를 사용한 많은 도구나 모델이 각각의 모델간의 연동을 지원한다. 따라서 이러한 설계를 바탕으로 코드를 생성하면 설계를 기반으로 코드를 쉽게 작성할 수 있도록 도울 수 있다. 본 연구는 이러한 UML 설계와 요구사항의 정보를 메타모델링한 설계 메타 모델을 바탕으로 자동으로 가이드 코드를 생성한다[1]. 또한 구현할 언어에 맞도록 가이드코드를 자동으로 생성하여 이를 통해 개발자들이 설계를 기반으로 개발을 편하게 할 수 있도록 돕는다.

2. 관련 연구

초기의 요구사항은 정형화 되지 않은 자연어의 집합이다. 따라서 이를 설계와 구현된 소프트웨어에서 추적하는 것이 어렵다. 이를 위해 비정형된 요구사항인 자연어를 문장 의미 모델을 생성한다[2]. 생성된 의미 모델은 요구사항으로부터 생성한 테스트 케이스를 검증하거나 생성할 때 사용할 수 있다. 이처럼 요구사항에 기반한 모델을 사용하여 요구사항을 추적한다. 또한 요구사항에서 설계를 생성하기 위해 Fillmore의 메커니즘을 사용하여 유스케이스를 추출한 연구가 있다[3]. 또한 소프트웨어의 개발 비용을 줄이기 위해 기존의 펌웨어의 절차적 설계기법에 usecase 접근 기반으로 테스트 케이스를 추출하는 연구가 있다[4]. 이

처럼 Usecase와 UML은 소프트웨어를 설계할 때 사용되는 가장 대표적인 설계 모델이다.

메타모델은 모델을 모델링 한 모델이다. 이는 유사한 두 개의 모델을 하나로 통합 및 변환하거나 모델기반의 개발에 사용된다. HS Solar energy 회사에서 이 기술을 사용하여 서로 다른 프로토콜과 데이터를 통합하여 관리하여 예측에 사용한다[5]. 또한 MDA기법을 통해 시스템과 컴포넌트를 활용해 통합 서비스를 효율적으로 개발 방법을 연구한 결과가 있다[6]. 요구사항과 설계를 반영한 가이드 코드를 발생시키기 위해 아래 그림 1과 같이 설계 메타모델을 정의하였다[1].



(그림 1) 설계 메타 모델

3. 설계 메타모델링 기반 가이드 코드 생성

기존에 제안한 방법은 java기반의 가이드 코드 만을 생성한다. 이를 개선하여 설계 모델을 기반으로 가이드 코드를 구현할 타겟 언어(Java, C++, Python, Lua, sol)로 생성한

다. 가이드 코드는 객체지향의 구조로 각 언어에서 지원하는 클래스 구조와 메서드 구조를 작성한다. 또한 설계를 기반으로 생성된 클래스와 메서드에 기반이 된 설계 정보를 주석으로 제시한다. 또한 메서드의 내부 구조를 use-case spec과 시퀀스 다이어그램의 메시지를 기반으로 제안한다. 아래는 생성된 java 및 Python 가이드코드의 예시이다.

```
package default;

/** UC-001-01: "문장 입력은 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-02: "시스템은 사용자 입력을 다항형 방식으로 제공한다."
 * UC-001-03: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-04: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-05: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-06: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-07: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-08: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-09: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 * UC-001-10: "시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다."
 */
public class MainSys
{
    /** UC-01
     * 시스템은 사용자로부터 언어를 제공한다.
     */
    public void main()
    {
        // 사용자로부터 언어를 제공하고, UC-01-P1-1
        MainSys.inputword();

        // 시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다. UC-01-P1-2
        MainSys.getword(word);
    }
}
```

(그림 2) Java 가이드 코드의 일부

Java언어에 맞는 document 구조로 요구사항 및 Usecase의 event flow(메시지)에 대한 정보와 ID가 주석으로 제공되며 호출할 메서드의 정보 또한 대략적으로 제공된다. Python의 경우도 java와 유사하게 기본적으로 클래스의 구조를 정의하여 가이드 코드가 생성할 수 있다. 클래스 정의에 따라 self 파라미터를 수정해준다. Python 역시 설계 정보가 document 구조로 주석이 생성된다.

```
class MainSys:
    """ UC-001
    시스템은 사용자로부터 언어를 제공한다.
    """
    def __init__(self):
        """ UC-01
        시스템은 사용자로부터 언어를 제공한다.
        """
        # 사용자로부터 언어를 제공하고, UC-01-P1-1
        self.inputword()

        # 시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다. UC-01-P1-2
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-3
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-4
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-5
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-6
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-7
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-8
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-9
        self.getword(word)

        # 사용자로부터 언어를 제공하고, UC-01-P1-10
        self.getword(word)

    def inputword(self):
        """ UC-01-P1-1
        사용자로부터 언어를 제공하고, UC-01-P1-1
        """
        return word

    def getword(self, word):
        """ UC-01-P1-2
        시스템은 '종료' 명령을 입력하면 종료, 시작, 정지, 시작, end 구조를 제공한다. UC-01-P1-2
        """
```

(그림 3) Python 가이드 코드의 일부

4. 결 론

본 논문은 Usecase 명세와 UML 설계를 메타모델링한 설계 모델로부터 다양한 언어의 가이드 코드를 생성하는 것을 제안한다. 이를 통해 소프트웨어를 구현하기 위한 다양한 언어를 제공하고 설계를 기반으로 소프트웨어의 구현을 할 수 있도록 돕는다. 또한 이렇게 생성된 가이드 코드는 동일한 설계를 각각 다른 언어로 이해할 수 있으며 구현할 언어의 선택에도 도움을 줄 수 있다.

ACKNOWLEDGMENT

이 논문은 교육부 및 한국연구재단의 4단계 두뇌한국21 사업의 지원(F21YY8102068)과 2022년도 정부(교육부)의 지원으로 한국연구재단의 지원(No. 2021R1I1A3A305040711, No. 2021R1I1A1A1A01044060)을 받아 수행된 연구임.

참 고 문 헌

- [1] 정세준. "Use case 명세서 기반 자동 가이드 코드 생성", 한국소프트웨어공학학술대회논문집, 24권 1호, 2022. pp.125-127
- [2] Jang, Woo Sung, Se Jun Jung, and R. Kim. "Design of Sentence Semantic Model for Cause-Effect Graph Automatic Generation from Natural Language Oriented Informal Requirement Specifications." Annual Conference on Human and Language Technology. Human and Language Technology, 2020.
- [3] Park, Bokyung, Hyoseok Yang, and Robert Youngchul Kim. "A Method to Identify Goal Use-Case(s) with Refined Fillmore's Case Grammar." Proceedings of the Korea Information Processing Society Conference. Korea Information Processing Society, 2013.
- [4] 장우성, et al. "Use-Case Approach 기반의 Test Case 추출을 위한 아두이노 펌웨어 설계 연구." 한국정보처리학회 학술대회논문집 20.2 (2013): 1015-1018.
- [5] Jang, Woo Sung, et al. "Best Practices on Metamodel-Based Photovoltaic Monitoring System with Prediction Method for Photovoltaic Power Generation." Applied Sciences 10.14 (2020): 4762.
- [6] 박영식, 김영철, 강윤희, 박보경, 장우성.(2019).MDA(Model Driven Architecture) 기반의 통합 아키텍처 연구 사례.JOURNAL OF PLATFORM TECHNOLOGY,7(4),3-8.