
목표 시나리오 기반의 동적 재구성을 위한 코드 자동 생성 기법

백수진[†], 심성호^{††}, 송영재^{†††}

An Automated Code Generation for Dynamic reconfiguration based on Goal-Scenario

Su-Jin Baek[†], Sung-Ho Sim^{††}, Young-Jae Song^{†††}

요약 오늘날 복잡해져가는 컴퓨팅 환경에서 자가치유를 위해 시스템에 발생한 문제를 스스로 인식하는 능력을 부여하는 연구가 중요한 이슈가 되고 있다. 그러나, 기존 방법론들은 새로운 요구사항에 대한 목표 시스템을 자가 치유하기 위해 개발자들이 제약조건을 모델링하고 분석해야 하는 노력이 크다. 따라서, 본 논문에서는 시스템에서 발생한 문제를 인식하고 감지된 문제를 해결하기 위해 목표 시나리오 기반의 분석하고, 전·후 전략을 적용하여 동적으로 재구성되도록 자가 치유 방법을 제시하도록 한다. 이러한 제안 방법론을 통해 새로운 요구사항에 대한 자가 치유 개발자의 분석의 부하를 줄이고, 변경 정보를 파악하여 동적으로 재구성 되도록 지원하도록 한다.

주제어 : 목표-시나리오, 동적 재구성, 자기적응형 소프트웨어, 적응 전략, 자가 치유

Abstract Today, the computing environments is very complex, so researches that endow a system with the self-healing's ability that recognizes problem arising in a target system are being an important issues. However, the existing methodology, the goal for the new requirements for self-healing system developers to model and analyze the constraints that must be greater efforts. Therefore, in this paper are aware of problems detected by the system to solve the problem is the analysis of goal-based scenarios. In addition, there is a pre and post applying a strategy to be dynamically reconfigured to show you how to self-healing. These proposed new requirements for methodology, self-healing reduces the load on the developer's analysis.

Key Words : Goal-Scenario, Dynamic reconfiguration, Self-Adaptive Software, Adaptive Strategy, Self-Healing

1. 서론

최근의 소프트웨어는 실행 환경이 점차 복잡해지고, 실행 중에 예측하기 어려운 상황들이 빈번하게 발생하고 있다. 이것은 기존의 컴퓨팅 패러다임에서 개발자들이 잘 정의되고 고정된 실행 환경을 가정하여 소프트웨어를 설계하였기 때문이다. 이로 인해 실행 중에 실패가 쉽게 발생되었으며, 외부 환경의 예외적인 상황에서의 기능을 항상 유지하는데 어려움이 발생한 것이다. 이에 대한 해결 방법으로 소프트웨어가 실행 중에 외부의 실행 환경

과 자신의 상태를 파악해서 '적응'을 할 수 있도록 하는 적응형 소프트웨어 기술이 필요하다. 자기 적응형 소프트웨어란 "동작 환경의 변화에 스스로 대처하여 자신의 행위를 수정할 수 있는 소프트웨어"를 말한다. 즉, 주변 환경이나 사용자의 요구사항이 변화함에 따라 자신의 행동을 동적으로 변화시킬 수 있는 소프트웨어이다. 이를 위해서는 실행 중인 시스템의 상태를 모니터링하고, 시스템에 발생한 문제를 분석하고, 적응의 필요성을 판단하는 평가, 발생한 문제를 해결하기 위한 적응 전략을 찾아내고 이에 따라 실행 중인 시스템의 구조와 행위를 동

[†] 경희대학교 컴퓨터공학과 박사과정(교신저자)

^{††} 경희대학교 컴퓨터공학과 박사과정

^{†††} 경희대학교 컴퓨터공학과 교수

논문접수 : 2012년 1월 18일, 1차 수정을 거쳐, 심사완료 : 2012년 2월 17일

적으로 재구성해야 한다[1]. 그러나, 자가 적응 시스템을 생성하기 위해서는 많은 경험과 지식이 필요하므로 시스템 구축의 어려움이 문제가 되고 있다. 오늘날 시스템들은 목표 시스템의 신뢰성, 강건성, 가용성 확보를 위해 추가적인 자가 적응 모듈의 설계와 비용이 고려되고 있다. 이러한 측면에서 중대한 시스템 내부에 존재하는 자가 적응 모듈로 인해서 발생하는 성능 저하 문제가 해결되어야 한다. 따라서 시스템의 문제를 스스로 인식, 진단하여 문제에 대한 전략을 적용하여 시스템이 자율적으로 복구하도록 자가 치유 연구가 필요하다[2]. 자가 치유 시스템에 대한 기존 연구들은 컴포넌트 기반[3], 모델기반[4], 로그 기반 방법론[5]으로 구분될 수 있다. 이 연구들의 공통적인 문제점은 자가치유 개발자가 내부를 알 수 없는 목표 시스템을 직접 분석해야 한다는 점이다. 자가 치유 개발자가 목표 시스템을 이해하는 정도에 따라 목표 시스템에 부여되는 자가치유 능력의 정도가 다를 수 있다. 따라서 목표 시스템의 설계 단계에서 생성된 산출물들을 이용하여 자가치유 능력을 부여하는 것이 가능하다면 자가치유 개발자의 분석의 부하를 줄이는 것이 가능하다.

본 논문에서는 시스템에서 발생한 문제를 인식하고 감지된 문제를 해결하기 위해 목표 시나리오 기반의 분석하고, 전·후 전략을 적용하여 동적으로 재구성되도록 자가 치유 방법을 제시하도록 한다. 외부 상황에 대한 문제 분석 및 평가는 목표 시나리오 기반으로 분석하며, 휴리스틱 알고리즘을 통해 유사 모델을 추출하며, 전·후 전략을 통한 자가 치유 기법으로 변경 정보와 제약 규칙에 따라 동적 재구성 되도록 자동으로 코드를 생성 하도록 한다.

2. 관련연구

2.1 자가치유에 대한 기존 방법론들

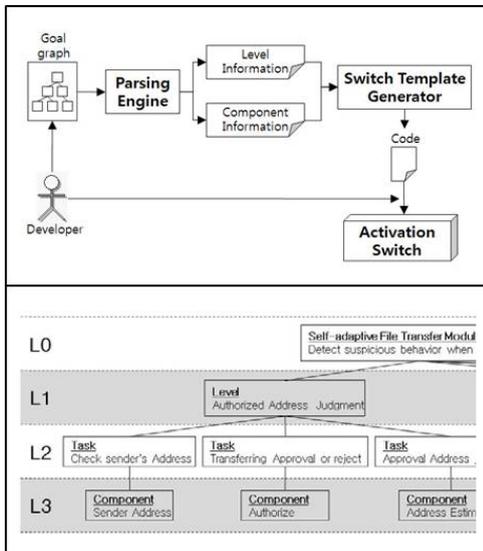
자가 치유 시스템에 대한 연구들 중 외부 환경을 위한 모델 기반의 자가치유 방법론[4]은 시스템을 모니터링, 해석, 분석, 재구성하기 위해 사용될 수 있는 계층화된 외적환경 관점의 구조적 분석 모델을 표현하였다. 이 방법론은 ADL(Architecture Description Language)을 기반으로 목표 시스템의 외부 관점에서 최적화된 자원 환경이나 프로세스를 모델링한 것이다. 그러나 자가 치유 개발자가 목표 시스템을 모델링하기 위해서 내부 정보 없

이 외부 자원 환경에 대한 분석의 부하가 매우 높다. 컴포넌트 기반의 자가 치유 방법론[3]은 소프트웨어 아키텍처의 강건성을 위해 컴포넌트 내부에 서비스 계층과 치유 계층을 각각 개별적으로 설계하여 컴포넌트의 강건성을 보장한다. 이 연구들은 서비스 계층 내부에 있는 태스크들의 행동을 구체화한 상태 모델을 통하여 컴포넌트 기반의 동적 재구성을 용이하게 하고, 재구성 후 상태 모델을 기반으로 컴포넌트가 정상 동작이 가능한지 아닌지의 여부를 판단할 수 있는 기초를 제공한다. 그러나 이들은 내부 상태에 대한 이벤트에 관하여 초점을 두고 있기 때문에 컴포넌트 구동환경인 자원 부족에 관한 문제 결정에 어려움이 있으며, 이로 인해 발생하는 고정을 치유할 수 없다. 로그 기반의 자가 치유 방법[5]은 다양한 타입의 시스템들에서 생성된 이벤트 로그들을 자동 수집하여 자가 치유 가능하게 하는 시스템을 제안하였다. 이 방법은 목표 시스템의 내부 상태 관점에 관한 에러 로그 발생 시에 관련 문제를 찾아 사후 관리를 위한 개선점을 제공하는 것이 용이하다. 또한, 로그의 내용이 실시간으로 기록되기 때문에 데이터베이스로 구조화하기 쉽고 상황식 분석이 용이하다. 그러나 로그가 발생하지 않는 경우, 시스템의 내부 관점을 모니터링하고 분석하고 치유하기 위한 제약조건을 자가 치유 개발자가 직접 모델링 해야 하고, 로그의 내용만으로는 하나의 컴포넌트로 인해 발생하는 문제가 다른 컴포넌트에게도 영향을 미치는지에 관한 연관성 분석이 쉽지 않다. 위에 언급된 기존 연구들의 공통적인 문제는 목표 시스템의 내부부 상태에 대한 문제 결정 수준을 개발자가 직접 결정하기 때문에 이에 대한 목표 시스템에 대한 분석의 부하가 높고, 목표 시스템의 외부 환경을 파악하기 위한 제약조건 모델링의 부하가 높다. 또한, 새로운 요구사항에 대한 목표 시스템의 내부 상태를 파악하기 위한 상태 모델링의 부하가 높고, 특정 컴포넌트의 문제가 발생했을 때 문제가 발생한 컴포넌트와 연관된 컴포넌트들과의 연관성 분석이 어렵다. 따라서, 본 논문에서는 기존 연구들의 문제점을 해결하기 위해서 새로운 요구사항에 대한 목표 시스템의 외적 환경과 내적 환경을 만족하는 자가치유 기법을 기반으로 목표 시나리오 기반으로 문제를 분석하고, 동적 재구성을 위한 코드 생성 기법을 제안한다.

2.1. 목표 그래프를 이용한 코드 생성 기법

목표 그래프를 이용한 코드 생성 기법[6]에서는 개발

자가 자가 적용 모듈을 직접 작성하고 관리하는 노력을 덜기위하여 목표 그래프를 이용한 외부 요인 평가 기법 [7]의 목표 그래프를 응용하였다. 이것은 자가 적용 모듈의 성능을 개선하는 방법을 추상화하였다. [그림 1]은 목표 그래프를 이용한 코드 생성 구조와 이를 통한 목표 그래프의 예를 나타낸 것으로 목표 그래프를 개발자가 그린 후, 자가 적용 모듈을 생성하는 과정을 나타내고 있다. 이러한 과정을 통해 생성된 코드는 개발자의 수정을 거쳐 최종적으로 자가 적용 모듈을 완성한다.



[그림 1] 목표 그래프를 이용한 코드 생성 구조 와 이를 적용한 목표 그래프의 예

3. 목표 시나리오 기반의 분석

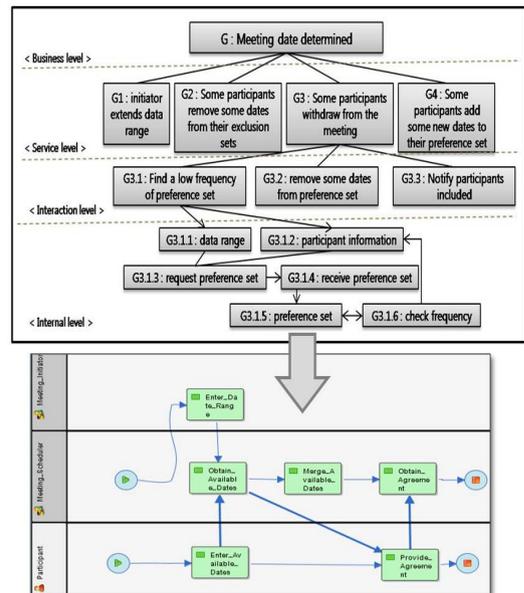
3.1. 요구사항 분석 및 목표 시나리오 기반 프로세스 모델 생성

프로세스 관리 시스템을 통해 계획단계에서 새로운 시나리오를 분석하여 변경해야 할 목표 요구사항을 파악한다. 요구사항 목표는 적어도 몇 개의 비즈니스 프로세스가 결합하여 수행해야 달성할 수 있다. 즉, 요구사항 목표를 달성할 수 있도록 비즈니스 시나리오를 정의하여 추상적인 목표 요구사항을 추출하도록 한다.문제에 대한 정확한 분석을 하기 위해서는 목표를 구체적으로 결정하고 기존 모니터링을 통해 값들에 대한 상태를 파악한 후 그에 맞는 시나리오를 작성해야 한다. 목표 비즈니스 시

나리오 정의가 이루어지고, 목표 시나리오를 생성하기 위해서는 시나리오에 필요한 수준별 세부 구성 요소들을 파악해야 한다. 그러기 위해서는 자연어 형태의 요구사항을 목표와 시나리오 기법을 통해 구조화하도록 한다.

요구사항의 추상화는 도메인에 대한 궁극적인 목표를 정의하고 기술하는 비즈니스 수준, 도메인이 사용자에게 제공해야 하는 서비스의 범위를 정의하는 서비스 수준, 서비스 범위에 대해 서비스가 실현되기 위해 사용자 또는 외부 시스템과 대상 시스템과의 상호작용을 표현한 상호작용 수준, 상호작용을 만족시키기 위해 내부적으로 시스템이 가져야 할 기능에 중점을 둔 내부 수준으로 나누어 표현한다.

[그림 2]는 기존의 미팅 스케줄링의 모델 중 “Meeting arranged” 부분에서의 미팅 날짜 결정이라는 새로운 요구사항이 발생했을 때의 분석 및 문제점 해결방안에 대한 목표 모델 일부를 보이고 있다. 새로운 목표 요구사항 및 그에 따른 시나리오 형식에 맞춰 작성된 목표 시나리오 모델의 일부로 G3 날짜 충돌 해결 부분에 초점을 두고 분석하였으며, 이를 통해 목표 시나리오를 기반으로 프로세스 모델을 생성한 것이다.

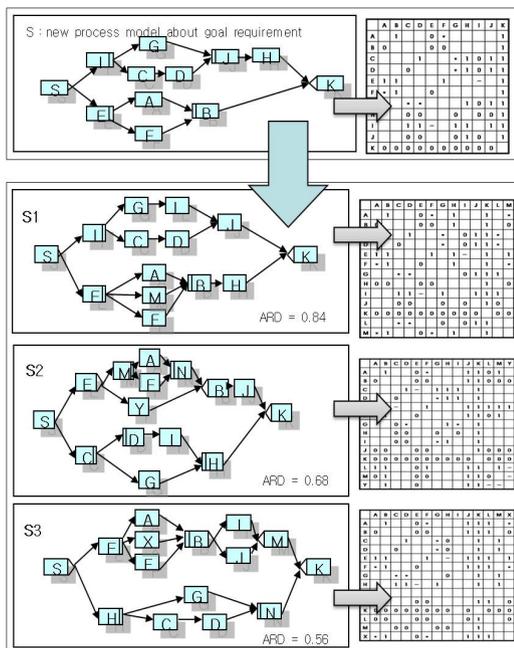


[그림 2] 목표 시나리오 기반한 MSS의 목표 모델 일부

3.2 휴리스틱 알고리즘을 통한 유사 모델 추출 목표 시나리오에 대한 프로세스 모델을 기반으로 기

존 프로세스의 전체 액티비티 관계를 분석하지 않고 액티비티의 출현 유무를 벡터공간 모델 유사도 방법으로 계산을 하여 필요한 정보만을 분석하여 변경하도록 한다. 이 논문에서의 액티비티의 유사도(ASD)는 목표 시나리오를 이용한 비즈니스 프로세스 외부상황 평가 모델[8]을 참조하여 분석하였다. ASD값은 적합한 프로세스를 찾는 데 이용되며 값이 클수록 우선순위가 높다. 0 일 경우는 기존 프로세스에서 목표 요구사항에 부합되는 프로세스가 없는 경우이므로 새로 프로세스를 생성하도록 한다. ASD가 높은 순으로 유사 프로세스 모델을 검색 한 후 액티비티 관계 분석을 하였다.

[그림 3]은 새로운 요구사항에 대한 프로세스 모델을 통해 유사도(ASD)를 측정하여 추출된 유사 프로세스 모델들 중의 일부이다.



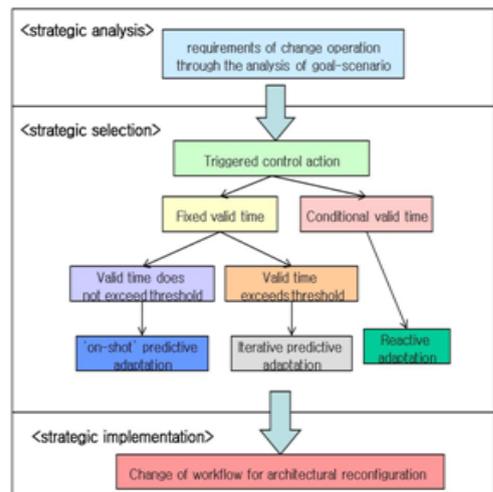
[그림 3] 유사 프로세스 모델

4. 동적 재구성을 위한 코드 자동 생성

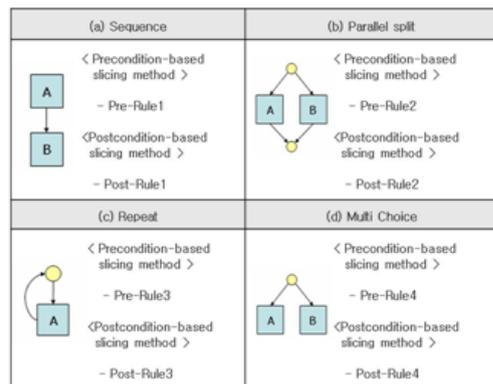
위의 목표 시나리오 기반 분석 정보를 토대로 동적 재구성을 위한 코드를 생성하도록 한다. 이를 위해 시간을 기준으로 전·후 전략을 프로세스의 변경 정보에 적용하여 필요한 코드를 생성하도록 한다.

4.1 자가 치유를 위한 전·후 전략 정의

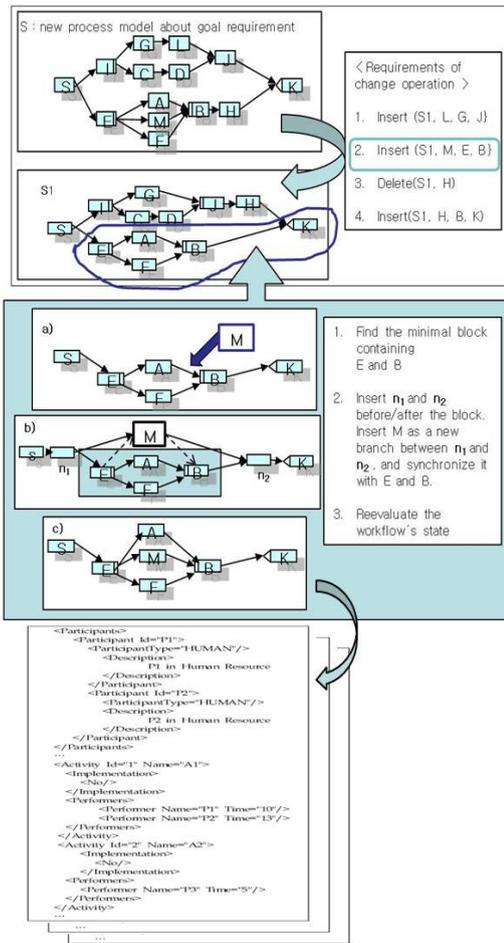
목표 시나리오 기반으로 분석한 정보를 기반으로 문제결정 수준에 맞추어 전략을 생성하고 이에 따라 문제 해결을 위한 코드를 생성하여 재구성되어야 한다. 자가 치유 방법론의 기본 전략들은 대체 전략이나 재시작 같은 기본적인 재구성 전략을 가지고 있다. 본 논문에서는 시간에 따른 전·후 전략을 생성하여 이를 통해 효율적인 재구성을 지원하도록 한다. [그림 4]는 목표 시나리오 분석을 통한 정보를 기반으로 시간에 따른 적용 전략 방법을 보이고 있으며, [그림 5]는 [그림 4]에서 선택된 시간 적용 전략에 전·후 슬라이스 조건을 적용한 각각 규칙 및 방법을 설명하고 있다.



[그림 4] 적용 전략에 따른 선택



[그림 5] 동적 재구성을 위한 전·후 조건의 제어 패턴



[그림 6] 동적 재구성을 위한 변경 정보 및 코드 생성

전 조건 기반 슬라이스(Precondition-based slice)는 전향적, 정적 슬라이스으로써, 프로세스가 Pre(선행 조건)을 만족하는 프로세스 상태에서 수행을 시작한 경우, 수행될 수 있고, 수행된 후 프로세스의 상태를 변경할 수 있는 프로세스와 제어 조건들의 집합으로 이루어진다. 후 조건 기반 슬라이스(Postcondition-based slice)는 후향, 정적 슬라이스으로써, 주어진 후행 조건 Post(후행 조건)에 대한 최악 선행 조건에 영향을 미칠 수 있는 프로세스와 제어 조건들의 집합으로 이루어진다. Pre-Rule 1은 프로세스 상에서의 순차적 문장 결합에 적용되며, 주어진 선 조건 Pre에 대한 두 프로세스 모델의 효과가 서로 상쇄될 수 있기 때문에 상태 변화의 누적 효과를 고려해야 한다. Post-Rule 1은 예상 출력에 관한 후 조건에 따라 선 조건 규칙과 같은 방법으로 순차적 문장 결합에 의해 발생하는 상태 변화의 누적 효과를 고려한 경우이

다. Pre-Rule2는 프로세스 상에서의 대입문과 같은 역할에 적용되며, 프로그램 상태가 바뀌지 않는다면 대입문을 skip하여 처리할 수 있다. Post-Rule2는 후 조건을 기준으로 하여 선 조건 규칙과 같이 프로세스 상태가 영향을 받지 않으면 skip하여 처리할 수 있다. Pre-Rule3는 프로세스 모델 상에서의 반복문과 비슷한 역할에 적용되며, 반복문의 조건식이 선 조건에 의해 반복문 내의 프로세스에 영향을 주지 않고 변경되지 않으며, 선 조건에 의해 반복문의 종류가 보장된다면 반복문 전체가 skip될 수 있다. Post-Rule3는 주어진 후 조건이 루프 불변자(loop invariant)이고, 반복문의 종류가 보장된다면 반복문 전체가 skip될 수 있다. Pre-Rule4는 프로세스 모델 상에서의 조건문과 비슷한 역할에 적용되며, 조건문의 분기 조건이 만족하지 않는다면 skip될 수 있고, 조건문의 조건이 만족하는 쪽의 분기문으로 치환될 수 있다. Post-Rule4는 후 조건에 따라 어떤 분기가 선 조건에 영향을 미치지 않으면, 그 분기는 skip으로 치환될 수 있다. 위와 같은 전략을 통해서 동적 재구성에 필요한 코드에 대한 변경 정보를 얻는다.

4.2 변경 정보에 따른 코드 생성

전략을 통해서 생성된 변경 정보를 이용하여 동적 재구성을 위한 코드를 생성하고 변환하도록 한다. 이것은 새로운 요구사항의 목표 시스템의 외부 환경 문제와 내부 상태 문제를 모니터링 할 수 있는 기반이 된다. [그림 6]은 예기치 못한 외부 상황의 문제 해결을 위해 목표 시나리오 기반으로 프로세스를 분석한 후 유사도를 통해 프로세스 모델을 중점으로 유사 부분을 추출한 결과이다. 전·후 규칙을 통해 문제해결을 위한 필요한 작업들을 함께 나타내었으며, 생성된 코드의 일부도 보이고 있다. 이러한 결과는 기존의 실행중인 모니터링에게 전달되며 지속적인 서비스를 하도록 정보를 제공한다. 또한, 기존 연구에서는 자가 치유를 구현하기 위해서 개발자의 역량이 크게 작용했으나 제안 사항은 목표 그래프를 통해 구조를 추상화 하였으며, 설계하는 과정의 일부를 시각화하여 세부적인 상호 연관 관계를 파악할 수 있도록 하였다.

5. 결론

본 논문에서는 시스템에서 발생한 문제를 인식하고 감지된 문제를 해결하기 위해 목표 시나리오 기반의 분

석하고, 전·후 전략을 적용하여 동적으로 재구성되도록 자가 치유 방법을 제시하도록 한다. 이러한 제안 방법론을 통해 새로운 요구사항에 대한 자가 치유 개발자의 분석의 부하를 줄이고, 시스템의 외부 환경뿐 아니라 내부 상태 문제에 관한 변경 정보를 파악하여 동적으로 재구성 되도록 지원하도록 한다. 또한, 자가 치유를 위한 확장 설계도 가능하다. 그러나, 제안한 방법은 새로운 요구사항에 대한 시나리오를 기반으로 하고 있기 때문에 실행 중인 기존 시스템과 외부 상황에서의 요구사항에 대해서 문서화되어 있어야 한다. 또한, 유사성 분석을 위해 작성된 목표 다이어그램 작성 시 초기 목표 식별이 어렵고, 추출된 요구 사항을 검증할 수 있는 방안이 미흡하다.

앞으로의 연구 방향은 부분적인 시나리오 기반 분석뿐 아니라 전체적인 여러 시나리오를 기반으로 한 관계 분석과 유사도 평가, 변경에 따른 영향력 추출에 대해서 연구할 계획이다. 또한 변경 요구 사항에 대한 적정 범위 산출 방법에 대해서도 고려할 것이다. 마지막으로, 목표 시나리오 기반 그래프를 이용하는 코드 생성 기법에서 동작의 단계를 명확하기 나눌 수 없는 문제와 변경을 위한 다양한 자가 치유 전략을 고려하여 문제를 해결하도록 향후 연구에서 다룰 예정이다.

참고 문헌

[1] David Garlan, and Bradley Schmerl, "Model-based Adaptation for Self-Healing Systems," ACM SIGSOFT Workshop on Self-Healing Systems (WOSS'02), November 18-19, 2002.

[2] Jeffery O. Kephart, David M. Chess, "The Vision of Autonomic Computing," IEEE Computer Society, Vol. 36, pp.41-50, Jan., 2004.

[3] Michael E. Shin, and Jung Hoon An "Self-Reconfiguration in Self-Healing Systems," Proceedings of the 3th IEEE International Workshop on Engineering of Autonomic & Autonomic Systems, pp.89-98, Mar., 2006.

[4] David S. Wile and Alexander Egyed, "An Externalized Infrastructure for Self-Healing Systems," proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture, pp.285-288, Sep. 2004.

[5] Jeongmin Park, Giljong Yoo, Chulho Jeong, and Eunsoek Lee, "Self-Management System based on Self-healing Mechanism," LNCS 4238, pp. 372-382, Sep. 2006.

[6] Jeongmin Park, Joonhoon Lee, and Eunsoek Lee, "Goal graph based Performance Improvement for Self-Adaptive Modules," ACM SIGKDD International Conference on Ubiquitous Information Management and Communication (ACM SIGKDD ICUIMC2008), pp.68-72, Feb., 2008.

[7] Jasun Kim, Sooyong Park, "Goal-based Evaluation of Contextual Situations for Self-adaptive Software, The Korean Institute of information Scientists and Engineers, Vol. 33, No.3, pp. 316-334, 2006.

[8] 백수진, 고종원, 송영재(2011). 목표 시나리오를 이용한 비즈니스 프로세스 외부상황 평가 모델, 한국콘텐츠학회 논문지 11(8), 43-50.

백수진



- 2002년 : 경희대학교 전자계산학과 (학사)
- 2004년 : 경희대학교 전자계산학과 (석사)
- 2004년 ~ 현재 : 현재 경희대학교 컴퓨터공학 박사과정
- 관심분야 : 소프트웨어 프로세스, 소프트웨어 재사용, 적응형 소프트웨어

· E-Mail : croso@khu.ac.kr

심성호



- 2002년 : 관동대학교 전자계산학과 (학사)
- 2005년 : 경희대학교 전자계산학과 (석사)
- 2005년 ~ 현재 : 현재 경희대학교 컴퓨터공학 박사과정
- 관심분야 : 웹 서비스, 소프트웨어 재사용, CBSE

· E-Mail : shim5800@khu.ac.kr

송 영 재



- 1969년 : 인학대학교 전자공학과(학사)
- 1976년 : 일본 keio 대학교 전산학과(석사)
- 1980년 : 명지대학교 전산학과(공학박사)
- 1976년 ~ 현재 : 현재 경희대학교

컴퓨터공학 교수

- 관심분야 : 소프트웨어 재사용, CASE 도구
- E-Mail : yjsong@khu.ac.kr