

객체 재사용성을 위한 레거시 시스템 인터페이스 기반 객체 추출 기법

이창목*, 최성만*, 유철중*, 장옥배*

*전북대학교 컴퓨터통계정보학과

e-mail: {cmlee, sm3099, cjoyoo, okjang}@chonbuk.ac.kr

Techniques to Extract Object Based on Interface of Legacy System for Object Reusability

Chang-Mog Lee*, Seong-Man Choi*, Cheol-Jung Yoo*, Ok-Bae Chang*

*Dept of Computer Science, Chonbuk National University

요 약

본 연구는 레거시 시스템의 인터페이스 정보로부터 의미 있는 정보를 파악하여 새로운 시스템에 통합될 수 있도록 하기 위한 기존 레거시 시스템의 인터페이스에 기반한 객체 추출 기법(이하 TEILOR ; Techniques to extract Object based on Interface of Legacy System for Object Reusability)을 제안한다. 본 논문에서 제안하는 TEILOR는 인터페이스 사용사례 분석 단계, 인터페이스 객체 분할 단계, 객체구조 모델링 단계, 객체 모델 통합 단계 등 4단계로 구성되어 있다. 인터페이스 사용사례 분석 단계는 인터페이스 구조, 레거시 시스템과 사용자간의 상호작용 정보를 획득하는 단계이다. 인터페이스 객체분할 단계는 인터페이스 정보를 의미 있는 필드들로 구분하는 단계이며, 객체구조 모델링 단계는 인터페이스 객체들간의 구조적 관계와 협력 관계를 파악하여 모델링하는 단계이다. 마지막으로 객체 모델 통합 단계는 객체 단위의 단위 모델들을 통합하여 추상화된 정보를 포함한 상위 수준의 통합 모델을 유도하는 단계다. TEILOR에 의해 생성된 객체 통합 모델은 역공학 기술자들의 레거시 시스템 이해와 레거시 시스템의 정보를 새로운 시스템에 적용하는데 있어 효율성을 극대화할 수 있다.

1. 서론

객체지향 패러다임이 제시된 이래 오늘날 대부분의 시스템을 개발하는데 객체지향 패러다임이 폭 넓게 적용되어가고 있는 추세이다. 그 이유는 객체지향 패러다임이 갖는 재사용성과 새로운 시스템에 자연스럽게 통합되어질 수 있는 유연함 때문이다[1]. 그러나 아직도 대부분의 기업에서는 비 객체지향적이거나 혹은 객체지향 언어를 사용하여 시스템을 개발하였으나 객체지향 개념이 정확히 적용되지 않은 레거시 시스템을 그대로 사용하고 있다. 왜냐면, 기업에서 사용하고 있는 현 시스템이 가장 안정적이기 때문이다. 그러나 이러한 구시대적으로 개발된 시스템은 급변하는 업무 환경에 적합하게 대처할 수 없는 경우가 대부분이다[2, 3]. 본 논문에서는 레거시 시스템의 자원 중에서 인터페이스 정보로부터 차세대 시스템에 적용될 수 있는 의미 있는 객체 정보를 추출해서 새로운 시스템과 통합될 수 있도록 하기 위한 역공학 프로세스 즉, TEILOR(Techniques to

extract Object based on Interface of Legacy System for Object Reusability)를 제안한다. 레거시 시스템으로부터 인터페이스에 관한 지식을 습득하고 습득된 지식을 이용하여 객체지향 모델을 생성해내는 것이 본 논문의 목적이다. 인터페이스는 최종 사용자 위한 사용자 인터페이스이다. 다시 말하면, 애플리케이션을 사용하는데 있어 최종 사용자로 하여금 가장 일반적으로 사용되는 공통된 수단이다[4]. 애플리케이션 시스템 대부분의 지식은 인터페이스와 그 인터페이스를 통해서 시스템과 대화하는 상호작용(interaction)에 의해 정보가 발생한다. 따라서 TEILOR는 최종 사용자와 시스템간의 상호작용 과정을 통하여 발생된 인터페이스 정보를 자연스럽게 분류 가능한 것이다. 본 논문의 구성은 다음과 같다. 먼저, 2장에서는 관련연구를 3장에서는 역공학을 이용한 TEILOR 구조에 대하여 설명하고 또한, 다른 기존의 역공학 방법론과 비교를 4장에서는 결론 및 향후 연구과제에 대하여 설명한다.

2. 관련연구

2.1 DB 스키마를 입력 자료로 하는 역공학의 예

DB 역공학에 관한 예는 관계형 DB 스키마를 ER 다이어그램으로 변환하는 방법, 네트워크 스키마를 ER 다이어그램으로 변환하는 방법, 계층형 스키마를 ER 다이어그램으로 변환하는 방법, 객체지향 DB 스키마를 바이너리 관계 다이어그램으로 변환하는 방법 등 여러 가지가 있다. 관계형 DB 스키마를 ER 다이어그램으로 역공학 하는 방법은 함수적 종속성(functional dependencies)과 내포 종속성(inclusion dependencies)을 기반으로 한다[4]. 또한 DBMS(Database Management System)내의 카탈로그(catalog) 정보와 기존 DB의 튜플(tuples)을 이용하여 DB의 개념적 다이어그램을 유도해낸다.

2.2 소스 프로그램을 입력 자료로 하는 역공학의 예

소스 프로그램을 입력 자료로 하는 역공학의 예는 소스 코드의 모듈 구조나 프로그램 논리 구조를 유도하는 것이 목적이다[1]. 이를 위해 소스 프로그램을 분석하고 이를 다시 의미 있는 정보 단위로 분할하는 것이다. 대표적인 예로 RE2(Reuse Reengineering) 프로젝트가 있다. RE2 연구 프로젝트는 CNR(Italian National Research Council)에 의해 투자되었으며, Naples 대학교의 DIS(Departments of 'Informatica e Sistemistica')와 Durham 대학교의 CSM(Research Institute in Software Evolution)에서 공동으로 수행되었다. RE2 프로젝트는 소프트웨어 재사용을 목적으로 하고 있으며, 후보 기준(candidature criterion)을 설정하여 활용한다. 그러나 이를 위해서는 데이터 추상화를 시켜야 되는데 처리규칙에 기반한 기준이 제안되었으나 기준을 설정하는 것이 모호하다.

3. 객체추출 기법: TEILOR

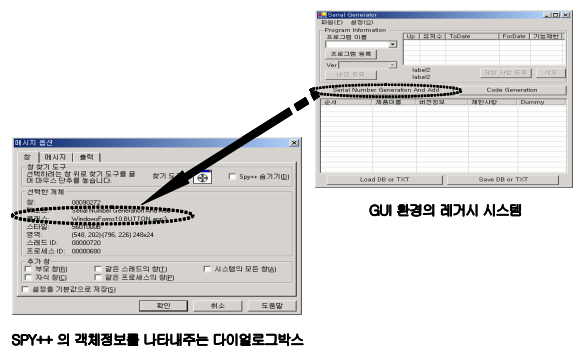
에이전트 기반 정보 수집기는 TEILOR를 위해 지원될 모니터링 프로그램의 한 종류이다. 이를 통해 수집된 정보는 인터페이스 지식 저장소에 저장되며 다음 프로세스에서 이러한 정보를 이용할 수 있다.

3.1 에이전트 기반 정보 수집기의 정보 수집 예

레거시 시스템으로부터 객체 정보를 얻기 위한 과정은 레거시 시스템의 인터페이스 환경이 CUI와 GUI(본 논문에서는 마이크로소프트사의 운영체제를 표준 플랫폼으로 가정했음)인 형태가 있다. 본 논문에서는 CUI 환경의 레거시 시스템에 초점을 맞춰 객체 추출 기법을 설명하고 있으나 GUI 환경이라 하더라도 GUI 환경에서 객체 정보를 추출하는데 그다지 어렵지 않다. CUI 환경에서는 VGA 그래픽카

드의 경우 B000:B800 은 텍스트 디스플레이를 위한 비디오 프레임 버퍼 공간이다. 따라서 CUI 에서 실행되는 에이전트를 램상주 프로그램으로 작성하고, 특정 이벤트(일반적으로, 단축키를 누르면) 발생시 프레임버퍼를 스캔한다. 이렇게 하면 화면 정보를 그대로 불러올 수 있다. B000:B800 메모리는 <속성, 문자코드> 의 2바이트 정보가 반복되어 있기 때문에, 속성은 배경색, 전경색, 강조, 깜빡임 등의 속성이고, 문자코드는 씌어지는 문자의 코드 값이다. 예를 들어 화면에 “date:[]” 라는 형식으로 되어 있다면 date를 변수로 잡을 수 있으며, [] 같은 문자나 색상이 달라지는 것을 통해서 입력을 위한 공간을 예측할 수 있다.

두 번째, GUI 환경에서는 GUI에 등록되어있는 객체 정보를 얻어낼 수 있는 소프트웨어(마이크로소프트사의 SPY++와 같은)를 이용하여 인터페이스 정보를 얻어낼 수 있다. 마이크로소프트사의 윈도우 환경이라면 SPY++라는 소프트웨어를 이용하여 인터페이스 객체정보를 획득할 수 있으므로 에이전트는 SPY++를 통해 획득된 정보를 토대로 객체정보를 추출한다. 다음은 SPY++를 실행시켜 GUI 환경의 레거시 시스템으로부터 획득한 정보를 추출하는 과정을 보여준다. 그림 1은 GUI 환경에서 실행되는 애플리케이션의 한 종류이다. 애플리케이션의 해당 객체 정보를 다이얼로그 박스에 나타내어주는 메시지 창이다. 여기에서는 “Serial Number Generation and Add”라는 버튼에 해당되는 정보를 나타내주고 있다.



GUI 환경의 레거시 시스템

SPY++ 의 객체정보를 나타내주는 다이얼로그박스

그림 1 애플리케이션의 객체 정보 파악

그림 2는 애플리케이션의 인터페이스에 나타나있는 객체정보(그림 2에서는 버튼 객체만을 예로 하였음)의 등록 사항을 나타내주는 그림이다. 그림에서 보는 바와 같이 각 버튼 객체와 인터페이스의 구성들도 알기 쉽게 등록되어있다는 것을 한눈에 볼 수 있다. 에이전트는 이러한 SPY++의 객체정보를 텍스트 파일로 변환한 다음 이를 인터페이스 저장소에 저장하여 다음 단계인 인터페이스 객체 분할 단계에서 이용된다.



그림 2 SPY++를 통한 애플리케이션의 객체 등록 형태

3.2 TEILOR의 객체 추출 기법

TEILOR은 4단계의 세부 프로세스 즉, 인터페이스 사용사례 분석 단계, 인터페이스 객체 분할 단계, 객체구조 모델링 단계, 모델 통합 단계 등으로 구성되어 있다. 그림 3은 TEILOR의 구조도를 도식화한 것이다.

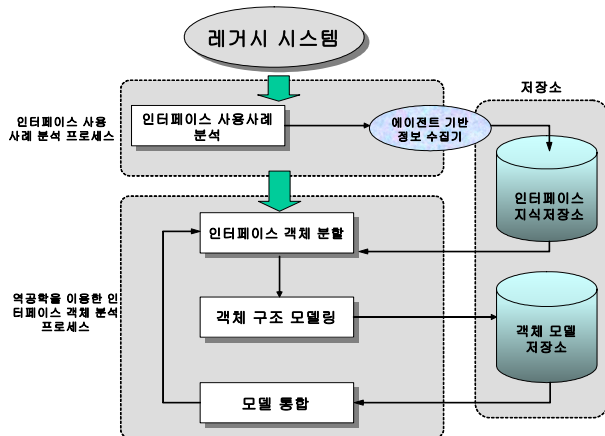


그림 3 TEILOR의 구조도

첫 번째 인터페이스 사용사례 분석 단계는 레거시 애플리케이션의 인터페이스 필드 정보와 인터페이스를 통해 사용자가 시스템과 상호 작용한 정보를 얻기 위한 단계이다. 사용자가 레거시 시스템의 인터페이스를 통한 상호작용의 예는 입력 필드에 어떠한 데이터를 입력하는 행위, 입력한 데이터를 처리하기 위해 처리 시스템에 보내기 위한 이벤트 발생 행위, 이벤트에 의해 발생되어진 입력 데이터를 처리하는 연산행위, 연산행위를 통해 처리된 데이터를 사용자 인터페이스에 다시 반환하는 행위, 데이터베이스로부터 데이터를 인터페이스에 넘겨주는 행위 등 여러 가지의 예가 있다. 이러한 상호작용 정보와 인터페이스에 관한 정보가 에이전트 기반 정보 수집기에 의해 자동 수집된다.

표 1 인터페이스 사용사례 분석 단계

과정	분석절차 기술	획득 정보
인터페이스 사용 사례 분석	① 레거시 시스템에서 단일 인터페이스 단위로 분할	· 인터페이스 구조 정보
	② 에이전트 시스템을 통해 사용자와 시스템 간에 상호작용 정보를 획득	
	③ 획득된 인터페이스 정보를 저장소 내에 저장	· 사용자/시스템 상호작용 정보
	④ 레거시 시스템의 인터페이스를 이용한 모든 정보가 완료될 때까지 ②, ③ 단계 반복 수행	

에이전트 기반 정보 수집기에 의해 얻어질 수 있는 인터페이스 구성 정보유형은 다음 네 가지의 유형이다. 첫째, 인터페이스 영역유형은 인터페이스의 영역별 기능을 나타내는 유형이다. 따라서 데이터의 항목을 나타낼 수 있는 항목별 필드 영역부분과 항목의 수를 카운트 할 수 있는 집계 필드 영역 또는, 그리드(grid) 형태로 나타낼 수 있는 그리드 영역, 그리고 이벤트를 컨트롤 할 수 있는 이벤트 컨트롤 영역, 마지막으로 조건에 따른 입력을 허용하는 조건부 입력 영역 등으로 세부 분류될 수 있다. 둘째, 입력지원 컨트롤 유형은 사용자가 인터페이스에 데이터를 입력하는 방법에 여러 유형이 있을 수 있는데 상황에 따라 사용자가 편리하게 데이터를 인터페이스에 입력할 수 있도록 한다. 셋째, 인터페이스의 필드를 편집할 수 있는 인터페이스 편집필드가 있다. 키 입력 필드와 데이터 입력필드, 조회와 수정 둘 모두 가능한 필드, 조회만 가능한 필드, 이벤트를 발생시킬 수 있는 필드, 마지막으로 시스템 관리를 위한 시스템 관리 필드의 유형이 존재할 수 있다. 넷째, 처리 적용형태에 따른 유형이 있다. 인터페이스 상에서 일어날 수 있는 처리 적용형태는 앞서 설명한 조회, 등록, 수정, 삭제의 크게 4가지가 있다. 이는 다시 인터페이스 상에서 이루어 질 수 있는 처리건별로 단일건 처리와 다수건 처리로 구분될 수 있으며 이는 또다시 데이터베이스를 기준으로 해서 다른 테이블과의 조인 또는 무조인으로 구분될 수 있다.

TEILOR의 두 번째 단계는 객체분할 단계이다. 이번 단계에서는 전 단계의 에이전트 프로그램에 의해 수집되었던 인터페이스에 관한 정보를 이용한다. 인터페이스에 관한 정보가 저장되어있는 인터페이스 지식 저장소로부터 인터페이스에 관한 다양한 분류 정보를 입력받는다. 그런 다음 인터페이스에 나타나 있는 다양한 필드들의 정보를 영역별(항목, 집계, 그리드, 이벤트, 조건부 영역 등) 및 입력지원 컨트롤(콤보박스, 서브 인터페이스, 라디오 버튼, 체크버튼

등) 그리고 인터페이스 편집필드 유형(키 입력 필드, 데이터 입력필드, 조회 및 수정 필드, 조회 필드, 이벤트 발생 필드, 시스템 관리필드) 등의 인터페이스 정보의 유형에 따라서 인터페이스에 관한 지식을 의미 있는 정보 단위 객체로 분할한다.

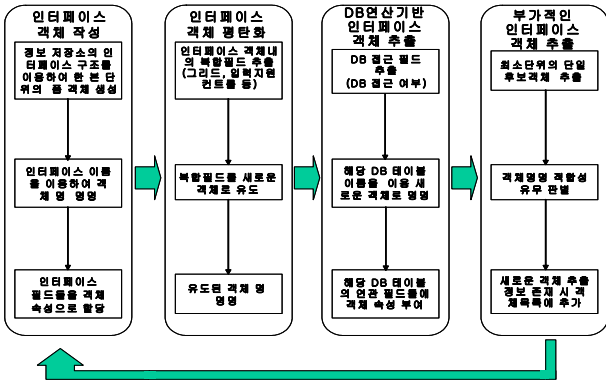


그림 4 인터페이스 객체 분할 단계

세 번째 단계는 객체 구조 모델링 단계이다. 이 단계는 이전 과정의 단계에서 나타난 결과로부터 객체들을 파악하는 단계이다. 또한 객체들간의 초기 수준의 구조적(structural) 관계와 협력(collaboration) 관계가 유도되는 단계이다. 이 단계에서 나타나는 결과물은 유도된 객체의 객체 명, 객체 속성, 그리고 구조적 관계로 구성된 객체구조 모델이 형성된다. 또한 객체구조 모델을 나타내기 위해 사용한 표기법으로는 현재 모델링 언어의 표준인 UML을 이용하여 나타낸다.

마지막 네 번째 단계는 모델 통합 단계이다. 모델 통합 단계의 목적은 전 단계의 결과물인 객체정보를 나타내는 단위 모델들을 통합하여 보다 상위 수준의 통합 모델을 제시하고자 함이다. 결과적으로, TEILOR의 과정을 통해 얻어지는 최종 결과물은 객체 구조 모델이다.

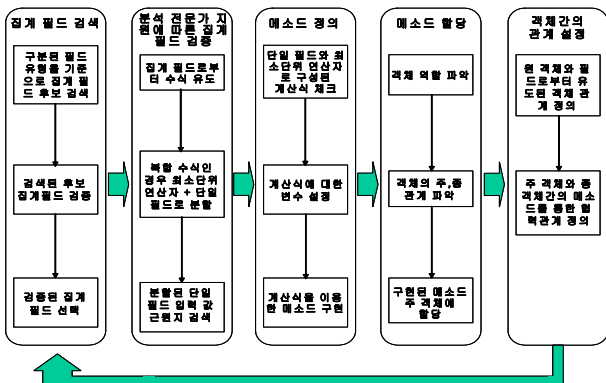


그림 5 객체 구조 모델링 단계

표 2는 관련 연구에서 알아보았던 몇 가지 역공학

방법론 과 본 프로세스를 비교해본 결과표이다.

표 2 기존 역공학 방법론과의 비교

방법론 항목	Chiang	RE2	FORE	본 프로세스
시스템 관점	데이터	프로세스	객체	객체
목적	· 물리적 DB에서 의미 데이터 복구	· 레거시 시스템으로부터 재사용 모듈 생성	· 레거시 시스템으로부터 객체 구조 유도	· 레거시 시스템으로부터 데이터, 프로세스를 이용한 객체구조 유도 · 통합 모델 추출
입력 데이터	· 물리적 DB 스키마	· 레거시 시스템의 모든 자료	· 입력 화면 양식과 사용자의 상호작용 정보	· 레거시 시스템의 인터페이스 정보 · 레거시 시스템/사용자간의 상호작용정보
결과 모델	· EER 모델	· 특별한 모델 없음	· ECRC를 이용한 객체 모형	· UML에 기반한 객체(클래스) 다이어그램 · 통합 모델

4. 결론 및 향후 연구

본 논문의 연구 결과로는 첫째, 입력자료는 사용자/시스템간의 가장 기본적으로 일어나는 인터페이스에 기반 한 상호작용 정보를 활용한 역공학 연구이고 둘째, 인터페이스에 관한 지식을 지식 저장소에 저장할 수 있는 방법을 알고리즘화 된 단계로 제시하였으며, 셋째, 데이터 및 프로세스를 동시에 다루고 있다는 것이다.

향후 연구사항으로는 에이전트 기반 정보 수집기를 완성해야 하며, TEILOR에서 보여준 객체 분석 과정을 자동화할 필요성이 있다.

5. 참고문헌

[1] H. Lee and Ch. Yoo, "A Form Driven Object-oriented Reverse Engineering Methodology", Information Systems, Vol. 25, pp. 235-259, May, 2000.
 [2] Aiken, P., A. Muntz, R. Richards, "A framework for reverse engineering DoD legacy information systems", Proceedings Working Conference on Reverse Engineering, Baltimore, Maryland, May, pp. 180-191, 1993.
 [3] Umar, A., Application Engineering Building Web-Based Applications and Dealing with Legacies, Prentice Hall, 1997.
 [4] Batini, C., S. Ceri, S. B. Navathe, Conceptual Database Design An Entity Relationship Approach, Benjamin/Cummings, Redwood City, 1992.