

소프트웨어 재사용시스템

1. 시스템 명 : 소프트웨어 재사용시스템

1) CARS(Computer-Aided Reuse System)

Version 1.0 / Version 2.0

2) 부품 개발 지원

2. 제 작 자

주관연구기관 : 중앙대학교

공동연구기업 : (주)유니온시스템, (주)한국정보시스템

위탁연구기관 : 강원대학교

협동연구기관 : 한국과학기술연구원 시스템공학연구소

연구기간 : 1990. 7 - 1993. 7 (3년)

1차년도 : 1990. 10 - 1991. 10

2차년도 : 1991. 10 - 1992. 10

3차년도 : 1992. 10 - 1993. 10

연구책임자 : 중앙대학교 컴퓨터 공학과

소프트웨어 공학 연구실 이 경 환

3. 시스템 개요

소프트웨어 작성 과정은 항상 새로운 설계나 코드의 생성이기 보다는 기존의 유사한 통용 분야의 산물인 설계 개념이나 코드 뿐 아니라 문서의 재사용 과정이라고 볼 수 있다. 소프트웨어 개발 과정의 생산성 제고를 위하여 소프트웨어 부품을 작성하고 재사용에 의해서 소프트웨어를 개발 할 수 있는 환경을 구축하였다. 본 재사용 시스템은 재사용의 쏠단계를 효과적으로 지원하기 위한 검색 시스템, 이해 시스템, 라이브러리 관

리 시스템을 구축하였다. 그리고 본 라이브러리에 저장하는 부품의 생산 방법은 객체 모델링 기법에 의해 표준화함으로써 재사용의 효율을 극대화시켰다. 부품 작성을 위한 객체 모델링 과정을 표준화 시킴으로써 소프트웨어 자동 생산에 관한 실현을 유도할 수 있게 하였다.

4. 개발 단계별 기간 및 소요 인원

석사학위 수여자 19명 \times 2년 = 38(M/Yr)

박사학위 수여자 8명 \times 4년 = 32(M/Yr)

5. 프로그램

1) 재사용 시스템

CARS 1.0 : 33본

CARS 2.0 : 28본

2) 부품개발 지침서

4개의 단계

20개의 활동(Activity)

6. 사용언어

CARS 1.0 : Boriand C++ 3.0

Boriand C

CARS 2.0 : UNIX C

7. 사용 시스템

CARS 1.0 : PC : 80386-DX

OS : MS-DOS Ver 5.

CARS 2.0 : IBM RISC /6000

OS : AIX 3.1

X-Window System

Motif Wldget

8. 직접효과 :

1) 전산화 효과

가. 생산성

CARS 시스템의 라이브러리에 저장할 부품을 개발하고 목적하는 시스템으로 조립할 경우 첫번째 개발에서는 1.3배 정도의 자원이 도움된다. 두번째 유사한 소프트웨어를 개발할 경우 50%의 재사용을 한다면 2개의 소프트웨어 개발에 1.8배 자원이 소요되므로 20%의 생산성을 가져온다. 계속되는 소프트웨어 개발에서 그 효과는 급증한다. 그 예는 사례집을 통하여 확인할 수 있다.

2) 공동 활용성

가. 공동 활용을 위한 본 연구 개발팀의 역할

- CARS와 부품 생산 도구 (객체 모델링)을 통합한 ICASE 개발
- 객체지향 Framework 구축 : C++ 이상의 기능을 가진 객체지향 언어개발, 객체지향 DBMS 개발 및 윈도우즈 NT와 같은 OS개발
- 공개적인 교육훈련 : 부품 생산방법 및 CARS에 의한 소프트웨어 개발

나. 공동 활용을 위한 일반 사용자의 역할

- 교육을 통한 기술 전수
- 독자적인 응용영역별로 부품을 생산하여 라이브러리 구축
- CARS에 의한 소프트웨어 개발

9. 간접효과

1) 소프트웨어의 자동생산

CARS와 그 부품 생산도구를 ICASE로 워크벤치를 만든다면 생산성의 효율은 기하급수적으로 늘어난다.

2) 공동 활용성

하나의 라이브러리를 응용영역별로 관련자들이 공유하고 그 부품을 공동으로 활용할 수 있다. CAD 부품의 검색에서 본 것처럼 여러 사람이 동일 프로젝트를 가지고 작업을 수행해 나갈 수 있다. 워크벤치를 재사용 프레임으로 확장한다면 완전한 소프트웨어 자동생산 시스템을 발전시킬 수 있다.

가) 공동활용을 위한 공개 소프트웨어 CARS 1.0

- MS/DOS 및 K/OS용으로 제작된 CARS 1.0을 공개하여 관심있는 전문가들이 활용할 수 있게 한다.
- ATM, 로봇, CIM등 이외에도 새로운 응용영역을 발굴하여 객체지향에 의한 소프트웨어를 개발하고 공개한다.

3) 유지보수 및 확장성

부품 개발 지침서에 따라서 부품을 개발하면 모든 부품이 일관성을 유지하게 되고 여러 사람이 동일한 영역의 부품을 개발하여 공유할 수 있다. 따라서 유지 보수 작업은 승계에 의한 클래스의 확장으로 수행될 수 있으므로 매우 쉽게 수행되고 확장해 나가기도 쉽다. 또한 CARS 시스템의 기능을 확장하기 위해서도 부품 개발 지침서를 도구로 만들어 CARS의 Front-end 시스템으로 개발하면 ICASE가 된다.

4) 운영의 효율성

검색 기능과 라이브러리 관리 기능을 Motif를 사용하여 툴킷 형식으로 개발함으

로써 사용이 매우 간편하다. 또 동의어 사전과 이해 시스템은 사용자를 도와서 부품의 이해와 검색을 지원할 수 있다.

5) 소프트웨어 관리 방법

CARS 라이브러리는 소프트웨어 베이스를 구축하고 있으므로 소프트웨어 부품과 그 시스템에 관한 모든 정보(개발자, 개발일자, 버전 관리, 사용빈도, 클래스로 확장된 영역 등)를 라이브러리 관리 서브시스템이 수행해 준다.

10. 개발기술 및 특기사항

가. CARS 개발의 기반 기술

1) 객체 지향 기술

- 추상화 • 다중연산
- 캡슐화 • 정보은닉
- 승계 • 소프트웨어 분류방법론
- 관계

2) 객체모델링 기술

- 객체기술 양식
- 속성과 행위기술의 양식
- 모델링 카드 설계
- 객체 모델링의 Procedure 표준화
- 객체의 상태와 이벤트의 모델링

3) ICON 인터페이스 기술

4) 사용자 인터페이스 툴킷 설계

5) 객체의 클러스터링

6) 객체의 이해 방법론

7) 부품의 합성이론

8) 객체의 품질 평가기법

9) 클래스의 상속 모델링

10) 클래스 라이브러리의 재구성 방법론

11) CARS의 GUI 구축 방법론

12) ADT 재사용을 위한 지식구조와 관계정보의 검색

나. CARS의 기능적 특성

1) CARS 사용자 인터페이스

1. 자연언어의 처리기능
2. 기능을 기반으로한 검색

3. 기능을 기반으로한 계층 구성도 제공
 4. 클래스의 계층 구성도 제공
 5. 사용자 인터페이스 및 그 구성요소를 객체로 설계하여 재사용이 편리함.
 6. Browsing에 의한 부품의 수정가능
 7. Pull-down 메뉴제공
 8. 부품 및 그 정보의 검색 지원
- 2) 라이브러리 관리 서브시스템
1. 부품의 등록과 삭제기능
 2. 부품별 재사용 정보의 추출
 3. 재사용 정보를 사용자별로 관리
 4. 부품간의 기능적 유사도의 정보추출
- 3) 라이브러리
1. 응용영역별 부품코드를 작성하여 검색을 지원
 2. 부품의 인덱스를 정하여 검색을 지원
 3. 부품의 유사도에 관한 정보제공
- 4) Term Dictionary
- 사용자가 검색을 위해서 질의어를 사용하는 용어(term)를 CARS에서 정의한 용어로 바꾸어서 용어를 통일시킨다. 즉, 동의어를 하나의 표준화된 언어로 바꾸어서 검색을 지원한다.
- 5) 검색 서브 시스템
- 검색을 지원하는 4개의 윈도우를 정의하였다.
1. Query Window : 부품의 기능 기술
 2. Component Window : 질의어를 바탕으로 하여 부품에 관한 인덱스, 유사도, 부합도 등을 보여준다.
 3. Browse Window : 질의어에 해당된 부품을 순서대로 나열하고 사용자가 가장 합당한 부품을 선택하게 한다.
 4. Class Window : 클래스의 계층도를 나타내어 승계 관계를 알려주고 새로운 객체의 생성을 지원한다.
- 6) 이해 서브 시스템
- 검색 시스템이 검색해준 부품을 재사용하기 위해서는 그 부품을 충분히 이해해야 한다. CARS는 이를 위해서 여러가지 정보를 제공
1. 부품에 관한 모든 정보
 2. 클래스 다이어그램

3. 클래스 관계 다이어그램
4. 입출력 차트
- 7) 부품의 품질평가 서브시스템
 1. 부품별 coupling을 체크하여 사용자의 부품선택을 지원
 2. 부품별 Cohesion을 체크하여 사용자의 부품선택을 지원
 3. 모든 부품의 품질에 등급을 정해서 사용자 선택을 지원한다.