

차세대 CASE

한국소프트웨어품질(연)
강원대학교

양해술*
이하용**

● 목 차 ●

- | | |
|------------------|--------------|
| 1. 서 론 | 4.2 개발환경의 통합 |
| 2. CASE의 현상과 문제점 | 4.3 프로세스 관리 |
| 2.1 CASE의 현상 | 4.4 정보저장소 |
| 2.2 CASE의 문제점 | 4.5 리엔지니어링 |
| 3. CASE의 발전 방향 | 4.6 개방형 CASE |
| 4. 차세대 CASE | 5. 결 론 |
| 4.1 개발방법론의 활용 | |

1. 서 론

소프트웨어에 대한 사용자의 요구가 다양화 고품질화되면서 소프트웨어 개발에 소요되는 비용과 노력이 급격히 증가한 반면 소프트웨어의 신뢰성과 생산성 및 품질은 사용자의 요구를 적절히 반영하지 못하는 수준에 머무르고 있다.

그리고 개발자들은 사용자의 요구 및 수요에 충분히 대응하면서도 품질과 신뢰성을 갖춘 소프트웨어를 개발하기 위한 방법 즉, 비용 대 효과가 높은 방법을 개발하고자 노력하게 되었고, 이러한 시대적인 조류에 부응하여 CASE (Computer Aided Software Engineering)는 노동 집약적인 수준에 머물러 왔던 소프트웨어 개발 방식을 부분적으로 자동화하여 소프트웨어의 개발을 기술집약적이며 효율적으로 지원하고 있다.

그러나 현재 개발되어 있는 대다수의 CASE 도구들은 특정 방법론의 일부 기능을 지원하거나 생명주기 단계의 후반부인 구현과 테스트

단계를 지원하는 하위 CASE가 주류를 이루고 있다.

이와 같은 도구들을 이용하는 경우 소프트웨어 개발에 대한 완전자동화를 기대할 수 없을 뿐만 아니라 일관성있는 작업을 할 수 없으며, 개발작업의 중복성과 불필요한 시간과 비용이 추가적으로 소요된다. 이로 인하여 인터페이스의 다양화, 데이터의 다양화, 제어의 다양화, 개발 도구와 관리 도구의 불일치 등의 문제점들이 발생하게 된다. 이러한 문제점을 해결하기 위한 방안으로 소프트웨어 생명주기 전단계를 지원하는 통합형 CASE에 대한 연구의 필요성이 대두되고 있으며 통합형 CASE에서 요구되는 특성들에 대한 보완이 활발히 진행되고 있다.

본 고에서는 현재 CASE에 대한 연구 개발이 어느 정도 진전되는지에 대한 CASE의 현상을 소개하고 아직까지 산재해 있는 CASE의 문제점을 살펴보고, 이러한 문제점들을 개선하고 CASE가 어떠한 방향으로 발전해 나가는 것이 바람직한가를 기술하고 통합화와 개방화를 지향하는 차세대 CASE의 모습을 살펴보고자 한다.

*중신회원

**학생회원

2. CASE의 현상과 문제점

2.1 CASE의 현상

지금까지의 CASE는 주로 구조적분석과 설계방법론을 지원하여 왔으나 최근에는 정보공학방법론의 부상으로서 인하여 정보공학방법론 지원 도구가 보편화되고 있다. 또한, 분산기술의 발전에 힘입어 분산개발환경에 알맞은 클라이언트/서버 환경으로 이전함에 따라 CASE 도구들이 단지 메인프레임 중심이 아니라 클라이언트용, 서버용 등으로 분류되면서 UNIX, OS/2 등으로 지원 플랫폼을 클라이언트/서버 환경으로 방향을 전환하고 있다.

이하에서는 이와 같은 CASE의 최근 동향으로 MIT의 Cusmano는 저서 "일본의 소프트웨어 전략"에서 소프트웨어 공장방식의 공통요소로서 "체계적인 재이용"과 "컴퓨터 이용 도구"로 분류하여 그 중심이 되는 CASE로 개발한 EAGLE과 SEWB를 소개하기로 한다.

먼저, EAGLE은 초기에는 호스트형의 CASE로서 대화처리에 의한 설계지원, 부품에 의한 프로그램의 합성, 데이터 사전에 의한 보수 문서의 작성과 같은 주요한 특징들을 가지고 있다. 또, 데이터 항목을 중심으로 한 설계기법인 데이터 중심 접근기법을 실현하는 데이터 항목 사전이 있으며 그림 1과 같이 프로그램 골격인 패턴과 서브루틴인 기능부품들을 사용하여 프로그램을 합성하고 있다.

그리고 SEWB는 분산형 CASE로서 1983년에 개발한 UNIX WS(Workstation)에서 가동하는 CASE이며 현재 지역분산개발의 진전에

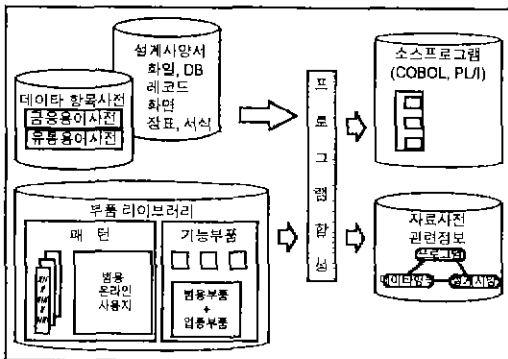


그림 1 통합형 CASE EAGLE의 구성

따라 EAGLE로부터 SEWB에 의한 개발로 주체가 이동하고 있다. 최근에 SEWB3의 최신 버전이 출하되었으며 그 특징은 다음과 같다. 클라이언트/서버 시스템 환경의 실현

- CASE 플랫폼의 실현
- 저장소에 의한 자원의 통합 관리
- 소프트웨어의 일괄 생산
- 데이터 중심 접근
- 온라인 프로토타이핑
- 리얼타임이팅의 실용화

SEWB3는 그림 2와 같이 국제표준모델에 대응한 CASE 플랫폼의 실현으로서 그룹 작업을 지원하며 그 일환으로 프로세스 관리지원, 프로젝트 관리지원, 개발작업 지원 기능을 가지고 있으며 프로세스 관리지원으로는 다음과 같은 사항을 지원한다.

- 프로세스(개발절차, 성과물, 규칙) 정의와 레이아웃
 - 시나리오에 의한 작업유도
 - '관련'을 이용한 관련정보의 참조
 - 명세서간의 무결성, 동기성을 유지하는 변경 통지
- 그리고 프로젝트 관리지원으로는 프로젝트 관리정보의 자동수집 기능을 지원하며 개발작업 지원으로는 다음과 같다.
- 설계문서의 정확한 정리(목차부터 설계 명세서를 모아 제본)
 - CASE 지원도구를 워드프로세스 도구와

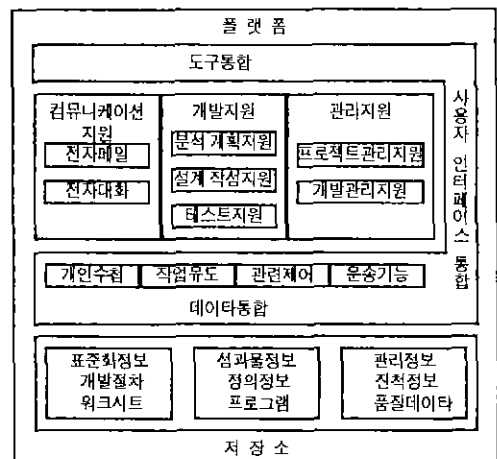


그림 2 CASE SEWB3의 플랫폼

연계

- 명세서와 워드문서(설계메모, 설계이유)인 부정형 데이터의 축적

또한, SEWB3는 저장소를 서버에 탑재하고 클라이언트(WS 또는 X단말)에서 지원 도구를 가동시킬 수 있으며 소프트웨어의 일괄생산을 위해 데이터 중심의 접근방법으로 데이터를 표준화하여 표준 데이터의 유지 프로세스를 중심으로 설계하는 방식을 사용한다.

그림 2의 CASE 플랫폼은 NIST/ECMA의 참조 모델에 기초한 도구 통합, 사용자 인터페이스 통합, 데이터 통합, 프로세스 통합의 메카니즘을 제공한다. 그리고 사용자 인터페이스는 OSF/Motif에서 통합하고 프로세스 통합에서는 프로세스의 정의와 제정, 계획을 행하며, 프로세스의 정의내용(개발절차, 성과물, 규칙(rule))에 따라 작업 유도(navigation)를 한다.

2.2 CASE의 문제점

먼저, CASE 사용자의 입장에서 CASE를 바라 본 경우에 특히 엔드유저컴퓨팅을 기반으로 하여 시스템 구축을 진행하기 위해서는 사용자 인터페이스의 이해용이성은 물론이고 업무 시스템의 개발 기법이나 완성되는 시스템의 구조가 사용자가 이해하기 쉬워야 한다는 점이 중요하다. 이해용이성의 포인트는 이미 가지고 있는 통상의 업무지식이나 시스템 이용경험을 기초로 할 때에 그것과 큰 차이가 없는 스타일에서 새로운 시스템을 이해할 수 있고 구축할 수 있는 것이다. 현재의 CASE가 사용자에게 받아들여지기 어렵다고 생각되는 점을 정리하면 다음과 같다.

- CASE에 따라 조작성이나 데이터 표현 형식이 통일되어 있지 않다. 따라서 각 도구마다 제공하는 조작이나 데이터 표현의 개념을 이해할 필요가 있다.
- 데이터 중심설계나 객체지향 설계와 같은 기술적인 키워드에 대해서는 추상적인 개념으로 되어있어 접근하기 어려운 점이 있다.
- CASE를 사용한 개발과 종래 기법과의 차이나 효과를 정량적으로 알기 어렵다.

따라서 도입시의 구입비나 교육비 등을 클 것으로 생각할 수 있다.

- CASE에서는 표준적인 설계 기법이나 명세표현을 대부분 지원하고 있지만, 각 이용조직에 적합한 최적화된 환경을 구축하는 것과 같은 수주 지원 기능이 부족하다.

그리고 기술적인 관점에서 바라본 경우 현재의 CASE가 포함하고 있는 문제를 정리하였다.

- CASE의 지원 레벨이 아직 문서화 지원의 틀을 넘지 못하고 있다. 물론, 도표로 표현된 명세의 무결성 체크나 다음 공정으로의 정보운송 등의 기능이 충실하다 해도 지적설계 활동을 직접 지원하고 있다고 말하기 어렵다.
- CASE의 보급에는 그 기반이 되고 있는 개발기법의 보급과 밀접한 관계에 있지만, 기법 자체가 널리 보급되어 있지 못하다.
- 엔드유저 컴퓨팅이나 신속한 개발 언어로 대표되는 것처럼 생산성을 혁신하기 위한 연구가 불충분하다.

현재의 CASE가 가지고 있는 문제점을 기술적인 관점과 사용자의 관점에서 요약하면 다음과 같이 정리할 수 있다.

(1) 기술적인 관점

- 지적활동의 직접지원이 되고 있지 않다.
 - 문서화 지원의 범위
 - 일부는 무결성의 체크나 하위로의 정보전달을 가짐
 - 사고결과와 정리지원이 주이고, 사고 프로세스의 가이드는 미약
- 기법이 확립 정착되어 있지 않다.
 - 업계의 표준 부재
 - 기법의 미성숙, 설계 정석집 등의 권위 있는 집대성이 필요
- 생산성 혁신으로의 효과가 불충분
 - 엔드유저 컴퓨팅과 신속한 개발에 대한 요구/기대가 큼

(2) 사용자의 관점

- 도구에 따라 조작성이나 데이터 호환성이 없음
 - 습득 비용의 중복
 - 장래성에 대한 불안
- 이용효과가 확실하지 않다
 - 상위에 대해서는 정량화가 곤란
 - 품질의 생산성을 도모하는 척도가 필요
- 기본적인 기술개념의 이해 곤란
 - 추상적인 언어(객체지향 등)
- 효과에 비해 비용이 높다
 - 도입비용, 교육비용
- 사용자 환경으로의 최적화가 곤란

3. CASE의 발전 방향

미래의 CASE는 전체로서는 표준화가 진척되는 것과 더불어 몇 개의 분야로 분할되면서 발전할 것으로 생각할 수 있다. 종래에는 COBOL을 중심으로 한 업무 응용프로그램 개발 지향 CASE나 C를 중심으로 한 조립 소프트웨어 또는 엔지니어링 응용 지향 CASE로 분류하였다. 그러나 앞으로는 CASE 이용자층에 대해 보다 전문적으로 분할된 고도의 서비스를 제공할 것으로 기대된다.

관련 업무의 개발은 앞으로 상당한 레벨의 기술을 가진 소프트웨어 기술자에 의해 개발이 진행될 것이다. 기술적으로는 대규모 조직에서 개발하는 것을 지원하기 위해 필요한 공동 작업 지원기능(그룹웨어), 프로젝트 관리기능과 생산계 기능의 밀접한 연계 등이 앞으로의 방향으로서 추천될 수 있다. 그리고 기간업무 중에서도 금융계, 유통계 등의 각 분야 전문 시스템을 만드는 방법의 구조가 정리되고, 시스템의 패키지화를 지금 이상으로 가속화하여 생산효율을 높일 필요가 있다. 또한, 다운사이징 중에서 업무 시스템의 재구성이 진행되는 타이밍에서 다양한 방식이 실현되고 있는 것으로 생각할 수 있다.

한편, 말단 정보계의 개발에 대해서는 정보 이용부문의 인간이 직접 개발할 수 있도록 엔드유저에 의한 개발을 적극적으로 지원하는 CASE가 도입되고 있다고 생각할 수 있다. 엔

드유저 지향 CASE는 GUI의 구축 도구나 데이터베이스 어플리케이션 구축 도구, 정보 프리젠테이션 도구 등의 기술을 통합화한 것이 될 것이다. 처리의 기술을 간단히 하기위해 스크립트 언어와 같은 간이 언어나 시각적인 프로그래밍 환경의 발전이 필요하다.

그리고 CASE를 더욱 효과적으로 활용하기 위한 과제로 우선 CASE를 사용하는 개발문화를 조직적으로 정착시킬 필요가 있다. 또한, 최근에 ISO9000 시리즈에서 소프트웨어 개발 프로세스로의 관심이 높아지고 있다. 이 기회에 여러 가지 조직의 개발 스타일의 재정의에 참여 현상의 개발 형태가 안고 있는 문제점을 구체적으로 파악함으로써 도입하는 도구의 선택이 가능해질 수 있다.

4. 차세대 CASE

대규모 정보 시스템은 신규 개발형의 프로젝트는 감소하고 은행 온라인 시스템과 같은 대형 시스템은 패키지나 기존 소프트웨어 리엔지니어링에 의한 재이용형으로, 기업업무나 계획업무 등의 정보계 시스템은 엔드유저 개발형으로 이동하고 있다. 또 다종다양한 정보시스템의 단기개발 요구가 증대하므로 이와 같은 상황에 대응하여 다음과 같은 소프트웨어 생산기술이 필요해 진다.

- 리엔지니어링 기술

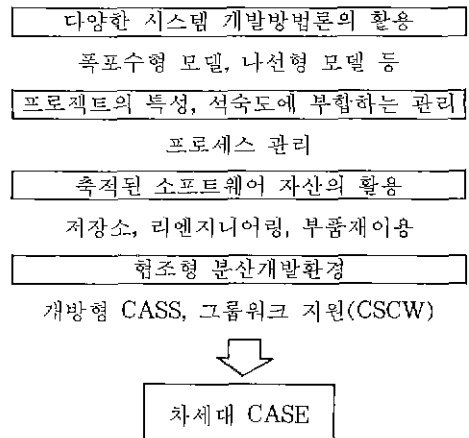


그림 3 차세대 CASE를 이루는 공통요소

- 엔드유저 개발 기술(EUD:End User Development)
- WS/PC 기반의 유연한 개발환경 구축 기술
- 객체지향 부품 재이용 기술

이들 신기술을 기반으로 미래의 소프트웨어 공장에서는 그림 3과 같은 공통요소가 중요해 지므로 이하에 이들 공통 요소에 대해 설명하기로 한다.

4.1 개발방법론의 활용

CASE는 개발하고자 하는 대상 시스템이 무엇인가에 따라 선택의 기준이 달라질 수 있다. 특히 우선적으로 고려할 사항은 정보시스템용 CASE와 시스템 소프트웨어용 CASE를 구분하는 것이다.

정보시스템용 CASE는 기업의 전략 정보를 처리하기 위한 시스템을 구축하기 위해 사용되며, 시스템 소프트웨어용 CASE는 과학 및 기술분야의 소프트웨어를 개발하기 위해 사용한다. 정보시스템용 CASE는 Martin의 정보공학 이론을 바탕으로 정보활동모형으로부터 출발하여 기업과 사업목표를 존중하는 정보요구를 추출하고 이로부터 개별 정보시스템을 구축하는 형태의 방법론을 사용하고 있으며, 시스템 소프트웨어용 시스템은 전통적인 소프트웨어 공학이론을 바탕으로 하향식 접근방법, 모듈화와 구조적 프로그래밍 기법, 최근에는 객체지향 개발 방법론도 많이 사용되고 있다. 그러나 객체지향 방법론을 적용한 CASE 도구들은 구조적인 방법론과 객체지향 방법론의 모델 표현을 지원하는 정도이다. 앞으로 모델의 표현은 물론이고 일관성 및 무결성 진단과 통합 CASE 도구로서의 발전과 함께 실시간 소프트웨어 개발과 같은 시스템 소프트웨어 개발에도 직접 응용될 수 있는 도구로 발전하리라 기대되며, 개발하고자 하는 대상 시스템의 특성이나 개발 형태에 알맞은 복수의 개발 모델을 지원할 수 있는 방향으로 나아갈 것이다.

4.2 개발환경의 통합

개발환경을 통합하기 위한 여러 가지 방법이 제안되고 있으나 크게 구분하면 다음의 3가지

접근방법으로 추진될 수 있다.

(1) 통합 CASE 환경의 접근방법

CASE 환경의 개발자를 중심으로 하여 개발되고 있는 각종의 CASE 환경을 통합해 가는 접근방법으로 그림 4와 같은 4가지 통합 접근방법이 제안되어 있다.

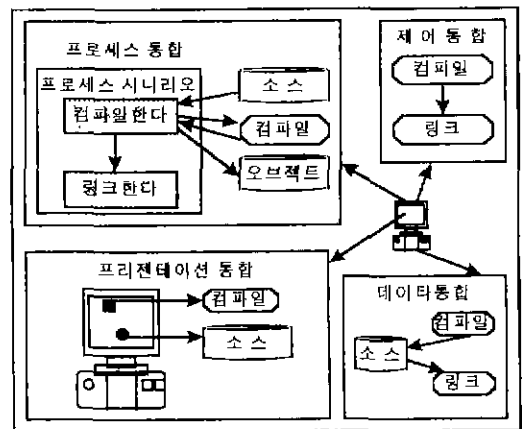


그림 4 CASE 환경을 통합하는 접근방법

(2) 개발 프로세스의 접근방법

소프트웨어 개발 프로세스의 모델화와 그 지원 연구로 발전하여 개발 프로세스를 WS상에 표시하고 가이드하며 필요한 도구를 자동적으로 기동하는 방법과 지원환경이 개발되고 있다. 따라서 개발 프로세스는 CASE 환경과 프로젝트 관리와의 가교역할로서 프로세스 통합이 이루어지고 있다.

(3) 그룹워크 CSCW의 접근방법

협조작업 CSCW(Computer Supported Cooperative Work)를 컴퓨터에 의해 지원하는 접근방법은 소프트웨어 개발뿐만 아니라 광범위한 비즈니스의 작업도 대상으로 하여 소프트웨어 개발에 적용하고 있다. 예를 들면, 워크플로우 관리는 소프트웨어 개발에서는 개발 프로세스의 관리에 해당되면서 또한, 그룹에 의해 결정되는 의사결정과도 연계되고 그룹 의사결정지원이나 커뮤니케이션 지원은 분산형 개발환경을 통합하는 여러가지 기술을 제공할 것으

로 기대된다.

4.3 프로세스 관리

프로젝트의 특성에 따른 개발 프로세스를 정의하여 실현하며 특성에는 성숙도, 개발형태, 프로젝트 구성 등이 있다. 프로세스의 정의, 분석, 개선, 예측, 재이용을 도모하는 프로세스 관리자가 전문직화한다. 개발 프로세스를 향상시키기 위해서는 개발 프로세스의 현재 수준 및 문제점을 파악하기 위해 개발 프로세스에 대한 측정이 필요하다. 프로세스를 측정하고자 할 때 가능한한 도구를 사용하여 자동적으로 자료를 취득할 수 있도록 하는 것이 바람직하지만 기존의 도구들은 대부분 개발 프로세스의 자동적인 측정에 대해 사전에 배려하지 않은 문제점이 있다. 이러한 점을 해결하기 위해 프로세스 정보취득 기구에 대한 연구가 이루어지고 있다. 즉, 프로세스를 파악해야 하는 중요한 프로젝트 과정은 수동으로 측정 정보를 전달할 시간적인 여유가 없으므로 종합환경상에서 대부분의 프로세스 측정은 도구를 개입시켜 자동화하는 것이 바람직하다. 그림 5는 개발 프로세스와 워크플로우 관리를 추진하는 방법의 예를 표현한 것이다.

4.4 정보저장소

정보저장소는 소프트웨어 생명주기 동안 집적된 모든 시스템 정보가 관리되고 저장되는 곳으로 각 도구들, 개발단계들, 사용자들, 응용

프로그램들 간에 시스템 정보의 공유를 지원한다. 또한 정보저장소에 모든 시스템 요소 및 정보가 저장되므로 시스템의 유지보수를 용이하게 해 준다. 정보저장소는 재사용 가능한 소프트웨어 요소들의 목록과 이들에 대한 이해 및 접근방법을 함께 유지함으로써 소프트웨어의 재사용을 가능하게 해 준다. 정보저장소에 저장되는 정보는 다음과 같다.

- 논리적 자료 및 처리모형
- 물리적 정의와 구현코드
- 기업 모형
- 프로젝트 자료와 규칙
- 개체간의 관계
- 정보저장소 메타모델과 검증 규칙
- 프로젝트 관리와 감사정보

CASE 도구에서는 생명주기 각 단계를 지원하는 도구에서 생성된 정보를 저장하고 공유하는 역할을 효과적으로 지원할 수 있어야 한다. 현재 관계형 데이터베이스를 이용한 정보저장소 및 자료사전의 구현과 표준화 및 개방형 시스템에 대한 적용 및 지능형 정보저장소에 관한 연구가 활발히 진행되고 있다.

저장소는 개발의 중심이 되며 정보 시스템을 구성하는 복수의 데이터 모델을 저장한다. 구체적으로는 비즈니스 모델(프로세스, 데이터, 규칙(rule), 조직 등), 시스템 모델(설계명세, 정의정보, 프로그램 등), 컴퓨터 운용을 위한 운용관리 모델의 3종류이다. 그림 6은 최근에

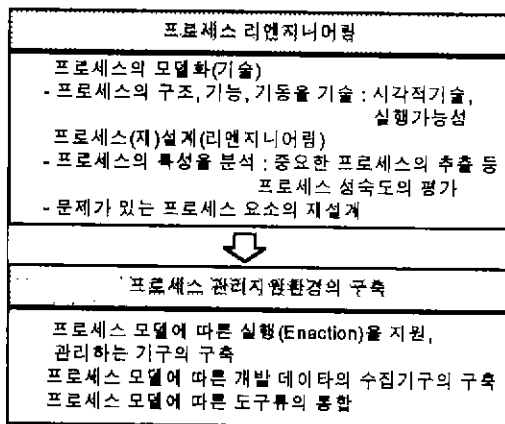


그림 5 개발 프로세스와 워크플로우의 관리추진

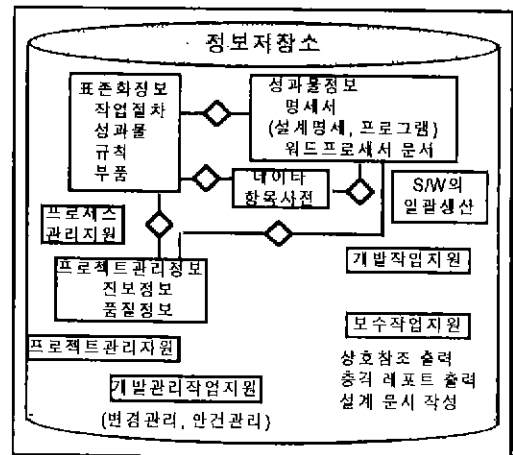


그림 6 정보자원의 통합관리

출시한 SEWB3의 정보저장소로서 정보자원의 통합관리를 추구하고 있다.

4.5 리엔지니어링

기존에 개발되어 있는 소프트웨어의 설계나 코드를 재사용하면 생산성과 품질면에서 큰 효과를 발휘할 수 있다. 그림 7과 같은 리엔지니어링은 기존의 소프트웨어를 변화된 환경에 적합하도록 변환하는 것으로 재사용면에서 활용 가치가 높다고 할 것이다. 리엔지니어링 분야에서 시스템의 변경과급 해석이나 프로그램의 표준화 기술 등은 이미 실용화되어 있지만 앞으로는 보다 상위 명세를 추출하여 비즈니스의 재구축이나 보수작업에서 이용하게 된다. 즉, 코드의 재사용뿐만 아니라 상위 수준의 재사용(요구사항, 분석, 설계 등의 재사용)이 촉진될 것이다. 추출한 비즈니스 규칙 등은 저장소에 등록되어 재이용된다. 그림 8은 기존 소프트웨어 표준화에 의한 보수와 재이용의 지원 예를 표현한 것이다.

4.6 개방형 CASE

소프트웨어 객체의 경계에서 그 객체가 가진 의미가 완전히 공표되어 있는 경우 이 객체를 개방형 객체라고 한다. 공표된 의미의 추상도에 계층을 설정할 필요가 있다(예를 들면, 통신 프로토콜에 대한 ISO 참조 모델처럼). 또, 의미영역의 정질에 의해서 개방도는 계속된 4영역 즉, 데이터의 의미, 제어의 의미, 표시의

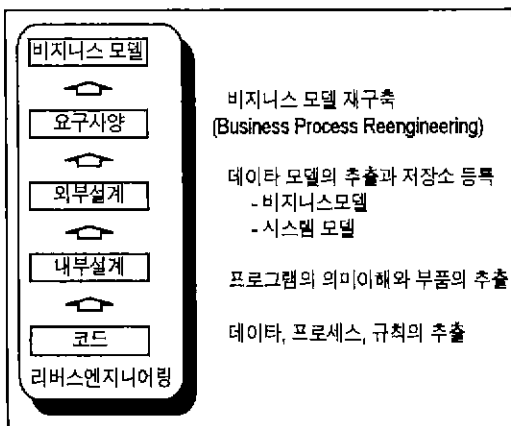


그림 7 리엔지니어링

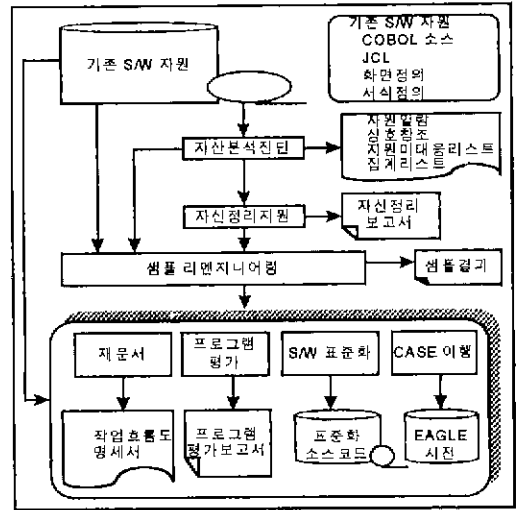


그림 8 S/W 표준화에 의한 보수와 재이용의 지원

의미, 프로세스의 의미로 나누어진다.

CASE 환경의 개방화에 대한 활동은 1980년대 전반부터 시작하고 있으며 같은 종류의 객체에 대해서 공통의 의미 영역을 정하는 것을 통합(integration)이라고 한다.

우선, 데이터 통합을 중심으로 시작되었으며 데이터 통합의 목적은 통합형 CASE 도구에서 각 도구들의 동작과 처리를 위해 일관성있는 정보를 제공하기 위한 것이다. 각 기능들을 지원하는 도구들이 처리하는 데이터가 일관성 없이 서로 다르게 저장되는 경우에 많은 오버헤드가 소요되고 또한 데이터의 일관성과 통합된 도구로서의 기능을 바르게 제공할 수 없다.

1990년대에 들어서는 제어통합이 중심으로 진행되고 있다. 유연성있는 기능 결합을 지원하기 위해 통합형 CASE 도구는 기능이 분할되며 분할된 도구들을 상호 동작할 수 있도록 제어해야 한다. 예를 들면 각 도구들의 처리를 위해 데이터를 요구하거나, 각 도구에서 데이터나 데이터의 참조를 전달하는 기능을 처리하기 위해서 제어가 통합되어야 한다. 이와 같은 것을 고려한 제어 통합은 데이터 통합을 보완하는 작용을 한다.

통합형 CASE 도구를 개발하기 위해서 예상되는 각 도구들의 프로세스 집합들을 통합하여 구현하는 데에는 많은 어려움이 있다. 이러한

통합형 CASE 도구를 개발할 때 고려해야 할 문제점으로 각 도구들에 관련된 동일의 프로세스 스텝의 통합을 고려하여야 한다. 표시통합은 OSF, X/Open 등, CASE와는 다른 국면에서 진행되고 있다.

또한, 데이터 통합은 그 성질상 데이터가 가진 의미의 크기(granularity) 및 의미영역의 종류에 의존하기 때문에 그 표준화에 대해서 ECMA-PCTE, ISO-IRDS, EIA-CDIF로 대표되는 3개의 흐름이 병행하고 있다. 첫째는 조립도, 둘째는 정밀도, 셋째는 의미영역의 종류 분류에 기초한 통합을 지향한다.

제어통합에는 도구간의 제어정보 관점을 기점으로 한 CASE Communique와 운영체제의 관점을 중시하는 CASE Interoperability Alliance의 2가지 흐름이 있고, ANSI-X3에서 융합이 도모되고 있다. 국제표준의 CASE 플랫폼을 기초로하여 다양한 CASE나 OA 도구 등 다른 분야의 도구를 통합화한다.

이상과 같이 소프트웨어공장의 입장에서 CASE의 현상과 차세대 CASE의 제문제를 고찰하였으나 CASE에 의한 소프트웨어 공장자동화를 성공시키기 위해서는 생산기술의 개발에 대해 기업 정책, 조직적 대응, 교육 투자, 모럴의 향상이 요구된다. 이것들은 현재에도 미래에도 밀접히 관련되어 지속적으로 발전하는 중요한 요소라고 생각한다.

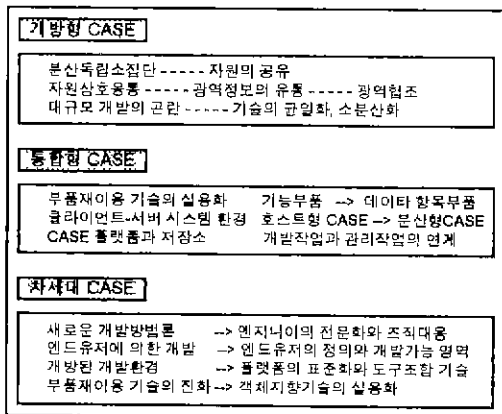


그림 9 개방형/통합형/차세대 CASE의 진행 방향

5. 결 론

CASE 도구의 개발은 소프트웨어의 생산성과 품질을 향상시키는 물론 개발에 소요되는 비용 및 노력을 현저하게 감소시킬 수 있는 방법으로 연구되어 왔고 이미 상용화되어 있다. 그러나 대부분의 CASE 도구들은 소프트웨어 생명주기의 일부만을 지원하거나 단계의 후반부만을 주로 지원하는 하위 CASE가 주류를 이루고 있는 실정이므로 소프트웨어의 생산성을 향상시키고 소프트웨어에 대한 수요와 신뢰성 및 품질에 대한 요구에 적절히 부응하기 위해서는 생명주기 전단계를 지원할 수 있는 통합형 CASE의 개발이 절실히 필요하다고 볼 수 있다. 통합 CASE 환경은 도구들간의 개발 정보를 교환할 수 있도록 개발되어야 하고, 새로운 방법론에 쉽게 적용할 수 있도록 개발되어야 하며 다른 도구들간에 이식성을 높여 차세대 CASE로의 전환이 용이하도록 해야 한다.

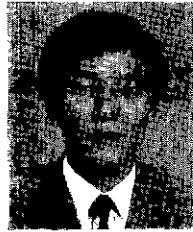
본 고에서는 CASE의 현상과 문제점을 살펴보고 CASE의 발전방향과 차세대 CASE의 제반 문제들을 살펴보았다. 소프트웨어 공학자들은 차세대 CASE를 개방형 시스템을 이용한 지능형 통합 CASE의 시대로 예측하고 있다. 이러한 발전된 차세대 CASE를 사용함으로써 개발 생산성을 향상시키는 것을 물론이고 유지보수성, 이식성, 사용자의 이해용이성, 호환성 등을 높이고 소프트웨어 개발 프로젝트에 대한 효율적인 통제를 할 수 있게 될 것이다.

참고문헌

- [1] D. Brumbaugh, "Object-Oriented DEvelopment : Building CASE Tools with C++", John Wiley & Son, Inc. 1994.
- [2] "A Tools Integration Standard", Digital Equipment Corporation, Feb, 1990.
- [3] M. Prabandham, W. Selfridge, D. Mann, "The Role of the IRDS", Database Programming and Design, April 1990.
- [4] ATIS, ANSI X3H4 Working Draft, Information Resource Dictionary System, Feb.

1990.

- [5] J. Rumbaugh, "Going with the flow : Flow graphs in their various manifestations", Journal of Object-Oriented Programming, June. 1994.
- [6] Alan Hecht and Matt Harris, "A CASE Standard Interchange Format : Proposed Extension to EDIF 200", Cadre Technologies, Inc., White Paper.
- [7] ANSI, "A Technical Overview of the IRDS system, Annex 1B, Draft", ANSI, 1985.
- [8] 本位田, "オブジェクト指向 システム開発", 日経BP出版センター, 1992.
- [9] 本位田, 小高, "知的ソフトウェア開発環境を目指す", コンピュータ科学, Vol. 2, No. 2, 1992.
- [10] 양해술, "객체지향 CASE 환경의 기초 기술", 한국정보과학회 정보과학회지 제9권 제2호, 1991. 4.
- [11] 양해술, "I-CASE의 도구 통합과 방법", 한국정보과학회 정보과학회지 제12권 제2호, 1994. 3.
- [12] 양해술, 이용근, 이하용, "객체지향 CASE의 기술 동향", 한국정보처리학회, [정보처리], 1996. 1.



양 해 술

- 1975 홍익대학교 공과대학 전기공학과 졸업(학사)
- 1878 성균관대학교 정보처리학과 정보처리 전공(석사)
- 1991 日本 오사카대학교 기초공학부 정보공학과 소프트웨어공학 전공(공학박사)
- 1975~79 육군중앙경리단 전자계산실 시스템분석장교 근무
- 1986~87 日本 오사카대학교 객원연구원

1993~94 한국정보과학회 학회지 편집부위원장
 1980~95.5 강원대학교 전자계산학과 교수
 1994~95 한국정보처리학회 논문지편집위원장
 1994~현재 한국산업표준원(INS) 이사
 1995~현재 한국소프트웨어품질연구소(INSQ) 소장
 관심분야: 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질감리, 품질컨설팅, OOA/OOD/OOP, CASE, SI), 소프트웨어 프로젝트관리



이 하 용

- 1993 강원대학교 전자계산학과 졸업(이학사)
- 1995 강원대학교 전자계산학과 소프트웨어공학 전공(이학석사)
- 1995~현재 한국소프트웨어품질연구소 전임연구원
- 1996~현재 경희대학교, 강원대학교 전산공학과 강사

관심분야: 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질감리, 객체지향 프로그래밍, 객체지향 분석과 설계, CASE)

● 제23회 정기총회 및 추계학술발표회 ●

- 행사일정 : 1996년 10월 25(금)~26일(토)
- 행사장소 : 한국외국어대학교(용인)
- 문 의 처 : 한국정보과학회 사무국

Tel. 02-588-9246, Fax. 02-521-1352

서울시 서초구 방배3동 984-1(머리재빌딩) ☎137-063