

확장 컴포넌트 개념에 의한 정보시스템 개발방법론

남기형*, 김선호**

The Methodology for Information System Development by the Concept of Extended Components

Ki-Hyung Nam, Sunn-Ho Kim

Abstract

This paper presents an efficient methodology for information system development by adopting the concept of extended components which can achieve the improvement of development productivity, reduction of development time periods, quality improvement of information systems, etc. when information development projects are carried out. First of all, for efficient development of information systems, development steps of current methodologies for information system analysis are subdivided into components in detail and outputs required at each step are defined. The components are extended by types and various application methods are proposed in each step of information system development. In addition, a method which uses advantages of current methods such as SDLC, prototyping, and RAD, is also proposed. In this research, a CASE tool named S-Designer is used for the development of extended components and their interface, and a system development method called DSDM for the deviation of detailed steps and corresponding outputs.

* 대우정보시스템(주)

** 명지대학교 산업공학과

1. 서론

기업들은 급변하는 경영환경과 정보기술의 발전속에서 기업 경쟁력을 확보하기 위해서 정보기술의 적극적인 도입과 활용을 모색하게 되었으며 BPR (Business Process Reengineering)의 경영혁신을 통해 전체 업무를 일원화하고, 각종 업무 단위의 효율성 증대 및 전사적인 자원에 대한 효과적인 운영과 관리를 모색하게 되었다. 또한 기업 경영 전략에 대한 효율성 극대화를 목적으로 전사적인 정보시스템 개발 도입을 추진하게 되었다. 이런 전사적인 정보시스템 개발 도입은 기존의 폐쇄적이고 중앙 집중적으로 운영되어오던 비 효율적인 업무 프로세스의 개선과 새로운 기술환경에 적합한 형태로의 기업체질 변형 및 정보화에 의한 기업 생산성 향상에 중점을 두고 있다.

이를 위하여 시스템 통합(System Integration : SI) 기술들이 적용되고 있으나 국내 여건상 아직까지는 이 기술이 성숙단계에 있지 못하고, 기술력 부족과 종합적인 문제 해결책 제시 능력의 부족 등으로 경영전략 컨설팅에서부터 전사적인 정보시스템 개발까지의 총체적인 서비스를 제공할 수 있는 업체가 극소수에 불과한 실정이다. 또한 개발단계별 요소기술의 개발이 부족하고 개발된 요소기술들의 활용 방법이 정립되어 있지 않다.

본 논문에서는 위와 같은 문제점들을 해결하기 위하여, 전사적인 정보시스템 개발 및 구축 프로젝트 추진시 개발생산성 향상, 개발기간 단축, 개발시스템 품질향상 등을 도모할 수 있도록 확장된 컴포넌트를 활용한 효율적인 정보시스템 개발방법론을 제시한다. 우선, 전사적인 정보시스템의 효율적인 개발을 위

해서 기존의 정보시스템 분석방법론의 개발 단계를 확장된 컴포넌트 개념으로 세분하여 재정의하고, 각 세부 단계별로 필요한 산출물들을 정의한다. 그리고 기존의 컴포넌트들을 유형별로 확장하여 정의하고 정보시스템 개발 단계별로 다양한 활용 방법을 제시한다. 기존 개발방법들의 장점들을 효과적으로 도입하는 방법도 제시한다.

본 연구에서는 확장된 컴포넌트의 개발 및 이들간의 인터페이스 해결방법을 제시하기 위해서 PowerSoft사의 CASE 도구인 S-Designer의 ProcessAnalyst, DataArchitect, AppModeler, Metaworks 모듈을 활용하여 분석, 설계, 개발하였다. 또한 정보시스템 개발방법의 세부적인 단계정의 및 산출물들을 정의하는데 D사의 정보시스템 개발방법론(DSDM)을 근간으로 정의하였다.

2. 개발방법론 연구 및 기술동향

2.1 정보기술

최근 정보기술의 발전 속도는 과거와는 달리 그 변화 속도가 급속도로 빠르게 진행되고 있으며, 정보기술의 발전과정은 인과 관계로 발전하여 현재의 정보기술 체계를 형성하였으며, 앞으로 더욱더 발전 속도가 가속화될 전망이다. 또한 정보기술은 전략수립기술, 조직 및 환경 정비기술, 시스템 접근기술, 시스템 요소기술, 네트워크기술, 시스템 구축기술, 시스템 구현기술, 시스템 평가 및 감사기술등으로 구성되어 체계화 되었다 [주헌 1996]. 정보기술의 컴포넌트 활용기술에는 소프트웨어 재사용, 제4세대 언어, 프로토타이핑, CASE 도구등이 있다.

소프트웨어 재사용이란 이미 개발되어 그 기능, 성능 및 품질을 인정받았던 소프트웨어의 전체 또는 일부분을 다시 사용하여 새롭게 개발되는 소프트웨어의 품질과 생산성 및 신뢰성을 높이고 개발기간과 비용을 감소시켜 주는 것을 의미한다 [McI 1967] [Fre 1987]. 한 통계에 따르면 개발된 소프트웨어의 50%만이 새로운 것이라고 밝혀졌고, 2000년대에는 85% - 90%가 이미 개발된 소프트웨어라고 예측되고 있다 [McC 1989].

제4세대 언어(fourth generation languages : 4GL)란 컴퓨터 프로그래밍에 대한 전문지식이 없더라도 원하는 업무를 처리하는 프로그램을 간단히 개발해 낼 수 있도록 해 주는 초고급언어(very high level language)이다. 또한 비절차언어(non-procedural language)이며 제3세대 언어(COBOL 등)에 비해 몇배 이상의 생산성을 갖는 응용소프트웨어 개발언어이고 최종사용자전산(end user computing)언어의 일면을 갖고 있다 [철현 1989].

프로토타이핑(prototyping)은 향후 개발 완성하고자 하는 시스템의 본질적 특성을 갖는 초기의 실험적 모형으로 소프트웨어 공학에서는 신속한 프로토타이핑(rapid prototyping)이라 한다. 또한 시스템이 갖추어야 할 기능들의 설명문구나 단순한 외적 모형보다는 개발자들과 사용자들과의 의사소통의 효과를 크게 증진시키는 동적시각모형(dynamic visual model)이다. 프로토타이핑의 가장 중요한 특성은 개발자와 사용자간의 상호이해 및 지식교환을 위한 의사소통의 도구라는 점이다 [충교 1986].

CASE(Computer Aided Software Engineering)는 소프트웨어 개발방식을 자동화함으로써 소프트웨어 개발 생산성 증대를 목적으로 하고 있으며, 초기의 단순한 분석이나

문서화 도구의 개념에서 소프트웨어 개발 전 과정에 걸쳐 자동화된 지원시스템의 의미로 확장되어 왔다 [경환 1991]. 아직까지는 완벽하게 모든 개발단계 즉, 분석에서 유지보수까지 지원하는 시스템을 찾기 어렵다.

2.2 J. Martin 정보공학 방법론

정보시스템의 계획과 구축시에 현업의 요구사항을 명확하게 정의하고 이를 시스템으로 구현하는 과정에서 개인의 경험이나 능력에 의존하는 부분을 줄이고, 최신 정보기술을 효과적으로 반영시킬 수 있음은 물론, 다양한 능력과 견해를 가진 인력을 조직적이고 효과적으로 참여 시키고자 하는 일관성있는 개발접근 방법론이다. 정보공학은 CASE 환경에 적절한 개발방법론의 결과가 가져온 결과물로서 C. Finkelstein 에 의해 고안되어 J. Martin에 의해 널리 보급되었다. J. Martin은 정보공학 방법론을 기업이나 기업의 핵심부분이 요구하는 정보시스템의 계획, 분석, 설계 및 구축에 필요한 정형화된 방법들의 응용으로 정의하고 있다 [Fin 1991] [Mar 1991]. 즉, 정보시스템 개발주기를 이용한 대형 프로젝트를 진행시키는 체계적인 방법론이다. 이 정보공학방법론의 특징은 다음과 같다.

- 계획단계, 분석단계, 설계단계, 구축단계의 4단계 개발주기 전략을 갖고 진행된다.
- 기업정보와 기업의 업무절차에 따라 시스템 데이터와 자동화 절차들이 모형화된다.
- 기업에 유통되고 있는 데이터와 흐름을 철저히 이해하고 모형화 한다.

시스템 구축과정은 {표-1}과 같이 데이터 측면과 활동측면으로 구분하여 진행한다.

[표-1] J. Martin의 정보시스템 구축과정

단 계	데이터 측면	활동 측면
정보전략수립계획	효과적인 기업경영의 정보전략	기업의 경영향상을 위한 기술활용 전략
업무영역분석	정규화된 데이터 모형	기업활동에 필요한 절차들의 상관관계
시스템설계	절차들에 의해 이용되는 데이터	업무수행을 위한 절차들의 설계
시스템구축	응용프로그램에서 보는 데이터	코드생성을 위한 상세 프로그램 논리

2.3 정보시스템 분석방법론

정보시스템을 구축하기 위해서는 시스템에 대한 기능 및 필요정보의 요구사항이나 데이터들의 처리방안이 수립되어야 하고, 이들의 분석을 위해서는 기능모델링, 프로세스 모델링 및 데이터모델링이 실시되어야 한다. 또한 이러한 분석방법들은 정보 데이터베이스의 구축 및 시스템 설계를 목표로 갖고 있으며 모델구축을 위해서 설계방법론에 의한 CASE 도구들의 지원을 받고 있다. 이들 방법론들의 특징은 다음과 같다.

- 주어진 문제에 대해서 여러 가지 시각을 제공하며, 업무시스템의 획득, 분석, 설계, 실험이 가능하며, 토론 및 의사소통을 위한 표준방식을 제공하여 합의를 가능하게 한다.
- 초보자라도 전문가의 역할을 가능하게 하며, 팀들의 활동을 통합한다.
- 코딩 또는 프로그램 전에 새로운 업무시스템의 설계를 지원한다.
- 문제해결을 위한 대안들의 평가가 가능하다.

2.3.1 IDEF 방법론

IDEF(Integration DEFINition)방법론은 1970년대에 미 공군에서 Manufacturing Productivity를 향상시키는데 목적을 두고 사용한

ICAM(Integration Computer - Aided Manufacturing) Program의 ICAM Definition에서 유래하였고, 미래형 공장의 구조를 정의하는데 기초가 되었다. 또한 Enterprise Integration을 위한 모델화 작업을 수행하는데 사용하여 시스템 통합에 참가하는 사람들의 의사소통을 원활히 하기 위한 목적으로 개발된 기법이다 [IDEF 1997]. 이 방법론은 실물을 추상화하여 모델을 작성하는 방법론의 군들로 구성되어 작성되며, 모델의 체계적인 분석을 가능하게 함은 물론, 문제점을 도출하여 미래모델의 설계를 가능하게 하는 동시공학(Concurrent Engineering)과 업무재설계(Business Reengineering Application)을 지원하는 방법론이다 [Gay 1991] [God 1989].

IDEF방법론의 장점은 다음과 같다.

- 시스템 분석에 필요한 기능(Function), 프로세스(Process), 정보(Information) 측면을 모두 제공하며, 하향식(Top-Down), 모듈화, 계층적 분석방법을 제공하여 구조적인 분석이 가능하며, 미 국방성(Department of Defense)의 표준 모델링 방법론이다
- CE, TQM, BPR을 지원하는 방법론중 하나이며, CALS의 문서표준을 수용하고 있다.
- IDEF3의 경우 워트니스로 시뮬레이션이 가능하므로 BPR검증이 가능한 도구이다. IDEF방법론의 단점은 다음과 같다.
- IDEFO, IDEFIX, IDEF3등 각 모델들간의

상관관계가 약하다.

- IDEFIX에서는 추상화 능력이 약하고 도메인 제약조건을 확인하기 어렵다.
- IDEF3에서는 하부 모델에서 반복적으로 발생하는 상황에 대한 추상화 능력이 부족하고, Decomposition시 $3 \leq D \leq 6$ 제약에 의한 계층과의 유연성이 부족하다.
- 기능의 그룹화 작업이 어렵고, 통합 View를 제공하지 못한다.

2.3.2 ARIS 방법론

ARIS(Architecture of Integrated Information Systems)방법론은 독일 IWI의 Scheer교수에 의해 정립되어 BPR에서부터 정보시스템 구현까지의 프로젝트 각 단계별 모델제공과 모델링 VIEW의 통합적인 프레임워크 제공 및 전체적인 관점에서의 비즈니스 프로세스 통합을 목적으로 IDS사에 의해 개발 되었다 [ARIS 1997].

ARIS 방법론은 조직, 데이터, 기능, 통제 의 4가지 View로 구성되어 통합적인 VIEW를 제공하며, 이들 각각의 VIEW를 표현하는데 ARIS Toolset의 지원을 받는다. ARIS Toolset은 미국의 전문IT 연구기관이 Gartner Group에 의해서 BPR 및 시스템 구축을 위한 효과적인 업무분석 도구로서 선도적인 위치를 점하고 있는 것으로 평가되고 있다 [Gan 1996]. ARIS 방법론의 장점을 3가지 측면에서 살펴보면 다음과 같다.

- 기술적 측면 : 도구사용 학습곡선이 짧은 편이어서 기본교육후 사용이 가능하고, ODBC를 통한 응용 S/W와 연계사용이 가능하며, RDBMS를 통하여 데이터의 일관성이 유지된다.
- 방법론 측면 : 원론적 BPR접근 보다는 검증된 레퍼런스 모델을 통한 단계적 접근

방법으로 위험을 최소화할 수 있고, 조직 View, 프로세스View를 통한 효과적인 프로세스 모델링이 가능하며, 시스템 설계시 가장 많은 시간과 노력이 필요한 데이터 모델링 작업을 효율적으로할 수 있고, 모델설계시 시간단축, 품질향상, 구현시 위험성 제거효과가 있다.

- 레퍼런스 측면 : 업종별 레퍼런스 모델제공 및 SAP R/3 레퍼런스 모델을 제공하며, BPR부터 시스템 개발까지 프로젝트 수행을 위한 프로시저 모델을 제공하고, 유럽, 미국 및 국내(삼성, 현대등)에서 성공적인 활용사례가 많으며, 최고의 컨설팅 회사와 전략적인 제휴관계를 맺고 있다.

ARIS 방법론의 단점은 다음과 같다.

- 모델링 방법론을 이해하는 데에는 많은 시간이 걸리며, ARIS 도구 Set의 구입 비용이 많이 소요되며, 한국 기업의 현실에 맞는 레퍼런스 모델이 없다.
- BPR 단계에서부터 정보시스템 구현까지 과정을 전문가가 아니면 큰 효과를 볼 수가 없다.

3. 확장된 컴포넌트 구성

3.1 확장된 컴포넌트의 개요

기존 컴포넌트의 개념을 확장하여 정의하고 확장된 컴포넌트들을 유형별로 구분한다. 또한 다양한 컴포넌트 기반 개발 시나리오 접근방법에 의한 정보시스템 개발의 효율적인 활용접근 방법을 제시한다.

3.1.1 컴포넌트의 정의

컴포넌트(Components)란 객체지향 개념

(캡슐화, 상속성, 다형성)을 지원하는 소프트웨어의 객체로 정의할 수 있다. 여기서 객체란 필요한 데이터 구조와 그 위에서 수행되는 함수들을 가진 하나의 소프트웨어 모듈로서 프로세스와 데이터 모두의 관점을 갖고 있다. 근래에 와서 이들 컴포넌트의 개념은 상속성과 다형성의 지원여부와는 별도로 상품화되어 있는 객체, 4세대 언어에서의 "객체" 또는 캡슐화된 이원객체, 마이크로소프트사의 OLE(Object Linking and Embedding), OCX(OLE Control Custom)등을 의미하기도 한다. 즉, 컴포넌트는 어플리케이션 시스템을 구축하는데 다른 컴포넌트와 결합되어 사용될 수 있는 캡슐화된 소프트웨어 객체이거나 이들 객체들을 구성할 수 있는 부분품들로 정의할 수 있다. 이런 컴포넌트들을 정의하는데 중요한 요소는 다음과 같다.

- 즉시 사용가능성 : 개발자 중심의 요소로 RAD개발 방법에 부합되는 요소
- 재사용 : 컴포넌트 개발 센터 중심의 요소로 재사용 컴포넌트를 생성하는 것은 매우 어렵고 시간이 많이 소요되므로 RAD 개발방법에 역행하는 요소

이들 요소들은 결과적으로 개발자들의 개발생산성과 시스템의 품질향상 및 개발기간의 단축효과를 극대화시켜 기업의 가장 중요한 두가지 요소인 비용절감과 경쟁력 향상의 효과를 창출할 수 있게 한다. 즉 컴포넌트의 품질을 결정하는데 있어서 가장 중요한 것은 위의 두가지 중요 요소에 앞서 그들이 시스템을 구축하는데 다른 컴포넌트들과 결합되어 사용될 수 있는가 하는 문제일 것이다.

3.1.2 컴포넌트의 확장

기존의 컴포넌트는 개발자들의 생산성 향상, 개발기간 단축 및 비용절감을 위해 이

미 개발되어져 그 기능과 성능 및 품질을 인정받은 소프트웨어의 전체나 일부분으로 활용되었다. 그러나 서브루틴 및 프로그램 패키지 등 작은 규모의 일반적인 코드 컴포넌트 수준으로는 높은 비용절감이나 생산성 효과를 얻기에는 한계성이 있다. 이에 정보시스템 개발에 필요한 모든 사항들 즉, 일반적 지식, 설계정보, 데이터정보, 텍스트 문서등으로 컴포넌트의 개념을 확장 정의하여 정보시스템 개발 단계별로 활용함으로써 정보시스템 개발에서의 생산성 및 품질향상, 개발기간 단축 효과를 얻으려는데 그 목적이 있다. 즉, 확장된 컴포넌트(extended componemts)란 기존의 컴포넌트 개념을 프로그램 코드수준에서 정보시스템 개발에 필요한 모든 자원으로 확대하여 정의하는 것을 의미한다. 현재의 많은 시스템통합 전문업체들은 다양한 기업들의 각종 업무들을 체계적으로 객체화 및 컴포넌트화하는 방법과 기술 개발에 많은 비용과 시간을 투자하고 있다. 이것은 확장 컴포넌트들을 블럭처럼 조합함으로써 다양한 고객의 요구사항에 부합되는 전사적인 정보시스템을 신속하게 구축할 수 있음은 물론, 품질 향상의 효과를 극대화할 수 있는 효과를 얻기 위해서이다. 이처럼 컴포넌트 기반 개발은 시스템통합 전문업체의 중요한 기술로 간주되고 있다.

3.1.3 컴포넌트 활용도구

클라이언트/서버 환경의 정보시스템 개발에서는 적절한 분석 및 설계, 신뢰성 있는 데이터베이스 구축, 모델기반의 개발 등이 중요한 사항이다. 이들을 효율적으로 관리하고 고품질의 어플리케이션을 개발하기 위해서는 통합 CASE 도구의 선정 및 활용이 무엇보다도 중요하다. 본 논문에서는 확장된 컴포넌트

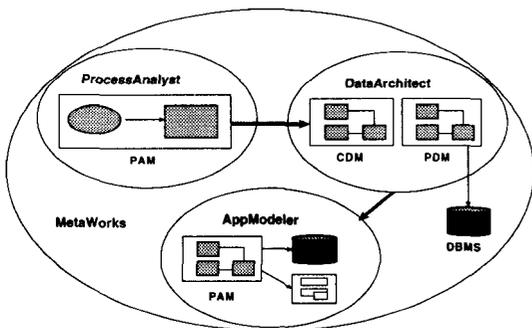
의 설계를 위해서 PowerSoft사의 CASE 도구인 S-Designer Toolset을 적용하였으며, 일부 확장된 컴포넌트의 개발을 위해서는 PowerSoft사의 4GL 개발도구인 PowerBuilder를 사용하였다.

CASE 도구인 S-Designer를 도입하여 활용한 이유는 국내의 전사적인 정보시스템 개발 사례를 검토한 결과 PowerSoft사의 4GL 개발도구인 PowerBuilder로 개발한 사례가 많아 개발도구와의 연계된 통합 활용측면에서 가장 큰 효과가 있기 때문이다.

S-Designer는 복잡한 클라이언트/서버 환경하에서의 데이터베이스 및 어플리케이션 개발을 위하여 필수적으로 요구되는 프로세스, 데이터 분석 및 모델링, 문서화, 프로토타이핑 개발을 가능하게 하는 통합된 모델링 도구이다. 또한 완전한 모듈 구조로 구성되어 있으며 각 모듈끼리는 데이터가 유기적으로 결합되므로 중복작업이 전혀 필요없다.

S-Designer의 구성은 [그림-1]과 같이 구성되며, 구성된 4개의 모듈의 기능을 요약하면 다음과 같다.

- ProcessAnalyst : 여러 업무에 내재된 모든



[그림-1] S-Designer의 구성

요건들을 발견하기 위해서 그 업무의 프로세스를 분석하여 모델을 설계하는 모듈이다. 또한 설계자가 프로세스와 데이터의 흐름을 묘사하고 전체구조를 파악하여 모델을 생성 및 유지 하기가 쉬우며, 개발 참가자들의 의사소통을 원활하게 하여준다.

- OMT's functional modeling과 전통적인 Data Flow Diagram 표기법제공.
- DataArchitect 모듈과 상호 연결되어 분석과 설계작업의 일관성 지원.
- 프로세스의 분해기능(Decomposition)제공.
- DataArchitect : 데이터베이스의 설계, 생성, 유지보수, Reverse-Engineering, 문서화 등을 포함한 데이터모델링을 위한 모듈이다. 또한 어플리케이션 데이터베이스 엔진을 통해 유연하고 효율적인 데이터 구조를 생성할 수 있다.
- Conceptual Data Model 및 Physical Data Model에 의한 여러 DBMS의 지원.
- 역공학을 통한 데이터베이스의 구조파악 및 통합 프로토타이핑 생성지원.
- AppModeler : 개발도구와 연계하여 개발을 가속화 시키고 어플리케이션의 질을 향상하는데 사용되는 모듈이다.
- PDM을 사용하여 4GL의 Object를 생성하여 프로토타이핑 개발지원.
- 30개 이상의 DBMS에 대하여 데이터베이스 생성.
- 기존의 데이터베이스를 역공학하여 데이터베이스 모델 생성.
- MetaWorks : 전체 프로젝트의 일관성과 안정성을 위하여 모든 모델링에 사용되는 오브젝트들을 관리하는데 사용되는 모듈이다.
- 사용자 정의, 접근 권한의 기능제공 및

- 유연한 디렉토리 브라우저 제공.
- 프로젝트의 모든 모델링 정보제공.

3.2 확장된 컴포넌트 유형

컴포넌트의 의미를 정보시스템 구축에 필요한 모든 객체, 설계명세서, 서비스등으로 확장하여 정의할 수 있으며, 이들 확장된 컴포넌트들은 어플리케이션 컴포넌트, CASE 모델 컴포넌트, 프레임워크 컴포넌트, 기능 컴포넌트 및 표준화 컴포넌트의 5가지 유형으로 구분할 수 있다.

3.2.1 어플리케이션 컴포넌트

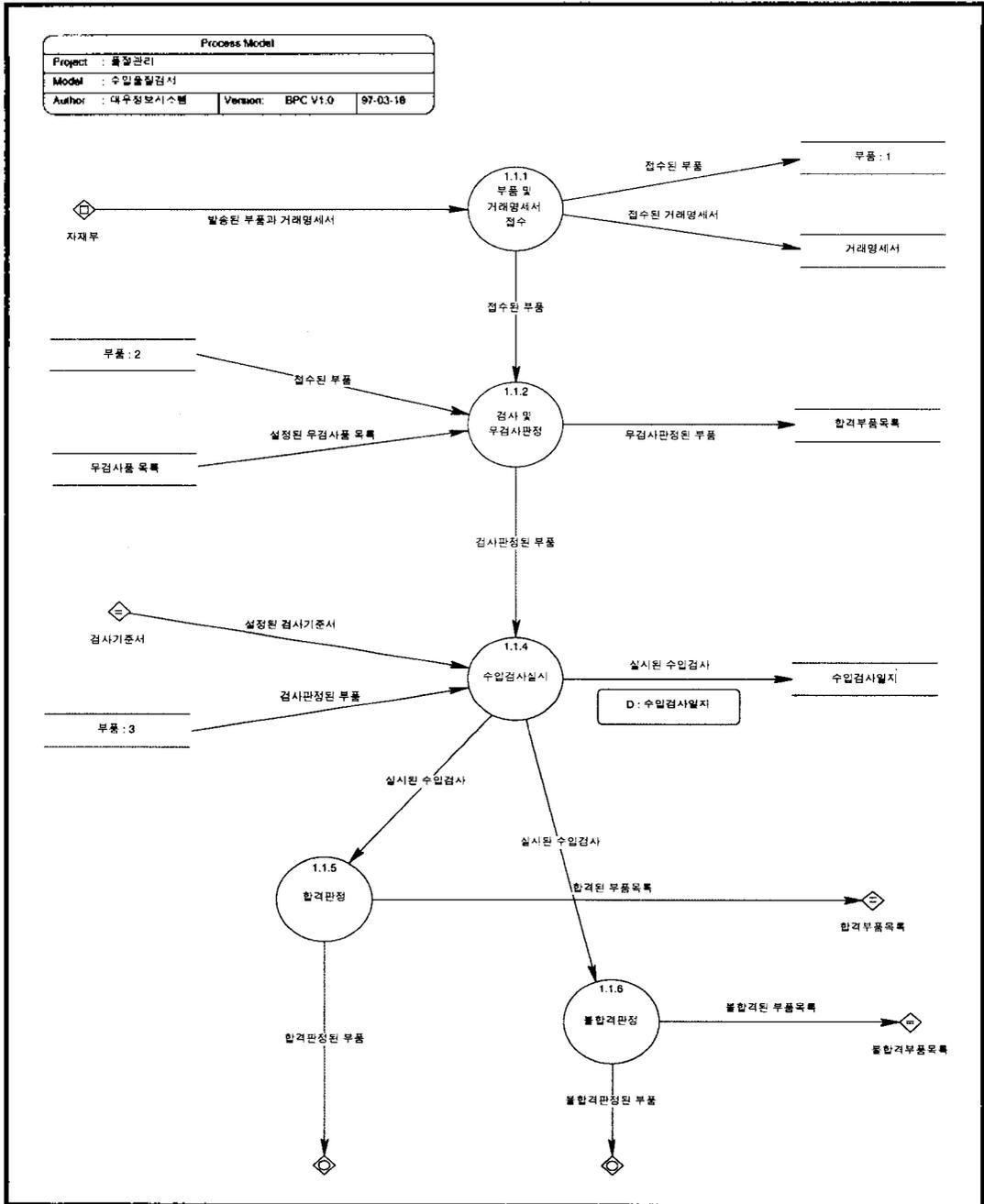
어플리케이션 컴포넌트는 정보시스템 개발전문 업체에 의해 특정 목적에 맞게 개발되어 그 품질과 기능이 검증된 소프트웨어 제품을 의미한다. 또한 이것은 그 규모에 따라 작게는 단위 어플리케이션에서부터 크게는 ERP(Enterprise Resource Planning)어플리케이션까지 다양하게 구성되어 있다. 본 논문에서는 ERP 어플리케이션의 정보구조및 기능구조를 어플리케이션 컴포넌트의 구성요소로 정의하여 정보시스템의 정보구조와 기능구조 설계에 활용하고자 한다.

어플리케이션 컴포넌트의 장점은 이미 검증된 정보구조 및 정보기능을 활용하여 개발생산성을 높일 수 있는 기초를 마련할 수 있다. 이해 반해 단점은 ERP 어플리케이션을 모듈로 구입하여 컴포넌트로 활용하는 경우 구입비용이 과다하게 지출되고, 재사용 가능성이 매우 낮기 때문에 고객의 요구사항을 충분히 충족시킬 수 있는 정보시스템을 구축하기에는 어렵다.

3.2.2 CASE 모델 컴포넌트

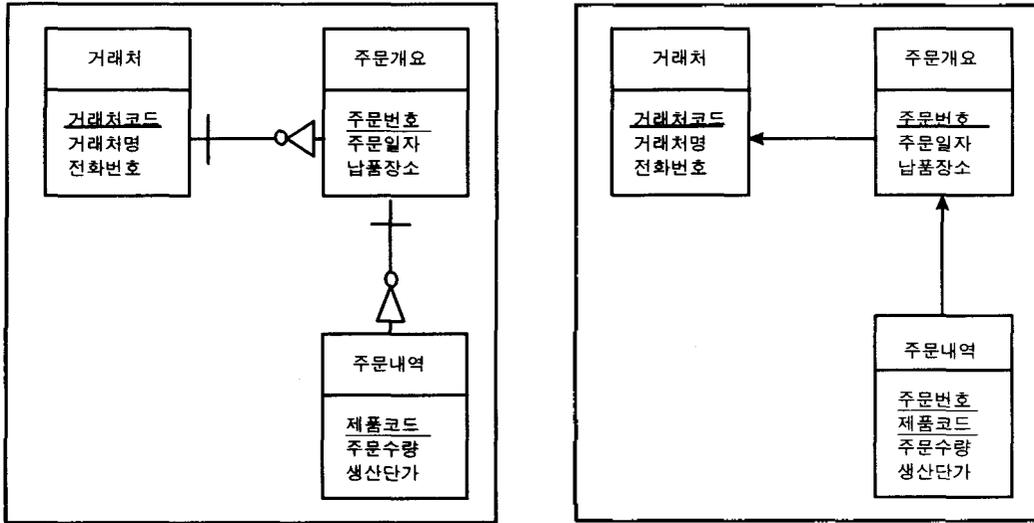
CASE 모델 컴포넌트는 정보시스템 개발을 위해 반드시 필요한 업무프로세스 및 데이터 모델 설계를 정의한다. 프로세스 모델 컴포넌트는 현업무를 분석 설계한 AS-IS 모델과, 현업무를 개선하여 설계한 TO-BE 모델 및 업무개선 모델중 정보시스템으로 개발하기 위한 프로세스와 데이터의 관계를 설계한 DFD 모델로 구성된다. 또한 데이터 모델 컴포넌트에는 개념적인 데이터 모델과 물리적인 데이터 모델로 구성된다.

[그림-2]는 CASE도구인 S-Designer의 ProcessAnalyst 모듈로 AS-IS 프로세스 모델을 설계한 사례이다.



[그림-2] ProcessAnalyst 모듈을 이용한 설계모델 사례

[그림-3]은 CASE도구인 S-Designer의 DataArchitect 모듈로 논리적인 데이터 모델과 물리적인 데이터 모델을 설계한 사례이다.



[그림-3] DataArchitect 모듈을 이용한 설계모델 사례

CASE 모델 컴포넌트의 장점은 고객의 요구사항을 충분히 반영한 업무프로세스 및 데이터 모델설계를 근간으로 정보시스템을 구축하여 고객만족을 극대화할 수 있으며, 많은 설계 경험의 축적으로 다양한 업종에 대한 레퍼런스 모델을 생성할 수 있다. 이해 반해 단점은 설계 작업에 많은 경험과 숙련된 CASE 도구의 활용이 요구되기 때문에 모델 작성시간이 많이 소요되고, 효율적인 모델 설계를 위해서는 프로세스와 데이터의 연계성을 고려해야 한다. 그래서 단기간의 정보시스템 개발에는 효율성이 매우 떨어지고 모델 설계 작업에 숙련된 전문성이 요구된다.

3.2.3 프레임워크 컴포넌트

프레임워크 컴포넌트는 다른 컴포넌트 자원들과 연계하여 컴포넌트의 생성 및 통합

을 원활히 수행하기 위한 개발환경을 제공한다. 이들 대부분은 흐름통제를 다루는 형식화된 소프트웨어 모듈로 설계된다. 프레임워크 컴포넌트는 프로세스와 연계하여 데이터의 흐름을 통제하기 위한 데이터 풀 컴포넌트와, Naming Rule의 검증과 전체적인 프로세스에 대한 데이터의 흐름을 검증하기 위한 자원검증 컴포넌트로 구성된다. 데이터 풀 컴포넌트는 개략적인 데이터 풀 컴포넌트와 상세적인 데이터 풀 컴포넌트로 구성되며, 자원검증 컴포넌트는 데이터 사전 컴포넌트와 프로토타이핑 컴포넌트들로 구성된다.

프레임워크 컴포넌트의 장점은 프로세스 및 데이터의 흐름통제 및 검증을 통하여 개발시스템의 품질을 향상시켜주며, 초기에 분석, 파악된 정보의 유실을 최소화하여 개발기간을 단축하고 생산성을 향상시킬 수 있다.

이해 반해 단점은 다른 컴포넌트간의 인터페이스에 어려움이 있으며, 정형화된 활용방법 없이는 실질적으로 정보시스템에 활용하기에는 오류가 발생할 가능성이 많으며, 인터페이스의 해결을 위해서는 목적에 부합되는 소프트웨어 모듈을 개발해야하는 어려움이 있다.

3.2.4 기능 컴포넌트

기능 컴포넌트는 정보시스템 개발사 개발도구인 4GL의 한계성을 극복하기 위한 하나의 방법으로 기능 및 활용면에서 이미 검증된 캡슐화된 소프트웨어 모듈인 기능OCX 컴포넌트와 이미 설계되어 검증된 업무로직 알고리즘으로 구성된다.

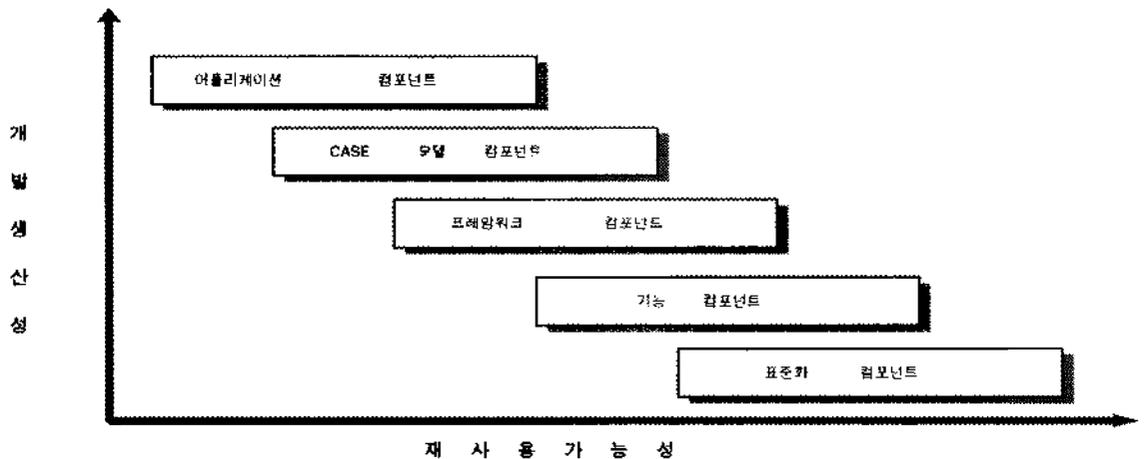
기능 컴포넌트의 장점은 재사용가능성이 매우 높기 때문에 4GL 도구의 한계성을 극복

할 수 있는 전문적인 소프트웨어 활용 및 개발기술이 필요하다.

3.2.5 표준화 컴포넌트

표준화 컴포넌트는 Naming Rule 컴포넌트, 화면설계 컴포넌트, 스크립터 기술 컴포넌트, 라이브러리 컴포넌트로 구성된다.

표준화 컴포넌트의 장점은 정보시스템 개발의 공공작업에서 발생하는 많은 문제점들을 제거할 수 있으며, 개발자간의 의사소통을 원활하게 하고, 향후 유지보수에 소요되는 시간 및 비용을 절감할 수 있다. 이해 반해 단점은 표준화의 질적 향상을 위해서 많은 시간과 갱신의 부가적인 작업이 필요하며, 정보기술의 발전에 신속하게 대응하기 위한 지속적인 노력이 필요하다.



[그림-4] 컴포넌트의 유형별 생산성과 재사용성

할 수 있으며, 다양한 업무로직 레퍼런스들을 생성하여 활용할 수 있다. 이해 반해 단점은 다른 컴포넌트에 비해 개발생산성이 낮으며, OCX 컴포넌트의 활용을 위해서는 인터페이스와 적용에 따른 문제점들을 고려하여 해결

컴포넌트의 유형별 특성을 보면, [그림-4]와 같이 규모가 클수록 생산성이 높은 반면에 재사용성은 낮다. 이에 반해 규모가 작을수록 재사용성은 높으나 생산성은 떨어진다.

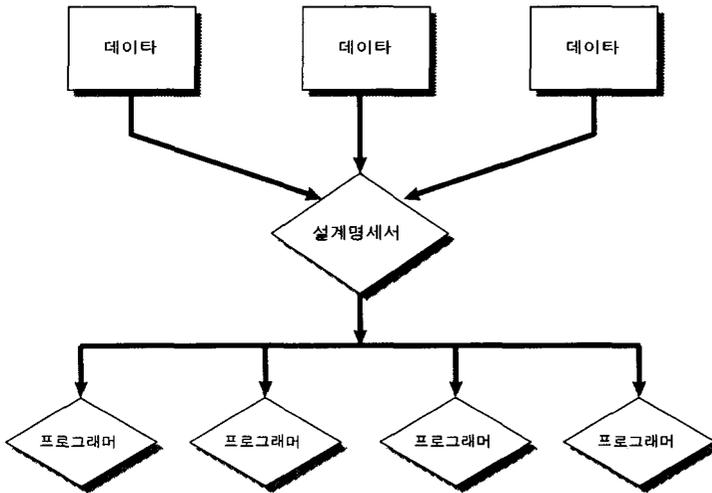
3.3 컴포넌트 기반 개발 시나리오

일반적인 정보시스템 개발은 비 컴포넌트 기반 개발과 컴포넌트 기반 개발로 구분되어지며, 컴포넌트 기반 개발은 컴포넌트의 확장을 통하여 확장된 컴포넌트들을 근간으로 다양한 개발 시나리오에 의한 정보시스템 개발 접근이 가능하다.

확장된 컴포넌트 개발 접근 시나리오들의 특성 및 장점들을 파악하여 정보시스템 개발에서의 효율적인 활용 접근방법을 제시할 수 있다.

3.3.1 비 컴포넌트 기반 개발

비 컴포넌트 기반 개발은 [그림-5]와 같이 분석에 있어서는 프로세스와 데이터가 분리되어 설계된다. 따라서 분석단계의 설계명



[그림-5] 비 컴포넌트 기반 개발

세서는 따로 진행된 사용자 인터페이스 요구 사항과 설계된 프로세스와 데이터 구조가 혼합되어 작성된다. 설계명세서가 작성되면 메

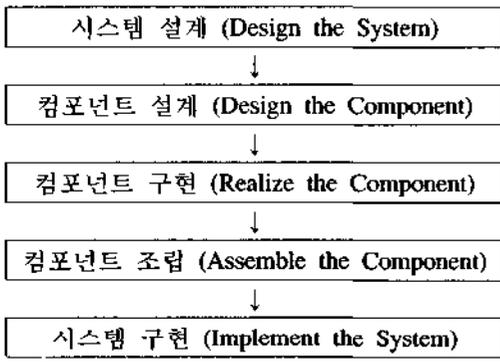
뉴 아이템별 또는 사용자 인터페이스 단위로 프로그래머들에게 프로그램 작업량이 할당되어 개발작업이 진행된다.

비 컴포넌트 기반 개발의 단점은 코딩 과정시 표준이 존재하지 않으면 중복되거나 일관성이 없이 구현되므로 통합 장애등이 초래될 가능성이 높다. 또한, 코드 생성은 프로그래머의 특성에 맞게 되므로 향후 프로그램 유지 및 코드 재사용이 매우 어려워지기 때문에 개발생산성 향상과 개발기간 단축의 효과를 기대할 수 없다.

3.3.2 컴포넌트 기반 개발

컴포넌트 기반 개발은 작은 애플리케이션 또는 컴포넌트들을 조립하여 정보시스템을 구축하는 것이다. 이와 같은 컴포넌트들은 개발 도구에서 제공되거나 개발자가 제3자로 부터 구매하거나 또는 개발자가 직접 생성하는 세가지 방법으로 구성된다. 컴포넌트 기반 설계명세서는 특정 개발시스템과는 독립적인 컴포넌트의 범주안에서 기초하여 작성된다. 따라서 특정 시스템의 설계는 이러한 컴포넌트의 통합을 위한 설계명세서에 기초하여 설계되며, 시스템의 개발은 기존의 생성된 컴포넌트의 통합으로 구현된다. 이러한 컴포넌트들이 [그림-6]과 같이 기능적인 범주안에서 전문가에 의해 설계되고 구현되므로, 다른 어플리케이션에 재사용이 가능하게 되어 개발기간의 단축 및 생산성 효과를 얻을 수 있게 된다.

컴포넌트 기반 개발의 단점은 기존의 코드 수준의 컴포넌트 기반으로는 전사적인 정보시스템 개발에서 높은 생산성 및 개발기간 단축의 효과를 기대하기 어렵다는 것이다. 또한 정보시스템 개발단계별로 다양한 컴포넌트들이 구성되어 있지 않기 때문에 단계별로 컴포넌트의 재사용성이 떨어진다.



[그림-6] 컴포넌트 기반 개발

3.3.3 확장된 컴포넌트 기반 개발

가. 확장된 컴포넌트 기반 개발

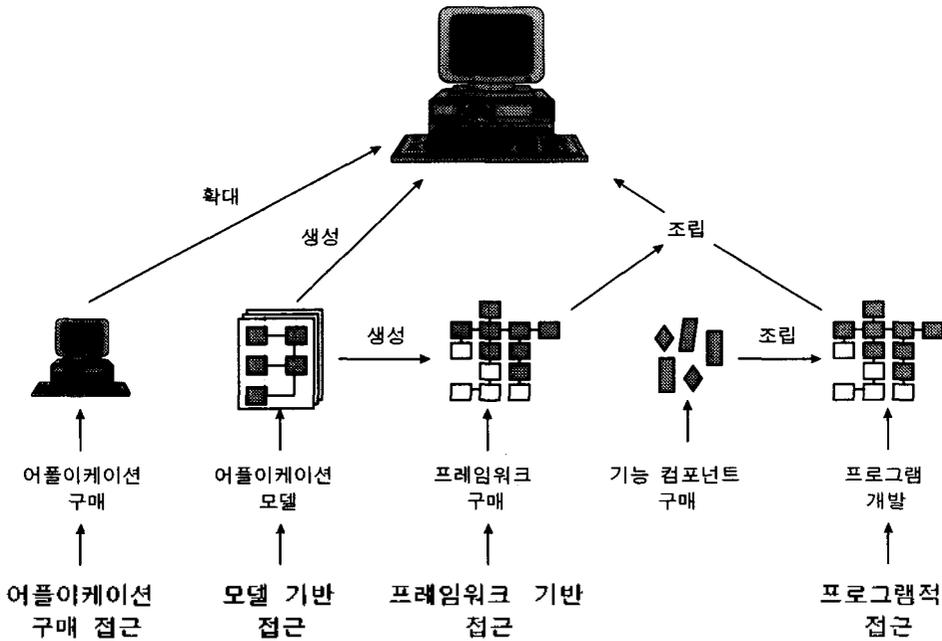
확장된 컴포넌트 기반 개발은 앞에서 정의된 확장된 컴포넌트를 기초하여 일반적으로 다음 쪽의 [그림-7]와 같이 구분할 수 있다. 이들의 개발 시나리오별 특성은 [표-2]와 같다. 이 중에서 효율적인 정보시스템 개발 시나리오는 고객의 만족도 증대, 시스템 품질의 향상, 큰 비용효과, 다양한 확장된 컴포넌트 활용 및 시스템의 유연성을 높일 수 있는 모델 기반의 시스템 개발 시나리오이다. 그러나 모델 기반 개발 시나리오는 모델 설계의 복잡성, 숙련된 전문성의 요구 및 개발단계별 확장된 컴포넌트의 활용방법이 정립되지 않고서는 위와 같은 효과를 얻기 어렵다. 또한 개발 기간이 많이 소요된다.

[표-2] 확장된 컴포넌트 기반 개발 시나리오별 특성

시나리오	프로그램 기반	프레임워크 기반	모델 기반	구매 기반
접근방법	상향식 (Bottom Up)	상향식 (Bottom Up)	하향식 (Top Down)	하향식 (Top Down)
활용 컴포넌트	기능, 표준화	프레임워크, 기능, 표준화	모델, 프레임워크, 기능, 표준화	어플리케이션
시스템 규모	소규모	중규모	대규모	대규모
개발패러다임	프로그래밍/조립	생성/조립	모델링/생성	구매/확대
저해요인	제한된 활용 제한된 분석	개발자 지향 제한된 분석	복잡함 전문성 요구	공급사 종속 적은 유연성
개발기간	중간	짧다	길다	짧다
비용효과	크다	중간	크다	적다
고객요구 만족도	상	중	상	하
시스템 품질	하	중	상	상
유연성	크다	중간	크다	작다

모델 기반 개발 시나리오의 효율성을 극대화 하기 위해서는 다음과 같은 사항들이 필요하다.

- 모델설계의 복잡성과 전문성을 줄일수 있는 CASE모델 컴포넌트의 활용방법 제시
- 확장된 컴포넌트 접근 측면에서의 정보시스템 개발단계의 세부적인 단계 정의
- 세부적인 개발 단계별 확장된 컴포넌트의 활용방법 제시



[그림-7] 확장된 컴포넌트 기반 개발 시나리오

4. 효율적인 정보시스템 개발방법

4.1 기존 개발방법의 장점도입

현재 사용되고 있는 개발방법은 프로젝트의 형태에 따라 SDLC(Software Development Life Cycle), 프로토타이핑(Prototyping), RAD (Rapid Application Development) 개발방법으로 구분되어 적용되고 있다. 이런 방법들은 점차 전통적인 정보공학 개발방법을 근간으로 나머지 방법들의 장점을 수용하는 형태로 발전하고 있으며 전체적인 통합구조에 기반을 두면서 개발과정을 신속하게 하기위해 통합 CASE의 활용을 필수로 하면서 복합적인 정보시스템 개발방법으로 정립되어 가고 있다.

SDLC는 정보시스템의 개발절차 또는 개발단계의 반복현상으로 시스템 개발주기로

정의된다. 프로토타이핑은 향후 완성하고자 하는 시스템의 본질적인 특성을 갖는 초기의 부분적 모형으로 정의된다. 프로토타이핑 개발방법은 품질이 좋은 정보시스템을 신속하게 개발하기 위한 실제적인 방법으로 프로젝트 중심이라기 보다는 결과 중심적이며 사용자 중심적이다. RAD는 제한된 범위의 단독시스템을 CASE의 활용으로 신속하게 개발하는 방법으로 개발대상 시스템의 독립적인 기술 설계가 가능한 경우에만 적용할 수 있다.

이 개발방법들에서 도입한 장점은 다음 쪽의 [표-3]과 같다.

4.2 정보공학 개발방법의 단계정의

정보시스템 구축을 위해 데이터, 절차, 기술 및 지원적인 측면에서 접근방법을 정의

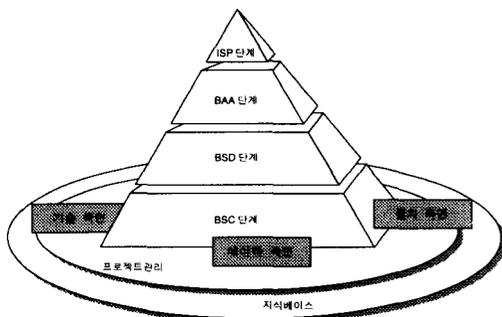
[표-3] 도입된 기존 개발방법의 장점

구분	SDLC 개발방법	RAD 개발방법	프로토타입 개발방법
도입 장점	<ul style="list-style-type: none"> - 단계별 승인과검증 - 개발단계별 산출물 정의 - 오류검출시 재귀성 개발주기에 의한 개발 단계정의 - 폭포수 모형 구조 	<ul style="list-style-type: none"> - 공동요구정의 (JRP) - 개발기간 단축 - 우선순위 및 범위 선정 - 사용자공동설계 (JAD) - 기존 시스템 재사용 	<ul style="list-style-type: none"> - 개발 생산성 향상 - 시스템 품질 향상 - 개발모형 제시 - 유지/보수비용 절감 - 변경 요구시 부작용 최소화

하고, J. Martin의 정보공학 개발방법을 근간으로 실질적인 개발프로젝트 기간(국내 : 평균 6개월 - 1년)에 부합되고 확장된 컴포넌트의 접근 측면에서 정보시스템 구축단계를 재정의한다. 또한 각 추진 단계별로 세부추진 단계의 정의 및 산출물을 정의함으로써 정보시스템 개발 프로젝트의 단계별 진행과정을 보다 생산적이고 효율적인 접근이 가능하도록 한다.

4.2.1 정보시스템 구축의 접근방법

정보시스템에 대한 개발단계별 접근측면



[그림-8] 정보시스템의 구축방법 구성

과 지원측면은 [그림-8]과 같다. 즉, 정보시스템은 데이터, 절차, 기술의 측면에서 접근하

고 개발단계는 계획, 분석, 설계, 구축으로 분류된다. 이 모든 과정을 위해 프로젝트관리와 지식베이스가 존재한다. 여기서의 지식베이스는 통합케이스의 정보저장소가 된다.

4.2.2 정보전략 계획수립단계(ISP)

정보전략 계획수립(Information Strategy Planning : ISP)단계는 기업의 사업목표 및 주요성공요인을 달성하기 위한 전략적 정보시스템 구축계획을 수립하는 단계로 기업의 최고경영목표 및 주요성공요인을 결정하고 기업에 필요한 정보 및 기업모형을 정의한다. 여기서 기업모형이란 기업전체를 목표, 주요성공요인, 조직이라는 전략적인 측면과 데이터, 절차, 기술이라는 시스템적인 측면에서 모형화한 것이다. ISP단계에서 정의된 기업모델은 조직모델(Organization Model), 전략모델 (Strategy Model), 업무절차모델(Process Model), 정보모델(Data Model)로 구성된다.

4.2.3 업무영역 분석단계(BAA)

업무영역분석(Business Area Analysis : BAA)단계는 기업 전체의 관점에서 벗어나 기업의 특정업무 영역을 분석하는 단계로 이

는 주어진 업무영역별로 상세한 분석을 통하여 정보시스템의 물리적인 구조에 의존하지 않는 독립된 논리적인 모형을 구축하는 단계이다. 또한 시스템에 대한 사용자의 요구사항을 분석 및 공동참여를 통하여 시스템에 대한 요구를 명확하게 정의하여 이에 대한 프로세스 모델, 기능 모델, 논리적인 데이터 모델을 작성하여 향후 물리적인 시스템 설계 기초를 형성하며, 패키지 구입계획을 입안한다.

4.2.4 업무시스템 설계단계(BSD)

업무시스템 설계(Business System Design : BSD)단계는 논리적인 시스템 설계를 물리적인 시스템 설계로 전환시키는 단계로 사용자의 안목에서 바라본 시스템 설계 즉, 화면 설계, 출력보고서 설계, 물리적인 데이터베이스 설계와 시스템 내부의 논리적인 기술설계 즉, 모듈설계, 업무로직설계가 중심이 된다. 이들을 설계할 때에는 사용자와 공동작업을 유도하며, 프로토타이핑 기법을 사용하여 향후 유지보수의 문제점을 최소화 함은 물론 개발기간 단축의 효과를 얻는다. 또한 정보시스템의 업무체계를 설계하는 것도 포함함은 물론 시스템의 시험계획 수립 및 시스템 구축계획을 입안한다.

4.2.5 업무시스템 구축단계(BSC)

업무시스템 구축(Business System Construction : BSC)단계는 확정된 설계명세서로부터 데이터베이스 생성기와 프로그램코드 생성기를 이용하여 데이터베이스와 실행 가능한 프로그램 코드를 생성하는 단계로 통합시험 및 운영교육, 매뉴얼 작성 등을 포함하고 있으며 향후 유지보수 계획수립을 입안한다.

정보공학 개발방법의 단계별 작업절차를 요약하면 다음 쪽의 [그림-9]와 같다.

4.3 확장된 컴포넌트 활용방법

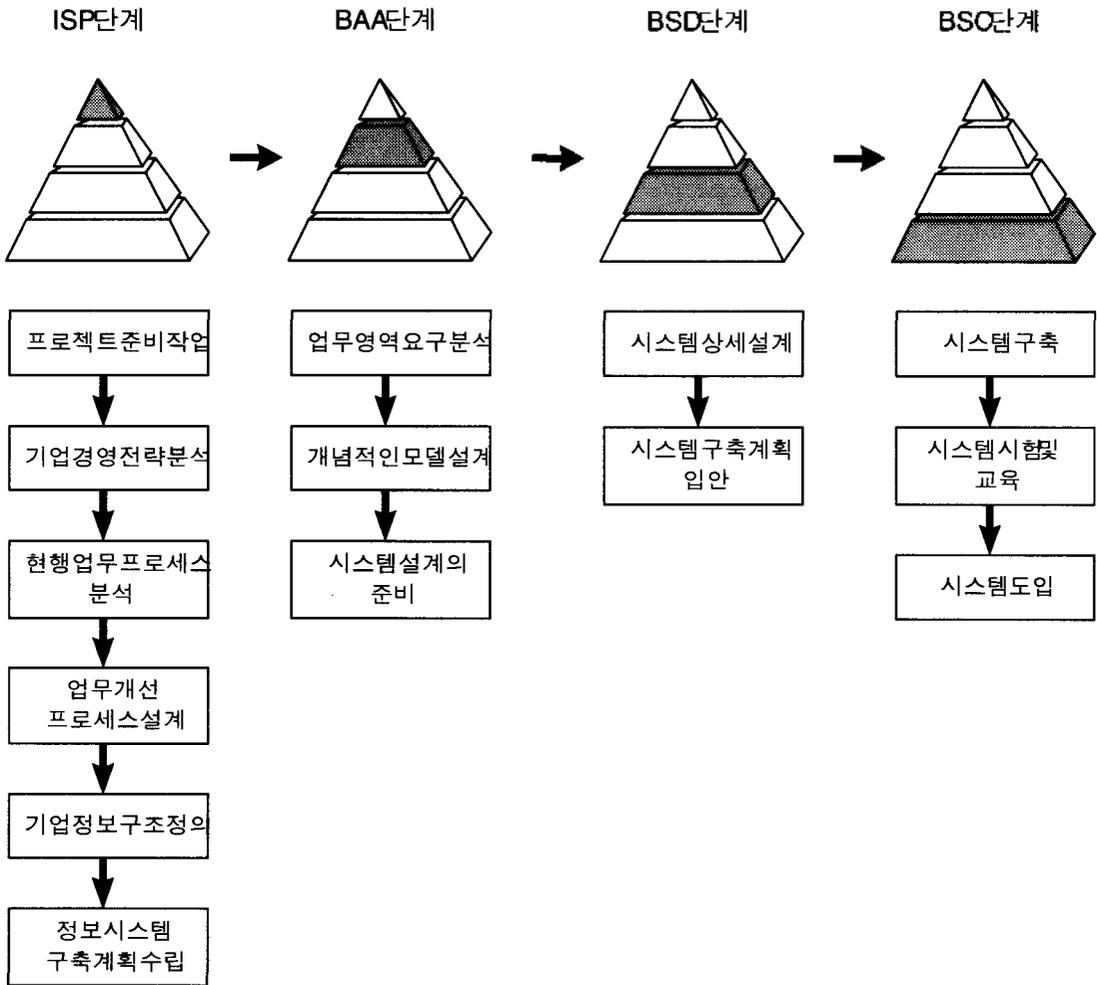
정보공학 개발방법의 단계별로 확장된 컴포넌트를 프로세스 측면, 데이터 측면, 기능 측면으로 구분하여 활용함으로써 정보시스템 개발을 위한 컴포넌트의 활용 효과를 극대화시킬 수 있으며, 또한 각 단계간의 인터페이스를 CASE도구를 이용하여 계획단계에서 생성된 자료를 실질적인 개발단계까지 활용할 수 있도록 하는 방법을 제시한다. 확장된 컴포넌트의 효과적인 활용은 정보시스템 개발의 생산성 및 품질향상, 개발기간 단축의 효과를 극대화할 수 있다.

4.3.1 프로세스 측면의 활용방법

프로세스 측면에서의 컴포넌트 활용방법을 정보공학 개발방법의 단계별로 정의함으로써 계획단계에서부터 구축단계에 이르기까지의 업무절차중 기업의 목표달성에 핵심적인 역할을 수행하는 부분들을 시스템의 논리적인 절차로 전환하는데 목적이 있다. 또한 정보시스템 개발과정중 분석단계에서 설계단계로 전환되는데 소요되는 시간과 중복된 작업을 최소화 하고 계획단계에서 설계된 모델을 시스템 구축단계까지 실질적으로 전환하는 인터페이스를 CASE 도구인 S-Designer ProcessAnalyst를 활용하여 제시한다.

정보공학 단계별로 활용되는 컴포넌트간의 인터페이스를 CASE 도구인 S-Designer을 이용하여 [그림-10]와 같이 논리적인 업무절차를 시스템적인 업무절차로 전환하여 실질적인 프로그램 코드를 생성한다.

- AS-IS 프로세스 모델설계 : 현업의 업무



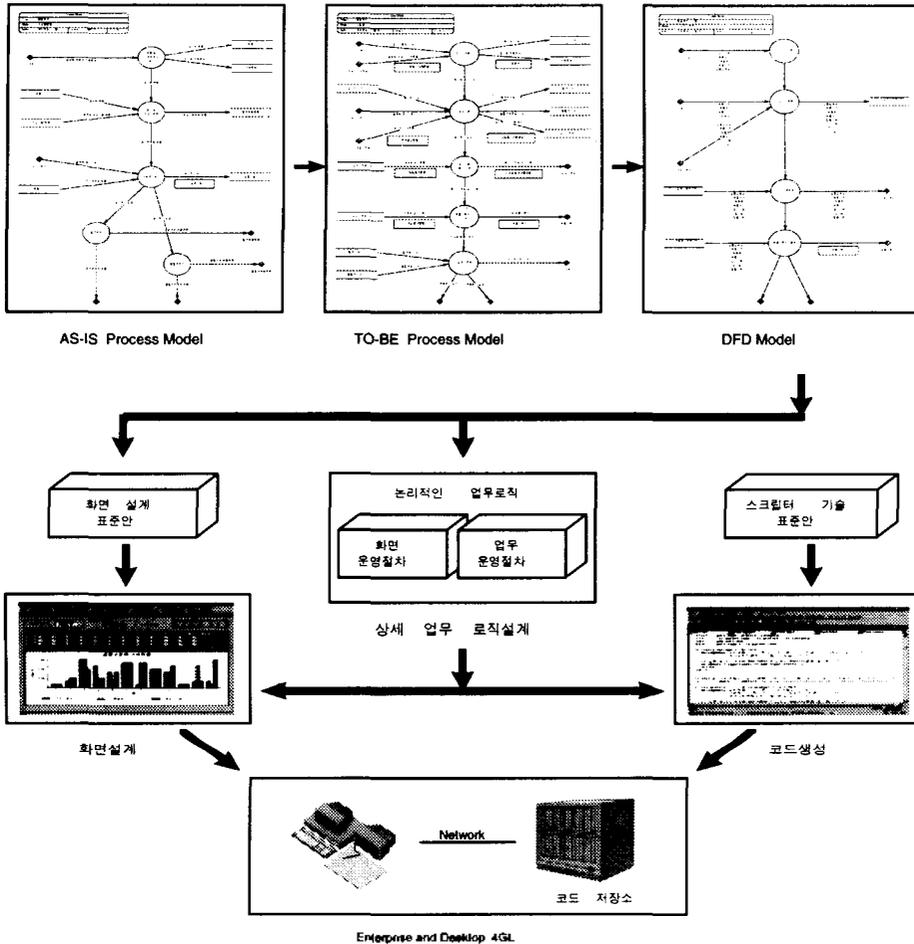
[그림-9] 정보공학 개발방법의 단계별 작업절차

분석의 결과로 설계한다.

- TO-BE 프로세스 모델설계 : 설계된 AS-IS 프로세스 모델을 소프트웨어적인 복사를 이용하고 레퍼런스 TO-BE 모델을 활용하여 설계한다.
- DFD (Data Flow Diagram)모델설계 : TO-BE 모델을 소프트웨어적인 복사를 이용하여 요구분석 사항을 반영하여 설계한다.
- 화면설계 : 설계된 DFD 모델과 화면설계

표준안을 근간으로 화면설계 및 운영절차를 설계한다.

- 업무운영절차설계 : 설계된 DFD 모델을 근간으로 설계한다.
- 실질적인 프로그램 코드 생성 : 설계된 화면과 업무운영절차를 근간으로 스크립터 표준안을 활용하여 프로그램 코드를 생성한다.



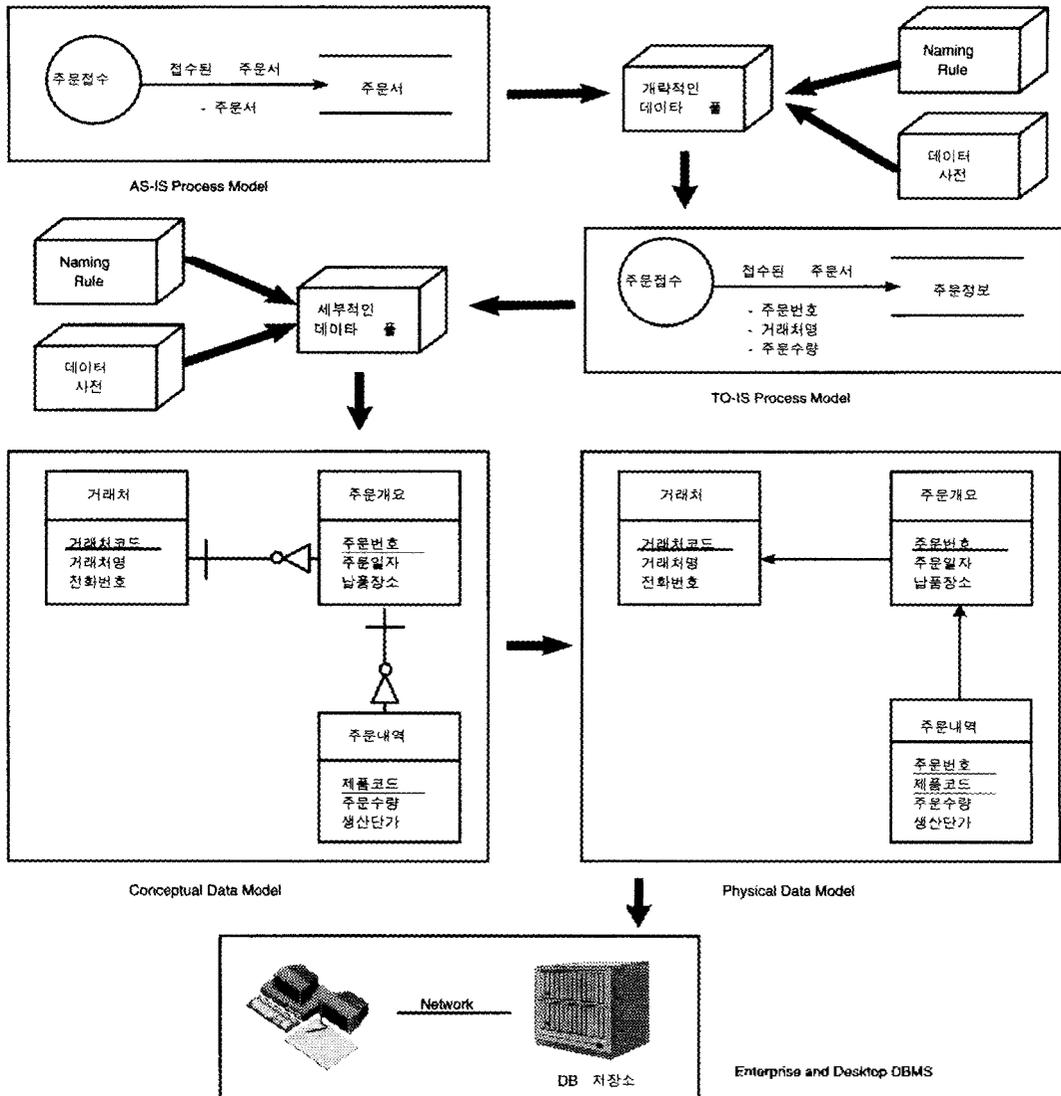
[그림-10] 프로세스 측면의 확장된 컴포넌트 인터페이스

4.3.2 데이터 측면의 활용방법

데이터 측면에서의 컴포넌트 활용방법을 정보공학 개발방법의 단계별로 정의함으로써 계획단계에서부터 구축단계에 이르기까지의 수집 및 분석된 데이터의 유실을 최소화 하는데 목적이 있다. 또한 정보시스템 개발과정에서 상당한 시간과 인력이 소요되는 데이터베이스 구축 과정에서의 중복된 작업을 최소화 시키는 것은 물론 계획단계에서 생성된 데이터

풀의 정보가 구축단계까지 실질적으로 전환되는 인터페이스를 CASE 도구인 S-Designer의 DataArchitect를 활용하여 제시한다.

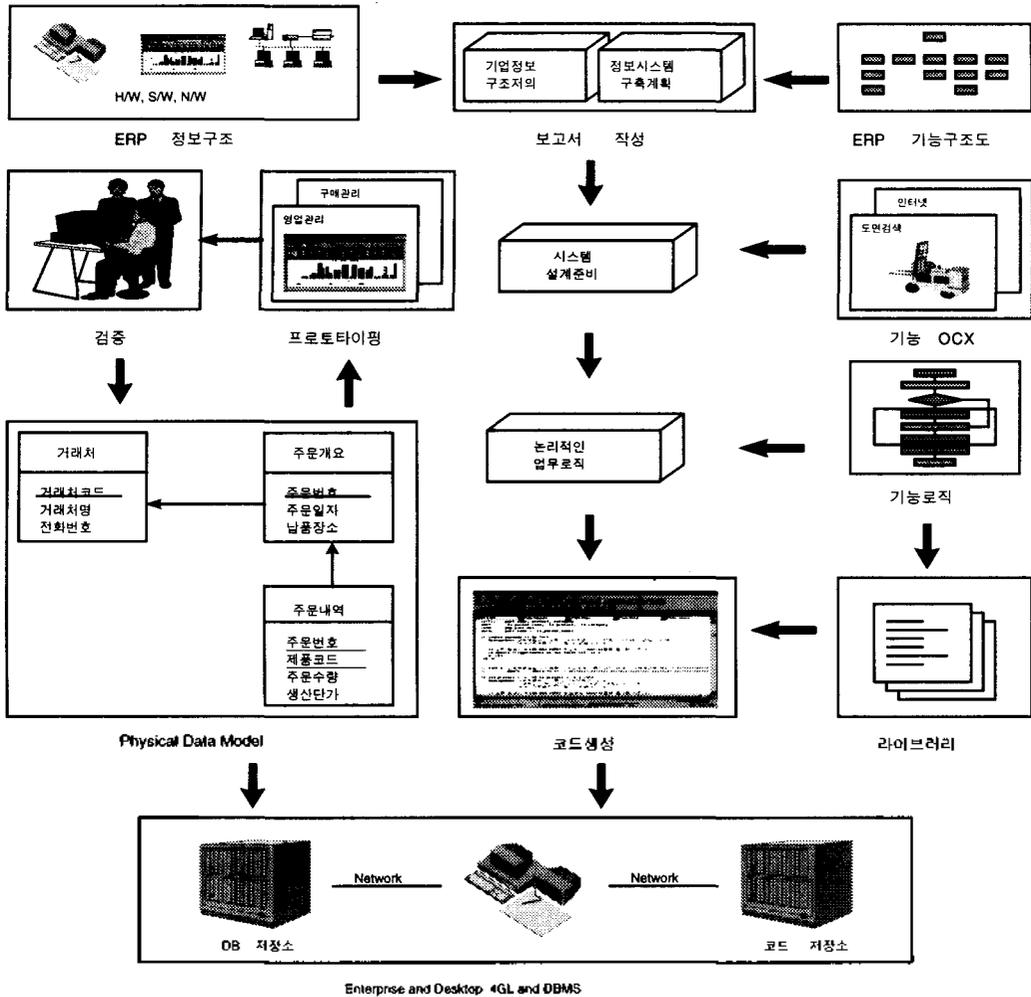
정보공학 단계별로 활용되는 컴포넌트간의 인터페이스를 CASE 도구인 S-Designer를 이용하여 [그림-11]과 같이 실질적으로 데이터를 전환하여 데이터베이스를 구축한다(다음 쪽의 [그림-11] 참조).



[그림-11] 데이터 측면의 확장된 컴포넌트 인터페이스

- 개략적인 데이터 풀을 생성 : Naming Rule 및 개발된 데이터 사전에 의해 개략적인 데이터 Item(한글명, 코드, 형식) 풀을 생성한다.
- 세부적인 데이터 풀을 생성 : Naming Rule

- 및 개발된 데이터 사전에 의해 개략적인 데이터 풀의 Item들을 세부적인 Item(한글명, 코드, 형식) 풀로 분해해서 생성한다.
- 논리적인 데이터 모델설계 : 세부적인 데이터 풀을 Export받아 이들 세부적인 Item



[그림-12] 기능 측면의 확장된 컴포넌트 인터페이스

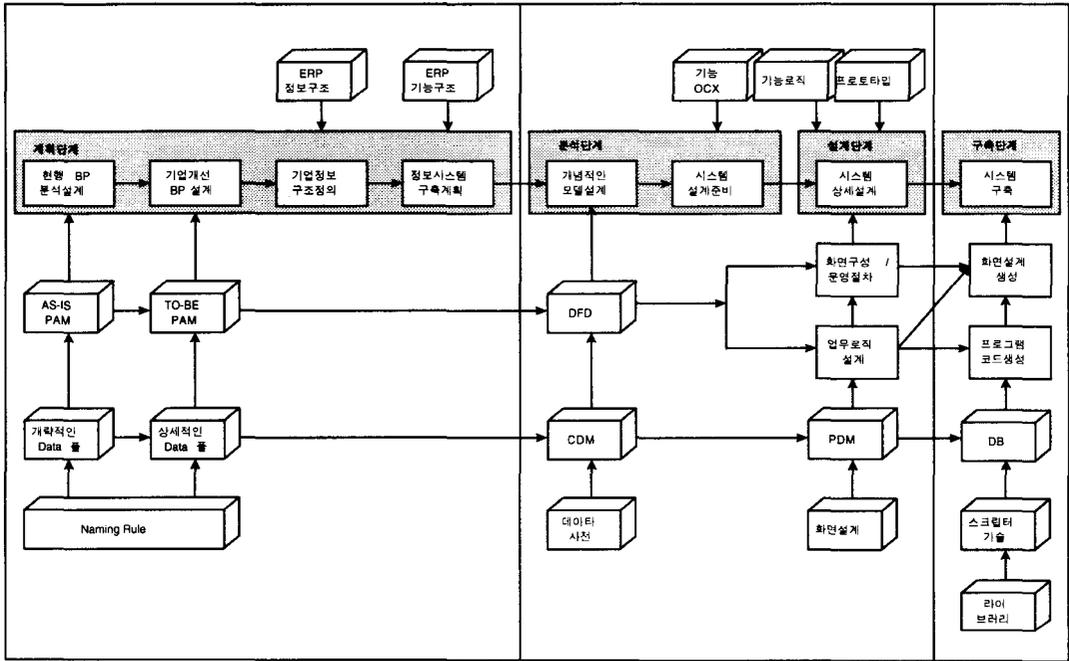
들을 이용하여 엔티타와 속성을 설정하고, 이들 엔티타들의 관계를 정의함으로써 논리적인 데이터베이스 모델을 설계한다.

- 물리적인 데이터 모델설계 : 설계된 논리적인 데이터베이스 모델을 CASE 도구인 DataArchitect을 이용하여 Export시켜 물리적인 데이터베이스 모델을 설계 한다.
- 실질적인 데이터베이스 생성 : 물리적인

데이터베이스 모델을 개발할 DBMS로 전환하여 실질적인 데이터베이스를 구축한다.

4.3.3 기능 측면의 활용방법

기능 측면에서의 컴포넌트 활용방법을 정보공학 개발방법의 단계별로 정의함으로써 정보시스템 구축시에 개발기간 단축 및 검증



[그림-13] 정보공학 단계별의 확장된 컴포넌트 활용방법

DFD: Data Flow Diagram

CDM: Conceptual Data Model

PDM: Physical Data Model

된 기능에 의한 정보시스템의 품질향상을 목적으로 한다. 전 개발단계에서 생성된 기능 컴포넌트들을 불력화하여 정보시스템 개발사에 조립하듯이 재구성하여 개발 생산성을 극대화할 수 있다. 프로토타이핑 개발, 기존 시스템을 역공학에 의한 재사용 및 인터페이스를 위해 CASE 도구인 S-Designer의 AppModeler를 활용하여 제시한다.

정보공학 단계별로 활용되는 컴포넌트간의 인터페이스를 CASE 도구인 S-Designer를 이용하여 [그림-12]와 같이 상세설계의 검증은 물론 구축단계까지의 전반적인 정보시스템의 품질향상 및 기간단축의 효과를 얻을

수 있다.

- 기업정보 구조정의 및 정보시스템 계획수립 작성 : 정보구조 컴포넌트와 기능구조 컴포넌트를 참조하여 작성한다.
- 시스템 상세설계 계획수립 작성 : OCX컴포넌트 기능을 고려하여 작성한다.
- 상세설계검증 : 상세설계후 CASE 도구인 S-Designer의 AppModeler를 이용하여 역공학에 의한 프로토타이핑을 개발하여 검증한다.
- 업무로직설계 : 기능로직 컴포넌트를 활용하여 논리적인 업무로직을 설계한다.
- 프로그램 코드생성 : 검증된 표준화 컴포

넌트의 라이브러리 컴포넌트를 활용하여 프로그래밍 코드를 생성.

정보공학 단계별로 확장된 컴포넌트의 활용방법을 요약하면 [그림-13]와 같다.

4.4 효율적인 정보시스템 개발방법의 통합 VIEW

정보시스템 개발프로젝트를 정의된 정보공학 개발방법의 세부 단계별로 진행을 수행하면서 각 단계별로 정의된 산출물에 의해 단계별 검증 및 진행 실적을 평가한다. 또한 각 단계별로 확장된 컴포넌트의 활용을 데이터, 프로세스 및 기능 측면에서 접근하여 단계별 생산성 및 품질 향상의 효과를 극대화하면서 확장된 컴포넌트 간의 인터페이스 연계에 컴포넌트 개발 및 CASE 도구의 활용으로 전체 프로젝트의 개발기간을 단축함은 물론, 기존 개발방법의 장점들을 전체 개발 과정에 유연하게 도입함으로써 효율적인 정보시스템 개발을 성공적으로 수행할 수 있다. 정보시스템 개발방법을 정보공학 개발방법, 확장된 컴포넌트 활용방법, 기존 개발방법의 장점도입을 고려한 정보시스템 개발방법을 [그림-14]와 같이 통합 View로 구성하였다.

5. 결론 및 추후 연구과제

전사적인 기업 정보시스템을 개발하는데 있어서 전체 개발 프로젝트의 나침판과 같은 역할을 수행하는 개발단계 및 산출물들에 대해서 명확하게 정의하기가 쉽지 않기 때문에 프로젝트 기간중에 많은 위험요소들이 발생하게 되었고, 전체 프로젝트의 개발기간 단축

과 개발 시스템의 생산성 및 품질 향상의 효과를 거두기에는 기술적인 접근방법에 많은 어려움이 있었다. 본 논문에서는 이러한 어려움을 해결하기 위한 하나의 방법으로써 확장된 컴포넌트에 의한 효율적인 정보시스템 개발방법을 연구하였다.

우선, 정보시스템 개발단계를 현실적인 프로젝트 기간에 부합될 수 있는 계획, 분석, 설계, 구축의 네 단계로 구분하여 정의하였으며, 각 단계별로 세부적인 추진단계 및 산출물들을 정의하여 전체 프로젝트를 원활히 수행할 수 있도록 하여 전체 프로젝트 기간중에 발생될 수 있는 위험요소를 최소화 하였다. 또한 정보시스템 개발의 세부 단계별로 확장된 컴포넌트들의 활용을 도입하여 생산성 향상, 품질 향상 및 개발기간 단축의 효과를 극대화 하였다. 기존 컴포넌트들의 활용으로서는 이러한 효과를 극대화할 수 없기 때문에 기존 컴포넌트들의 개념을 확장하여 정의하였으며, 이들의 활용 방법도 프로세스 측면, 데이터 측면, 기능 측면에서 접근 하였다. 그리고 기존의 개발방법의 장점들을 도입하여 정보시스템 개발방법의 전체적인 관점에서의 질적 향상 및 운영 기준 설정에 고려 하였다.

추후의 연구과제로는 첫째, 정보시스템 개발단계가 더욱더 세분화 되어 정의되어야 한다. 즉, 개발과정이 업종별, 추진 목적별 및 개발진행 기간별로 차이가 있으므로 각각에 부합되는 정보공학 개발단계 및 산출물이 정립되어야 하고 또한 각각의 활용 컴포넌트가 정의 되어야 한다.

둘째, 확장된 컴포넌트들의 세분화 및 적용방법의 표준화이다. 본 논문에서 정의된 확장된 컴포넌트들은 개략적인 수준에 불과함으로 더욱더 세분화시켜 정의할 필요가 있으

며, 궁극적으로 향후에는 확장된 컴포넌트들이 표준화된 방법에 의해 활용되어야 한다.

셋째, CASE 모델 컴포넌트인 TO-BE 프로세스 모델과 개념적인 데이터 모델 컴포넌트의 한국적인 레퍼런스 모델의 구축이다. 지금까지는 독일 SAP R3와 같이 외국의 레퍼런스 모델을 국내 기업의 TO-BE 모델로 활용하고 있어 한국기업 실정에 부합되지 않는 부분이 적지 않은 실정이므로 한국기업 환경에 부합되는 레퍼런스 모델 구축이 매우 시급하면서도 중요한 과제이다.

넷째, 실질적인 통합 CASE 도구의 개발로 인한 단계간의 인터페이스 해결이다. 정보시스템의 전체 개발기간을 단축하는데 가장 큰 효과를 얻을 수 있는 것이 CASE 도구의

활용 측면이다. 그러나 현재 국내에 보급되어 사용되고 있는 CASE 도구들 대부분이 개발 단계 부터 구축단계까지 전체적인 단계를 지원하지 못하고 있는 실정이므로 개발단계간의 인터페이스 부분을 해결할 수 있는 CASE 도구의 개발 및 인터페이스를 위한 연계 소프트웨어 개발이 중요한 과제일 것이다.

마지막으로 정보시스템 개발 단계별 전문인력의 양성이다. 전문화된 인력에 의한 단계별 프로젝트 수행은 프로젝트 수행기간에 발생할 수 있는 위험요소를 최소화 시킬수있음은 물론, 예측하지 못한 위험에 대해 신속한 대응이 가능하므로 효율적인 정보시스템 개발에 있어서 가장 중요한 과제가 될 것이다.

참 고 문 헌

- [Fin 1991] Finkelstein, C., "Information Engineering," Computer World, 1991.
- [Mar 1991] Martin, J., Information Engineering, Prentice-Hall, 1991.
- [Gar 1996] Garter Group RAS Services, Selected BPR Tool vendors, 1996. 6.
- [Gay 1991] Gay, R.K.L., Lim, R. and Shing, L.W., "Using IDEF Methodology," International Conference on Computer Intergrated Manufacturing, pp. 97-100, 1991.
- [God 1989] Godwin, A.N., Gleeson, J. W. and Gwillian, D., "An Assessment of the IDEF Notations as Descriptive Tools," Information System, Vol. 14, No. 1, pp. 13-28, 1989.
- [Fre 1987] Freeman, P., "A Perspective on Reusability," Tutorial : Software Reusability, IEEE, 1987.
- [McI 1967] McIlroy, M.D., "Mass Produced Software Components," Software Engineering, Naur and Randell(eds.), 1969.
- [McC 1989] McClure, C., CASE Is Software Automation, Prentice-Hall, 1989.
- [주현 1996] 이주현, "정보기술의 체계", 법영사, pp. 56-63, 1996.
- [철현 1989] 맹철현, "생산성 향상의 도구로 등장한 4세대 언어", 경영과 컴퓨터, 1989.
- [충교 1986] 정충교, 프로토타이핑 방식을 통한 시스템 개발, 한국외국어대학교 경영정보 대학원, 석사학위논문, 1986.
- [경환 1991] 이정환, "CASE 환경에서 재사용 도구의 고찰", 정보과학회지, 1991. 4.
- [IDEF 1997] 대우정보시스템, IDEF 방법론, 대우정보시스템 정보기술부, 1997.
- [ARIS 1997] 대우정보시스템, ARIS 방법론, 대우정보시스템 정보기술부, 1997.
- [전략 1997] 대우정보시스템, ERP구축을 위한 경영전략 및 업무환경분석 보고서, 대우정보시스템 CIM사업부, 1997
- [현행 1997] 대우정보시스템, ERP구축을 위한 현행업무 프로세스분석 보고서, 1997.
- [대우 1997] 대우정보시스템, DSDM 개발 방법론, 1997.
- [G7 1996] 대우정보시스템, 경영관리 소프트웨어 기술개발, G7과제 보고서, 1996.
- [기태 19995] 신기태, 제조데이터베이스 구축을 위한 개념설계 지원시스템에 관한 연구, 서울대학교대학원 박사학위논문, 1995. 2.
- [에스 1997] 에스프리퀵설팅, CASE Tool S-Designer 5.0, 1997.
- [기술 1997] <http://expert.disc.co.kr/index.htm>, 기술동향, 대우정보시스템(주), 1997.
- [가트 1997] <http://expert.disc.co.kr/index.htm>, 가트너서비스, 대우정보시스템(주), 1997.