

중형컴퓨터시스템 개발의 시스템 공학적 접근†

박진원* · 이준석*

Systems Engineering Approach for Midrange Computer System Development †

Park, Jin-Won* · Lee, Joon-Seok*

ABSTRACT

Developing a midrange computer system in a country like Korea is a difficult task. It is because of the lack of computer related technologies, appropriate manpower and the sufficient domestic market. Thus, only a few country except the United States has succeeded in developing her own model of computer systems. Fortunately, Korea became one of them.

In this paper, we will present some special features of TICOM and how we successfully developed our own model of a midrange computer system with respect to systems engineering point of view. The Difficulties and suggestions from the experience of the development of TICOM will also be described.

1. 서론

정부는 지난 1980년대 초반에 행정전산망, 교육 및 연구전산망, 금융망, 국방망, 공안망 등으로 구성되는 국가기간전산망 구축 계획을 수립하여 1980년대 중반부터 이를 실천에 옮기기 시작했다. 상기 5개 분야 중에서 행정전산망(NAIS, Nati-

onal Administrative Information System)은 현재 정부의 지속적인 지원에 힘입어 가장 성공적으로 운영되고 있으며 이러한 행정전산망의 성공적인 운영은 이에 사용되고 있는 주전산기인 타이컴(TICOM : Tlger COMputer, 혹은 Tlghtly COupled Multiprocessor System)의 개발에 기인하고 있다.

† 타이컴(TICOM) 개발 사업은 정보통신부, 통상산업부, 과학기술처와 대우통신, 삼성전자, LG전자 및 현대전자산업의 공동 출연 연구개발사업으로 추진되었음.

* 한국전자통신연구소 시스템공학연구실

타이컴 개발 사업은 설계, 구현, 통합, 시험 및 생산의 전 과정이 한국전자통신연구소(ETRI, Electronics and Telecommunications Research Institute)와 대우통신, 삼성전자, LG전자 및 현대전자산업 등 국내의 기업체와 연구소에 의해 공동으로 추진된 대규모 시스템 연구 개발 사업으로 전체 시스템 구조는 외국 제품의 구조와는 전혀 독립적으로 설계되었고, 하드웨어는 ETRI 기술진이 설계, 구현 및 시험을 담당하였다. 또한 운영체제 등의 시스템 소프트웨어들은 세계적으로 표준화되어 있는 제품을 들여다 약간의 수정 및 개선을 거쳐 타이컴에 이식하였다. 그리고 데이터베이스 관리시스템과 UNIX 운용관리 소프트웨어들은 ETRI에서 독자적으로 개발되었다.

따라서 이러한 대규모 시스템 연구 개발을 성공적으로 수행하기 위해서는 많은 인력, 시간, 비용 등의 자원과 요소 기술 및 설계 방법론 등의 공학 기술이 유기적인 결합을 통해 효과적이고 종합적으로 관리되어야 한다. 특히 시스템적인 분석을 통한 절차나 방법론 등이 없이 규모가 큰 연구 프로젝트를 수행한다는 것은 매우 위험한 생각이며 이러한 관점에서 볼때 연구 개발 사업 시작에서 사업의 종료까지 시스템 공학적인 접근 방법이 필수적이며 선진국의 경우 각자의 고유한 시스템 공학적인 방법을 사용하고 있다[9, 11].

그러나 우리나라의 경우 연구 개발 측면에서 규모가 큰 연구 개발 사업을 수행한 경험이 많지 않아 시스템 공학적인 기술 축적이 미진하여 소수의 연구개발 사업의 경우에도 개발 환경이나 문화가 다른 선진국의 예만을 주로 적용하였기에 이에 따른 많은 시행착오와 어려움을 겪는 경우가 많았다[7]. 또한 기존의 개발방법론들[8, 10, 13, 14, 15]이 대부분 소프트웨어를 대상으로 하고 있기에 하드웨어와 소프트웨어가 혼합되어 있는 시스템의 경우 이를 적용하기에는 많은 무리

가 따른다.

이에 따라 본 논문은 국산 중형급 컴퓨터 시스템인 TICOM 개발 사업을 수행하며 습득한 경험과 공학적 기술을 토대로 연구개발 사업에서의 개발 방법론에 대한 시스템 공학적 측면을 분석, 정립해 본 것이다.

본 논문은 다음과 같은 구성을 따랐다. 1절 서론을 거쳐, 2절에서 타이컴의 특징과 타이컴 개발 사업의 특징을 서술한다. 3절에서는 타이컴 개발 방법론을 시스템의 요소별 분해와 사업의 단계별 수행이라는 두가지 방향에서 살펴보고, 4절에서 시스템 공학적인 측면에서 타이컴 개발 방법론 자체를 분석해 본다. 마지막으로 결론과 향후 연구 과제를 5절에서 다룬다.

2. 타이컴 개발 사업

본 절에서는 구체적인 시스템 공학적 분석에 앞서 타이컴 개발 사업에 대한 전반적인 이해를 돕기위해 타이컴 모습에 대해 간단히 소개하고 타이컴 개발 사업에 대한 주요한 특징을 살펴본다.

2.1 타이컴 규격 및 설계 개념

타이컴은 중형급의 다중프로세서 컴퓨터시스템으로서 UNIX System V Release 3.1을 기반으로 한 다중처리용 운영체제를 탑재하고 있다. 타이컴은 또한 DBMS(데이터베이스 관리시스템), 컴퓨터통신 소프트웨어, 7가지 프로그래밍 언어, UNIX 운용관리 소프트웨어와 각종 유틸리티 프로그램들을 탑재하고 있다. 타이컴에 대한 자세한 규격은 [표1]에 기술되어 있다.

<표 1> 타이컴 규격

하드웨어	
중앙처리장치 (CPU)	-MC68030(2~20개) -성능 : 10~80MIPS
부동소수점 연산기	-MC68882 -IEEE Std 754
주기억장치	-512MB(ECC 기능)
캐쉬메모리	-64KB /프로세서(Write-back)
시스템버스	-100MB /sec
디스크용량	-최대 40GB
단말기수	-최대 512대

소프트웨어	
운영체제	-UNIX SVR 3.1 -한글 코드지원(KSC5601)
프로그래밍 언어	-C, C++, Con-C, BASIC -FORTRAN, COBOL 등
컴퓨터통신	-LAN(Ethernet), WAN(X.25) -OSI
DBMS	-BADA(RDBMS)
유틸리티	-운용관리 S/W

타이컴의 설계 개념[3]은 이 사업에 참여했던 엔지니어들과 중간 관리자들이 1년 넘는 기간동안 여러 차례의 회의를 통해 확정되었다. 궁극적으로는 사업이 시작된 후 1년 6개월 정도가 지나서야 최종적으로 설계가 확정되었다[2]. 다음은 타이컴의 기본 설계 개념을 요약한 것이다.

- 국제 표준을 적용함
- 상용 마이크로 프로세서를 사용함(MC68030)
- 밀결합 다중프로세서 구조를 가짐

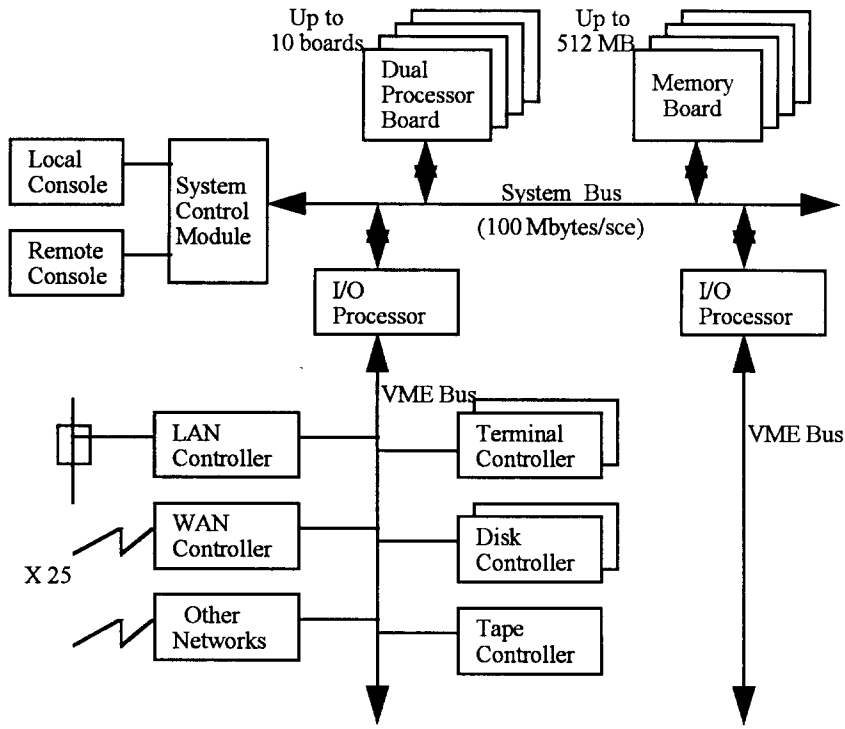
- 시스템 확장성을 가짐(프로세서를 20개까지 장착 가능)
- 트랜잭션 처리를 지원함
- 한글 처리 기능을 제공함
- 관계형 DBMS를 탑재함
- 각종 컴퓨터통신 소프트웨어를 탑재함

이와 같은 설계 개념에 따라 타이컴의 하드웨어 구조는 <그림 1>과 같은 모습으로 결정되었다. <그림 1>에서 보는 바와 같이 타이컴은 기본적으로 버스를 기반으로 하는 다중프로세서 컴퓨터시스템으로 분류할 수 있다.

2.2 타이컴 개발 사업의 특징

타이컴 개발 사업은 몇가지 특징을 갖고 있다. 첫째, 앞의 서론에서 언급된 바와 같이 타이컴의 모든 구성 요소가 타이컴 개발사업에 참여했던 ETRI(한국전자통신연구소)와 4개 참여 기업에 의해 독자적으로 개발된 것은 아니고 일부는 기술 도입에 의존하였고 일부는 독자적으로 개발했다는 점이다. 즉, 하드웨어 부분은 보드 수준에서 독자적으로 개발되었지만 주처리장치(CPU)는 미국 모토롤라사의 MC68030을 사용하였다. 또한 입출력보드(IOPBoard)에 사용된 프로세서도 모토롤라사의 MC68020을 사용하였다. 타이컴을 구성하고 있는 여러 요소들 중에서 직접 개발(Make)한 것과 구매(Buy)하여 사용한 것에 대한 분류는 [표 2]와 같다.

둘째, 타이컴 개발 사업은 산, 학, 연 및 정부가 모두 참여하는 다소 복잡한 방식으로 진행되었다. 이 사업은 우선 중형급 컴퓨터가 국내에서 개발되어 행정전산망에 주전산기로 사용되어야 한다는 생각에서 시작되었다. 이 생각은 정부가 제안하고, 정부출연 연구기관인 ETRI가 구체적인 사업 계획을 수립하였으며, ETRI와 국내 가전 4개사



〈그림 1〉 타이컴 하드웨어 구조

〈표 2〉 타이컴 구성요소중 개발과 구매분류

구성요소	개발/구매
하드웨어	개발
CPU	MC68030 구매
운영체제	UNIX SVR 3.1 구매 및 부분개발
언어	구매/이식(Porting) 및 부분개발
DBMS	개발
통신SW	구매 및 부분개발
유틸리티	운용관리SW(개발) 병렬유틸리티(개발) 한글유틸리티(개발) 그밖의 유틸리티(구매)

가 공동으로 연구개발 사업을 수행하는 것으로 진행되었다. 타이컴 개발 사업은 1987년 6월에 시작되어 1991년 7월까지 4년동안(중간에 2개월의 공백이 있었음) 228억원의 연구비와 연인원 714명의 연구 인력이 투입되어 수행되었다.

셋째, 타이컴 개발 사업은 하드웨어, 소프트웨어가 복잡한 방식으로 혼합되어 있어 지금까지 통용되던 소프트웨어 개발 방법론은 그 적용에 한계를 보여 주었다. 이는 시스템의 모든 요소가 개발되는 것이 아니라는 점, 시스템 구성 요소들 간에 독립성이 매우 강하다는 점과 더불어 새로운 시스템 개발 방법론을 정립하고 적용해야 하는 필요성을 제기시켰다.

다음 절에서는 타이컴 개발 방법론을 시스템의 요소별 분해와 사업의 단계별 수행이라는 방향에서 살펴본다.

3. 타이컴 개발 사업 개발방법론

타이컴 개발 사업은 투입된 연구비의 규모나 인력, 그리고 연구개발 수행 기간 등 여러면에서 우리 나라에서는 매우 큰 규모의 프로젝트이다. 따라서 이 사업에서 개발하고자 했던 타이컴이라는 전체 시스템을 작은 단위로 분해하여 각각을 개별적, 독립적으로 개발한후 나중에 이를 다시 통합하는 방법을 동원하였다. 개발 기간 역시 몇 단계로 나누어 각 기간의 종료 시점에 중간 연구 결과물을 산출하고 이를 전문가에서 평가받는 방법도 사용하였다[1].

여기서는 타이컴 개발 사업에서 사용된 개발 방법론이 내포하고 있는 두 가지 기본 철학, 즉 시스템의 수직적 분해(시스템의 요소별 분해)와 수평적 분해(사업 기간의 구분)에 대하여 살펴본다[3].

3. 1 시스템의 요소별 분해

타이컴 개발 사업이 본격적으로 추진되기 시작할 때 기존의 상용화된 중형급 컴퓨터 시스템이 참고용으로 조사되었다. 이와 동시에 타이컴을 개발하는 방법을 계획하기 위하여 기존의 개발방법론도 연구되었다.

일반적으로 대규모 시스템 연구개발 사업에서 자주 사용되는 방법으로 라이프사이클 모델이 있다. 타이컴 개발 사업에서 사용될 라이프 사이클 모델을 결정하기 위해 여러 형태의 라이프사이클 모델 즉, 전통적인 폭포수 모델(Waterfall model)[10, 13]과 프로토타이핑 모델(Prototyping model)[8, 12, 14] 등을 고려하였다.

이중 폭포수 모델은 소프트웨어 개발 사업에서 사용하는 모델이긴 하지만 하드웨어와 소프트웨어가 통합되는 컴퓨터 시스템 개발에 적합한 라

이프 사이클 모델이 없을뿐만 아니라, ETRI 내부의 다른 개발 팀에서 사용해 본 경험이 있고, 사업 수행 기간이 비교적 충분하기에 약간의 수정을 거쳐 채택하였다. 여기서 기존 모델에 수정을 한 이유는 규모가 큰 연구 개발 사업을 수행한 경험이 많지 않은 우리나라·실정상 개발 환경이나 문화가 상이하고 연구 개발 및 관리 수준이 높은 선진국의 예만을 주로 적용하였기에 이를 그대로 적용하기에는 무리가 따른다[7]. 전반적인 수정 사항은 3. 2 사업별 수행 단계에서 언급한다.

컴퓨터시스템은 통상적으로 하드웨어와 소프트웨어들이 상당히 독립적인 형태로 구성되어 있다. 타이컴은 일반적인 중형급 시스템의 범주에 포함되기 때문에 이를 개발하기 위하여 하드웨어에서 시스템 소프트웨어, 응용소프트웨어에 이르기까지 모든 요소들을 망라해야 한다. 타이컴의 개발을 원활히 수행하기 위하여 타이컴을 7개의 서브시스템으로 나누고 이를 다시 45개의 블록으로 나누었다. 7개의 서브시스템은 각기 하드웨어, 운영체제, 프로그래밍 언어, DBMS, 컴퓨터통신, UNIX 운영체제 소프트웨어와 OLTP 소프트웨어로 구성되어 있다.

서브시스템은 사업내에서 독립적인 개발 및 조직의 단위를 기준으로 분해하였다. 따라서 각 서브시스템은 다른 서브시스템과 사용자 요구사항, 설계, 구현 및 시험 측면에서 거의 독립적인 성격을 갖고 있다. 이에 따라 각 서브시스템은 개발기간이 막바지에 이르러 시스템으로 통합되기까지는 상호 관련이 있는 인터페이스 부분만을 공유하고 관리할뿐 서로 독립적으로 개발되고 시험되었다.

블록은 독립적인 동작, 교체 및 관리가 가능한 단위를 기준으로 분해되었다. 하나의 서브시스템마다 평균 5개에서 9개까지의 블록으로 분해되었

으며 이들 블록은 각기 필요에 따라 여러개의 기능 모듈로 다시 분해되기도 했다. 그러나 기능 모듈은 사업 차원에서 공식적인 것은 아니고 각 개

발팀이 필요에 따라 설정할 수 있게 하였다. [표 3]은 타이컴을 구성하고 있는 7개 서브시스템과 45개 블록을 정리한 것이다.

〈표 3〉 타이컴 서브시스템과 블록 항목

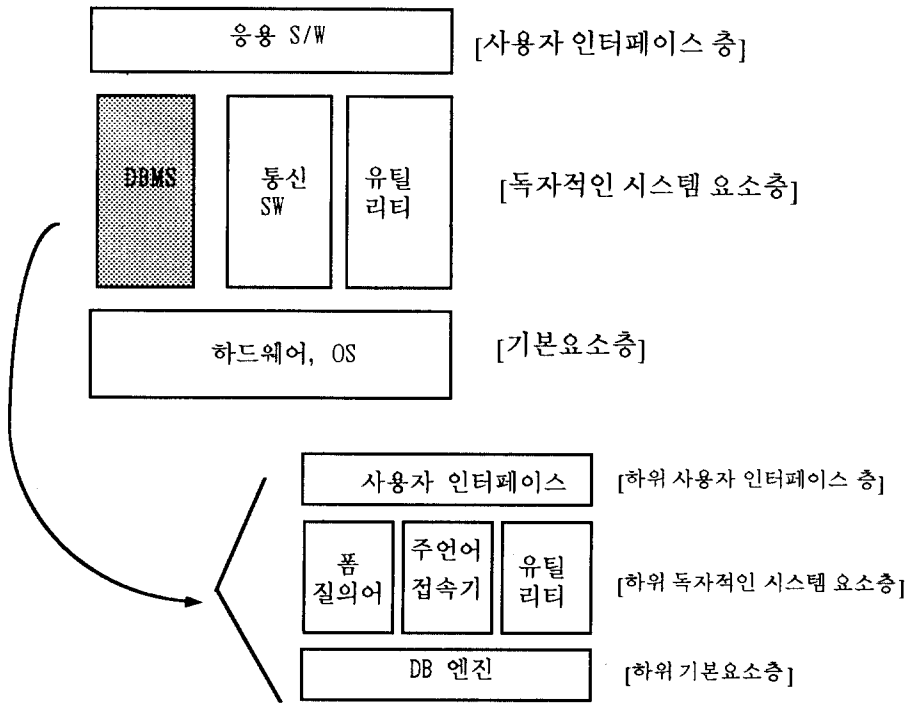
서브시스템	블록
하드웨어	중앙처리장치, 시스템버스, 주기억장치, 시스템제어기, 입출력처리기, 펌웨어, 진단, 패키징
운영체제	커널, 명령어 및 라이브러리
언어	C, C++, Con-C, FORTRAN, COBOL, BASIC
DBMS	메뉴시스템, 스키마관리기, 질의어처리기, 보고서작성기, DB유틸리티, 폼처리기, SQL처리기, 데이터접근관리기
통신 SW	MHS, FTAM, VT, PT/SS, TCP/IP, LAN, WAN
운용관리 SW	감시, 진단, 사용자 인터페이스, 일상관리
OLTP	TP커널, TP명령어 및 라이브러리, ISAM

타이컴의 구성 요소를 서브시스템을 중심으로 계층별로 살펴보면 〈그림 2〉와 같다. 〈그림 2〉에서 가장 위에 해당되는 부분에는 사용자가 존재하고 있음을 의미한다. 따라서 시스템 구성 요소가 윗 부분에 자리잡고 있을수록 그 요소가 사용자와 관련성이 높다는 의미를 내포한다. 〈그림 2〉에서 보는 바와 같이 맨 윗 부분은 사용자 인터페이스 요소이며, 중간 부분은 독자적인 시스템 구성 요소들을 나타내고 맨 아래 부분은 시스템의 기본 구성 요소를 나타낸다. 일반적으로 사용자 인터페이스 요소, 하드웨어 및 소프트웨어로 구성되는 시스템은 〈그림 2〉와 같은 로마 숫자 III과 같은 모습을 보인다. 이는 흔히 통신 시스템이나 컴퓨터 시스템에서 시스템의 계층별 구조를 나타내는데 유용하게 사용되고 있다.

본 개발 사업에서는 이러한 III자 형태의 시스

템 구성 모습을 바탕으로 컴퓨터 시스템의 구조를 표현하였다. 그리고 한가지 주목할 만한 점은 시스템을 구성하는 각 부분들이 자체내에 다시 작은 형태의 III자 형태를 갖고 있어 일종의 Nested Layer model로서 표현할 수 있다는 것이다. 예를 들면 타이컴 전체로는 사용자 인터페이스를 시스템 요소로 각종 응용 소프트웨어들이 시스템 구성 요소로 인식되며 하드웨어와 운영체제는 기본 구성요소를 나타내는 것으로 설명할 수 있다. 또한 DBMS를 다시 살펴보면 다시 〈그림 2〉와 같은 형태로 그 구조를 표현할 수 있는데 맨 위에 사용자 인터페이스 부분, 중간에 폼 질의어, 주언어 접속기, 유틸리티 등이 자리잡고, 맨 아래에 DB 엔진 부분이 자리잡고 있는 것으로 설명할 수 있다.

이와 같은 Nested Layer Model은 복잡한 시스



<그림 2> 타이컴의 Nested Layer 구조

템을 구조적으로 파악하고 이를 설계, 분석하는데 도움이 될것으로 보인다. 앞으로 좀더 다양한 시스템들을 대상으로 이 모델의 유용성을 검증해 보는 일이 필요하다고 생각된다.

3. 2 사업의 단계별 수행

타이컴은 1987년 6월부터 1991년 7월까지 4년간 개발되었다. 전체 개발 기간은 결코 짧은 시간이 아니며 따라서 단계적인 개발 전략이 필요로 하게 되었다.

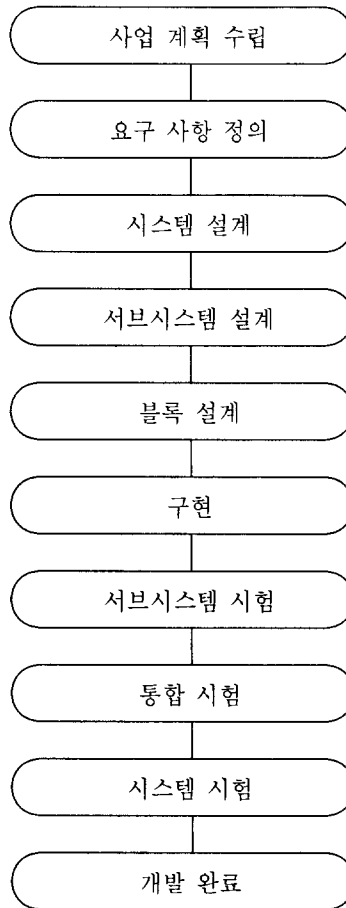
ETRI에서 이 사업을 시작할때 사업 참여 엔지니어들은 사업 수행에 관한 다음 3가지 원칙을 정립했다. 첫째, top-down 방식의 설계, 둘째, 서브 시스템들 간의 서로 독립적인 구현, 셋째, bottom-up 방식의 통합 및 시험이 그것이다. 이러한

원칙에서 전체 사업 기간을 개발 시작에서 부터 개발 완료 단계까지를 [그림 3]과 같이 전부 10개 단계로 나누고 각 단계별로 수행해야 할 업무를 다음과 같이 정의하였다[3].

(1단계) 사업 계획 수립 단계

1) 개요

사업 계획의 수립은 주전산기 개발의 출발점으로 개발에 대한 전반적인 계획을 수립하는 단계로서 사용자로부터 요구된 조건을 분석하고 사업적 측면의 제반 정보를 체계화시키기 위하여 기술 조사를 통한 개발의 필요성과 타당성 검토, 개발의 내용과 범위 설정, 개념적인 측면에서의 시스템 모델링, 그리고 추진 전략의 일환으로 투입 자원 및 일정 계획을 수립한다.



〈그림 3〉 타이컴 개발 순기

2) 산출 결과물
사업 계획서

(2단계) 요구 사항 정의 단계

1) 개요

이 단계에서는 사업 계획서를 기반으로 하여 개발하고자 하는 시스템을 사용자의 관점에서 명확히 정리한 요구 사항 정의서(Requirements Definition Document)를 작성한다. 또한 요구 사항 정의서는 서브시스템 설계 또는 블록 설계 단

계에서 작성되는 사용자 지침서를 참조하여 시스템 시험 단계까지는 최종판을 완성한다. 그리고 시스템이 개발 완료된 후 이를 시험하기 위한 시스템 계획서도 작성한다. 요구 사항 정의 단계에서 작성한 시스템 시험 계획서는 최종판이 아니며 개발이 진행되면서 계속 수정 보완하여 시스템 시험이 시작되기 전까지 최종판을 완성한다.

2) 산출 결과물

요구사항 정의서

(3단계) 시스템 설계 단계

1) 개요

시스템 설계는 시스템에 대한 설계를 수행하는 단계로서 사용자의 관점에서 작성된 요구 사항 정의서를 분석하여 이를 충족시키기 위한 시스템의 기능, 성능 및 구조를 제시하고, 시스템을 차하위 구조인 서브시스템으로 분해하여 이들의 상호 연관 관계 및 각각의 설계 지침(기준)을 제시한다. 또한 이를 시험하기 위한 통합 시험 계획서도 작성한다. 그러나 이 단계에서 작성하는 통합 시험 계획서는 통합 시험이 수행되기 직전까지 지속적으로 수정·보완되어야 한다.

2) 목적

- 사용자 관점의 요구 사항을 시스템 개발자 관점의 '무엇(What)'에 해당하는 제공 기능, 성능 및 시스템 구성 요소 등으로 분해
- 시스템을 독립성이 강한 서브시스템으로 분해
- 조직 구성의 기준(서브시스템)을 정함
- 서브시스템 설계를 위한 설계 지침(기준) 제시
- 통합 시험 기준 제시

3) 서브시스템(Subsystem) 분해 기준

시스템을 여러개의 서브시스템으로 분해하는 기준은 각 서브시스템이 독립성이 매우 강하고, 조직 구성의 기준이 되며 개발 완료 후 타 서브시스템의 영향을 최소한으로 받으면서 독립적인 시험이 가능할 수 있도록 한다.

4) 산출 결과물 명

시스템 설계서(System Design Specification), 통합시험 계획서(Integration Test Plan)

(4단계) 서브시스템 설계 단계

1) 개요

서브시스템 설계는 시스템 설계 단계에서 제시된 서브시스템 설계 지침과 서브시스템이 제공해야 할 기능과 성능을 충족시키기 위한 방안을 제시한다. 이를 위하여 서브시스템의 전체 기능을 명확히 서술하고자 하위 구조인 블록으로 나누어서 이들의 상호 연관 관계 및 각각의 설계 지침을 제시한다. 또한 서브시스템 설계의 전제가 되는 사용자 지침서(최종 사용자를 위한 것으로 서브시스템 단위로 작성함)를 작성하며 추후 단계에서 지속적인 수정, 보완 작업을 수행한다. 그리고 서브시스템이 구현된 후 이를 시험하기 위한 시험 기준도 제시한다.

2) 목적

- 시스템 설계 단계에서 제시한 각 서브시스템 별 제공 기능, 성능을 구체적으로 어떻게 구현할 것인가를 제시.
- 서브시스템을 비교적 독립성이 강한 블록으로 분해.
- 조직 구성 및 관리의 최소 단위 기준(블록)을 정함.
- 블록 설계를 위한 설계 지침 제시.
- 서브시스템 시험 기준 제시.

3) 블록(Block) 분해 기준

블록은 시스템을 구성하는 기본 단위이다. 따라서 블록은 가능한 한 명확히 정의되어야 한다. 서브시스템을 블록으로 분해할 때에는 다음의 기준을 적용한다.

- 독립적인 관리의 단위가 되도록 함.
- 시스템, 서브시스템, 블록은 구조적 체계를 갖게함.
- 블록은 내부 응집력을 최대로 하고 블록 간의 결합력을 최소로 하여 상호연관성을

최소화

- 블록은 가능한 한 독립적으로 동작할 수 있도록 하고 쉽게 교체할 수 있게 하여 오류발견시 다른 부분에 영향을 미치지 않고 격리시킬 수 있어야 함.
- 블록은 구현이 완료된 후 가능한 한 독립적인 시험이 가능해야 함.

4) 산출 결과물

서브시스템 설계서(Subsystem Design Specification), 사용자 지침서(User's Manual : Subsystem 단위), 서브시스템 시험 계획서(Subsystem Test Plan)

(5단계) 블록 설계 단계

1) 개요

블록은 시스템을 구성하는 기본 단위로서 독립적인 관리의 단위이며 비교적 독립적으로 동작할 수 있고 오류 발생시 쉽게 교체할 수 있어야 한다. 또한 구현이 완료되면 각 블록에 대한 독립적인 시험도 가능해야 한다. 블록 설계는 서브시스템 설계 단계에서 제시된 기능, 성능등을 실현시키기 위한 구체적인 실천 방안을 제시한다. 따라서 블록 설계가 완료되면 전체 시스템에 대한 설계가 마무리 되고 구현 단계로 넘어가게 된다. 블록 설계가 완료된 후 서브 시스템 설계 단계에서 작성하지 못한 사용자 지침서를 작성하고 이에 따라 요구 사항 정의서도 수정, 보완한다. 그리고 타 개발 팀(Immediate User)을 위한 설명서도 작성한다.

2) 목적

- 시스템을 구성하는 기본 단위(블록)를 명확히 정의함.
- 개발자로 하여금 시스템이 제공하는 기능과 성능의 범위와 한계를 명확히 알게함.

- 시스템에서 제공하는 기능과 성능을 어떻게 구현할 것인가를 구체적으로 서술함.

- 블록 시험의 기준 제시

3) 산출 결과물

블록 설계서(Block Design Specification), 블록 시험 계획서(Block Test Plan : 블록 시험 프로그램 포함), 설계 도면, Algorithm 설계서(펌웨어), Recording spec.(porting), Algorithm 설계서(독자 개발), 사용자 지침서, 설명서

(6단계) 구현 단계

1) 개요

구현 단계에서는 블록 설계 단계에서 작성된 블록 설계서에 따라 블록을 구체적으로 실현시킨다. 하드웨어인 경우 PBA(Printed Board Assembly)를 개발하고 소프트웨어인 경우 원천 코드(Source Code)를 작성한다. 그리고 개발된 PBA나 원천 코드가 블록 시험 계획서에 규정한 바와 같이 그 블록의 기능과 성능을 만족시켜 주고 있는가 시험한다. 구현이 끝나면 기술 보호가 필요하다고 인정되는 사항에 대하여 특허 출원 및 소프트웨어 등록을 추진한다.

2) 산출 결과물

PBA 및 기구물 구현용 문서(하드웨어), 설계 관련 CAD 출력(하드웨어), 원천 코드(소프트웨어), 블록 시험 결과 보고서

(7단계) 서브시스템 시험 단계

1) 개요

서브시스템 시험은 구현 완료된 블록들을 통합하여 서브시스템을 구현하는 과정에서 각 블록들 간의 연결이 무리없이 이루어지며, 통합된 서브시스템이 서브시스템 설계서에서 제시한 기능과 성

능을 만족시키기가 시험한다. 서브시스템 시험 단계에서 유의할 점은 각 서브시스템의 독립성이 매우 강하고 최종 사용자가 사용할 기능이 이미 구현된 상태이므로 사용자 측면을 충분히 고려하여 시험을 수행해야 한다.

2) 산출 결과물

서브시스템 시험 결과 보고서, 서브시스템 인증 결과 보고서

(8단계) 통합 시험 단계

1) 개요

통합 시험 단계에서는 서브시스템 시험 단계에서 시험이 끝난 각 서브시스템들을 필요에 따라 수평적, 수직적으로 통합하여 최종 시스템이 형성될 때까지 필요한 시험을 수행한다. 또한 시스템 설계서에서 제시한 시스템 기능 및 성능들이 제공되는가 시험한다. 그리고 통합 시험이 끝나는 시점에는 특허 출원 및 소프트웨어 등록 추진이 완료되어야 한다.

2) 산출 결과물

통합 시험 결과서, 기술 전수 계획서, 특허 출원 및 소프트웨어 등록 신청 결과서

(9단계) 시스템 시험 단계

1) 개요

시스템 시험 단계에서는 통합 시험 단계를 거쳐 완성된 시스템이 요구 사항 정의서에서 제시한 사항들을 만족시키는지의 여부를 시험한다. 그리고 시스템 구성 변화에 따른 성능 변화 측정, 통합 시험에서 제기된 성능 향상 방안의 실천, 더 나은 시스템을 만들기 활동을 수행한다. 또한 사용자 지침서를 수정 보완하여 최종판을 완성하고 이를 바탕으로 요구사항 정의서를 수정 보완한다.

그리고 마지막으로 운용, 관리 및 보존 지침서를 작성한다.

2) 산출 결과물

시스템 시험 결과 보고서, 사용자 지침서(최종판), 시스템 운용, 관리 보존 지침서

(10단계) 개발 완료 단계

1) 개요

개발 완료 단계에서는 통합 시험, 시스템 시험이 모두 완료된 후 그 동안 수행되어 온 연구 개발 업무를 총 결산한다. 이 단계에서는 기업에게 시스템을 생산하는데 필요한 자료를 인계하며 연구개발 과정에서 작성된 문서를 최종적으로 정비한다. 마지막으로 기술 전수 체계에 의거하여 기업체에게 기술 전수를 수행하고, 문서를 검토하여 책자화 할 것을 정리하고 이를 수행한다.

2) 산출 결과물

최종 문서 목록, 수정 완료된 결과물, 기술 전수 결과서, 사업 결과 보고서

이와 같이 타이컴 개발 방법론의 핵심은 폭포수 모델(Waterfall Model)에 기반을 둔 시스템의 요소별 분해(System Division)와 단계별 수행(Time Division)으로 요약해 볼수 있다. 이러한 개발 방법론이 타이컴 개발사업을 수행하는 최적의 것이었는가는 검증하기 매우 어렵다. 그러나 이와 개념적으로 비슷한 개발 방법론은 ETRI 내의 TDX 개발팀에서 사용하여 성공적으로 TDX(국산 전자교환기) 개발사업을 완성시킨 바 있고, 타이컴도 성공적으로 개발되어 상용 시장에서 성공을 거두고 있다[1]. 타이컴 개발 방법론의 원형은 스웨덴의 전화 회사인 ERICSSON과 ETRI의 TDX 개발 사업에서 사용된 것이다. 그러나 이들의 개발 방법론은 본질적으로 전자교환

기나 기타 통신 장비를 개발하는 사업에서 개발된 것이므로 TICOM 개발 사업을 수행하면서 컴퓨터 개발 사업의 특성에 맞도록 지속적으로 수정, 보완되어 왔다.

다음 절에서는 타이컴 개발 방법론이 시스템 공학적인 측면에서 어떤 특징을 갖고 있는지 분석해 본다.

4. 타이컴 개발 방법론 분석

타이컴 개발 사업 수행 과정을 자세히 살펴보면 시스템 공학적인 측면에서 다음과 같은 3가지 특징을 발견할 수 있다. 이는 2장에서 언급한 바와 같이 타이컴이라는 중형급 컴퓨터 시스템 자체가 갖는 특징과, 타이컴 개발 사업 수행상의 특징에 따라 결정된 바가 크다고 판단된다.

- Middle-out 전략
- 부분 개발, 부분 기술도입
- 시스템 요소별 연구 인력 투입 비율

한가지 주목할 점은 타이컴 개발 방법론이 갖고 있는 이러한 측면들이 타이컴 개발 사업을 시작하던 초기 단계부터 정리된 것이 아니라 사업을 수행해 나가면서 자연스럽게 형성되었다는 점이다. 이제 이들 3가지 사항에 대하여 자세히 살펴보자.

4. 1 Middle-out 전략

전체적으로 타이컴 개발 사업이 앞 장에서 서술한 순서에 따라 진행되기는 했지만 실제로는 각 단계의 수행 내용이 전혀 변경없이 진행된 것은 아니다. 개발 목표 시스템인 타이컴은 하드웨어, 시스템 소프트웨어 및 응용 소프트웨어의 집합체이다. 따라서 타이컴은 한사람의 천재적인 엔지니어가 하드웨어 구조부터 응용 소프트웨어의

사용자 인터페이스 부분까지 모두를 일괄적으로 설계할 수는 없다. 그러므로 타이컴 설계, 개발 과정에서 divide-and-conquer 형태의 전략을 택할 수 밖에 없었다. 시스템, 하드웨어, 시스템 소프트웨어, 응용 소프트웨어 각 분야에 걸쳐 전문가들이 모여서 그들이 생각하는 시스템 설계 개념을 제시하고, 이를 토대로 최종 시스템 설계를 완성하였다. 그러나 각 분야의 전문가들은 시스템 차원에서 협동하여 시스템 설계서를 완성했지만 각각의 서브시스템 설계는 상당히 독립적으로 이루어졌다.

이와같이 목표시스템을 설계하고, 구현 및 통합하여 최종 시험 과정에 이르기까지의 전과정이 철저하게 Middle-out 방식에 따라 이루어졌다. 이는 전통적인 시스템 개발 방식인 Top-down 방식이나 Bottom-up 방식을 택한 것이 아니라 이를 혼합한 중간 형태를 택한 것이다. 이러한 방식으로 타이컴 개발 사업을 수행한 것은, 아주 유능한 컴퓨터 시스템 설계 전문가가 없었기 때문이거나 컴퓨터 시스템 자체가 갖고 있는 구성 요소들의 강한 독립성 때문이라고 생각된다.

4. 2 부분 개발, 부분 기술도입

대규모 시스템이 개발될 때 그 시스템을 구성하는 모든 요소가 개발 사업을 담당하는 곳에서 개발되지는 않는다. 그보다는 시스템을 구성하는 주요 요소는 개발 주체가 직접 개발하고 그 이외의 요소는 선택적으로 외부의 것을 기술 도입하든지 단순히 구매하여 사용한다.

타이컴 개발 사업은 이 점에서 전형적인 컴퓨터 시스템 개발 방식에 따라 진행되었다. 특히 컴퓨터 기술 개발은 사용자의 요구라는 측면보다는 개발자의 새로운 기술 개발(예를 들면 펜티엄 마이크로프로세서, 윈도우 95의 운영체제 등)에 의

해 주도되므로 기술 표준의 개방화에 대한 세계 기술 추이를 따라야 하며, 각 요소 기술별 세계적 수준과의 격차가 상이하게 다르기 때문에 각 요소들에 대한 개발 전략은 달라야 한다.

◦ 독자 개발의 경우

기술 이전이나 노하우(Know-how) 공개를 꺼리는 부분은 독자적으로 개발한다. 예로 컴퓨터 시스템 아키텍처 설계 분야를 들 수 있다.

◦ 구매의 경우

기술적인 격차가 심하거나 굳이 개발하기보다는 구매하여 사용하는 것이 개발 기간의 단축이나 연구개발비의 절감 측면에서도 용이한 부분은 그대로 구매하여 개발 시스템에 장착한다. 중앙처리장치(CPU), 디스크 드라이브 등을 예로 들 수 있다.

◦ 부분 도입 부분 개발

거의 국제적으로 표준화·개방화되었으며, 우리나라와 현격한 기술 차이가 있지만 그대로 국내에서 사용하기에는 맞지 않는 분야의 기술들은 도입하여 개량화하여 사용한다. 운영체제와 같이 한글화, 기능적 개선 등을 통해 개발하는 분야가 있다.

타이컴의 경우 탑재되는 중앙처리장치 자체(MC68030 사용)를 제외한 하드웨어는 독자적으로 개발되었고 운영체제로서 국제 표준인 UNIX가 기술도입되어 다중프로세서용으로 수정, 보완된 후 타이컴에 탑재되었다. 프로그래밍 언어를 비롯한 대부분의 시스템 소프트웨어들은 미국의 소프트웨어 개발 회사들로부터 도입되었다.

부분 개발, 부분 기술도입 전략은 타이컴 개발 사업이 시작되던 당시에 우리나라 컴퓨터 산업계가 기술적으로 충분히 성숙해 있지 못하였고, 따라서 이 분야에서 유능한 엔지니어를 찾기도 쉽지 않았기 때문에 불가피하게 선택한 전략이었다. 그러나 이러한 부분 개발 전략은 기존의 시스

템 공학에서 다루고 있는 시스템 개발 방법을 그대로 사용하기 어렵게 만들었다. 왜냐하면 일반적인 시스템 개발 방법들은 전혀 새로운 시스템을 사용자 요구사항부터 새롭게 정의하여 마지막으로 시스템을 통합, 시험하기까지 모든 부분을 개발하는 경우를 가정하기 때문이다[9, 11]. 이러한 이유 또한 기존의 라이프 싸이클 모델을 그대로 적용하지 못하는 한 예가 될 것이다.

4. 3 시스템 요소별 연구 인력 투입 비율

타이컴은 크게 하드웨어와 소프트웨어로 나누어 생각해 볼 수 있다. 그러나 컴퓨터와 같이 여러 구성 요소를 갖는 시스템을 개발하는데에는 반드시 시스템 차원의 업무가 발생한다. 여기에는 시스템 설계, 통합, 시스템 시험, 성능 평가, 개발 환경 정립 및 개발 사업 관리 등의 업무가 포함된다. 이러한 업무의 수행 결과는 시스템을 구성하는 요소가 되지는 않지만 전체 시스템이 완성되는데 반드시 필요한 업무이다.

타이컴을 개발하는 과정에서 시스템/하드웨어/소프트웨어 분야에 투입된 인력은 대략 1:1:2를 보이고 있다. 이 비율은 사업 전체 기간동안 정확히 유지된 것은 아니지만 사업 종료 시점에서는 정확히 이 비율을 유지하고 있었다. 흥미로운 사실은 1:1:2 비율이 TDX 개발 사업에서도 똑같이 유지되었다는 점이다. 이는 타이컴 개발 방법론이 TDX의 그것을 원용하였기 때문에 발생한 자연스러운 결과이거나, TDX와 TICOM이 모두 전자분야의 시스템이라는 점에서 공통점을 찾을 수 있다. 그러나 일반적인 시스템 연구 개발 사업에 대한 자료를 좀더 수집해서 분석해 보면 이 비율은 일반화될 수 있는 가능성이 있다고 판단된다.

5. 결론

타이컴 개발 사업은 시스템 공학적인 측면에서 몇 가지 흥미있는 특징을 보여주었다. 그것은 컴퓨터 시스템 구조를 표현하는 방식으로 III자 모양의 Nested Layer Model을 설정하여 설명하는 것이 유용하다는 점과, 타이컴 개발 사업에서 시스템, 하드웨어 및 소프트웨어 부분에 대한 연구 인력 배분비가 1:1:2였다는 점인데 이 비율은 타 연구 개발 사업의 예를 좀더 수집해서 분석해 보면 일반화될 수도 있어 보인다. 또 하나는 기술적으로 크게 앞서 있지 못한 국가에서 대규모 연구 개발 사업을 추진할때 국내 개발과 기술 도입을 적절히 혼합하여 수행하는 방안이 성공적일수 있는 예를 보여주었다는 점이다.

마지막으로 대규모 연구 개발 사업에서 가장 중요시해야 하는 면은 인간적인 측면이다. 한 개인이나 팀이 팀 동료나 다른 팀과 고립되어 연구 개발을 수행해 나가면 대부분의 경우 아무리 뛰어난 능력을 가진 연구원이라도 자신의 임무를 수행하기 어려웠다는 사실이다. 이는 원만한 인간 관계가 사업의 성공적인 수행에 가장 중요한 요소가 되었음을 의미한다.

본 타이컴 개발 방법론은 단순히 부분적인 요소들을 통합하여 개발하는 연구사업보다는 독립적인 요소가 강한 서브시스템들이 모여 개발 시간 추이면에서나 구조면에 있어서 계층적인 성격을 갖는 규모가 큰 시스템을 개발하는 환경에 적용될 수 있으리라 생각한다.

연구 개발 사업도 점차 대형화, 복합화되어 가는 경향을 보이고 있다. 타이컴 개발 사업 수행 과정에서 개발된 시스템 공학적 접근 방법이 이와 유사한 연구 개발 사업에서는 어떻게 진행되었는가에 대한 자료를 수집하는 것이 필요하다. 이러한 자료들을 바탕으로 대규모 연구 개발 사

업 수행을 위한 시스템 공학적인 공통 부분을 정립하면 앞으로 이와 유사한 사업을 추진하는 연구팀에게 보탬이 될 것이다.

끝으로 타이컴은 개발이 완료된후 1992년 초부터 상기 4개 기업은 6개월여의 상용화 작업을 완료하고 국내 시장에 시판하기 시작했다. 타이컴은 국내 중형컴퓨터 시장에서 1995년 3월 현재 500대를 넘는 판매 실적을 보이고 있어 일단 성공적인 제품으로 평가받고 있다. 이러한 타이컴의 성공적인 개발에 이어 ETRI와 참여 4개사는 계속해서 주전산기 III(고속중형 컴퓨터)을 개발 완료했고, 이어서 주전산기 IV(고속병렬컴퓨터, SPAX)를 94년 2월부터 개발하고 있으며 이에 따라 시스템 공학적인 측면의 공학 기술도 함께 발전하고 있다[4, 5, 6].

참고 문헌

- [1] 박진원, "TDX와 TICOM 개발체계 비교", 「전자통신」, 제14권, 제4호, (1991), pp.88-96.
- [2] 박진원, 천유식, "TICOM 구조 및 성능", 「전자공학회지」, 제18권, 제7호, (1991), pp. 31-39.
- [3] 이준석, 박진원, "컴퓨터시스템 개발과 라이프사이클 모델", 「전자통신」, 제13권, 제3호, (1991), pp.25-32.
- [4] 이준석, 박진원, "고속병렬컴퓨터 개발을 위한 개발 방법론", 「대한산업공학회 '94추계학술대회 발표논문집」, (1994), pp249-256.
- [5] 이준석, 박진원, "컴퓨터시스템 개발을 위한 통합 형상관리체계", 「정보처리」, 제1권, 제3호, (1994), pp.39-47.
- [6] 이준석, 박진원, "주전산기 개발을 위한 계층적 라이프사이클 모델", 「한국경영과학회/대한산업공학회 '95추계학술대회 발표논문집」,

- (1995), pp.376-382.
- [7] 천유식, 시스템 개발 방법론, 컴퓨터 월드, (1991), pp243-261.
- [8] Agresti, W. W., *New Paradigms for Software Development*, IEEE Computer Society Press, 1986.
- [9] Blanchard, B. S., Fabrycky, W. J., *System Engineering and Analysis (2nd ed.)*, Englewood Cliffs, N. J.: Prentice-Hall, (1990), pp.16-33.
- [10] Boem, B. W., "Software Engineering," *IEEE Transactions on Computer*, Vol. C-25, No. 12, (1976), pp.1226-1241.
- [11] Fairley, R. E., *System Engineering Concepts*, McGraw-Hill, (1985), pp.52-53.
- [12] Lantz, K., *The Prototyping Methodology*, Englewood Cliffs, N. J.: Prentice-Hall, (1986).
- [13] Ramamoorthy, C. V., "Software Engineering: Problems and Perspectives," *Computer*, (1984), pp.191-209.
- [14] Rock, P., "Controlling Software Projects," *Software Engineering Journal*, (1986), pp.108-117.
- [15] Royce, W. W., "Managing the Development of Large Software Systems," *Proceedings of IEEE WESCON*, (1970), PP.1-9.