□정보산업동향□

Y2K(밀레니엄 버그)

최 성†

◈ 목 차 ◈

요 약

인류의 역사는 앞으로 400여일이 지나면 새로운 천년을 맞이하게 된다. 세계는 2000년을 기점으로 도약하기 위하여 VISION 2000을 제시하는 등 세기말을 바쁘게 지내고 있다. 그러나 희망으로 다가오는 2000년은 시한폭탄이 되어 다가오고 있다. 그것은 산업사회에서 정보화 사회로 이전되면서, 우리가 모르는 사이에 디지탈 시대에 살게 되었기 때문이다. 디지탈의 세계에서는 메모리의 절약과 표현의 간소화를 위하여 연도표기를 마지막 2자리 숫자로 사용하여 왔다. 그로인해 2000년이 되면 컴퓨터에서 처리되는 연도가 '00'으로되어 1900년과 구별되지 않게되었다. 은행거래, 각종 공과금 계산의 오류는 물론이고 산업계 전역에 설치되어 있는 자동화기기의 RTC(Real Time Clock)오동작으로 국가 기반시스템까지 마비될 가능성을 파생시켰다. 이러한 Y2K문제(일명 : 밀레니엄 버그)는 크게 하드웨어, 시스템 소프트웨어,애플리케이션, 비정보처리계시스템(자동화기기)등 4가지 종류에서 발생하고 있다. 본고에서는 이들 4가지 종류에맞는 해결안을 마련하고 실현하는 구체적인 계획을 제시함은 물론 이어 최종적으로 이를 다시 통합해서 테스트하는 3단계방식으로 Y2K문제를 해결하도록 제안하고자 한다.

1. Y2K의 발생배경 및 해결 필요성

1950년대 정보화가 시작된 이래 컴퓨터의 보급은 급격히 증가하게 되었고, 산업전반의 변화도 빠른 속도로 진행되고 있다. 당시에는 컴퓨터의 메모리 및 디스크의 비용이 매우 비싸 이를 효과적으로 사용하기 위한 노력 또한 다양한 방법으로 연구되었다. 컴퓨터를 활용한 정보시스템의 구축도 이때부터 진행되어 왔으며, 고가의 메모리를 절감한다는 점과 컴퓨터의 생명주기(Life Cycle)가 최대 10년이므로 10년 이내에 재구축한다는 점을 고려하여 데이타 등의 구축시 4자리 연도표기를 앞의 2자리를 생략하고 뒤의 2자리만 표기하게 되었다. 당시 연도표기 형태를 2자리로 표현한 정보시스템에서 4자리 날짜표기로 바꾸는 것은 매우 간단하다고 생각하여 2자리 표기논리는 계속되었다. 그러나 21세기인 2000년대에는 1998년을 98년으로 표기하는 식의 방법 사용이 불가능해지기 때문에 여러가지 문제점이 발생한다.

- ○ 모든 조회화면이나 보고서 상에서 2000년은 00으로 표시된다.
- ○ 00은 99보다 작은 수로 인식되어 대부분의 정렬(Sort) 프로그램이 제대로 작동되지 않을

† 종신회원 : 남서울대학교 전자계산학과 교수

것이며, 두자리 연도를 사용하는 모든 비교나 연산조작이 엉뚱한 결과를 산출한다.

- ○ 많은 프로그램들이 99보다 큰 연도는 없다고 가정하거나 서기 2000년을 윤년으로 처리하지 못하거나 혹은 00을 잘못된 입력으로 간주하여 에러로 처리한다.
- ○ 21세기를 미리 가정하여 아예 '19'가 프로그램에 내장되어 사용되고 있다.

정보화는 점차 자원의 공유, 클라이언트/서버 개념의 도입, global network화 되어가는 추세에 따라 정보시스템의 변경은 타 정보시스템에도 영향을 미치게 되고 이로인한 파급효과가 발생하고 있다. 2000년 문제의 특성은 확실한 deadline이 설정되어 있고 시간은 얼마남지 않은 상황에 있다. 따라서 문제해결을 위한 수요자는 짧은 시간에 급격하게 증가될 것이며 이에 대한 공급 가능한 기술자의 수는 제한적이어서 비용 또한 급격히 증가할 것이다.

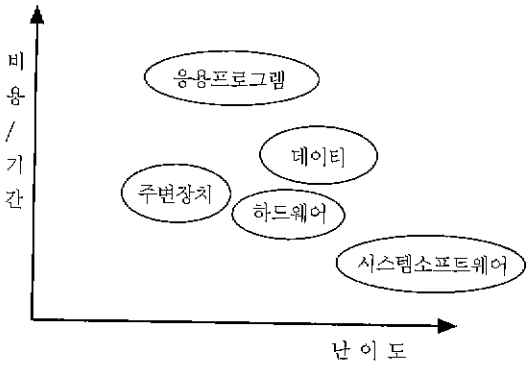
조사회사인 가트너그룹의 분석에 따르면 이 문제를 해결하는 데에는 프로그램 한 라인당 평균 $0.5~$1.0이상의 비용이 소요되어 서기 2000년까지 전세계적으로는 최고 6천억달러 정도의 엄청난 비용이 필요할 것으로 예상되고 있다. 이러한 문제점들은 COBOL이나 파일 구조에 국한된 문제가 아니라 심지어는 시스템 클럭까지 연관이 있고, 서기 2000년 1월 1일 00:00 이전까지는 반드시 해결되어야 한다. 이처럼 모든 시스템이 피해 갈수 없다는 사실이 문제의 심각성을 더해주고 있다. 1999년을 시험운영기간으로 간주할 때 이제 남은 시간은 3개월에 지나지 않는다. 그런데도 아직까지 문제에 대한 인식을 하지 못하고 있는 이용자도 있다. 2000년의 문제는 H/W, 주전산기, 패키지소프트웨어, 응용소프트웨어, 통신, 자동화기기, 실험기기 등 다양한 분야에서 발생될 수 있으며 주어진 시간에 이를 해결하기 위해서는 주도면밀하게 계획을 수립해야 한다.

1.1 Y2K 문제의 유형

정보시스템의 연도문제를 해결하지 못했을 경우 대 고객관련 은행업무 및 보험서비스 업무에서는 이자계산의 오류, 각종 연산 결과의 오류, 만기 보험료 산정의 오류가 발생한다. 공공 및 행정 서비스 업무에서는 날짜와 관련한 대소의 비교 오류, 연도출력의 오류, 세금계산의 오류 등이 발생한다. 기타 일반 산업부문에서도 각종 통계자료의 오류, 단위시스템간 통합성 파괴, 퇴직금 계산의 오류 등이 발생할 수 있다. 주민관리시스템의 경우 1950년생이 2001년이 되는 경우 나이는 01-50=-49서 태어나기 49년전을 의미하며, 120세의 할아버지가 2000년에 20세로 인식되어 병역통지서를 고지 받을 수도 있다. 또한 106세의 할머니가 2000년에 6세로 인식되어 입학통지서를 고지받을 수도 있을 것이다. 부동산관리시스템의 경우 1998년에 A에게서 B가 소유권을 양도받은 후 2000년에 소유권이 B에서 C로 양도될 경우 뒤의 2자리 비교시 98 > 00 이므로 소유권이 B에게 있는 것으로 인식하여 소유권의 처리에 오류가 발생한다. 부동산관리시스템의 화면이나 통계자료 입력시 '00'은 1900으로 들어가도록 프로그램이 되어있어서 통계자료의 추출 또는 프로그램내의 연도를 체크할 때 잘못된 결과가 발생될 수도 있다. 자동차관리시스템의 경우에서는 1997년에 자동차 등록 후 계속 세금을 내지않고 2000년에 자동차를 팔았을 때 세금계산은 00 - 97 = -97로서 국가에서 자동차 주인에게 오히려 97년에 해당하는 세금을 주어야 하는 것이다.

1.2 요소별 문제의 중요성 파악

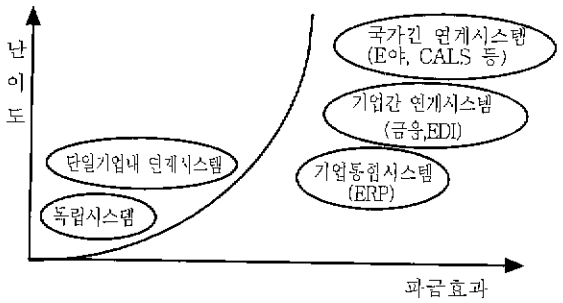
정보시스템에서 연도문제로 발생될 수 있는 요소는 전반적이라고 볼 수 있다. 하드웨어적인 문제, 시스템 소프트웨어, 응용프로그램, 데이터, 주변장치, 패키지 등의 문제해결을 위한 비용, 난이도를 나타낸 것이 (그림 1)이다.



(그림 1) 정보시스템 구성요소별 연도문제 해결 난이도 및 비용곡선

(그림 1)을 살펴볼 때, 응용프로그램이 비용과 기간에서 많은 비중을 차지하고 있으며, 시스템 소프트웨어의 경우는 비용과 기간은 적게 소요되나 난이도가 큰 것으로 분석되고 있다.

응용시스템을 유형별로 난이도 및 영향도를 분석한 것이 (그림 2)이다. (그림 2)을 살펴볼 때, 독립된 시스템 내의 응용프로그램의 경우 비교적 간단하다고 볼 수 있으나 CALS, EDI 등 국가간에 연계하여 사용하는 응용시스템의 경우 가장 복잡하여 문제해결의 실패로 인한 영향은 매우 클 것으로 판단된다.



(그림 2) 응용프로그램 성요소별 연도문제 해결 난이도 및 비용곡선

2. Y2K 기술적 해결방법

Y2K를 기술적으로 해결하는 방법을 나타낸 것이 <표 1>이다.

<표 1>의 4가지 방법 중에서 상대날자방법과 압축방법은 데이터를 access하는 모든 프로그램을 변경해야하는 등 장점보다 단점이 훨씬 많아 확장방법과 연도창방법만을 생각해 보도록 하겠다. 확장방법과 연도창방법의 장단점을 비교한 것이 <표 2>이다.

〈표 1〉 Y2K의 기술적 해결방법

접근방법	내 용
확장방법 (expansion)	- 수정방법(Conversion)이라고도 함 - 데이터파일의 연도필드를 두자리에서 네자리로 확장하는 방법으로 "년월일"을 표시하는데 총 8바이트가 소요됨 - 데이터파일의 크기가 커지고 해당 프로그램을 이에 맞도록 수정해야 함 - 면도문제를 근본적으로 해결가능한, 가장 확실한 방안이나 기존자료(디스크, 테이프)를 재구성하여야 하므로 막대한 비용과 시간이 소요됨 - 시스템 및 관련시스템, 프로그램에 연쇄적인 영향이 발생하므로 변환시 철저한 inventory 및 영향분석이 준비되어야 함
연도창방법 (windowing or sliding scale or algorithm or procedural workaround)	- 연도표시는 두자리를 그대로 두고 네자리 연도로 해석하도록 프로그램 코드를 수정하는 방법으로 "년월일"을 표시하는데 총 6바이트가 소요됨 - 예를 들어 1950~2049년 등으로 연도의 범위를 정하고 YY가 50보다 크면 19nn이고 50보다 작으면 20nn으로 처리하는 방법 - 프로그램에서 연도표시하는 부분을 찾아 수정하므로 기존 데이터를 재구성할 필요없이 그대로 이용함으로 작업량이 적어 경제적임 - 시스템에 주는 영향이 가장 적지만, 다양한 형식의 연도를 구별하고 처리할 수 있는 정확한 연도변환 모듈이 필요함 - 100년내 연도만 처리 가능으로 문제소지는 남음
상대날짜방법 (relative dates)	- 특정일자로 부터의 날짜수를 포함하도록 파일을 수정하는 것으로 현재사용하는 형식과 동일하게 6자리를 사용함 - 예를 들어오늘은 1900년 1월1일부터 nnnnnn번째 날로 표시하고 이러한 날짜수를 YYYYMMDD의 형식으로 바꾸어 주는 간단한 변환모듈이 필요함 - 로직의 처리, 날짜 정렬 등의 처리 방법이 간단하고 파일의 확장이 필요없으나 동일한 기준날짜가 필요하게 됨
압축방법 (encapsulation)	- 데이터 파일의 두자리 연도필드에 네자리의 연도데이터를 16진수 따위로 코드화하거나 압축하여 저장하는 방법으로써 "년월일"을 표시하는데 총 6바이트가 필요함 - 적당한 코드화/압축 알고리즘을 찾아야하며, 데이터와 프로그램 모두의 수정이 필요함

〈표 2〉 Y2K 기술적 해결방법의 장단점 비교

접근방식	장 점	단 점	선 택 기 준
확장방법 (데이터 기초방식)	o 영구적이다. o 알고리즘변경 거의 없음 o 향후 유지보수비용 절감 o 프로그램 일관성유지	o 데이터 변환은 필수 o 부가적인 저장소가 필요할 수 있음 o 스크린, 보고서가 수용하기 어려울 수 있음 o I/O 증가로 추가적 튜닝이 필요할 수 있음	o 대부분의 경우 선택 (시간과 비용에서 가능할 경우)
연도창 방 법 (프로세스 기초방식)	o 데이터 변환이 필요없음 o 변환시간이 비교적 짧음 o 비교적 적은 비용투입	o 프로그램 로직변경 불가피 o 에러발생 가능성높음 o Sort기능 수행 불가 o 최대100년 범위내 가능 o 향후 유지보수 비용 높음 o 프로세스의 일관성상실 우려가 있음 o 철저한 시험 필요	o 시간/비용 부족시 o 업그레이드 계획이 있고, o 가장 오래된 데이터가 2000년 기준 100년이 되지 않고 100년안에 업그레이드 계획이 있을 경우 (가령 가장 오래된 데이터가 1910년일 경우 2010년까지 업그레이드 되어야함.)

3. 시스템 요소별 해결방법

3.1 PC 분야

1) 하드웨어 : Real Time Clock (RTC) 칩에서 년도를 2자리의 BCD로 표시하고 별도로 Century 필드를 2자리의 BCD로 가지고 있어 Century 필드가 시스템 clock에 동기되어 동작하지 않는다 따라서 시간의 경과에 따라 19에서 20으로 자동으로 올라가지 않아 발생하는 문제이다. 이는 2000년에 BIOS나 사용 운영체제인 DOS / WINDOWS-95에서 이를 강제로 Century 필드를 설정해야 한다.

2) 시스템 BIOS 문제

Century 필드는 시스템 clock에 동기되어 동작하지 않으므로, 2000년에 BIOS에서 자동 혹은 수동으로 설정해야 한다. 최근에 생산된 PC의 경우는 BIOS에서 이 문제를 자동으로 해결한다. 다음과 같은 간단한 테스트로 확인할 수도 있다.

* [RTC Rollover Test]

① DOS (혹은 WINDOWS-95) 프롬프트 상에서 date 명령어를 사용하여 현재 날짜를 1999년 12월 31일로 설정한다. ('date 1999-12-31'라고 입력 후 엔터키를 누름)

② DOS (혹은 WINDOWS-95) 프롬프트 상에서 time 명령어를 사용하여 현재 시간을 23시 58분으로 설정한다('time 23:58:00.00'라고 입력 후 엔터키를 누름)

③ 컴퓨터의 전원을 끈 뒤 3분 정도 기다린 후 전원을 다시 켠다

④ DOS (혹은 WINDOWS-95) 프롬프트 상에서 date 명령어를 사용하여 날짜를 확인한다 ('date'라고 입력 후 엔터키를 누름르면 2000년 1월 1일로 되어 있어야 함)

* [RTC Set Test]

① DOS (혹은 WINDOWS-95) 프롬프트 상에서 date 명령어를 사용하여 현재 날짜를 2000년 1월 1일로 설정한다 ('date 2000-01-01'라고 입

력 후 엔터키 누름)

② date 명령어를 사용하여 날짜가 설정되었는지 확인한다('date'라고 입력 후 엔터키를 누름)

③ 컴퓨터의 전원을 끄고 3분 정도 기다린 후 다시 전원을 켠다

④ date 명령어를 사용하여 날짜를 확인한다 ('date'라고 입력 후 엔터키 누름, 2000년 1월 1일로 되어 있어야 함)

RTC Rollover Test를 성공했을 경우는 BIOS에서 자동으로 설정해주는 경우이므로 2000년 문제가 발생하지 않으며, RTC Rollover Test는 실패하고 RTC Set Test만 성공했을 경우에도 2000년 1월 1일에 년도를 수동으로 설정하면 사용상에 문제가 없다. 그러나 두가지 모두 실패했을 경우에는 2000년에 사용이 불가능하기 때문에 해당 공급사와 협의하여 해결해야 한다.

3.2 주전산기분야

대부분의 UNIX machine에서 날짜를 읽거나 설정할 때 직접 RTC 칩에 접근한다. UNIX의 time() 시스템 호출은 time_t 변수로 현재 시간을 리턴하도록 되어있으며, time_t 변수값은 1970년 1월 1일 0시부터의 초 카운트를 32비트의 signed integer로 나타내고 있다. 따라서 2^31 - 1 = 2147463647(초) = 68(년)이므로 기준년도를 1970년도로 설정했을 때 1970 + 68 = 2038(년), 즉 2038년까지는 사용이 가능하며, 또한 그 시점을 기준으로 다시 기준년도로 계산하기 때문에 전혀 문제가 없을 것으로 예측된다. 그러나 유닉스 계열이 아닌 IBM기종, TANDEM, DEC, MV, CYBER, PRIME, VAX 등의 경우는 공급사와 협의하여 문제를 해결할 필요가 있다. 또한 시스템 소프트웨어의 경우, 주전산기 공급사로 부터 문제해결에 대한 지원을 받는 것이 바람직하다.

3.3 패키지소프트웨어분야

각종 패키지 소프트웨어의 경우도 연도문제가 있는 것으로 파악되고 있다. 대부분의 경우 공급사에서 본 문제와 관련하여 해결전략 및 계획을 수립하여 진행중에 있으며, 금년말까지 대부분 해결할 것으로 예측되고 있다. 각종 응용프로그램에서 사용하는 패키지 소프트웨어는 매우 중요한 미들웨어의 성격을 가지고 있어 사용자 임의로 고칠수 있는 사안이 아니며, 최대한 빠른 시간내에 공급사로 하여금 해결토록 요청해야 한다.

3.4 응용소프트웨어분야

가장 비용이 많이 들고, 시간도 많이 투입되어야하는 가장 큰 비중을 차지하고 있는 부분이다. 사용자 응용소프트웨어 이외의 경우는 대부분 공급사가 있어서 이를 해결하여 줄 것이나, 응용소프트웨어의 경우는 그렇지 않다고 볼 수 있다. 최종 주어진 시간을 고려할 때 시행착오는 곧 실패를 의미하므로 실수가 없도록 최적의 방법론을 적용하여 추진하는 것이 요구된다. 큰 정보시스템의 경우 이를 모듈화하여 전체의 팀을 보조하는 형태의 팀을 구성하여 추진하는 것도 중요한 방법이다.

3.5 클라언트/서버시스템분야

일반적으로 클라언트/서버 시스템은 업무 단위로 분산되어 있을 뿐만 아니라, 대부분 최근에(보통 2~3년이내)개발되어, 상대적으로 문제가 덜 심각할 것으로 예상된다. 그러나 하나의 클라이언트에서 잘못이 발생할 경우 그 영향이 다른 시스템까지 연쇄적으로 미칠 수 있고, 메임프레임과는 달리 시스템 전반에 걸친 분석 및 발생에 대처할 유용한 수단이 별로 없으므로, 발생 가능한 문제점들을 주의깊게 살펴보고 확인하여야 한다.

- 서버의 DB에서는 연도가 4 Digit로 표시되어 있으나, 클라이언트 프로그램에서는 2 Digit

로 사용되지는 않는가?
- 구형 PC인 경우, BIOS가 4Digit의 연도 표시를 지원하는가?
- 메인프레임과 연결된 3-TIER 클라이언트/서버 시스템인 경우, 호스트용 프로그램이나 파일이 수정되면 그 영향은 제대로 반영되는가?
- 서기 2000년을 윤년으로 처리하지 않는 프로그램은 없는가?
- 수정된 프로그램의 배포 및 관리는 적절하게 이루어지고 있는가?

4. 날짜표기 표준연구

4.1 국내 표준화 현황 및 적용방향

국내의 경우는 한국공업규격 KSC 5610 - 1992에 날짜 및 시각 표시 코드 (Identification code of dates and times)를 명시하고 있으며 국제표준 ISO 8601-1988을 수용하고 있다. 이 내용에서는 적용사례에서 하루를 역날짜로 정확하게 표시하기를 요구하는 경우 연도표시의 완전표시코드로 CCYYMMDD를 사용하도록 권고하고 있으며, 적용사례에따라 2개, 4개 혹은 6개의 숫자를 생략하여 표현하는 정확도를 줄인 표시코드로 CCYY- MM, CCYY, CC 등을 사용할 수 있도록 규정하고 있다. 적용사례에서 절단된 표시코드가 요구되는 경우, 절단표시코드는 YYMMDD, -YYMM, -YY, --MMDD, --M, ---DD등으로 사용하도록 권고하고 있다. 단기날짜표시코드를 단기지정자 [K]를 사용하여 [KCCYY]를 포함하여 표현하고 있으며, 태음력 날짜표시일 경우 태음력지정자 [N]을 사용하여 상기 네가지 방법을 이용하고 있다.

4.2 국외 표준화 현황

1) 국제표준 ISO 8601-1988에서는 날짜와 일일 시각에 관한 정보의 수치적 표기에 대한 명세를 포함하고 있는데, 서력날짜, 서수날짜, 주 수에 의해 인식되는 날짜, 시간주기, 일일 날짜와 시각의 조합, 지방시각과 세계시각과의 차의 표기에 대한 유사한 형식에 대한 표기방법을 기술하고 있다. 적용사례에서 하루를 역날짜로 정확하게 표시하기를 요구하는 경우 연도표시의 완전표시코드로 CCYYMMDD를 사용하며, 적용사례에 따라 2개, 4개 혹은 6개의 숫자를 생략하여 표현하는 정확도를 줄인 표시코드로 CCYY-MM, CCYY, CC 등을 사용하고, 적용사례에서 절단된 표시코드가 요구되는 경우, 절단표시코드는 YYMMDD, -YYMM, -YY, --MMDD, --M, ---DD등으로 사용할 수 있다고 규정하고 있다.

2) FIPS PUB 4-1 에서는 정보시스템간 자료교환을 용이하게 하기위한 연도날짜와 서수날짜의 표기방법에 대하여 알려 준다. 이 표준은 ANSI X3.30-1985(정보교환을 위한 연도날짜와 서수날짜 표기)를 적용하고 있다. 이 개정판은 FIPS PUB 4를 완전히 대체한다. FIPS PUB 58-1(정보교환을 위한 일일 지역시간 표기), FIPS PUB 59(정보교환을 위한 세계시각, 지역시간차등, 미국시간대의 표기), ANSI X3.30-1985(정보교환을 위한 달력날짜 및 서수날짜표기), ANSI X3.43-1986(정보교환을 위한 일일 지역시간 표기), ANSI X3.51-1975(정보교환을 위한 세계시간, 지역시간차등, 미국시간대참조의 표기), ISO 2014-1976(모든 수치형태에서의 달력날짜 표기), ISO 2711-1973(정보처리교환-서수날짜의 표기), ISO 3307-1975(정보교환-일일시간의 표기), ISO 4031-1978(정보교환-지역시간차등표기) 등을 참조하며, ISO 2014-1976(모든 수치형태에서의 달력날짜 표기), ISO 2711-1973(정보처리교환-서수날짜의 표기), ISO 3307-1975(정보교환-일일시간의 표기), ISO 4031-1978(정보교환-지역시간차등표기)는 ISO 8601-1988에 흡수된다.

2000년 연도표기 문제와 관련하여 이 표준은

미국의 정부기관들 사이에 교환을 목적으로 하는 어떠한 형태의 전기데이타에 있어서, NIST는 네자리의 연도요소를 사용할 것을 강력히 추천하고 있다. 연도는 앞의 두자리로 세기를 표시해야 하고, 뒤이어 두자리는 연도를 표시해야 한다고 명시하고 (예:1999,2000등) 더불어 ANSI X3.30-1985 (R1991)에 명시되어 있는 생략가능한 두자리 연도시간요소는 미국정부기관간 어떤 자료교환의 목적에서도 사용되지 않아야한다고 명시하고 있다.

3) FIPS PUB 58-1 에서 12시간, 또는 24시간 작업시간계를 갖춘 시스템에 기초한 단일 시간표기를 제공한다. 이것은 데이터시스템간의 정보교환을 목적으로하는 디지탈형태의 일일 지역시간 표기 방법을 제공한다. 또한, 시간요소와 일련의 배치, 시간요소간의 분리자사용 등을 명시한다. 이표준은 ANSI X3.43-1986을 적용한다. 이개정판은 FIPS PUB 58을 완전히 대체한다.

4) FIPS PUB 59에서는 ANSI X3.51-1975를 적용한다. 이 표준은 데이터시스템간 정보 교환에 활용하기 위하여 세계시각, 지방시각차, 미국 시간대참조의 표기를 위한 수단으로 제공된다.

5) ANSI X3.30-1985에서는 정보시스템에 대하여 정보교환을 위한 서력날짜와 서수날짜에 대한 표기방법을 명시(for information systems - repre- sentation for calendar date and ordianl date for information interchange)하고 있다. 역연도는 몇 세기인지를 알 수 있는 응용프로그램 내에서 앞의 두자리를 생략할 수도 있는 네자리로 표기되어져야 한다. 이같은 방법으로 네자리 연도는 아래 두자리만으로 사용할 수 있다. 연도가 두자리로 표기되어져 있는 경우 세기연도란 용어가 표기를 인식하는데 사용되며, 한자리로 표기되어있는 경우 십년대연도란 용어가 사용된다.

6) ANSI X3.43-1986는 정보시스템에 대하여 정보교환을 위한 일일 지방시각에 대한 표기(for information systems - representation for Local Time of Day for information interchange)로서 이 표준은 12시간, 24시간의 시간운영 시스템에 기초한 단일화된 시각표기를 설정하기위해 설계되어졌다. 이것은 자료시스템간 정보교환을 목적으로 디지털 형태의 일일 지방시각을 표기하는 방법을 제공한다.

○ 날짜와 시간표기의 결합 : 이 표준은 ANSI X3.30-1971과 함께 사용하기 위하여 설계되었다. 높은순위에서 낮은순위로, 즉, 연, 월, 일, 시, 분, 초로 배치되어져야 한다. 분리자는 필요하지 않으며 자료처리시스템간 교환을 위하여 사용될 때 날짜와 시간을 분리하여 사용해서는 안된다. 그러나 사람의 이해를 돕기위하여 분리자를 사용할 때는 하이폰이나 공백이 날짜의 낮은순위요소와 시각의 높은순위 요소사이에 사용된다. 시각표기 "141236"은 서력날짜 "1971-09-01"과 함께 "19710901141236"으로 표기되어지거나 분리자사용을 하여 "19710901-141236"으로 표기되어짐

○ ANSI X3.51-1975에서는 정보교환을 위한 세계시각, 지방시각차, 미국시간대 참조의 표기(representations of universal time, local time differentials, and United States time zone references for information interchange)를 제공한다. 이 표준의 목적은 자료시스템간 자료교환을 함에 있어 세계시각, 지방시각차, 미국시간대 참조를 표기하는 표준방법을 제공하는데 있다.

4.3 국내 적용방안

국내의 연도와 관련한 표현방법은 역일자, 순서날짜, 역주와 일수에 의하여 표현되는 날짜, 단기날짜 표시코드, 태음력 날짜 표시코드 등이 있으며 이들은 다시 완전표시코드, 정확도를 줄인 표시코드, 절단표시코드 등의 방법을 이용할 수 있다. 위 표시코드 중 역일자의 절단표시코

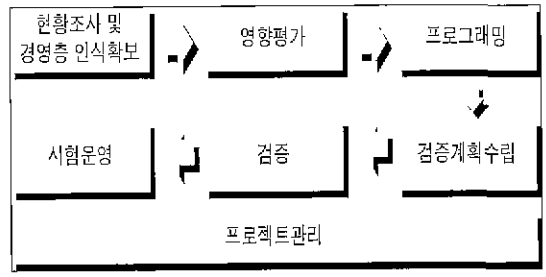
드(기본형식 및 확장형식), 순서날짜의 절단표시코드(기본형식 및 확장형식), 역주와일수에 의하여 표현되는 날짜의 절단표시코드(기본형식 및 확장형식), 태음력 날짜표시시 상기 세가지 방법으로 표현할 경우가 세기를 생략할 수 있게 규정되어 있다. 지방시각, 자정, 표준시각 등과 결합한 시각표현에 있어서는 세기가 생략되어선 안되게 규정되어 있다. 미 연방정부의 경우 연방정부간 또는 연방정부와 공공기관 또는 일반기관과의 정보교환에 있어서 기존의 세기 연도를 생략할 수 있는 조항을 따르지 않을것과 네자리의 연도를 사용하도록 규정하고 있다. 그러므로, 향후 정보시스템의 모호한 연도표기를 방지하기 위하여서는 절단표시코드의 연도 생략에 관한 규정이 삭제되어야 한다. 단체표준 즉 공공부문의 정보시스템 구축시 또는 자료교환을 위한 날짜표시는 네자리의 연도를 사용할 것 등의 표준적용과 기존 시스템에 대한 변환 또는 수정작업의 시한을 정하여 2000년 문제에 대비하여야 한다. 또한 날짜와 관련된 모든 코드들은 주관 부처에서 정리하여 이를 국가Y2K 종합대책반에 통보함으로써 이를 활용하는 모든 사용자들에게 알려지도록 하여야 한다. 현재 주민등록번호에서는 2000년이후에 발생하는 번호는 성별코드에서 남자는 3을 여자는 4를 넣도록 하였다.

4.4 날짜표기 인증적용

2000년 문제는 현재의 정보시스템의 경우는 물론, 미래에 구축할 정보시스템 들의 경우에도 간과하여 발생될 수 있다. 따라서 향후에 정보시스템을 도입할 경우 이러한 연도문제가 없어야 한다는 강력한 구매사양이 고려되고 있으며, 이러한 제재 조치로 미연방정부에서는 Y2K를 적용하지 않은 국가나 기업과는 무역거래를 제한하고 있다. 또한 구매했을 경우 이에 대한 인증 및 시험등이 관련기관에의 요청을 통해 점검되도록 하고 있다.

5. Y2K 해결 모델

5.1 구조적 접근 방법



(그림 3) 연도문제 해결단계

(그림 3)에서 현황조사 및 경영층인식확보단계(awareness)란 2000년 문제를 정의하고 경영층의 지원과 후원을 얻어내고 전담팀을 구성하여 전체적인 해결전략을 수립하여 조직내 모두가 문제를 잘 파악하는 단계를 의미한다. 영향평가단계(assessment)란 조직내에서 2000년 문제가 미칠 수 있는 영향을 평가하고 핵심 업무영역과 핵심 프로세스를 확인하고 핵심 업무영역에 해당하는 시스템을 분석하여 변환에 대한 우선순위를 결정하는 단계를 의미한다. 또한 데이터 교환문제, 데이터의 부족문제, 오류문제 데이터 등을 조정할 수 있는 일관된 계획이 수립되며 필요한 자원이 파악되어 준비되는 단계이다. 프로그래밍단계(renovation)란 해당 플랫폼, 응용프로그램, 데이터베이스, 유틸리티 등의 연도문제 관련 변환을 위한 설계 및 프로그래밍 하는 단계를 의미한다. 검증단계(validation)란 변환된 플랫폼, 응용프로그램, 데이터베이스, 유틸리티 등을 시험 및 검증하고 성능, 기능, 통합시험을 실시하는 단계를 의미한다. 시험운영단계(implementation)란, 변환된 구성요소들에 대한 시험적인 운영을 하는 단계를 의

미한다. 이러한 총체적인 단계를 일관성이 유지될 수 있도록 관리하는 것을 프로그램 및 프로젝트 관리라고 한다.

5.2 일정작성모델

적어도 1998년말 또는 1999년초까지는 검증 및 시험운영을 위한 프로그래밍단계에 이르러야 한다. 현재 국내 공공기관에서는 대부분이 계획수립단계 이전인 인식단계에 머무르고 있어 시급히 본 문제의 해결을 상기와 같은 단계를 적용하여 계획을 수립 및 추진해야 할 것이다. 따라서 문제해결 추진을 위하여 작성하게 될 일정은 17:12:25:4:42 비율로 산정할 것을 권고한다.

6. Y2K 실제 적용연구

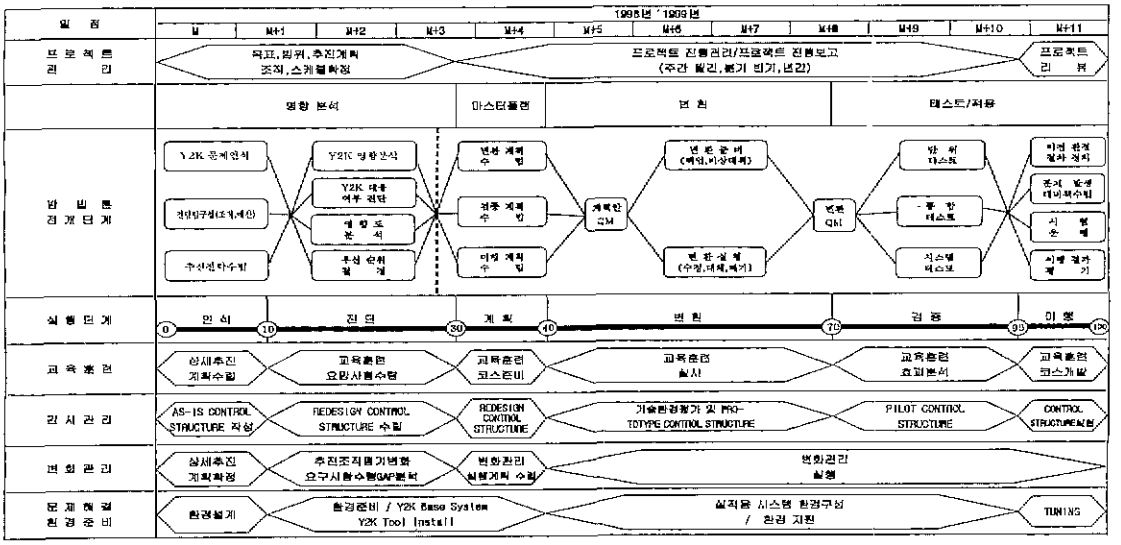
Y2K 문제에 대한 접근 방법을 찾아 고심하고 있는 프로그래머나 전산관리자를 위하여 실제 적용할 수 있는 방안을 시스템 프로우로 나타낸 것이 (그림 4)이다..

6.1 접근단계의 구분

제1단계 : 문제의 이해
- 계획 및 영향 평가(impact analysis)
- 보유 프로그램 조사/분류 및 범위결정
- 사용기술 및 프로그램 대체 전략을 포함하는 변환방법 결정
- 프로젝트 실행계획 수립

제2단계 : 해결방안기술
- 확인(assessment) 및 예측(estimation)
- 프로그램의 구조 및 시스템 구성분석
- 소요 작업량 예측 및 자원소요/제반사항 결정

제3단계 : 문제의 해결
- 실행(implementation) 및 검증(verification)
- 프로그램 및 데이터 수정
- 수정된 시스템의 테스트 수정



(그림 4) Y2K 문제해결 시스템 플로우

6.2 대상 프로그램 확인

- 자체개발한 프로그램 - 사용언어/데이터 파일 등
- 외부 구매한 프로그램 - 공급자가 아직도 계속 지원을 제공하는가, 소스 코드는 보유하고 있는가 등
- PC용 BIOS - 예를 들면 인텔팬티엄 BIOS버전 1.2는 2000년 지원이 어려운 것으로 알려지고 있음
- 마이크로 프로세서 시스템 - 엘리베이터, 구내전화 시스템 등

6.3 예상 소요비용 산정

- 산출기준은 가트너 그룹에서는 라인당 $05~$1.0이상(프로그램당 $450~$600)으로 계상하고 있으며, 엔더슨 컨설팅에서는 100만 라인당 2~4인/년 이상으로 산정하고 있다.
- 단계별 소요비용은 문제의 인식에 15%, 분석에 10%, 수정에 30%, 단위 테스트에 25%, 통합 테스트에 20%로 배분한다.

6.4 툴 선정시 유의사항

분석 및 변환 작업이 이루어지는 위치에 따라 메인프레임용 툴과 PC용 툴로 구분할 수 있는데, 일반적으로 호스트 컴퓨터의 부하 증가를 방지하고, 응용 프로그램의 DOWN-TIME을 줄이며, 나아가 GUI 등 다양한 기능들을 활용하기 위해서는 PC 로 다운로드하여 처리하는 방법이 보다 유리하다고 할 수 있다. 툴은 프로그래머의 생산성 향상은 물론, 나아가 변화작업 전체의 성패까지 좌우할 수도 있다. 따라서 선정시에는 다각적인 검토가 필요한데, 몇가지 유의사항을 살펴보면 다음과 같다.

- 툴의 종류
- 가격 및 기능 범위
- 사용의 편이성 및 친숙도
- 추후 활용성
- 시스템 부하 증가량
- PL/1의 지원 여부

6.5 검토 및 실행 단계의 착안사항

- 수정작업을 수행할 인력자원은 충분한가?
- CPU의 부하량은 추가적인 수정작업을 감당할 수 있을 만큼 충분한가?
- 대용량의 파일이나 데이터베이스 처리를 위하여 디스크 용량은 충분한가?
- 응용프로그램의 많은 부분이 변경될 경우 테스트 작업은 어떻게 수행한 것인가?
- SAS, FOCUS와 같은 외부 툴로부터의 입력은 어떻게 처리하고 테스트할 것인가?
- PC에 있는 사용자의 데이터의 및 프로그램 수정은 누가 지원할 것인가?
- 버퍼 크기와 맞지 않는 화정된 레코드로 인하여 발생할 수 있는 실행속도 문제는 어떻게 대처할 것인가?

6.6 툴 및 해결방안 공급 회사

- 변환서비스 제공 : Alydaar Software Corp.
  Peritus Software Services
  Inc. Cogni-CASE.
- DATE ROUTINES : Trans Century Data System.
- 분석 툴 : Revolve(Micro Focus)
  COBOL Analyst(SEEC Inc.)
  System Vision
  Year2000(Adpac Corp)
  ESW(Viasoft)
  Discover(Software Emancipation Technology Inc)
- 프로그램 변환 툴 : COBOL Workbench(Micro Focus)
- 데이타 변화 툴 : File-Aid(Compuware Corp)
  Century Conversion Software

(Quintic Systems Inc)

- 테스트 툴 : VIA/Smarttest & Validata (Viasoft), Tictor(Isogon) Xpediter & Xchange (Compuware Corp)
- 컨설팅 제공 : Micro Focus, EDS Andersen Consulting

6.7 피해야 할 10가지 함정

1) 지금 즉시 착수하지 않고 망설인다.
2) 변환작업에 착수하기 전에 사전 분석작업을 철저하게 하지 않는다.
3) 프로젝트 관리 및 인력자원 할당의 필요성을 낮게 평가한다.
4) 테스트 작업의 범위와 소요비용을 충분하게 책정하지 않는다.
5) 변환할 프로그램의 우선순위를 제대로 정하지 않는다.
6) Business의 문제로 보지않고 단순히 기술적 차원의 문제로 간주한다.
7) 내부 프로그램의 Career Path를 관리해 주지 않는다.
8) 협력업체와 전반적인 업무협조체제를 구축하지 않는다.
9) 타 언어나 시스템, H/W까지 전반적인 영향을 고려하지 않는다.
10) 외부전문가의 도움을 구하지 않는다

7. 향후 방안

가트너 그룹의 분석에 따르면, 1998년에는 전체 프로그램의 50% 정도가, 그리고 1999년에는 약 90% 정도가 Y2K 문제로 인하여 제대로 작동하지 않을 것으로 예상된다. 그러나, 서기 2000년까지는 이제 불과 15개월밖에 남지 않았다. 현재 미국에서도 약 50%의 회사들이 이미 Y2K 문제 해결을 위하여 착수하였으며, 30%정도가 실시를 검토중이고 나머지 20%는 아직 대처하지 못하고 있다. 2000년까지 남은15개월 이라는 기간이 긴 것 같지만, Y2K문제의 대응에는 많은 시간이라 할 수 없다. 그 영향 범위는 어플리케이션 프로그램에 그치지 않고 OS를 비롯한 PP나 벤더가 제공하는 각종 소프트웨어 등에 까지 이르고 있다. 이쯤되고 보면 방대한 소프트웨어 자산 전부를 빠짐없이 충분히 손질할 시간은 이미 없다고 보아도 될 정도이다. 'Y2K 문제'의 대응에서 가장 중요한 포인트는 '유효하고도 효율적인 대응을 위한 전체계획의 설정'이다. 이때 유의해야할 사항은 다음의 7가지 점이다.

1) 현재상황의 정리와 영향범위의 규명
2) 코어가 되는 서브시스템에 대한 충분한 샘플조사 실시와 필요한 변경량의 정량적 파악
3) 샘플조사의 결과를 바탕으로 각 컴퍼넌트마다 최적 수정방식 결정
4) 앞으로의 개발계획이나 대외적 영향의 중대함을 고려하여, 검토우선 순위의 설정과 실시체제의 구축, 또한 계획설정 후 구체적인 수정작업을 실시할 때엔, 다음과 같은 점에도 주의해야 한다.

5) 불요불급한 신규개발 안건의 동결 : 병행개발은 확인작업의 2중화나 테스트환경의 복수소유를 불가피하게 할 수 있기 때문에 동결 또는 필요 최소한으로 줄여야 한다.

6) 거래선과의 대응에 관한 조정 : 대응할 시간이 없음이 명확해졌을 경우에는 즉시 업무상 조정을 행한다.

7) 충분한 예산확보와 메이커·벤더와의 협력관계 구축 : 대응에 필요한 충분한 예산을 확보함과 아울러 타사와의 경합 때문에 사후 약방문이 되지 않도록 메이커·벤더와 대응멤버에 대한 협력체제를 이룩한다.

'Y2K 문제'에 대한 정확한 대응에는 방대한 공

수와 착실하고도 철저한 작업이 필요하다. 그러나 이들 작업은 새로운 차별화를 실현하는 신기능등 부가가치를 기대할 수 있는 것은 전혀 아닌 것이다. 다만, 제대로 움직이고 있는 시스템을 제대로 움직이게 하기 위한 필수적인 작업일 뿐이다. 이 문제는 결코 정보시스템 부문만으로 해결할 수 있는 사항이 아니며,'책임소재를 따질 여유도 없다. 경영측이 문제를 정확하게 인식하여 '현황조사부터 착수하는 전사대응 프로젝트의 발족'등 필요한 판단을 신속하게 내려야 할 것이다.

Y2K 연도문제의 해결을 위한 가장 중요한 요소는 시간, 비용, 기술자의 확보에 있다.

선진국과 비교할 때 우리나라의 경우는 2 ~ 3년 정도 늦게 시작하였기 때문에 선진국의 사례를 바탕으로 추진하기에는 남아있는 시간이 매우 부족한 실정이다. 따라서 우리나라의 경우 본 문제 해결을 위해서는 가장 체계적이고 효과적인 방법을 적용해야 한다.

사용자프로그램을 제외한 정보시스템의 구성요소들에 대해서는 각 업체들의 대응 현황을 살펴보고 문제의 해결을 위한 역할 정의, 소요비용의 확보 및 산정방법 등에 대해서도 심도있게 고려해야하며, 사용자 프로그램의 경우는 각종 SI업체 또는 컨설팅 업체를 활용하여 문제해결에 활용해야 한다.

참고문헌

[1] 한국전산원 연구보고서, "2000년 연도표기 해결 방안 연구", 1997.

[2] 목전의 '서기200년'혼란과 대책, 정보화 사회 특집, 정보통신진흥협회, 1997.1.

[3] '컴퓨터 2000년 연도표기 문제 현황과 대처방안', 정보화사회로가는 길, 정보문화센타, 1998.5.

[4] 김숙자 역, "2000년 컴퓨팅의 해결책", 성안당, 1996.9.

[5] 한국전산원, "'95년도 국가기간전산망 추진실적 평가", 1996.8.

[6] 한국전산원, 한국정보통신진흥협회주최 세미나 자료, "제1차 2000년 연도표기문제 세미나", 1997.5.

[7] Brian Kahin, Janet Abbate, "standards policy for Information Infrastructure", 1995.

[8] GAO(United States General Accounting Office), "Year2000 Computing Crisis:An Assessment Guide", 1997.2.

[9] "ISM Year 2000 Draft", 1997. 2.

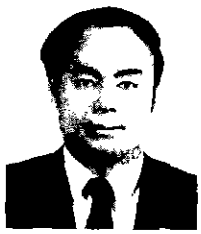
[10] William M.Ulrich, Ian S. Hayes, "The Year 2000 Software Crisis", Yourdon Press Computing Series, 1997.

[11] http://www.itpolicy.gsa.gov 의 2000년 관련 자료

[12] http://infosphere.safb.af.mil/~jwid/fadl/world/fedguide.html (SSA자료)

[13] 정보처리전문가협회, '관리자세미나자료 SDS 발표', 1998.7.

[14] 정보처리전문가협회, '관리자세미나자료 UNISYS 발표', 맹철현, 1998.3.



최 성

1970년-1976년 동국대학교 공업경영학과
1981년-1983년 연세대학교 산업대학원 전자계산전공
1994년-현재 강원대학교 대학원 박사과정수료

1994년- 현재 남서울대학교 전자계산학과 교수
한국정보처리전문가협회 총무이사
전) 한국생산성본부 OA추진사무국장, 제주은행 전산실장 역임

관심분야 : 시스템엔지니어링, 멀티미디어 게임, 멀티미디어통신, 데이타베이스