

밀레니엄버그의 해결 전략: 체신금융시스템 사례를 중심으로

A Solution Strategy on the Millennium Bug: Focusing on a Case Study with the Communications Banking System

원 달 수 (Dal Soo Weon) 현대정보기술 기술본부
오 재 인 (Jae In Oh) 단국대학교 경영정보학과

목 차

- | | |
|-------------------|-------------------------|
| I. 서 론 | V. 체신금융의 밀레니엄버그 해결사례 |
| II. 밀레니엄버그 문제의 개요 | VI. 밀레니엄버그의 문제점 및 해결 전략 |
| III. 선진국의 대처방안 | VII. 결 론 |
| IV. 체신금융시스템의 개요 | |

Keywords: Millennium Bug, Y2k, 체신금융시스템 자원조사, 영향평가, 변환 및 전환 검증 및 테스트

I. 서 론

서기 2000년을 불과 1년 남짓 앞둔 이 시점에서, 정보산업 분야에서 매우 심각한 현안임에도 불구하고 그 심각성이 제대로 인식되지 못하고 있는 것 중의 하나가 밀레니엄버그 문제이다. 컴퓨터의 역사를 살펴보면 1980년대까지만 하더라도 메모리나 디스크와 같은 부품이 상대적으로 고가였으므로, 프로그래머들은 서기력을 사용할 때 MM/DD/YYYY에서 년도 부분의 표기를 YY로 단축하여 사용함으로써 컴퓨터의 효율성을 높이도록 전력을 다하였다(Ulrich and Hayes, 1997).

이로 인하여 서기 2000년 1월 1일 이후에 야기될 부작용은 매우 심각하다. 예컨대 일자의 대소 비교가 불가능해지고, 일자의 장표 인쇄처리 및 연령 산출처리에 착오가 발생하며, 이자계산이 왜곡되고, 음력파

양력의 변환 처리도 힘들어 질 것이다. 1990년대 들어서면서 메모리나 디스크에 대한 비용 부담이 줄었음에도 불구하고, 2000년이 아직 요원하다는 이유로 기존의 YY 부분을 4자리로 고치지 않음으로써 밀레니엄버그 문제는 미리 예견된 것이었다고 할 수 있다(Computer Technology Research Corp., 1996).

밀레니엄버그 문제의 특성은 2000년이라는 확실한 시한이 주어져 있을 뿐만 아니라 또한 시간도 얼마 남지 않았다는 것이다. 더구나 이 문제가 심각한 이유는 1년이 채 남지 않은 짧은 시간에 수요는 급격히 증가하겠지만, 이를 해결할 수 있는 기술 인력은 제한적이어서 해결비용이 급상승할 것으로 예견되기 때문이다(OMB, 1997). 따라서 우리 기업이나 정부기관 그리고 개인 또한 밀레니엄버그 문제의 중요성을 직시하고, 이를 효과적으로 해결할 수 있는 방안을 모

색하는 것이 매우 시급하다고 할 수 있다. 하지만 이 문제에 대한 해결전략의 개발이 학계에서나 업계에서나 미미한 실정이다.

따라서 본 연구의 목적은 밀레니엄버그 문제의 특성을 파악한 다음 이 문제를 성공적으로 해결한 체신금융시스템 사례를 중심으로 분석하여 우리기업이나 정부에 효과적인 해결전략을 제시하는데 있다. 먼저 II장에서는 밀레니엄버그 문제의 개요를 살펴보고, 이 문제에 대한 선진 각국의 대처방안을 III장에서 고찰한다. IV장에서는 체신금융시스템의 개요를 탐색한 다음, V장에서는 밀레니엄버그의 해결 과정을 분석한다. 이러한 사례분석을 바탕으로 VI장에서는 밀레니엄버그의 문제점과 그 해결전략을 제시하고, 마지막으로 본 연구의 결론이 VII장에 요약되어 있다.

II. 밀레니엄버그 문제의 개요

밀레니엄버그 문제의 본질은 다음과 같이 7가지로 요약할 수 있다 (ICA, 1997).

첫째, 이 문제는 매우 광범위하게 펼쳐져 있다는 것이다. 즉 시스템의 기본인 운용체계, 하드웨어를 움직이는 Micro Code부분의 일자 정의, 업무시스템을 움직이는 응용프로그램 부분 등 전반에 걸쳐서 치명적인 문제를 야기할 수 있다는 것이다.

둘째, 밀레니엄버그 문제는 해결해야 할 시한이 정해져 있다는 것이다. 즉 어떠한 일이 있어도 2000년 이전에는 이 문제를 해결해야만 한다.

셋째, 동일한 문제에 많은 컴퓨터가 직면해 있어서, 밀레니엄버그 문제도 동시 다발로 해결되어야 한다. 하지만 많은 인원을 동시에 무리하게 투입하게 되면, 이에 따른 부작용이 우려된다.

넷째, 정보통신의 발달로 밀레니엄버그 문제는 어떤 특정 기업이나 정부기관의 문제가 아니라 국가적인 차원에서 문제를 접근해야 한다.

다섯째, 그 심각성에 비해서 문제를 해결할 획기적인 대책이 없다는 것이다. 따라서 해결책을 기다릴 수도 없는 실정이고, 밀레니엄버그 문제를 해결할 프

젝트를 각자의 위치에서 구성해야 한다.

여섯째, 업무가 컴퓨터시스템에 의존하는 정도가 심할수록 기업경영 차원의 문제와 직접 관련되고 있다는 것이다. 즉 이 문제는 업무의 연속성을 위협하는 동시에 주주 혹은 고객으로부터 소송을 당하거나 또는 회사의 존립까지도 위협 받을 수 있게 된다.

일곱째, 정보화시대에서는 정보가 경쟁력의 원천이기 때문에, 정보의 공급중단으로 인하여 전세계가 경제 공황을 겪을 수도 있다는 우려도 있다.

밀레니엄버그 문제를 해결하기 위한 프로젝트를 구성시 먼저 처리절차를 정의하고 각 단계별로 문제 해결을 위한 방안을 제시해야 하는데 그 절차는 다음과 같다 (한국전산원, 1997d).

- 하드웨어와 소프트웨어를 제어하는 운용체계에 서의 문제를 해결한다.
- 업무프로그램의 내용을 위한 기준을 설정한다.
- 우선순위를 결정한다.
- 문제해결에 들어갈 자원을 측정한다. 즉, 필요 인력과 기간을 산정하기 위한 Pilot조사를 해 보아야 한다.
- 각 단위에서 필요한 작업내용을 결정한다.
- 실제와 같은 상황에서 특별운용을 실시한다.
- 장애대응을 위한 방안을 강구한다. 미처 해결하지 못한 밀레니엄버그 문제를 예상한 장애대응 방안을 세우는 것이다.
- 실제로 문제해결 기간과 인력을 계산한다.
- 2000년 이후 문제의 잔재를 예측한다. 현실적으로 문제해결을 위한 주요 조치를 취한 후 항구적인 조치를 착수해야 한다.

밀레니엄버그 문제가 프로그램들에 미치는 영향을 평가하는 방법에는 다음과 같이 3가지가 있다 (한국정보통신진흥협회, 1997b).

첫째, 분석방식은 영향분석용 툴을 사용할 경우 높은 작업효율과 분석된 양의 정량적 계측, 변환연동가

능 및 해당 설명문서의 자동출력도 가능하다.

둘째, 문자열 탐색방식에 따르면 변수명이 표준화되어 있다. 일자관련 프로그램이 서브 프로그램화되어 있는 경우에는 좋은 방식이지만, 기본적인 방식보다는 보조방식으로 사용하는 것이 좋을 것이다.

셋째, Degradе Test방식은 상기의 2가지 방식을 적용하지 않아도 별 문제가 없다고 판단될 경우에 사용된다. 프로그램에서 문제가 되는 부분을 찾아내서 수정이 불필요한 부분을 거꾸로 찾아내는 방식이다.

또한 밀레니엄버그 문제가 발견되었을 때 이에 대한 해결하는 방법에는 다음과 같은 3가지가 있다(ICA, 1997).

- 년도 4자리 표기: 문제 해결의 방법 중 가장 좋은 것으로 년도표기를 4자리로 바꾸기 때문에 향후에 발생할 문제를 모두 방지할 수 있다는 장점이 있지만, 외부와 연결되는 경우에는 신중을 기해야 한다.
- 내부계산에서만 4자리를 사용: 비록 자료는 2자리지만 내부에서 일정기간을 정해서 2000년대와 1900년대를 나누는 방법이다. 이 경우는 프로그램이 약간 복잡해지고, 일자를 인덱스해야 하는 작업이 필요하며, 정렬업무(Sort) 등에서는 수행시 많은 어려움이 뒤따른다.
- 년도 차감방법: 프로그램의 입력부분에 년도가 입력될 경우에는 일정 년도를 차감하여, 마치 2000년이 도래하지 않은 것처럼 하는 것이다.

최종적으로 수행해야 할 사항은 철저한 테스트하는 것이다. 이 단계에서 중점적으로 고려해야 할 것은 2가지이다.

첫째, 하드웨어의 밀레니엄버그 문제가 해결되지 않고 업무적으로만 해결해야 할 경우에는 어려움이 많고 하드웨어에서 제공하는 일자 등을 전혀 사용하지 못하기 때문에 많은 테스트가 요구된다.

둘째, 하드웨어가 완벽한 경우의 테스트로서, 2000년으로 일자를 변경한 후에 실시하는데 테스트가 상대적으로 용이한 편이다.

Ⅲ. 선진국의 대처방안

선진국들은 이미 1990년대 중반부터 밀레니엄버그 문제에 대한 심각성을 인식하고, 많은 연구와 투자가 이루어져 왔다. 이 문제로 초비상이 걸려있는 이유는 미연에 대비는 해왔지만 상대적으로 컴퓨터 시스템의 범위가 넓고 자동화된 업무가 많기 때문이다. 미국, 캐나다, 영국, 일본 등 주요 선진국들은 정부 차원에서 대대적인 투자를 통하여 이 문제를 본격적으로 해결하고 있다.

3.1 미국

미국은 예산관리처(OMB)가 24개 연방기관의 대응 추진 상황을 분기별로 보고 받고 결과를 대통령과 국회에 보고한다 (OMB, 1997). 부진한 기관에 대해서는 제재조치를 취하고 있으며, 1998. 2. 4. 클린턴 대통령은 연방기관의 대응을 촉진하기 위해 행정명령인 Year 2000 Conversion을 발령하고, 대통령 직속으로 연방기관의 대표자가 참여하는 Year 2000 Conversion Council을 설립하여 추진 상황을 분기별로 점검하고 있다.

미국의 바이어소프트웨어가 밀레니엄버그 문제의 영향을 조사해본 결과 프로그램의 70%, 데이터의 1% 이상이 해당되며 프로그램 총 스텝수의 약 2%가 영향을 받는 것으로 판명되었다. 본 연구에서는 국방성과 증권업계를 중심으로 미국의 밀레니엄버그 해결사례를 살펴본다.

최첨단 기술을 이용하고 있는 미국방성은 근래 몇 년 동안 국방예산이 45%나 삭감된 반면 업무량은 약 10년 동안에 500%나 증가했다고 한다. 현재의 예산은 250만불로 그 중 65%는 인프라에 대한 고정경비에 해당된다. 구매능력 또한 75%나 감소했다고 한다. 그러나 미군의 작전능력을 증강하지 않으면 안되므로 밀레니엄버그 문제는 뒤로 미루어 왔다. 국방성도 밀레니엄버그 문제의 대응에 늦기는 마찬가지였다.

국방성은 특별예산을 가지고 있지 못하지만 대응

을 서두르지 않으면 안되었기 때문에 기존 예산 중에서 전용할 수 밖에 없는 실정이었으므로 신규개발 및 진행중인 프로젝트를 축소해서 밀레니엄버그 문제에 대응하기 위한 예산으로 사용하고 있다. COBOL 프로그램은 톨을 사용할 경우 수정 및 테스트 비용으로서 라인당 1.5\$~2\$정도 소요되는 것으로 판단하였으나 수정할 수 있는 프로그래머 수가 줄어 들고 있어 부족한 상황이고 시간이 지날수록 수요증가로 인건비는 상승하고 있다.

공군에서 이용되고 있는 JOVIAL언어는 관련 전문가가 소수이고 라인당 7~8\$의 비용이 들것으로 예상된다. 또한 미군의 구성시스템으로 사용되고 있는 소프트웨어들은 규모도 알 수 없을 뿐만 아니라, 특수하기 때문에 수정할 수 있는 사람이 없을 가능성도 있다. 전차와 전투기, 군함, 잠수함, 미사일 등 특수 시스템이 다수 사용되고 있어 밀레니엄버그 문제는 대단히 심각한 실정이다. 따라서 국방부에서는 2000년 문제에 완전히 대응할 수는 없으므로 사명감을 갖고 아주 중요한 부분만이라도 우선은 쓸 수 있도록 노력한다는 말까지 하고 있을 정도이다.

최근 미국 증권업협회(SIA)는 시계를 1999년 말과 2000년 초로 맞춰놓은 컴퓨터를 이용하여 모의 증권 거래 실험에 성공했다고 밝혔다. SIA는 이날 28개 증권업체들이 컴퓨터 모의실험을 통해 2000년도에서도 거래를 할 수 있었다면서, 최종적인 실험은 내년에 실시할 것이라고 한다.

이번 실험에서 컴퓨터 시계는 1999년 12월 29일과 30일, 금세기 마지막 개장일인 31일, 그리고 21세기 첫 개장일인 2000년 1월3일로 설정됐다. 또한 28개 증권업체들은 4일 동안 실시된 이번 실험기간 중에서 하루 평균 각각 5백여건씩 모두 1만여건의 거래를 했다. 미국 증권업체들은 5천만명이 넘는 투자자들의 위탁계좌를 직접 관리하고 있으며 기관들을 통해 수천만 계좌를 간접 관리하고 있다. 8백개 증권업체들을 회원으로 두고 있는 SIA는 내년 3월과 4월에 밀레니엄버그 해결을 위한 최종 실험을 실시할 것이라고 밝혔다.

3.2 캐나다

캐나다의 밀레니엄버그 대처사례는 가장 성공적인 로얄은행을 중심으로 살펴보기로 한다 (한국정보통신진흥협회, 1997d). 로얄은행은 48,000명의 종업원, 1,670억\$의 자산, 1,500개의 지점을 가진 캐나다 최대 규모의 은행으로 Data Center를 캐나다에 7개, 해외에 10개를 보유하고 있다. 밀레니엄버그 문제 대처하기 위하여 1995년에 초기 영향분석을 위한 Pilot Project를 실시하고 기업전체에 대한 영향 평가도 수행하였다.

Pilot Project를 통하여 로얄은행은 전체 적용업무의 현상을 파악하고, 이로 인해 영향을 미칠 수 있는 데이터 변수와 프로세스를 분석 및 평가하고 수정에 필요한 시간과 경비도 예측하였다. Pilot Project에서 3가지의 적용업무를 분석한 결과 600개의 프로그램중 절반인 300개 프로그램을 수정해야 한다는 것을 알게 되었다. 또한 60종에 이르는 각종 컴퓨터 언어로 쓰여진 25,000개의 프로그램과 약2,500만 스텝에 영향이 있고 통합 테스트를 제외하더라도 1,300명의 인원이 필요한 것으로 판명되었다.

로얄은행은 1996년부터 1997년에 걸쳐서 밀레니엄버그 대응계획을 작성하여 1998년부터 1999년까지 본격적인 작업을 진행하고 있다. 밀레니엄버그 대응에는 막대한 경비가 들어가게 됨에도 불구하고, 2000년이 되어서도 문제를 일으키지 않는다는 소극적인 효과 밖에 없다. 따라서 기업전체의 정보를 회사의 자산이라는 차원에서 재검토하는 좋은 기회가 될 수도 있기 때문에 장래에는 Data Warehouse나 Internet을 이용한 신시스템을 구축할 예정이라면, 밀레니엄버그 대응을 기회로 하여 정보시스템의 전면적인 재구축을 추진하는 것도 좋은 방법이라는 결론을 얻었다.

3.3 기타

영국에서는 밀레니엄버그 문제에 관한 정보 교환을 촉진하기 위해 중앙컴퓨터통신처(CCTA) 산하에 MBDG (Millennium Bomb Discussion Group)를 구성하여 대처하고 있다 (한국전산원, 1997b). 중소기업에만

3,000만 파운드를 지원하고, 정부 홍보산업인 '액션 2000'의 예산도 100만 파운드에서 1,700만 파운드로 크게 늘렸으며, 블레어 총리는 앞으로 1년간 2만명의 컴퓨터 프로그램 전문인력을 집중 양성하겠다고 발표한 바 있다.

싱가포르 증시는 최근 밀레니엄버그를 대비한 시스템 시험을 성공적으로 마쳤다고 증시 관계자가 밝혔다. 시행된 시험에서 모든 컴퓨터의 날짜를 2000년 1월 3일로 맞추어 놓고 시스템이 제대로 운영되는지 확인해 본 것으로 결과는 성공적이었다. 싱가포르 정부는 대기업이 대부분 밀레니엄버그 문제를 해결했다고 밝히고 있지만, 컴퓨터 전문가들은 금융기관을 포함한 많은 싱가포르 기업들이 아직도 이 문제로 인해 일어날 수 있는 부작용에 대처할 능력이 부족하다고 지적한다. 한편 싱가포르 컴퓨터 전문가들은 일부 금융기관에게 밀레니엄버그로 인해 일어날 수 있는 만약의 사태에 대비하여 2000년 처음 며칠동안 회사의 문을 닫고 상황을 살필 것도 조언하고 있다.

일본의 정보서비스산업협회는 1996년 5월에 '서기 2000년 문제위원회'를 발족하는 등 민간의 정보통신업계를 중심으로 대응하고 있다. 다만 중소기업에 대하여는 통산성, 대장성에서 문제해결 비용을 손비로 관리하는 등 세제 지원을 중심으로 지원하고 있다.

IV. 체신금융시스템의 개요

체신금융시스템은 정보통신부 산하 약3,000개 우체국에서 운영하는 은행 및 보험 등과 같은 업무용 분산시스템으로써 도서벽지를 포함하여 대국민 금융서비스를 담당하고 있다. 한편 국내 컴퓨터 산업 및 신기술 육성, 지방자치제의 정착에 따른 지역정보화 선도, 특정 시스템에 대한 종속회피 등을 목적으로 H전자가 주개발자로 선정되어, 1997년 7월부터 1999년 12월까지 개방형 분산시스템으로의 전면 재구축을 추진하게 되었다.

이 프로젝트는 1995년부터 이미 검토되었으나 정책적 판단, 기술성숙도, 분산시스템에 대하여 광범위한

의견을 수렴하느라 착수가 늦어지게 되었다. 프로젝트의 위험도가 매우 큰 점을 고려하여 2000년 이전에 분산시스템이 가동되지 못할 경우도 대비하여, 일종의 보험차원에서 현행시스템에 대한 밀레니엄버그 문제 해결에 착수하게 되었다.

먼저 본 시스템의 구현 목표는 살펴보면 다음과 같다. 첫째 중앙집중에서 분산처리로 전환한다. 둘째 메인프레임에서 UNIX 환경을 지향한다. 셋째 폐쇄형 네트워크에서 TCP/IP네트워크로 변경한다. 넷째 GUI 단말환경으로 전면 교체한다. 다섯째 각종 구성장비의 표준화 및 개방형 시스템을 채택한다. 여섯째 무장애 시스템을 구현한다.

또한 업무구현 목표는 다음과 같이 8가지로 요약할 수 있다. 첫째 계정과목 중심에서 기능중심으로 전환한다. 둘째 내부업무처리 중심에서 고객중심을 지향한다. 셋째 국산금융 미들웨어를 개발하여 적용한다. 넷째 다양한 첨단 고객서비스의 기능을 구현한다. 다섯째 연중무휴 그리고 24시간 고객에게 서비스를 제공한다. 여섯째 계정계 및 정보계 시스템을 동시에 구축하고 전환한다. 일곱째 은행 및 보험 업무를 동시에 구축하여 시행한다. 여덟째 신상품 개발 및 유지보수 노력을 최소화시키는 방향으로 전환한다.

시스템 구성 내용을 살펴보면, 체신금융분산시스템은 기존의 폐쇄형 시스템을 개방형 시스템으로 전환하며, 1개 중앙센터에서 집중 처리하던 업무를 8개 지역센터로 분산 처리하게 된다. 클라이언트/서버 방식으로 계정계, 정보계, 대외계 시스템을 구축하며 관계형 데이터베이스를 활용한 분산DB를 구축하는 것이 가장 기본적인 골격이라고 할 수 있다.

또한 응용프로그램부문도 공통기능을 통합시켜 활용하는 체신금융 공통소프트웨어 개발과 프로그램 이식성 및 확정성을 보장하는 형태로 개발하고 있다. 소프트웨어 시스템은 중앙센터, 지역센터, 감독국, 우체국등에 유닉스서버, PC 클라이언트, 감독국BP용 유닉스서버 등 상이한 기종을 플랫폼에 구축한다. 다양한 기술구조 플랫폼에서 이뤄질 응용프로그램은 내부 개념적인 구조로 통일시켜 개발함으로써 개발을 효율

적으로 추진하도록 유도하며 앞으로 운영 및 유지보수가 용이하도록 설계하였다.

소프트웨어 언어는 크게 시스템소프트웨어, 공통소프트웨어, AP 등 3가지로 나누고 있다. 첫째 시스템 소프트웨어 계층은 분산시스템에서 요구되는 기본적인 서비스를 지원한다.

둘째 공통소프트웨어 계층은 다시 공통 인터페이스, 업무공통 모듈, 통합운영 소프트웨어로 분류된다. 공통 인터페이스는 응용프로그램에서 요구되는 기능과 시스템소프트웨어계층의 중간에 소켓 형식으로 개발하여 서비스 응용프로그램을 특정제품과 분리시킨다. 업무공통모듈은 응용프로그램에서 공통적으로 사용되는 기능을 모듈형태로 제공하는 것으로 프로그램에서 불러내 사용함으로써 중복을 피하고 관리를 쉽게 하는 파라메타 POOL방식으로 구성한다. 통합운영 소프트웨어는 중앙센터와 지역센터서버에서 사용하는 것으로 공통모듈과 체신금융 분산시스템 전체를 운용할 때 필요한 것이다.

셋째 AP계층은 예금계, 보험계, 대외계, 정보계 응용프로그램등 분야별로 개발된다. 통합운영프로그램은 개시일거래, 마감일거래 등 전체 응용프로그램에 연관되는 프로그램이다. 하드웨어 및 네트워크 구성은 개방형 분산시스템구조를 적극 수용할 수 있도록 산업표준네트워크인 TCP/IP, X.25등을 수용하는 네트워크 인프라를 기반으로 표준 운영체제인 UNIX SVR4.2의 호환성에 기반을 둔 운영체제로 구성하였다.

체신금융분산시스템의 효율적인 운영방안은 시스템통합관리, 백업관리, 소프트웨어분배관리, 보안관리, 장애처리관리 등 5가지로 나누고 있다.

첫째 시스템통합관리는 별도의 시스템관리서버를 중앙 및 지역에 설치하여 중앙센터 지역센터에서 이뤄지는 상황에서 발생하는 각종 이벤트를 감시한다. 특히 각 분산되어 있는 서버들간의 소프트웨어 자동분배, 보안, 배치작업자동화, 이벤트 처리, 백업 및 복구대책관리를 중앙의 통합콘솔을 통해 시스템 전체상태를 파악하고 시스템이용도 모니터링한다.

둘째 백업은 시스템백업 및 데이터베이스백업으로

구분하며 서버 각각의 백업은 해당지역서버의 백업용 장치를 이용하여 지역센터에서 직접 수행한다. 작업 실행을 위한 백업용 스크립트는 지역센터의 백업서버로 보관, 관리되며 백업작업에 대한 결과 및 정보는 지역 및 중앙센터에 통보되어서 통합적으로 관리된다. 작업진행에 관련된 상황모니터링은 지역센터 시스템관리 서버에서 관리하고 백업작업을 수행할 때 발생된 장애사항을 조치해준다. 백업작업이 이뤄진 뒤에 성공 또는 실패여부에 대한 결과는 지역센터 및 중앙센터의 시스템관리서버로 동시전송하고 중앙센터 운영담당자에게 보고한다. 데이터베이스의 백업주기는 일간, 시스템파일 백업주기는 주간으로 하는 것을 원칙으로 하고 있다.

셋째 소프트웨어의 분배관리는 대상기기로 분배하는 전 과정을 일목요연하게 처리하고 지역센터, 감독국, 우체국 단말에 중앙 집중적인 관리가 요구된다. 이 분야의 경우 신뢰성, 편의성, 기능성, 이식성 등의 기능을 원활하게 수행하는 것이 특징이다.

넷째 서버의 보안관리는 감독국에서 발생하는 시스템 접근사항 및 보안위반사항에 대해 지역센터 및 중앙센터의 보안 담당자용 콘솔로 해당내용을 동시에 기록해서 보관하며, 지역센터에서 발생하는 모든 위반사항은 중앙센터로 기록된다. 특히 비인가자 및 등록되지 않은 사용자의 접근시 해당내용을 보안담당자가 빠르게 발견할 수 있도록 표시를 한다.

이것은 인가된 사용자의 시스템 접근사항 및 위반사항에 대해 센터 및 각 센터콘솔을 통해 감시 및 조치하게 된다. 또 비인가자의 시스템 접근에 대해 콘솔에 적색표시로 담당자에게 쉽게 전달하게 된다. 특히 일정한 횟수로 연속위반시 사용자ID의 사용을 일시적으로 중단시킴으로써, 사용자를 감시하고 보안 담당자와 연결 후 자동복구가 가능하다.

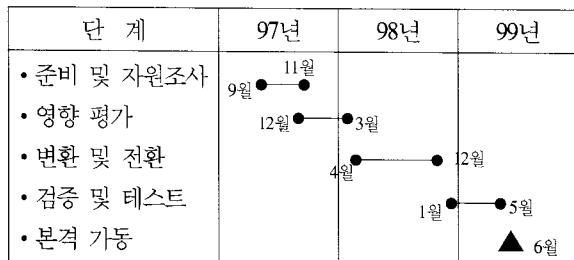
다섯째 장애처리에 대한 관리는 지역센터, 감독국, 우체국에서 장애가 발생될 경우 중앙센터의 통합서버로 표시되고 HELP DESK 서버에 의해 이뤄진다. 통합콘솔은 장애와 이벤트를 감지해 HELP DESK 서버에 등록하고 콘솔화면에 표시한다. 운영실 담당자

는 해당문제를 파악하는 스태프들을 호출하고 담당스태프는 HELP DESK와 상호연락을 통해 장애내용을 파악한 다음 조치한다. 발생한 모든 이벤트와 티켓은 장애유형별로 로그로 보관돼 앞으로 장애발생시 추적이 가능한 시스템 구축을 위한 통계자료로 활용된다.

V. 체신금융의 밀레니엄버그 해결사례

1997년 8월에 착수한 체신금융시스템의 밀레니엄버그 문제를 해결하기 위한 프로젝트는 년도문제와 밀접한 금융업무였다. 규모를 살펴보면 IMS DB/DC의 PL/1 약 7,000본과 CICS의 COBOL 약 800본으로 구성된 총 4백만 STEP으로 구성되어 있다. 당시로서는 2000년 문제에 대한 경험부족, 업무량의 정확한 예측불가 등으로 인하여 전체적인 계획수립이 불가능한 어려움 가운데 착수되었다. 체신금융 밀레니엄버그 프로젝트는 자원조사 (3개월, 약 30M/M투입)와 시스템 영향평가 (3개월, 약 50M/M투입)를 먼저 1차로 추진하였다. 1차 추진사항 결과를 종합적으로 평가한 후 변환 및 전환, 검증 및 시험을 2차로 추진하기로 하였다 (<표 1> 참조).

<표 1> 프로젝트 일정표



5.1 자원조사 및 영향평가

밀레니엄버그 대응 여부를 파악하기 위해 먼저 필요한 조사양식 등을 정의하여 작성하고, 자원 조사 및 영향분석을 위한 작업환경을 구축하였다. 또한 밀레니엄버그 관련 자원을 조사하고 영향 분석도 실시하였다. 이를 위하여 미국 보험회사인 Kemper, 컨설

팅회사인 Costello, 전기통신회사인 MCI Systemhouse, 밀레니엄버그 톨개발 회사인 Micro Focus 등을 벤치마킹하였다. 국내에서는 제일은행, 국민은행, 동아생명, 한국전력 등과 같이 금융권을 중심으로 방문하여 변환 기법 및 밀레니엄버그 지원도구에 대한 활용사례를 조사하였다.

<표 2> 응용프로그램 분석결과표

업 무 구 분	본 수	스텝 수	날짜영향 스텝수	비 고	
예금계	전자종합	1,511	1,003,319	38,546	3.84%
	근로자장기저축	81	53,106	4,079	7.68%
	근로자우대저축	84	37,975	1,533	4.04%
	세금우대	288	212,237	11,945	5.63%
	학생장학적금	292	139,830	6,611	4.73%
	가계장기저축	125	40,623	2,390	5.88%
	금융공동망	705	418,689	9,050	2.16%
	누계	3,086	1,905,779	74,154	3.89%
총괄계	계산관리	1,000	502,318	14,407	2.87%
	우편환	345	183,853	6,031	3.28%
	자기앞수표	17	162,581	5,372	3.30%
	경조환	12	3,110	39	1.25%
	신용카드	10	5,883	204	3.47%
	조사	338	197,179	6,908	3.50%
	누계	2,022	1,054,924	32,961	3.12%
대체계	연금	144	103,907	2,712	2.61%
	우편대체	435	193,459	5,508	2.85%
	은행지로	179	95,853	1,982	2.07%
	누계	758	393,219	10,202	2.59%
보험계	체신보험	908	592,246	20,042	3.38%
	누계	908	592,246	20,042	3.38%
국고계	관세EDI	72	42,066	1,399	3.33%
	국민연금	85	36,363	1,270	3.49%
	통신문	12	2646	85	3.21%
	국고	290	169,177	5,927	3.50%
	누계	459	250,252	8,681	3.47%
회계계	증서보급	70	30,010	580	1.93%
	누계	70	30,010	580	1.93%
개발	우편주문판매	60	27,010	993	3.68%
	누계	60	27,010	993	3.68%
운영	시스템	40	23,334	817	3.50%
	누계	40	23,334	817	3.50%
지원계	총 계	7,403	4,276,74	148,430	3.47%

밀레니엄버그 관련 날짜변수와 식별에 대한 조사 및 분석을 실시하여 업무용 프로그램에서 사용하는 날짜변수의 패턴을 파악하고, 날짜변수 패턴에 의한 변수간 그리고 프로그램과 DB간 영향을 분석하였다. 단위업무를 대상으로 파일럿 프로젝트를 실시하여 프로그램 변환단계의 작업에 대한 사전 검사도 추진하였다.

이 과정을 통해 관련범위를 3가지 분야로 설정하였으며, 이의 해결 방안을 다음과 같이 조정하였다. 첫째 IBM Mainframe 등 센터기기에 대한 대책은 정보통신부에서 해당 공급업체와 협의하여 직접 해결한다. 둘째 응용프로그램 및 데이터에 대한 대책은 용역업체에 의뢰하여 해결한다. 셋째 단말기, 자동화기기 등에 대한 대책은 정보통신부에서 해당 공급업체와 협의하여 직접 해결한다.

한편 전체 7,403개 프로그램의 4,276,724스텝 중 날짜와 관련된 스텝이 148,430으로 3.47%에 해당되었다. 또한 소스프로그램이 없이 로드모듈만 남아있는 프로그램 재코딩이 필요한 프로그램은 업무분석, 설계과정을 거쳐 신규 작성하는 부분도 상당수 필요한 것으로 나타났다.

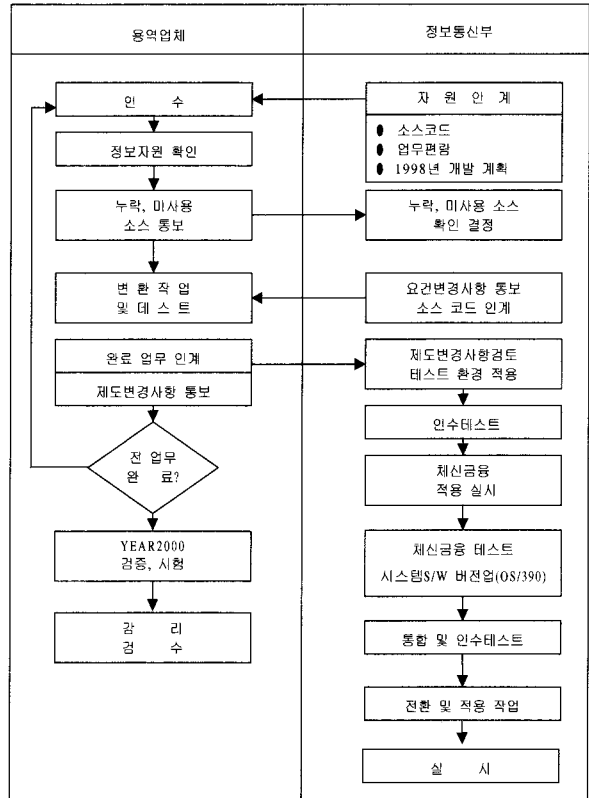
또한 예금 데이터의 경우는 데이터베이스 내에 년도표시가 2자리로 구성되어 있어 이를 변환하는 것이 무리가 있다고 판단되었다. 또한 분산시스템이 구축 중에 있어서, 한시적으로 사용될 예정이므로 년도창 기법에 의해 처리토록 하였다. 응용프로그램에 대한 분석 결과는 <표 2>와 같다.

5.2 변환 및 전환

변환된 프로그램은 현행시스템 온라인환경에서 우선 적용하여 날짜 관련 부분의 문제를 해소하도록 하고 있다. 체신금융시스템에서 채택한 절차를 정리하면 <그림 1>과 같다. 먼저 정보통신부는 용역업체인 H전자에 자원을 인계하면 H전자는 이를 인수하여 확인한 후 누락되었거나 미사용된 소스를 통보한다.

정보통신부는 요건에 대한 변경사항이 있으면 H전

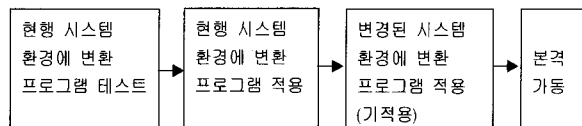
자에 통보하고 소스코드를 인계한다. H전자는 이를 변환하고 테스트까지 마친 다음 인계하면 정보통신부는 제도변경사항을 검토하고 테스트환경에 적용한다. H전자는 전업무가 완료되면 밀레니엄버그에 대한 검증과 시험을 하고 마지막으로 감리를 받고 검수를 거친다.



<그림 1> 변환작업 및 현행 온라인 적용 흐름도

5.3 검증 및 테스트

본 단계는 밀레니엄버그 문제를 해결하기 위한 절차 중에서 가장 중요한 단계이다. 체신금융시스템에서는 다음 <그림 2>와 같은 방법을 채택하고 있다.



<그림 2> 일정 계획표

또한 검증 및 테스트를 위하여 체신금융시스템에서 실시하고 있는 구체적인 일정은 <그림 3>과 같다. 먼저 업무별 적용계획을 1998년 12월 중으로 수립한다. 둘째 통합 및 인수에 대한 테스트를 1999년 1월부터 3월까지 수행한다. 셋째 1999년 4월에서 5월까지 전산관리소에서 온라인으로 적용한다. 넷째 1999년 6월에 감리를 7월까지 검사를 완료한다.

	8	9	10	11	12	1	2	3	4	5	6	7
업무별 적용 계획수립					●	●						
통합 및 인수 테스트 (미래날짜 포함)						●	●	●				
온라인 적용 (전산관리소)									●	●		
감 리											●	●
검 사											●	●

<그림 3> 일정 계획

VI. 밀레니엄버그의 문제점 및 해결 전략

이제까지 탐구한 체신금융시스템의 밀레니엄버그 문제해결 사례를 바탕으로, 우리 기업이나 정부기관이 흔히 가지고 있는 문제점은 정리하면 다음과 같다.

첫째, 기존시스템의 복잡한 로직에 대한 문서도 부재하고 초기 개발에 참여했던 프로그래머들마저 없다면, 개발후 시스템을 수정하는 경우가 태반이기 때문에 밀레니엄문제 자체를 해결하기가 매우 힘들게 된다는 것이다.

둘째, 정보기술의 다운사이징 및 아웃소싱의 영향으로 COBOL, PL/1, ASSEMBLER 등과 같은 대형컴퓨터 프로그래밍언어에 대한 전문가가 줄어들고 있는바, 이러한 현상이 밀레니엄버그 문제해결에 지대한

걸림돌이 되고 있다는 점이다. 대규모 프로젝트를 관리하는 것도 어려움이 따르기 때문에 문제를 자체적으로 해결하기 위한 시간과 비용은 점점 급증하게 되고 별도의 해결책이 제시되지 않는다면, 결국 우리 기업 나아가 우리나라의 국제경쟁력까지도 실추시킬 수 있을 것이다.

본 연구에서는 이러한 문제들을 성공적으로 대처하고 밀레니엄버그 문제를 해결하는 전략을 다음과 같이 종합적으로 제시하고자 한다.

첫째, 밀레니엄버그 문제의 심각성을 경영진 및 현업에 분명하게 인식시키고, 조속히 정보화 프로젝트에 공식적으로 반영해야 한다.

둘째, 솔루션 틀을 적절하게 활용하여, 안정성과 3-4배 이상의 생산성 향상효과를 거둘 수 있도록 한다.

셋째, 별도의 테스트 환경 및 Data Conversion을 위한 디스크 등이 필요한바, 차질이 없도록 충분한 자원을 미연에 확보하여야 한다.

넷째, 날짜 로직은 시스템 전체에 산재되어 있기 때문에 많은 자원을 필요로 한다. 따라서 자원투입에 인색해서는 안되며, 밀레니엄버그를 자체해결하는 아웃소싱을 하든 예산을 충분히 확보하여야 한다.

다섯째, 시간이 갈수록 자원의 한계로 비용상승이 예상되므로 전담팀을 구성하여 조기에 프로젝트를 착수하는 것이 비용을 절감하는 지름길이다.

나아가 우리 기업이나 정부기관이 고려해야 할 중요한 사항들은 크게 기술적인 측면과 관리적인 측면으로 나누어 정리하면 다음과 같다.

기술적인 측면으로 변환 및 전환후의 테스트를 위한 시스템 환경구축, 3단계의 날짜 변경 테스트(Current Date 테스트 및 Future Date 테스트 즉 1999년과 2000년 간의 변경 테스트)와 이들 테스트를 위한 시나리오 준비(테스트 Date의 입력과 이에 대한 서면 및 리포트를 통한 출력데이터 정의)가 요구된다. 외부기관과의 데이터 교환에 대한 표준 및 지침의 필요성 문제가 대두되고, 이로 인한 영향평가에 대한 분석이 통합적으로 실시되어야 하기 때문에 전환계획 수립이 결코 용이하지는 않다.

둘째, 관리적 측면으로 밀레니엄버그 프로젝트를 추진하기 위한 예산 및 전문 인력의 확보와 프로젝트 진행과정에서 발생할 수 있는 위험요소에 대한 위험 관리 방안, 프로젝트 진행과정의 비용 관리 및 변경 되는 소스프로그램의 관리 지침이 필요하다. 뿐만 아니라 이를 위한 비용 및 시간의 문제를 보면 밀레니엄버그 문제는 접근할수록 점차 그 비용이 증가하게 된다.

따라서 검증된 방법론과 일정관리가 부재한 상태에서 대응할 경우 더욱 극명하게 나타남으로, 명확한 계획의 결정과 비용면에서 효과적이고 구현가능한 해결방법을 제시하는 정형적인 방법론을 채택하는 것이 매우 중요하다. 실패가 허용되지 않는 밀레니엄버그 문제의 특성상 프로젝트의 전반에 걸쳐 일관적이고 효율적인 관리가 가능하게 하는 방법론의 도입은 대단히 중요하다고 할 수 있다.

방법론을 도입할 경우 또 하나의 장점은 프로젝트를 Tool 위주가 아닌 Task 위주로 진행할 수 있다는 점이다. 전체 프로젝트를 하나의 시나리오상에서 구현함으로써 특정 Tool에 종속되지 않고 미리 확정된 Task 위주로 진행하는 것이 가능하게 된다. 따라서 관리의 용이성과 효율성 그리고 일관성을 확보할 수 있게 된다.

VII. 결 론

밀레니엄버그 문제의 해결과정은 새로운 기술을 개발한다거나 새로운 서비스를 창출하는 등과 같이 창의를 요구하는 작업은 아니다 (Ulrich and Hayes, 1997). 하지만 이 문제를 해결하는 과정에서 예상하지 못한 문제가 돌발적으로 발생할 가능성이 상존한다는 점에서 해결과정에서 세심한 주의가 필요하다.

본 연구에서는 H전자가 개발업체로 참여하고 있는 체신금융시스템의 밀레니엄버그 해결 사례를 중심으로 그 문제점과 해결 전략을 우리기업에 제시하였다. 먼저 밀레니엄버그의 본질과 일반적인 해결절차를 분석한 다음 선진 각국의 대처방안을 탐구하였다.

우리나라는 IMF의 영향으로 이 문제가 조기에 부

각되지 못하였지만 늦게나마 성공적인 해결전략이 모색되어야 한다는 점에서, 체신금융시스템의 개요를 살펴본 다음 밀레니엄버그 해결 사례를 자원조사 및 영향평가, 변환 및 전환, 검증 및 테스트 등과 같이 단계별로 분석하였다.

사례분석의 결과 밀레니엄버그 해결과 관련하여 우리기업이나 공공부문이 안고 있는 문제는 크게 시스템 개발시 문서화의 미비, 대형컴퓨터 프로그래밍 언어의 전문가 부족 등이었다. 이러한 걸림돌을 성공적으로 대처하기 위한 전략으로 본 연구에서는 밀레니엄버그 문제의 공식화, 솔루션 툴의 적절한 활용, 충분한 자원의 확보, 충분한 예산의 확보, 조기에 착수 이상 5가지를 제시하였다.

결론적으로 밀레니엄버그 문제는 세밀한 계획, 시스템 전문가, 문제해결을 위한 전략과 방법론 등 막대한 자원과 세심한 계획이 요구될 뿐만 아니라 제한된 기간 내에 해결해야 하는 시급한 과제이기도 하다. 설상가상으로 주어진 기한 내에 이 문제를 해결하려는 수요는 급증할 것인 반면에, 해결할 수 있는 전문인력의 공급은 한정되어 있기 때문에 우리기업이나 정부의 조속한 대처가 요구된다.

참 고 문 헌

- 동아일보, 서울신문, 조선일보, 경향신문 1997. 8. 26.
- 미연방 OMB(Office of Management and Budget) 보고서 1997. 8. 15.
- 전자신문 1997. 9. 12.
- 중앙일보 1997. 11. 8.
- 하이테크 정보 1997. 9. 5.
- 하이테크 정보 1997. 10. 20.
- 한국전산원, "제1차 2000년 연도표기문제 세미나", 1997a. 5. 9.
- 한국전산원, "2000년 연도표기문제 해결방안 연구", 1997b. 6.
- 한국전산원, "컴퓨터 2000년 연도표기 현황분석 보고", 1997c. 10.

한국전산원, “컴퓨터 2000년 연도표기문제 해결을 위한 워크샵”, 1997d. 11. 14.

한국정보통신진흥협회, “96년 정보통신산업통계연보”, 1996.

한국정보통신진흥협회, “97 소프트웨어 개발산정기준 해설”, 1997a.

한국정보통신진흥협회, “제2차 2000년 연도표기문제 세미나”, 1997b. 6. 24.

한국정보통신진흥협회, “제3차 2000년 연도표기문제 대응전략 부산지역 세미나”, 1997c. 9. 26.

한국정보통신진흥협회, “국내외 2000년 문제 대응, 해결 사례집”, 1997d. 11.

Computer Technology Research Corp., “The Year 2000 Crisis”, 1996.

Help for Small and Medium Enterprises to be Year 2000 Ready

(<http://www.ncb.gov.sg/ncb/press/ay2k.asp>).

ICA(International Council for Information Technology in Government Administration) Year 2000 Workshop, Copenhagen, 1997. 8., 14~15.

OMB(미 예산관리국), “Progress on Year 2000 Conversion”, OMB, 1997. 8.

Statskontoret Information system - Year 2000 (<http://www.statskontoret.se/2000/indexe.htm>).

The Millennium Bomb (http://www.open.gov.uk/ccta/mill/dis_gro.htm).

Ulrich, William M. and Ian S. Hayes, “The Year 2000 Software Crisis”, YOURDON PRESS, 1997.

Year 2000 Committee (<http://www.ogit.gov.au/committees/yr2000.html>).

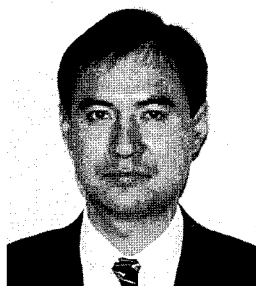
Year 2000 Plan of Action, Denmark (<http://www.fsk.dk/fsk/publ/1997/2000-initiativer/uk/>).

● 저 자 소 개 ●



원 달 수 (dsweon@shinbiro.com)

현재 현대정보기술(주) 기술본부에 재직중이다. 숭실대학교 전자계산학과를 졸업하고 숭실대학교에서 공학석사를 취득하였으며, 국민은행 정보시스템부에 재직시 전산기획, 시스템, 개발 등 금융관련 업무전반을 수행하였으며, 특별히 정부 및 한국은행 주관의 공동연구작업분야에 주로 참여하였다. 현재 정보통신부 체신금융분산시스템구축 프로젝트의 PM을 담당하고 있으며, 배화여자대학 겸임교수, 단국대학교 경영학과 시간 강사로 활동하고 있다. 주요 관심분야는 분산시스템, 차세대금융시스템, ISP, ERP등이다



오 재 인 (jioh@dankook.ac.kr)

현재 단국대학교 경영정보학과 부교수로 재직 중이다. 서울대학교 경영학과를 졸업하고, 미국 볼링그린주립대학에서 경영학석사와 휴스턴대학에서 경영정보학박사를 취득하였다. 미국 텍사스에이앤엠(플레이리비우)에 재직시 아메리칸캐피털, 랜덜스 등 미국 우수기업들의 전략정보시스템 평가프로젝트에 참여하였다. 주요 관심분야는 전략적 정보관리, 지식경영, 전자상거래, ISP, ERP 등이다. 또한 *International Journal on Policy and Information*, 경영정보학연구, 경영과학회지, 한국 CALS/EC학회지 등 학술지와 INFORMS (DSI), TIMS/ORSA 등에서 연구 논문들을 발표해 왔다.