

# 유럽 정보통신산업의 주요 기술 및 표준 동향(II)

조사부

## 4. 소프트웨어 기술

소프트웨어 기술은 새로운 개발의 결과로 다른 기술의 개발을 가져옴에 따라 계속적으로 끝없는 혁신의 흐름을 겪고 있다.

다음의 상호 의존적이고 밀접하게 관련되는 세가지 예가 새로운 개발의 핵심을 보여준다.

- 목적 중심의 소프트웨어 기술
- WEB와 hypertextual로직
- 빠른 개발 어플리케이션과 시각정보 중심의(Visual-oriented) 개발 환경

인터넷과 소프트웨어 콤포넌트의 재이용성의 강화로 소프트웨어 개발에 대한 새로운 패러다임의 이전이 활발해질 것이다.

실질적인 이용 가능성, 수정 및 보안 등 많은 문제들이 아직도 해결되어야 하지만 전세계적으로 이용할 수 있는 수십억가지 중에서 선택된 모듈을 모아서 어플리케이션을 개발하는 능력은 소프트웨어 개발 방법, 로직, 시간과 비용에 있어서 대변혁을 일으키게 될 것이다.

모든 분야의 소프트웨어 기술표준에서 또한 상당한 발전이 이루어지고 있는데 특히 소프트웨어의 전체적인 질과 제한 기준과 관련된 면에서 두드러진다.

1) 시스템 소프트웨어와 유틸리티

### 1) 시스템 소프트웨어와 유틸리티

여기에는 원래 엄격하게 관련되는 소프트웨어가 지배하는 오퍼레이팅시스템, 시스템 유틸리티, 시스템 관리가 포함된다.

시판되고 있는 제품은 종종 이 모든 기능을 하나의 패키지에 포함시키고 있다. 다른 것들은 이들 기능 중에서 한가지만을 지원하거나 PC에서 메인프레임에 이르기까지 다양한 플랫폼에서 이동 및 측정 가능하다.

이 부분의 큰 흐름은 다음과 같

다.

- “개방성(openness)”과 “Client/server” 개념의 폭넓은 수용과 확산
- 현재 server측에서 이용가능한 몇가지 메인프레임 기능으로 기능적인 능력증가
- 고도의 cross-platform scalability
- 사용의 용이성
- 낮은 가격對 성능비

앞으로의 시장 기회에 따라 소비자 중심(consumer-oriented)과 기업중심(enterprise-oriented)시스템의 두가지 흐름으로 주요 기술적인 혁신이 이루어지고 있다.

첫번째 소비자 중심 시스템의 핵심은 멀티미디어 기능, 낮은 가격대, 모든 동작의 단순화, 그리고 사용하기 쉽지만 확실하고 신뢰성이 있어야 한다는 사실과 함께 매우 친근감있는 시각정보(visual) 인터페이스를 포함한다는 것이다. 두번째 기업중심 시스템의 경우에

는 위의 모든 측면이 필요하지만 다른 플랫폼 위에서의 높은 Scalability와 Portability뿐 아니라 고도의 보안성, 이용가능성, 신뢰성에 대한 요구도 있다.

### ① 오퍼레이팅 시스템

PC와 워크스테이션은 IT시장의 주도적인 세력으로서뿐 아니라 정보 시스템 아키텍처 개발의 핵심 요소로서도 중심적인 역할을 한다. 따라서 기술혁신의 측면에서 가장 큰 노력이 이루어지고 있는 곳이 이 분야이며 시장 지배권 싸움이 치열한 곳도 이 분야이다.

그러나 향상과 혁신은 또한 소형/중형 및 대형 컴퓨터 분야에서 급속히 이루어지고 있다.

현재 오퍼레이팅 시스템의 급속한 발전은 다음에 기초한다.

- Microkernel 어프로치
- 목적중심 기술의 광범위한 이용
- 하드웨어 플랫폼과의 독립성 증대
- Plug and Play로직
- OS커널에 새로운 보안 메카니즘을 포함시키는 것을 채택
- 다른 통신망에서도 다른 환경의 상호동작성을 허용하는 다른 프로토콜 파일의 이용가능성을 갖는 개방성(Openness)의 로직
- 새로운 세대의 CPU로의 전환(16bit에서 32bit로, 32bit에서 64bit CPU로)

마이크로커널은 거의 모든 오퍼레이팅 시스템의 필수적인 핵으로서 모든 하드웨어에 따르는 특성

을 분리한다. 따라서 멀티프로세서나 분산된 시스템의 다른 노드에 이식 가능하다. 개념상 마이크로 커널 자체는 제외하고 대부분의 오퍼레이팅 시스템은 매우 플렉시블한 어플리케이션으로서 운용된다.

마이크로커널은 종래의 수직적 성층(Stratification) 기능을 수평적으로 대체 해준다.

마이크로커널 로직은 최신의 첨단 오퍼레이팅 시스템에 이미 채택되고 있다. 더우기 마이크로커널 로직에 아주 적합한 목적(object)의 사용은 오퍼레이팅 시스템에 뛰어난 기능과 다양한 기능을 제공해주며 어떠한 특정한 런타임에서도 필요한 기능을 능동적으로 결합시켜준다.(dynamic binding).

가장 중요한 특징은 Window NT 사용의 큰 증가, Unix의 실질적인 위치 유지, 그리고 IBM 메인프레임 오퍼레이팅 시스템의 급격한 감소이다.

위의 세번째 사실은 기존의 메인프레임 역할의 변화를 확실히 보여주는 것으로서 앞으로 메인프레임은 특수한 데이터 베이스 서버나 임무 중심의 호스트 컴퓨터로서 살아남게 될 것이다.

### ② PC와 NOS

PC와 NOS(Network Operating System)의 OS경쟁에서 기술적인 평가 기준이 되는 것은 현재 Windows환경이 Unix세계를 좌우하며 중간급 시스템을 지배하는 주축인 Microsoft이다. 주요

경쟁자로서 IBM, Apple, Novell, Sun 등이 있다.

1995년은 분명 구형 DOS(16bit 세계)와 새로운 32bit NT환경 사이의 다리 역할을 하는 Microsoft OS인 Windows 95의 한해였다.

Microsoft의 NT서버는 특히 자사의 멀티플랫폼 하드웨어 지원, 어플리케이션 서버가 될 수 있는 자사의 능력, 고객측과의 일반적인 인터페이스, 그리고 기본 사양으로 내장된 매우 풍부한 기능 세트를 강조하며, Novell의 NOS 시장 주도권을 공격하려 하고 있다.

### ③ Unix세계

Unix의 발전은 주로 모든 새로운 플랫폼을 포함하는 방향으로 나아가고 있다. 그리고 시장에서는 가장 많이 사용되는 모든 DBMS에 제공하는 우수한 지원 덕택으로 데이터 베이스 서버의 역할을 차지하고 있다.

현재 Unix는 거의 모든 중소형 플랫폼에서 사용될 수 있으며 주로 네트워킹과 DBMS지원을 목적으로 기존의 메인프레임 세계로 진출하고 있다. 몇몇 Unix제품은 또한 PC/WS 영역에서 그들의 지위를 통합하고 있으며 또한 Windows OLE(Object Linking and Embedding)의 호환성을 지원하고 있다.

1995년 11월에 Novell, SCO, HP는 PC, WS, LAN Server용의 단일한 Unix 버전에 집중하기 위해 계약을 체결했다.

이 계약은 그래픽 인터페이스를

갖는 R.4.2를 향상시키고 우세한 MS-Windows 환경과의 호환이 가능해지면서 Unix의 실질적인 사실상의 표준의 발전 통합에 기초적인 단계가 될 수 있을 것이다.

#### ④ 중대형 시스템의 오퍼레이팅 시스템

중대형 시스템 전용 오퍼레이팅 시스템의 발전은 주로 다음 사항에 기초하고 있다.

- 새로운 RISC 마이크로 프로세서 아키텍처의 지원
- Multi-thread와 Object 로직
- DECI(분산 컴퓨팅 환경 : Distributed Computing Environment) 아키텍처와의 조화
- 다른 플랫폼에서의 Scalability와 Portability

주요 중대형 시스템 제조업체들은 거의 모두 같은 하드웨어 플랫폼에 대한 전용 Unix OS를 공급하고 있다.

#### 2) 시스템 운영

각각의 단일 시스템이 점점 복잡해지고 LAN과 WAN이 분산 컴퓨팅에 빠르게 보급됨에 따라 점점 네트워크 중심화되고 있는 전체 시스템의 제어 및 관리에 효율적이고 플랫폼불한 강력한 틀이 필요하게 되었다.

현재 시판중인 제품의 대다수가 독립적인 제품들이라 하더라도 OSI(Open System Interconnection) 참고 모델에 대한 ISO NMS(Network Management Sy-

stem)과 TCP/IP SNMP(Simple Network Management Protocol)은 법률상, 실질상의 표준이다. 분산 환경에는 DCE내의 OSF DME(Distributed Management Environment)가 아직도 주요 참고점이다.

기능적인 관점에서 향상되어야 할 주요 분야는 구성과 교체 관리, 동작과 문제관리, 성능관리, 그리고 보안 관리이다.

이 분야의 혁신은 주로 다음의 광범위한 사용과 관련된다.

- 그래픽 인터페이스와 우수한 맵 배열(Geographical maps)
- 문제 결정과 문제 해결을 위한 전문가 시스템(Expert System)과 퍼지 로직같은 지능적인 로직
- 네트워크와 시스템 운영의 논리적, 기능적 통합
- 최종 사용자가 원격지에서 필요한 정보와 자원을 찾을 수 있도록 해주는 중앙 및 분산 디렉토리의 주요 역할 : 참고 표준은 X.500Recommendation이다.

- 강력한 점점 자동화되는 Help desk와 이것의 역할이 정보센터의 확산과 Outsourcing으로의 이전 증가에 중요한 부분을 차지할 것이다.

보안 문제는 분산된 환경에서 점점 더 중요한 측면이 되고 있으며 전자무역, EDI, 공공행정 서비스 등에서의 유저가 증가함에 따라 보안 처리의 요구가 늘어나고 있다. OSI와 DCE가 보안을 처리하고 있는데 OSI는 NMS Sy-

stem-Management의 다섯 가지 기능적 분야(시스템 구성, 오류, 성능, 회계, 보안관리)의 하나로써 보안 운영을 포함하고 있으며 아키텍처적인 서비스와 구조를 상세히 설명하고 있다.

대화자와 교환된 메시지의 분명한 입증은 디지털 표시, 대칭과 비대칭의 암호작성 및 해독(cryptography), 칩과 스마트카드, 유저확인을 위한 생체측정 제어(Biometric Control)와 같은 기술과 장치로 해결될 수 있다.

## 2. 어플리케이션 틀

이 분야에서는 목적중심, 시각(Visual)프로그래밍, Client/server 아키텍처와 관련 데이터 베이스의 사용, 툴 통합 수준의 향상, 상호조작성, 재사용, 유지보수성 등에 의해 혁신이 이루어지고 있다. “더 빠르고 더 값싸게”가 개발자들의 절대적인 요구로서 RAD(Rapid Application Development)의 방향으로 진행되고 있는 새로운 노력의 목적인 목적기술, 소프트웨어 구성 부분의 재이용, 엄청난 수의 소프트웨어 업체에 대한 세계적인 접근 등이 이미 가까운 미래에 대비하여 소프트웨어 개발과 중요한 CASE 향상에 실질적인 변화를 허락하고 있다.

RAD 툴은 개발자들에게 그들이 프로토타입을 세워서 이를 생산화시키는 데, 즉 완전히 기능적이고 조작적인 어플리케이션과 환경으로 만드는 데 필요한 모든 것

을 제공한다.

디자인은 인터페이스로부터 출발하여 윈도우, 메뉴, 버튼 등을 시각적인 방법으로 어셈블하여 이루어진다. 모든 요소들은 그래픽의 자체 설명적인 가시적 아이콘(icon)으로서 개발자들은 이용가능한 부분을 간단히 어셈블한다. 즉, 단 몇개의 프로그래밍 문장이 쓰여지는 것이 필요하며 이로 인해 프로그래밍과 디버깅 시간의 엄청난 단축이 초래된다.

소프트웨어 개발을 위한 설계 과정 또한 변하고 있는데, 기존의 엄격한 톱다운(top-down) 방식(분석, 목표설정, 상세한 기능적인 사양과 시스템의 디자인, 프로그래밍, 테스트, 디버깅 그리고 나서 생산으로 이전하는 방식)에서 덜 연속적인 방식으로 바뀌고 있다.

즉 Client 단계에서의 인터페이스 설계와 Client와 Sever의 양쪽 단계에서의 엔진 설계는 프로토타입 같은 방법을 사용하여 병렬적이고 독립적으로 이루어질 수 있다.

그래서 모든 것이 최종 사용자에게 의해 피드백되어 승인되었을 때 이러한 많은 요소들이 통합, 조정되어 생산단계로 넘어갈 수 있다.

한가지 흥미로운 면은 최종 소프트웨어가 다른 크기와 다른 OS를 가진 다른 시스템(예를 들면 메인프레임)에서 실행될지라도 개발 및 시험 환경은 항상 PC나 LAN 단계에서 이루어진다는 것이

다. 두번째 흥미로운 점은 이러한 개발 환경은 종래의 언어(Cobol이나 Pascal같은 즉 목적중심이거나 그렇지 않은)와 C++나 Java같은 새로운 언어 위에서 수립된 시각적인 도움을 이용하여 더 높은 추상적인 개념의 단계에서도 작업할 수 있다는 것이다.

### 1) 목적기술의 통합

유럽 정보기술전망 95(EITO 95)보고서에서는 목적중심의 기술을 자세히 분석하여 IBM, Microsoft, Apple, Next, HP 등의 제조 집합으로 특정되었다.

예를 들어 보험회사의 경우 자동차나 생명보험 증권같은 상품 대신에 고객, 보험료, 보험금 지급별로 체계화된 자료의 집합을 말한다. 이렇게 수집된 자료의 통합되어 비소멸적인 연대순으로 체계화되어 현재의 운영 요구를 지원해준다.

데이터가 거의 업데이트되지 않음에 따라(어떠한 새로운 데이터라도 이미 저장된 것에 기본적으로 추가됨) 단지 "read" 업체들의 다른 방법을 기술적으로 비교한 바 있다.

그로부터 1년후 상황은 크게 바뀌지 않아서 거의 모든 업체들은 아직도 그 단계에서 활동하고 있다. OLE 인터페이스 사용의 일반화 덕분으로 Microsoft가 Window와 client/Server 어플리케이션의 이용을 확대하고 있지만 아직 단독 승자는 없다.

확실한 것은 목적중심 기술이

단지 일시적인 유행이 아니라 실질적인 혁명으로 모든 정보통신기술의 구도 변화를 가져온다는 것이다.

Cobol 또한 곧 그 OO를 갖게 되리라는 것은 중요하다. ANSI는 1993년에 OOCobol을 발표했으며 1997년에는 표준화될 수 있을 것으로 예상된다. 그동안 몇몇 OO Cobol 컴파일러가 1995년에 시장에 나타났다.

### 2) 데이터베이스, 데이터 창고(Ware house) 다차원(multi-dimension) 데이터베이스와 OLAP

상관 데이터베이스 판매업체들은 ODBMS(즉 멀티미디어 응용에 필요한 Binary Large Objects(BLOB)를 저장하는 능력)와 경쟁하기 위해 몇가지 기능을 그들의 제품에 추가하고 있다.

상관 데이터베이스와 목적 데이터베이스 사이에는 기본적인 차이가 있다. 상관 데이터베이스는 대량의 데이터를 잘 처리하지만 복잡한 데이터는 그렇지 못하다.

반면에 목적 데이터베이스는 대량의 데이터를 잘 처리하지만 복잡한 데이터는 그렇지 못하다. 반면에 목적 데이터베이스는 복잡한 데이터를 처리할 만큼 충분히 스케일이 증대되지 못한다.

데이터 창고(warehouse)의 정의는 1981년에 처음 도입되었지만 그 개념은 최근에 비즈니스의 맥락에 실제로 이용되고 있다.

데이터 창고는 세계의 비즈니스

와 어떤 회사의 역사에 관한 역사적인 자료를 포함하고 있어 쉬운 방법으로 찾아질 수 있으며 의사 결정에 유용하다.

Inmon은 데이터 창고를 어플리케이션이 아닌 엄밀하게 주체에 관련된 데이터의 할 수 있다.

따라서 이들에게 비소멸성을 제공한다. 데이터는 온라인으로 빈번하게 업데이트되는 다른 통용 데이터베이스로부터 수집되는데 이는 데이터 창고에 저장된 정보의 통합을 보증하기 위해 종종 다른 방법으로 기록된 그 내용이 동일한 구조로 이루어져야 함을 의미한다.

전형적인 데이터 창고는 매출, 마케팅 재정, 제조 및 인적 자원 등에 관한 기업의 데이터베이스이다.

의사 결정권자들에게 월간 총수입이나 송장 작성(invoicing)에 대한 간단한 정보는 더 이상 의미가 없다. 이들은 지역별, 산업분야별, 직·간접 경로별, 개별 상품 및 판매원별 총매상고를 분석할 수 있는 것이 필요하다.

이러한 데이터베이스의 사용자들은 간단한 방법으로 매우 빠르게 서로와 아주 다른 복잡한 조사를 할 수 있는 것이 필요하다. 그 결과로서 이러한 기록보관은 다차원 데이터베이스에 의해 이루어지고 있으며 조사의 일부는 온라인 분석처리(On-line Analytical Processing(OLAP))에 의해 이루어지고 있다.

전형적으로 데이터 창고에 보관

된 매우 많은 양의 정보는 또한 대량의 병렬 처리에 기초한 특정한 데이터 베이스 서버에 의해 지원된다.

다차원 데이터베이스는 위에서 서술한 기능과 성능을 제공할 수 있는 전통적인 상관 데이터베이스를 특별히 발전시킨 것이다.

다차원 데이터베이스는 요구되는 데이터가 여러 차원의 교차점에서 발견되어지기 때문에 다수의 개별 차원에 대한 복잡 매트릭스를 세우는 한 세트의 상관 구조를 기초로 하고 있다. 비즈니스 어플리케이션의 경우에 차원은 시간, 위치, 판매업체, 제품, 광고 방법 등이 될 것이다. 평가 기준은 예산, 전망, 송장, 총 수입이 될 것이다.

각각의 차원이 기초적인 항목(예를 들면 특정 판매업체)이나 복합적인 항목(예를 들면 국별, 지역별, 지방별, 자치시별로 구분된 위치 등의)이 될지도 모르기 때문에 아마도 의미가 없는 것은 제거해 가면서 현명한 방법으로 데이터를 관리할 수 있도록 해준다. 즉 어떤 판매업체는 몇몇 분야에서만 관계될 수 있으며 몇몇 제품은 일부 채널에 의해서만 판매될 수 있다는 것이다.

다차원 데이터베이스는 양적인 값과 측정 가능한 값(수치)과 서술적인 항목(제품의 색깔같은)에 의해 데이터를 체계화한다.

이 구조는 논리적으로 입방체(cube나 hyper-cube)로 보여질 수 있는데 각각의 면은 분류의 범

주를 나타낸다. 데이터 회전(data pivoting) 혹은 입방체를 회전하는 능력은 유저에게 다른 관점에서 필요한 정보를 빠르게 선택할 수 있도록 해준다. 예를 들면 결과의 연간 비교 10대 항목의 분류, 최고/최악의 성과 순위 등이다.

주로 역사적인 자료에 대해 이용되고 상관 데이터베이스보다 훨씬 많은 대량의 메모리를 사용하는 다차원 데이터베이스의 주요 목표는 가능한 변수와 차원내에서는 어떠한 요구에 대해서도 즉시적인 답을 제공하는 것이다.

On-line Analytical Processing(OLAP)은 역사적 데이터의 탐구와 분석에 대한 모든 기술을 포함하고 있어 처리되어야 할 데이터(transactional data)의 계속적인 업데이트를 엄격하게 지향하는 On-line Transaction Processing(OLTP)와는 크게 다르다.

OLAP는 기본적으로 수많은 종래의 데이터베이스에 의해 공급되는 데이터 창고(data warehouse)를 구성하는 다차원 데이터베이스에서 실행된다. OLAP는 최종 유저가 특별한 개발의 필요없이 확실히 데이터를 분석할 수 있도록 해준다.

따라서 다차원 데이터베이스와 OLAP은 데이터 창고를 특징짓는 엄격하게 상호 연관된 개념이다. 현재 합법적인 표준은 없으며 모든 시판 제품은 전용 제품이다.

몇몇 제품은 소프트웨어화되어

있으며 다른 플랫폼으로 전환될 수도 있고 client/server나 분산 환경에서 사용될 수도 있다.

또 다른 제품들은 전문화되고 매우 신뢰성있는 고성능 컴퓨터에서, 그리고 종종 멀티 프로세서 아키텍처에서 실행되는 특별서버 시스템이다.

### 3. 퍼지 시스템

퍼지 이론은 1965년 L.A.Zadeh에 의해 소개되었지만 최근에 와서야 종종 뉴런네트워크와 확률 이론과 함께 퍼지 시스템의 개발을 위해 널리 이용되기 시작하고 있다.

퍼지 시스템은 부정확함과 불확실함의 허용치를 개발하고 경험을 통해 배워 코스의 변화에 적응할 수 있는 매우 지능적인 시스템이다.

퍼지 시스템은 직관적인 생각과 관련된 복잡한 문제의 해결에 효과적이다.

모호집합, 이론, 알고리즘, 모델, 소프트웨어, 프로세스와 컨트롤을 포함하는 퍼지 시스템은 주어진 과업을 수행하는데 요구되는 정밀성을 최소화시키는 개념과 기술을 제공한다. 이 시스템은 정확한 추론 방법보다는 근사치를 사용하는 인간의 능력을 따르려고 노력한다.

현재 로봇공학에서 가정용 전자 기기, 자동차에서 항공우주 산업, 의학에서 경제학에 이르기까지 수많은 다양한 응용 분야에서 퍼지

시스템의 개발이 증가하고 있다.

넓은 의미에서 보면 퍼지 이론을 명확하게 정의된 경계를 갖지 않는 한 모호집단(퍼지세트)의 이론과 거의 같다. 어떠한 뚜렷한 이론이라도 그 이론내의 어떤 집합의 개념을 모호집단(퍼지세트)의 개념으로 일반화함으로써 퍼지화될 수 있다.

퍼지 시스템의 요점은 그 값  $[0.0, 1.0]$ 의 범위에 있다는 것이다. 여기에서 0.0은 절대적인 거짓이고 1.0은 절대적인 참값을 의미한다. 예를 들면 “알렉스는 어리다(Alex is young)”라는 진술은 만일 알렉스가 6세라면 0.90이라는 참값이 주어질 수 있다. 그리고 이것은 우리에게 퍼지 세트의 개념을 가져다 준다. 한 집합의 사람들의 나이를 예를 들면 다음의 표는 젊음의 정도를 각 원소에 관련시킨다.

이름	나이	정도 (등급)
마 크	45	0.1
알 렉 스	6	0.9
안 드 레 아	2	1

한 집합의 회원 함수는 각 원소를 1.0과 0.0 사이의 등급에 배치한다. 그리고 “Alex is a member of the set of young people”이라는 진술은 상징적으로 다음과 같이 표시될 수 있다.

$$m_{\text{YOUNG}}(\text{Alex})=0.9$$

여기에서 m은 젊은 사람들이라는 모호한 집합의 예에서 작용하는 회원 함수이다. 목적 집합

(X)에서 각 원소는  $X : X = \{X\}$ 로 표시될 수 있다.

X에서 모호집합 A는 각 원소를 실제로 0.0~1.0사이의 등급에 배치하는 회원 함수  $m_A(x)$ 에 의해 특징지어진다.  $m_A(x)$ 가 1.0에 더 가까워짐에 따라 x의 회원의 등급은 올라간다.

이러한 기본 개념에서 출발하여 다양한 연산과 성질을 끌어내는 것이 가능하다.

모호집단이 정규 집단의 확장인 것과 마찬가지로 퍼지 이론은 정상이론의 확장이다. 정상 이론처럼 퍼지 이론은 합집합(OR), 교집합(AND), 그리고 여집합(NOT)의 개념을 도입한다. 모호집합에서 원소의 등급은 퍼지 이론에서 참값의 정리에 해당할지도 모른다. 퍼지 이론은 A이면 B다 (If A then B :  $A \rightarrow B$ )와 같은 모호한 관계를 정의하며 두개의 모호집합(A와 B는 모호집합)사이의 관계로서 보여진다. 따라서 다음의 공식으로 정의하는 것이 가능하다.

만일 모든 X에 대해 A가 공집합이면

$$m_A(x)=0$$

만일 모든 X에 대해  $A=B$ 이면

$$m_A(x)=m_B(x), \text{ 간단히 } m_A=m_B$$

$$\text{여집합 } m_A=1-m_B$$

만일  $m_A \rightarrow m_B$ 라면 A는 B에 포함된다.

$$C = A \cup B, \text{ 여기서 } m_C(x) = \text{MAX}(m_A(x), m_B(x))$$

$C = A \cap B$ , 여기에서  $mC(X) = \text{MIN}(mA(x), mB(x))$

이 점에서 퍼지 시스템과 확률(probability) 사이의 차이를 이해하는 것이 중요하다. 둘다 똑같은 수치 범위를 사용하며 둘다 같은 값을 갖는 것 같다. 그러나 의미적인 관점에서 큰 차이가 있다.

위의 보기에서 확률 이론은 Alex가 젊다는 90%의 기회를 언급한다. 즉 “Alex가 젊은 사람들 집합의 일원이다”라는 퍼지 이론과는 반대로 “Alex가 다소 젊거나 그 값이 0.90에 해당하는 어떤 다른 자격자”임을 의미한다.

합집합(OR)과 교집합(AND)은 확률 이론과는 한층 더 분명한 차이를 보여주는 것으로 여기에서 OR연산은 덧셈(합계)이고 AND연산은 곱셈이다.

퍼지 이론에서 또 하나 중요한 부분은 언어상의 변수 개념이다. 우리가 이전에 예를 들었던 “나이”는 “어리다(young)”, “중년이다(middle aged)”, “늙었다(old)”의 값이 회원 함수로 정의되는 언어상의 변수로 생각될 수

있다. 언어상의 변수는 데이터 압축의 형태로서 보여질 수 있다. 즉 “Granulation”으로 언급되며 “Hedge”를 갖는 모호집합의 경우이다.

언어상의 “Hedge”의 예와 그 의미는 다음의 표에서 설명된다.

이러한 “Hedge”는 수학적인 계산을 통해 자연 언어와의 밀접한 관계를 유지하면서 퍼지값을 수정한다.

about, around,	실수로 표시할 수
near roughly	있는 수량에 가까움
above	퍼지 범위가 확대됨
almost, definitely, positively	대조 강화
rather, somewhat quite	퍼지 범위의 회박
very, extremely	퍼지범위의 강화

간단한 예로 “Alex is very young”이라는 진술을 생각해보자. 우리는 어떻게  $m$  “very” YOUNG(Alex)?를 계산할 수 있을까? very라는 “Hedge”는 정상적으로  $m(A)(x)$  2로서 정의된다. 따라서 만일  $mYOUNG(Alex) = 0.9$  이면  $m$ “very” YOUNG(Alex)

$= 0.81$ 이 된다. 다른 예로서 “more or less : 다소” 혹은 “about : 약”등이 있는데 이는 정상적으로  $\text{SQRT}(mA(x))$ 로서 계산된다.

퍼지 제어는 주어지는 적합한 입력(전형적인 것으로 시스템 자체로부터 측정)이 시스템 제어를 위한 아웃풋을 결정할 수 있는 퍼지 추론에 의한 기술이다.

퍼지 모델은  $M \times N$ 영역의 모호 범위의 근사값으로 여기에서  $M$ 은 퍼지제어 변수의 이며  $N$ 은 양립하는 범위이다.

지금까지 살펴본 것처럼 퍼지 시스템의 어플리케이션은 구체적으로 이미 많은 분야에서 사용되고 있다. 최초의 실질적인 어플리케이션은 화학분야의 생산 관리에 있어서 몇가지 실용적인 방법의 능동적인 이용을 요구하고 있는 사업의 의사결정 지원 시스템에 도입되었다. 퍼지 시스템은 소프트웨어로 실행되며 종종 전문가 시스템과 뉴런 네트워크에 밀접하게 관련된다.

## 값진희생 빛난공훈 깊이새겨 나라사랑