사용자 환경 접목한 맞춤 서비스 시대

애플리케이션 활용 · 구축 방법 숙지, 네트웍 유지보수 기술 중요

클라이언트/서버 개발에 있어 단일 서버용 솔루션만 다루던 과거와 달리 이제는 인터넷은 물론 다중 네트웍 운영체체 및 다중 플랫폼을 이용해 네트웍 중심의 분산 애플리케이션을 구축하고 있다. 따라서 애플리케이션이나 데이터베이스 서버, 디바이스, 파일 등의 네트웍 접속이 가능한 자원(resource)의 위치를 알아내고, 이를 활용하는 것이 다수의 시스템을 다루는 데 있어 매우 중요한 사안이 되고 있다. 더욱이 자원들을 해커로부터 보호하기 위한 보안 문제를 비롯, 사용자들이 원하는 각각의 자원에 대한 세부사항을 제공하는 일 또한 선결 과제라 할 수 있다. 〈편집자〉

디렉토리 서비스란 네트웍 자원의 위치(location), 식별(identification), 사용(use), 권한(authorization)을 제공하는 메카니즘이다. 디렉토리 서비스는 애플리케이션을 향한 하나의 엔트리를 제공하고, 디렉토리의 공유 세트를 사용할 수 있도록 지원하며, 서비스 표준을 규정한다.

즉, 사용자들이 수천개의 유용한 자원들 속에서 자신이 필요로하는 애플리케이션을 구축할 수 있도록 도와주는 일을 하는 것이다.

디렉토리 서비스가 아직까지는 사용자의 애플리케이션 기반 구조를 위한 확실한 방법을 제시하고 있지는 못하다. 최근 사용자의 욕구에 부응할 만한 몇몇 제품군이 출시돼 있기는 하다. 하지만 디렉토리 서비스 개념으로의 다양한 접근을 시도하고, 플랫폼 자원을 단계별로 차별화하며, 고유한 방식으로 실질적인 디렉토리 서비스 기준을 지원하는 수준에 머물러 있을 뿐 본래의 디렉토리 서비스 개념과는 상당히 다른 접근을 시도하고 있다. 이러한 상황에 실망스러울 수도 있겠지만 최근 몇년 간의 발전 상황을 살펴보면, 애플리케이션의 분산화가 구체화되면서 디렉토리 서비스의 발전 역시 계속 진척되고 있다.

가치

디렉토리 서비스는 사용자가 네트웍상에서 지능적으로 자원을 찾을 수 있는 애플리케이션을 구축할 수 있도록 한다. 디렉토리는 애플리케이션을 대신해 자원들의 위치를 파악하고, 형태를 바꾸거나 움직이고 삭제되면서 자원들을 순회한다. 예를 들어, 전자우편 애플리케이션은 유저 그룹의 위치를 알아내고, 워드프로세싱 애플리케이션은 프린터를 찾아내며, 클라이언트/서버 애플리케이션은 자원들이 네트웍 어디에 존재하는지에 관계없이 데이터베이스를 발견해 낸다.

애플리케이션 객체는 어느 특정 서버 상에 위치하는 것이 아니라 모든 네트웍 상에 존재한다. 따라서 개발자가 객체의 위치를 파악하기 위해 일반 인프라스트럭처를 확보하는 일은 필수적이라 할 수 있다. 객체와 데이터는 모두 디렉토리 서비스에 등록되어 있거나 디렉토리에 상주하고 있다.

만일 독자의 네트웍 관리자가 책임을 분산시키고 있다는 점에 대해 못마땅해 하고 있다면 그가 추진중인 디렉토리 서비스에 대해 알아보라. 디렉토리 서비스는 애플리케이션 인프라스트럭처를 통해 인위적으로 동기화시키지 않아도 중앙에서 유저의 추가, 변화, 삭제를 제공할 수 있어 독자의 걱정을 해소시켜 줄 수 있는 서비스이다.

서비스 기반

사실 디렉토리 서비스는 네트웍 상에 있는 자원들을 분류하기 위한 하나의 순차적인 방식에 지나지 않는

다. 실생활에서 이같은 예는 얼마든지 찾아볼 수 있다. 예를 들어, 생물학자는 모든 생물을 종, 속, 과, 목, 강, 문, 계로 분류한다. 이와같은 접근방식과 마찬가지로 디렉토리 서비스도 각 분야별 방향을 설정하기 위해 네이밍 시스템을 이용한 분류체계에 따라 컴퓨터 시스템을 식별, 분류한다.

현재 네트웍 디렉토리 서비스에는 DNS(Domain Name System), 노벨의 네트웨어 디렉토리 시스템(Netware Directory System)과 노벨 디렉토리 서비스(Novell Directory Service), 넷스케이프의 디렉토리 서버(Directory Server), 마이크로소프트의 액티브 디렉토리(Active Directory)와 X.500 등이 있다. DNS는 모든 인터넷 유저에게 서버 이름(www://dbmsmag.com 같은)을 분석할 수 있도록 도와주는 디렉토리 서비스를 제공함으로써 수년간 역할을 훌륭히 수행해 왔지만 하나의 단일 함수로 제한되어 있다는 단점이 있다. 인터넷이 번창하는한 디렉토리 서비스의 요구는 더욱 커질 것이다.

1988년에 국제전신전화자문위원회(CCITT)는 X.500 디렉토리 서비스의 두개 버전 중 최초의 것을 출간했다(갱신된 버전도 1993년에 나왔다). X.500은 세계적인 디렉토리 정보의 보고로 유저 이름과 패스워드, 사진, 위치, 액세스 컨트롤 메카니즘 등 귀중한 자료를 보유하고 있다. 애플리케이션이 자원의 위치를 쉽게 찾을 수 있는 메카니즘을 제공하는 X.500은 DNS 대체용으로 판매된 제품으로 DNS보다 적응하기 쉽고 기능과 특징이 더욱 다양하다는 장점을 갖는다.

X.500 디렉토리 모델은 정보의 논리적 데이터베이스 제공을 돕는 시스템 중 하나로 분산된 집단 시스템이다. 디렉토리 서비스 사용자는 이 정보를 이용하여 디렉토리 가입자에게 디렉토리 서비스를 액세스할 수 있다. X.500 디렉토리의 모든 정보는 '엔트리(entries)'에 존재한다. 각 엔트리는 적어도 하나 이상의 '객체 클래스(object class)'를 가지고 있다.

디렉토리 서비스는 네임스페이스(namespace)에 의존하고 있는데, 네임스페이스는 관련 정보(이름, 전자우편, 주소, 조직 등)의 집단화를 효과적으로 참조, 검색할 수 있도록 한다. X.500 네임스페이스에서는 논리적이고 체계적인 디렉토리가 제공되며, 엔트리의 구조를 다루고 있지만 유저에게 정보를 프리젠테이션하지는 않는다. X.500 DIT(Directory Information Tree)에 있는 모든 엔트리는 진정한 속성들의 모음이다. X.500은 별명과 국가, 지역, 조직, 조직 단위, 사람 등의 객체 클래스를 정의한다.

속성 타입에는 공통 이름(CN), 조직 단위 이름(OU), 조직 이름(O), 지역 이름(L), 주소(SA), 주(S), 국가(C) 등이 포함된다. 예를 들어, 다음처럼 적용시켜 필자를 찾아 보라.

C=US, O=Ernst-Young, OU=CTE, CN=Dave

이 주소가 X.500 분류체계로 작동하면 〈그림〉처럼 나타난다.

앞의 보기에서 처럼 디렉토리가 가능한 애플리케이션은 하나의 DUA(Directory User Agent)를 이용하여 디렉토리를 액세스한다. DUA는 DAP(Directory Access Protocol)를 이용해 요구들을 하나의 DSA(Directory System Agent)로 전달한다. DSA는 정보 공유를 위해 내부적으로 통신하거나 할 수 있는 '대상자(referral)'를 특정 DSA로 보냄으로써 임무를 완수한다.

그런데 이렇게 유용한 X.500이 왜 현재는 사용되지 않는 걸까? X.500도 단점이 많이 있기 때문이다. 우선 X.500 검색에는 상당한 시간이 소요되는데, 이는 거대한 용량의 디렉토리 데이터베이스와 데이터베이스 액세스를 위해 사용되는 오버헤드 디렉토리 액세스 프로토콜 때문이다. 둘째, DAP는 네트웍을 통한 링크를 위해 OSI 스택을 사용하고 있는데, 디렉토리 서비스를 포함하는 이 OSI 스택에는 엄청난 용량의 메모리와 자원이 요구된다. 몇몇 벤더가 X.500을 표준으로 삼고 있기는 하지만 대부분은 이 취약점들을 심각하게 받아들이게 되었고, 결국 몇년째 IT조직에서 밀려나는 결과를 초래했다.

DNS가 퇴화되면서(1997년 7월의 인터넷 DNS 주가 하락 현상을 상기하라) 글로벌화, 분산화되지 않은 인터넷 디렉토리 문제를 해결하기 위해서 좀더 진보된 디렉토리 서비스가 필요하다는 사실을 모두가 절감했다. 결국 X.500을 향상시키려는 시도를 꾸준히 해온 미시간대학에서는 1993년 LDAP(Lightweight Directory Access Protocol)를 개발했다.

당시 LDAP는 디렉토리 데이터베이스 액세스를 제공하는 X.500의 최대 장점을 취하고 있어 효율적이기는 했으나, DAP 일부분만을 사용하고

있어 규모가 작은 단점이 있었다. 결국 LDAP는 애플리케이션 부분을 버리고 디렉토리 정보의 일부에만 액세스할 수 있는 디렉토리 서비스로 완성되었다. 그러나 LDAP는 디렉토리 서비스 사용자를 만족시키기에는 여전히 이해하기 어려운 제품이었다. 이 제품 역시 중간 타협안을 제시한 것이다.

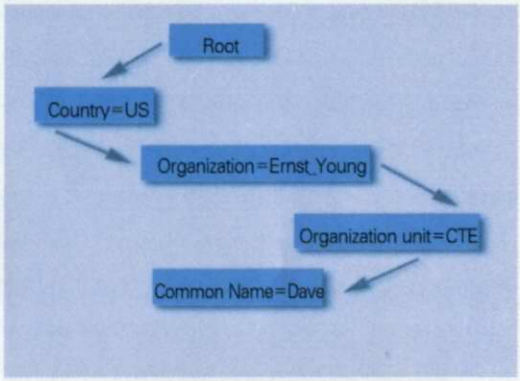
LDAP로 진입

이제 우리의 목표점을 알게 됐으니 우리가 도달한 곳에서 무엇을 찾아냈는지 자세히 알아보자.

LDAP는 대부분의 오버헤드 세션과 프리젠테이션 계층을 우회하여 TCP나 다른 전송 계층 프로토콜의 지렛대 역할을 수행할 수 있다. X.500을 이해하고 있다면 LDAP를 이해하는데 별 어려움이 없을 것이다. 이것은 X.500 디렉토리 모델 엔트리에 기초하고 있으며, 엔트리에 언급된 이름과 구별되는 이름을 사용하고 있다.

반면, 고도로 구조화된 X.500 데이터 인코딩 메카니즘을 사용하는 대신에 LDAP는 엔트리를 표시하기 위해 단일한 문자열에 기초한 접근방식을 시도하고 있다. 따라서 애플리케이션의 디렉토리 서비스 요구는 문자열이 기본 인코팅 규칙에 따른 부분 구조를 사용함으로써 훨씬 단순화되었다. LDAP는 엔트리가 객체 클래스 속성을 사용하게 함으로써 디렉토리 데이터베이스의 검색 방법을 훨씬 더 용이하게 했다. 일례로, 세분화된 질의로 검색을 제한할 수 있다.

우선 LDAP는 클라이언트가 TCP/IP를 사용하여 단일 LDAP 서버로 인터랙트할 수 있도록 설계되었다. 단일 LDAP 서버가 OSI를 이용할 경우 X.500의 단일 서버나 그룹과 대화가 가능하다. 원래 LDAP 사양은 RFC 1487이었는데, 이는 LDAP간의 커뮤니케이션이나 복제(replication), 선행 보안을 위해 제공되지는 않았다. LDAP는 디렉토리의 인터랙션과 유지보수를 제공하는데 있어서 여전히 X.500에 의존하고 있다.



〈그림〉 X.500 계층형에서 사용자는 목표를 설정하기 위한 트리로 옮겨갈 수 있다.

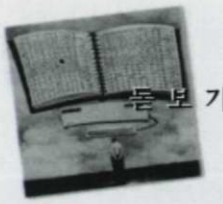
LDAP는 IETE 표준 프로세스(웹상에서 ds.internic.net/rfc/rfc1777.txt에 있는 RFC 1777 참고)를 통해 버전 2(LDAPv2)로 발전되었고, LDAP의 최신 사양인 LDAPv3가 현재 초안 상태에 있다. LDAP도 웹 진출을 목전에 두고 있다. LDAP URL 포맷은 RFC 1959에서 정의된 웹브라우저 유저가 표준 URL을 통해 LDAP 디렉토리 검색에 액세스할 수 있도록 지원한다. 예를 들어, "ldap://ldap.abccorp.com/ou=mis, o=abccorp, c=us?dave"라고 정의하면 디렉토리 트리를 사용해 LDAP 디렉토리 검색을 수행할 것이다.

보안 문제에 있어서는 명확한 텍스트 패스워드를 사용하여 인증하도록 하고, 커베로스(Kerberos)도 지원하고 있다. 버전 3(LDAPv3)는 X.509를 사용하여 강력한 인증을 지원함으로써 보안을 향상시킬 계획이며, 이로 인해 공용 키를 저장할 수 있는 특징이 있다.

그렇다면 무엇이 문제인가?

LDAP가 X.500의 디렉토리 검색 용량 중 하나의 서브 세트만을 제공하고 있다는 점이다. 예를 들어 보자. DAP는 질의를 프로세스하는데 여러 방식을 제공하고 있다. 클라이언트에게 디렉토리 정보를 정렬하여 전송하기도 하고, 질의를 처리할 수 없을 때 다른 서버에게 검색을 맡기거나 만족할 만한 서버를 추천하기도 한다. 현재 LDAP는 이점을 해결하지 못하고 있다. 이는 LDAPv3에서 보완될 것이다.

LDAPv2에서는 LDAP 서버가 X.500 디렉토리를 독립적으로 사용할 수 있도록 하여 LDAP가 오버헤드와 복잡도를 완화시키고 작업 완성도를 높일 수 있게 한다. LDAP 종속 서버는 마스터 디렉토리 서버에 링크되어 서브 트리의 디렉토리를 갱신할 수 있다. 이것은 회사내 각 분야(회계, MIS, 판매 등)의 마스터(master) 서버를 셋업할 수 있는 능력을 조직에 제공하고, 분류체계에서 다른 모든 서버들을 하위계급으로 분류한다. LDAPv3에서는 이러한 아키텍처가 달라질 예정이고, 중복 자원을 더욱 활용하게 될 것이다. 또한 LDAP 서버가 질의를 처리하다가 다른 디렉



토리 서버도 추천할 수 있게 될 것이다.

한편, LDAP에서는 다른 서버에 저장된 정보에 관해 더 많은 것을 알아내기 위해 다중 트리를 사용할 수 없다. LDAP 다음 버전에는 마스터 서버가 인덱스 작업을 활발히 수행할 수 있고, 다른 마스터 서버를 복사하는 일명 선행 인덱스(forwading Indexes) 작업이 가능하게 될 전망이다.

다른 사양이 부족함에도 불구하고 LDAP는 아직 개발의 여지가 많이 남아 있다. 복제와 같은 새로운 특징들이 부분적으로 표준화되면서 LDAP 성공의 열쇠는 다름아닌 벤더들의 지원에 있다고 할 수 있다. 지금까지 로터스 디벨롭먼트와 IBM, 노벨, 마이크로소프트, 넷스케이프 등은 자신들의 전용 디렉토리 안에 LDAP 지원을 추가하면서 LDAP 개발에 모두 협력할 것을 약속했다.

앞으로 강력한 글로벌 디렉토리 서비스 메카니즘을 구축하기 위해 LDAP가 각 업체별 전용 디렉토리에서 보유한 모든 정보를 바인딩할 수 있게 될 것이다. 이에관해 LDAP가 가능한 디렉토리 서비스 제품 중 몇가지 견본을 검사하는 일이 도움이 될 것이다.

액티브 디렉토리

마이크로소프트의 액티브 디렉토리는 현재 지원하고 있는 평면 모델에 계층적 도메인 아키텍처를 접목시킨 것이다. 액티브 디렉토리는 전용 제품이지만 넷스케이프나 노벨 등의 타사 디렉토리와 정보 교환을 위한 커뮤니케이션 프로토콜로서 LDAP를 지원할 예정이다. 액티브 디렉토리는 계층형의 경우에 사용자가 몇가지 도메인을 제어할 수 있게 해준다.

액티브 디렉토리 서비스 프로바이더 인터페이스는 액티브X를 이용하여 윈도우 NT와 외부의 전용 디렉토리 서비스간의 상호연결을 지원한다. 이는 마이크로소프트와 써드파티 벤더간의 통합된 공통 계층으로 서비스 제공자와의 인터페이스를 제공하는 기능을 말한다.

또한 노벨 NDS 바인더리 같은 타 디렉토리 서비스에 플러그 인하고 관리하는 것을 허용한다. 관리자는 마이크로소프트 매니지먼트 콘솔(Microsoft Management Console)을 통해 액티브 디렉토리를 제어할 수 있다. 액티브 디렉토리의 이점은 현재 데스크탑 시장과 서버 시장을 점령하고 있는 윈도우 95와 윈도우 NT를 모두 지원하는 것이다. 액티브 디렉토리의 약점은 비윈도우 플랫폼에서는 같은 수준의 지원을 제공하지 못한다는 점이다.

넷스케이프 디렉토리 서버

스위트스팟의 한 컴포넌트인 넷스케이프 디렉토리 서버는 LDAP를 기반구조로 사용한 디렉토리 서비스를 제공한다. 넷스케이프는 넷스케이프 커뮤니케이터(네비게이터 4)에서 LDAP를 지원하는데, 이는 파트너를 끌어들이고 표준화를 이끌어가기 위한 넷스케이프사의 디렉토리 전략 일환이다.

디렉토리 서버는 윈도우 NT 뿐만 아니라 SGI, IRIX, 솔라리스, AIX, HP-UX 등 몇몇 플랫폼 상에서 유용하다. 관리자는 모든 넷스케이프 서버에 있는 HTTP 기반 어드미니스트레이션 서버를 이용하여 디렉토리 서버를 관리할 수 있다. 디렉토리 서버는 관리자가 다양한 지리적 위치에서 마스터 서버 상에 있는 장치들을 조직화할 수 있도록 하는 중복 메카니즘을 제공한다.

이곳에서는 같은 서버 상에서 전형적인 상호 검색을 하는 유저들을 집단화하는 반면, 필요한 정보가 지역적이지 않을 경우 서버를 다른 마스터 서버에서 쿼리할 수 있도록 해준다. 이러한 기능은 글로벌 디렉토리 서비스와 함께 최상의 수행을 할 수 있도록 돕는다. 디렉토리 서버는 또한 모든 마스터 디렉토리를 구성하는 리플리카(replicas)로 관리자가 마스터 서버를 결합해 유저에게 완벽한 디렉토리 맵을 제공할 수 있도록 한다.

디렉토리 서버는 관리자가 키와 필드를 첨가할 수 있도록 커스토마이즈할 수 있는 데이터베이스를 제공한다. 디렉토리를 커스토마이즈하기 위한 요령은 LDAP 표준 속성을 따르지 않는 것이다. LDAPv2로는 비표준 디렉토리 스키마 사용법을 알 수 없을 것이다.

노벨 디렉토리 서비스

노벨 디렉토리 서비스(NDS)는 모든 기업을 전체적으로 검색할 수 있는 노벨 네트웨어 네트웍을 제공한다. NDS에서는 계층적 트리 구조 내에 있는 사용자, 그룹, 프린터, 볼륨, 기타 장치 등 각 자원 정보를 유지보수하는 기능이 있어 복잡한 기업 네트웍을 간단하고 논리적으로 검색할 수 있게 한다.

네트웍 어디에서나 폴트-톨러런트 로그인과 관리를 제공하기 위해 NDS는 복제된다. 이러한 폴트-톨러런트 기능을 제공하기 위해 디렉토리는 고도의 중복 서비스를 이용하여 관리가 가능하도록 세분화되고 네트웍 상에 분산된다. NDS 데이터베이스는 확장 가능하며, 사용자가 기업이나 애플리케이션의 요구에 부합하도록 NDS를 맞춤화할 수도 있다.

NDS는 관리자가 NDS 트리의 특정한 분자를 위해 보안을 정의하는 계층적 관리 기능을 제공하는데, 이로 인해 그 분야의 모든 객체는 보안 기능을 하는 기본 세트를 취하게 된다. 인증 서비스는 RSA 공용 키와 개인 키를 사용하는 암호화 기술로 네트웍 접근을 제어할 수 있다.

NDS는 또한 파일 서비스, 프린트 서비스, 보안 서비스 등 다양하게 분산된 서비스를 제공하는데, 관리자는 이를 이용하여 사용자가 중앙에 있는 네트웍 애플리케이션에 접근하는 것을 제어한다. 또한 이제는 관리자가 네트웍 클라이언트상에 있는 최신 또는 기존의 애플리케이션을 웍스테이션 없이 설치하거나 업그레이드할 수 있게 되었다.

오라클과 같은 데이터베이스 벤더들은 NDS로 직접 작업할 수 있는 방법을 모색 중이다. 오라클은 자사의 데이터베이스 소프트웨어와 NDS를 합성하여 사용자에게 단일한 사인-온과 네이티브 네이밍(native naming) 기능을 제공한다. 이러한 기능은 오라클 데이터베이스 기반의 운영체제에 로그인하는 사용자를 자동으로 인증하도록 할 수 있다. 네이티브 네이밍 기능은 사용자가 오라클 서비스 이름 대신에 디렉토리 트리에 있는 데이터베이스 객체 이름을 지정하여 오라클 데이터베이스에 접속할 수 있도록 한다.

자바의 향방

우리는 모든 것을 '자바화' 하고 있다. 하물며 디렉토리 서비스라고 예외일 수는 없다. 썬마이크로시스템즈의 자바소프트 부문은 현재 JDNI(Java Directory and Naming Interface)로 알려져 있는 표준 자바 디렉토리 상에서 작업하고 있다. JDNI는 자바 애플릿과 애플리케이션에 네이밍과 디렉토리 서비스를 제공하고 있는 네이티브 자바 API이다. JDNI는 명명된 자바 객체를 저장하고 검색할 수 있으며, 표준 디렉토리 운영을 수행할 수 있는 방법을 제시한다.

JDNI는 몇몇 특정한 디렉토리 서비스에 독립적이다. 자바 애플릿과 애플리케이션은 하나의 상용 API를 이용하여 다른 네이밍과 디렉토리 서비스에 액세스할 수 있다. JDNI용 애플릿과 애플리케이션은 LDAP, NDS, DNS, 네트웍 정보 서비스(Yellow Pages) 등 대부분의 디렉토리 서비스에 링크될 수 있다.

한편, 마이크로소프트도 자체 표준 자바 상에서 작업하고 있다. 지난 5월에는 마이크로소프트가 NC웨어 테그놀러지의 LDAP 자바 구현 기술을 라이센스한 바 있으며, 앞으로 마이크로소프트는 이 기술을 액티브 디렉토리 서비스 프로바이더 인터페이스에 통합할 계획이다.

자신의 방법을 찾아라

결론은 개발자라면 적어도 개념적으로라도 디렉토리 서비스를 이해해야 한다는 것이다. 그동안 많은 소규모 사업자의 웍그룹 애플리케이션에는 글로벌 네트웍 자원을 검토하고 액세스할 순차적인 메카니즘이 필요치 않았다. 하지만 이제는 그러한 소규모의 애플리케이션들이 설 자리를 잃어가고 있다.

현재 우리는 빌딩, 도시, 주, 국가에 이르기까지 모두 기업 수준의 애플리케이션을 구축하고 있다. 따라서 수백가지 네트웍에 액세스할 수 있는 자원을 이용하여 수천명의 사용자를 지원하고 있다. 물론 애플리케이션을 사용하고 구축하는 방법을 알아내는 것은 간단하지 않다. 하지만 모든 컴포넌트와 객체, 데이터는 글로벌 네트웍 상에서 효율적으로 바인딩되어 있다. 더욱이 사용자 구성, 단일 사인-온 메카니즘, 선행 보안 등의 네트웍을 쉽게 유지보수할 수 있는 메카니즘은 꼭 필요하다.

디렉토리 서비스 제품과 표준은 이러한 문제를 해결할 수 있다. 어떤 표준 세트를 가지고 전용 제품에서 사용할 때 디렉토리 서비스 제품을 확장하는 일은 혁명이라기 보다는 진화에 가깝다. 우리 모두 동의할 만한 표준을 사용하여 디렉토리 서비스 제품의 범용 제품을 보유하는데는 상당한 시간이 걸릴 것이다.

일반적으로 복잡한 분산 애플리케이션 개발시 기반구조부터 정비하는 순차적인 방법을 외면한 채 제품 개발에만 박차를 가하게 되는 경향이 있다. 그러나 이러한 성급한 실행은 매우 위험한 발상이다.

〈DBMS/USA〉