



(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。 Int. Cl.

G06Q 99/00 (2006.01)

G06F 17/50 (2006.01)

(11) 공개번호 10-2007-0076603

(43) 공개일자 2007년07월25일

(21) 출원번호 10-2006-0005639

(22) 출원일자 2006년01월19일

심사청구일자 2006년01월19일

(71) 출원인 충남대학교산학협력단
대전광역시 유성구 궁동 220번지 충남대학교

(72) 발명자 장경선
대전 유성구 전민동 청구나라아파트 104-901

(74) 대리인 김원준

전체 청구항 수 : 총 6 항

(54) 네트워크상의 카드 톨 자원 공유 방법

(57) 요약

본 발명은 네트워크상에서 디렉토리 서비스를 이용하여 카드 톨 자원을 여러 설계자와 효율적으로 공유하고 사용할 수 있게 해 주는 공유방법 및 시스템에 관한 것이다.

본 발명의 일 예에 따른 공유방법은 A) 서버 에이전트가 카드 톨 자원에 대한 정보/파일을 탑재한 카드 톨 자원 서버의 대행자를 디렉토리 서비스에 등록하고, 디렉토리 서비스로부터 서버 에이전트에 서비스 아이디가 부여되는 단계; B) 카드 톨 자원을 필요로 하는 클라이언트 에이전트가 디렉토리 서비스에 접속하여 대행자를 검색한 후 원하는 대행자를 선택하여 내려받는 단계; 카드 톨 자원 중 라이선스를 사용하는 경우, C) 클라이언트 에이전트가 내려받기한 대행자를 통해 라이선스 서버와 연결을 수립하여 라이선스를 요청하는 단계; D) 클라이언트 에이전트로부터 요청받은 라이선스가 있는 경우 라이선스 서버가 라이선스 정보 또는 파일을 클라이언트 에이전트에게 제공하고, 요청받은 라이선스가 없는 경우 라이선스 서버가 클라이언트 에이전트의 라이선스 요청을 거부하는 단계; 및 E) 클라이언트 에이전트가 라이선스 서버로부터 라이선스를 획득한 후, 클라이언트 에이전트가 획득된 라이선스를 클라이언트의 환경변수로 자동으로 설정하여 카드 톨의 사용을 가능하게 하는 단계; 및 카드 톨 자원 중 카드 톨을 사용하는 경우, F) 클라이언트 에이전트가 내려받기한 대행자를 통해 카드 톨 서버와 연결하여 설정정보를 요청하는 단계; G) 클라이언트 에이전트가 내려받은 설정정보를 이용하여 사용자 환경변수를 설정하는 단계; 및 H) 클라이언트 에이전트가 내려받은 대행자가 수행 가능한 대행자일 경우, 그것을 실행하는 단계로 구성된 것을 특징으로 한다.

또한, 본 발명에서 제안하는 일 실시예의 공유시스템은 A) 카드 톨 자원 서버와 통신하고, 서버의 대행자를 디렉토리 서비스에 등록하는 서버 에이전트; B) 카드 톨 자원 서버로부터 획득한 상기 카드 톨의 설정 정보 또는 라이선스 정보/파일을 환경변수로 자동 설정하는 클라이언트 에이전트; C) 카드 톨 설정, 또는 라이선스 설정 정보/파일이 클라이언트에게 제공되어 카드 톨의 사용이 가능하도록, 카드 톨 자원 서버의 정보 및 대행자가 등록·관리되는 디렉토리 서비스; D) 서버와 클라이언트를 연결하는 대행자 및 사용자 환경을 모두 포함하는 수행 가능한 대행자; 및 E) 카드 톨 자원 서버와 디렉토리 서비스 서버와 클라이언트 사이를 연결하는 통신망; 으로 구성된 것을 특징으로 한다.

대표도

도 6

특허청구의 범위

청구항 1.

카드 톨 자원 서버의 정보와 각 서버에 접근할 수 있는 대행자를 포함하는 서버 에이전트;

상기 카드 톨 자원 서버로부터 획득한 설정 정보(또는 라이선스 정보/파일)를 환경 변수로 자동 설정하는 클라이언트 에이전트;

상기 카드 톨 정보, 설정정보 또는 라이선스 정보/파일이 클라이언트에게 제공되어 카드 톨의 사용이 가능하도록, 상기 카드 톨 자원 서버의 정보와 각 대행자를 서버 에이전트를 통해 등록하고 관리하는 디렉토리 서비스;

상기 카드 톨 자원 서버에서 클라이언트와 카드 톨 자원 서버 사이의 통신을 위해 서버가 서버 에이전트를 통해 제공하는 대행자 및 카드 톨 공유의 경우, 상기 디렉토리 서비스에 등록된 대행자 중, 작은 클라이언트 프로그램을 특별한 설정 없이 바로 내려받은 후 바로 사용할 수 있는 수행 가능한 대행자;

상기 카드 톨 자원 서버와 디렉토리 서비스, 클라이언트 사이를 연결하는 통신망; 으로 구성된 것을 특징으로 하는 네트워크상에서 디렉토리 서비스를 이용한 카드 톨 자원 공유 시스템.

청구항 2.

제1항에 있어서,

상기 서버 에이전트는

카드 톨 자원 정보와 대행자를 상기 제1항의 디렉토리 서비스에 등록을 요청하여 서비스 아이디를 부여받는 것을 특징으로 하는 네트워크상에서 디렉토리 서비스를 이용한 카드 톨 자원 공유 시스템.

청구항 3.

제1항에 있어서,

상기 클라이언트 에이전트는

상기 제1항의 디렉토리 서비스에 연결을 성립하여 필요한 카드 톨 자원에 대한 정보를 조회하고, 필요한 대행자를 내려받아, 대행자를 이용하여 카드 톨 자원 서버에 연결을 수립하여 필요한 정보를 요청하여 획득하고, 획득한 정보를 클라이언트 에이전트가 환경변수로 자동으로 설정하는 것을 특징으로 하는 네트워크상에서 디렉토리 서비스를 이용한 카드 톨 자원 공유 시스템.

청구항 4.

제1항에 있어서,

상기 디렉토리 서비스는

상기 제1항의 서버 에이전트에 의해 캐드 툴 자원 서버에 대한 정보와 대행자를 등록하고, 클라이언트에게 캐드 툴 자원 정보와 대행자를 제공할 수 있는 것을 특징으로 하는 네트워크상에서 디렉토리 서비스를 이용한 캐드 툴 자원 공유 시스템.

청구항 5.

제1항에 있어서,

대행자는 서버 에이전트에 요청에 의해 디렉토리 서비스에 등록되며, 클라이언트 에이전트와 캐드 툴 자원 서버의 통신을 위한 인터페이스를 제공하고, 통신을 성립시켜 필요한 정보를 제공하고, 수행 가능한 대행자는 통신을 위한 인터페이스뿐만 아니라, GUI 형태의 사용자 인터페이스를 포함하는 것을 특징으로 하는 네트워크상에서 디렉토리 서비스를 이용한 캐드 툴 자원 공유 시스템.

청구항 6.

제1항에 있어서,

상기 공유 시스템은 캐드 툴 자원의 대행자, 정보/파일이 탑재된 것을 상기 제1항의 서버 에이전트를 이용하여 디렉토리 서비스 등록하고, 상기 제1항의 디렉토리 서비스에 자동으로 접속할 수 있는 클라이언트 에이전트가 상기 디렉토리 서비스를 조회하여 필요한 대행자를 다운로드 받고, 이것을 이용하여 서버와 1:1 연결을 수립함으로써 상기 클라이언트가 필요로 하는 캐드 툴 또는 라이선스 사용이 가능하게 하는 것을 특징으로 하는 네트워크상에서 디렉토리 서비스를 이용한 캐드 툴 자원 공유 시스템.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 네트워크상에서 디렉토리 서비스를 이용하여 캐드 툴 자원을 효율적으로 공유하고 사용할 수 있게 해 주는 캐드 툴 자원 공유 방법 및 시스템에 관한 것이다.

초고밀도 집적회로(VLSI) 디자인의 발전과 더불어 캐드(CAD) 툴의 사용이 기하급수적으로 증가하고 있다.

그러나 캐드 툴(소프트웨어, 프로그램)은 매우 고가이기 때문에 대부분의 설계그룹은 캐드 툴을 사용함에 있어 한정된 개수의 라이선스를 이용해 여러 명의 설계자가 공유하여 사용한다. 이와 같은 상황에서는 캐드 툴 자원에 대한 효율적인 공유 방법이 필요하다. 캐드 툴 자원은 크게 캐드 툴 자체와 캐드 툴 라이선스로 나뉜다.

첨부된 도 1은 전통적인 캐드 툴 자원 공유방식을 나타낸 것으로서 해당 설계그룹에 캐드 툴 자원을 관리하는 사람이 있어 직접 관리한다.

그러나 도 1과 같은 구조를 이용하여 캐드 툴 자원 중 캐드 툴 자체를 자동으로 공유하는 방법은 현재 존재하지 않는다.

캐드 툴 자원 중 라이선스를 관리하기 위한 대표적인 소프트웨어로는 FLEXlm이 있다.

FLEXlm은 라이선스를 관리하기 위한 클라이언트/서버 모델 방식으로, 아래의 두 가지 방식 중 어느 한 방식으로 캐드 툴 라이선스를 관리한다.

그 중 첫 번째 방식은 라이선스 파일을 사용자에게 배포하고, 사용자는 그 파일을 자신의 라이선스 환경변수로 수동 설정하여 사용하는 방식이다.

이에 반해, 두 번째 방식은 라이선스 서버를 만들고, 사용자에게는 라이선스 서버에 접근할 수 있는 포트 번호와 서버 이름을 배포하는 것이다. 사용자는 이 정보를 이메일 형태로 만들어 자신의 라이선스 환경변수로 수동설정하게 되는데, 현재 이 방법이 가장 널리 사용되는 라이선스 관리방식이다. (참고문헌: FLEXlm End User guide Version 9.2, Macrovision, 2003)

현재 특정 설계그룹에서 캐드 툴 자원을 공유하기 위한 자동화된 툴은 없으며, 일반적으로 사용자와 관리자 사이의 수동적인 방법에 의한 정보교환 후에 정상적인 사용이 가능하다.

캐드 툴 자원 중 라이선스의 경우, 캐드 툴 사용자들에게 제공된 라이선스의 관리를 쉽게 하기 위해 라이선스 관리 툴을 사용한다. 하지만, 종래의 라이선스 관리 툴들은 라이선스를 얻기 위해 사용자가 라이선스 파일 및 서버가 어디에 위치하고 있는지, 라이선스 서버에 접근하기 위해서는 어떤 포트로 접근해야 하는지 및 전체 라이선스의 개수에서 현재 사용가능한 라이선스가 있는지 등에 대한 정보를 미리 알고 있어야만 한다.

왜냐하면, 종래의 캐드 툴 라이선스 관리시스템은 도 2와 같은 클라이언트/서버 모델에 기반을 둔 시스템이므로, 사용자가 사전에 필요한 정보를 알고 있고, 설정을 마친 후 사용할 수 있기 때문이다.

결국, 사용자는 라이선스의 사용 가능 여부를 사전에 알기 힘들고, 만일 라이선스가 사용 가능하다는 것을 알게 되더라도 해당 라이선스에 대한 설정을 로컬 머신에 직접 해 주어야 하는 불편함을 겪게 된다.

이러한 불편함은 하나의 캐드 툴에 한 명의 사용자만이 사용 가능한 경우에는 문제가 되지 않지만, 도 1 과 같이 하나의 캐드 툴에 대해 여러 사용자가 사용 가능하도록 해주는 경우에는 라이선스 서버를 관리하는 관리자에게 요청이 집중됨으로써 라이선스 서버 관리자가 똑같은 요청에 대한 답변을 계속 해줘야 한다는 번거로움이 발생한다. 예를 들어, 라이선스를 사용할 수 있는 사용자들 또는 단체에서 캐드 툴을 사용하고자 하는 경우, 각 사용자들은 캐드 툴을 설치한 후 라이선스를 얻기 위해 라이선스의 파일 및 서버의 정보를 각자 확인해야 한다. 이런 과정에서, 사용자들은 라이선스 서비스 데몬이 실행되어 있는 라이선스 서버(컴퓨터)의 포트 번호와 이름을 관리자에게 요청하게 되고, 라이선스 서버 관리자는 매번 정보를 확인하여 사용 가능한 라이선스가 얼마나 남았는지를 사용자에게 알려주어야 한다. 또한, 사용자 각각은 얻은 정보를 바탕으로 자신의 컴퓨터에 환경변수로 수동설정을 해 주어야 한다.

현재 많이 사용되는 라이선스 관리 툴인 FLEXlm은 매크로비전(Macrovision)사에서 개발한 프로그램으로, 네트워크상에서 사용 가능한 소프트웨어의 라이선스를 사용자에게 허가하는 기능이 있다.

상기 FLEXlm에는 소프트웨어의 라이선스 처리방식에 따라 FLEXlm Unserved 라이선스 모델과 FLEXlm Served 라이선스 모델의 두 가지 처리모델이 있다.

그 중, 도 3의 FLEXlm Unserved 라이선스 모델은 과거에 많이 사용되었으나 요즘에는 잘 사용되지 않는 방식이다. FLEXlm Unserved License Model은 라이선스 서버가 특별한 역할을 하지 않으며 사용자의 컴퓨터 내에 라이선스 파일을 저장하여 캐드 툴을 사용하는 방식이다.

반면, 도 4의 FLEXlm Served 라이선스 모델은 최근에 거의 모든 캐드 툴에서 사용하는 방식으로, 라이선스 서버를 통해 라이선스를 얻는 방식이다. 이 방식은 라이선스 서버가 같은 컴퓨터에 존재할 수도 있지만 일반적으로는 같은 C 클래스 내의 네트워크상에 존재하면서 라이선스를 다른 컴퓨터에 제공한다.

FLEXlm Served 라이선스 모델은 도 4와 같이 FLEXlm 라이선스 응용프로그램과 라이선스 서버를 구성하는 라이선스 매니저 데몬 및 벤더 데몬, 그리고 라이선스 파일 등 4개의 컴포넌트로 이루어진다.

라이선스 매니저 데몬은 벤더 데몬과 연결되어 있고, 라이선스를 원하는 응용프로그램과의 초기 연결을 처리한다.

벤더 데몬은 FLEXlm을 통해 라이선스를 제공받아야 하는 응용프로그램들을 만든 벤더별로 하나씩 존재하며, 응용프로그램에 라이선스를 제공해주는 역할을 한다. 또한, 벤더 데몬은 현재 사용 중인 라이선스의 수와 라이선스를 누가 가지고 있는가를 검사한다. 즉, 응용프로그램은 라이선스 매니저 데몬을 통해 벤더 데몬과 연결하고, 벤더 데몬으로부터 라이선스를 획득하는 것이다.

라이선스 파일은 응용프로그램을 만든 벤더가 제공해 주는 것으로, 파일 안에는 라이선스 서버에 대한 정보와 벤더 데몬에 대한 정보, 그리고 적어도 한 개 이상의 라이선스를 위한 데이터(FEATURE 또는 INCREMENT)를 포함하고 있다.

응용프로그램은 라이선스 서버와 연결해 주는 FLEXlm 클라이언트 라이브러리와 연결되고, 벤더 데몬과의 통신을 통해 라이선스를 요청하는 주체이다.

FLEXlm은 도 5와 같은 컴포넌트를 사용해 라이선스의 요청/수락/거부를 처리하는데, 구체적인 라이선스 처리 절차는 다음과 같다.

- ① 클라이언트 응용프로그램 내의 라이선스 모듈이 호스트 이름과 라이선스 매니저 데몬(lmgrd)의 포트 번호를 포함하는 라이선스 파일(예컨대 license.dat)을 찾는다.
- ② 이어, 클라이언트는 라이선스 매니저 데몬과 연결을 수립하고, 라이선스 매니저 데몬에게 어떤 벤더 데몬과의 연결을 원하는지 알린다.
- ③ 그러면, 라이선스 매니저 데몬은 특정 머신의 포트와 벤더 데몬이 잘 연결되는지 결정하고, 그 정보를 클라이언트에게 보낸다.
- ④ 동 정보를 이용해 클라이언트는 벤더 데몬과의 연결을 수립하고, 라이선스를 위한 요청을 한다.
- ⑤ 이어, 벤더 데몬은 사용 가능한 라이선스가 있는지를 검사하여 승인 또는 거부한다.
- ⑥ 끝으로, 응용프로그램 내의 라이선스 모듈에서 최종적으로 반환된 값을 통해 승인 또는 거부를 확인한다.

FLEXlm은 이러한 6단계의 라이선스 요청절차를 통해 라이선스 서버에서 사용자에게 라이선스를 제공하게 된다.

그러나 FLEXlm을 이용하는 종래 라이선스 관리방식에 따르면, ②~⑥번 부분은 FLEXlm 내부에서 진행되는 과정이어서 캐드 툴 사용자나 서버 관리자가 관여할 수 없다 하더라도 ①번 과정 부분에서 다음과 같은 문제점이 발생할 수 있다.

첫째, 사용자는 라이선스 서버가 설치되어 있는 컴퓨터에 대한 정보를 미리 알고 있어야 한다는 점이다. 정보를 미리 알고 있지 못한 경우, 사용자는 라이선스 파일이 위치해 있는 라이선스 서버의 포트 번호와 서버 이름을 알아올 수가 없을 것이다. 따라서 사용자는 캐드 툴을 사용하기 전, 라이선스 서버의 정보를 알고 있어야 하고 그 정보를 유지해야 한다.

둘째, 라이선스 서버 관리자가 라이선스 서버에 대한 정보를 사용자에게 알려줘야 한다는 점이다. 이는 한 장소에서 여러 사람이 동일한 캐드 툴을 사용하는 경우에는 큰 문제가 되지 않지만, 한 장소에 여러 개의 캐드 툴이 설치되어 있거나 다른 장소에서 동일한 캐드 툴을 사용하고자 할 경우에는 라이선스 서버 관리자가 모든 사람들에게 라이선스 서버 정보를 알려줘야 하는 문제가 생길 수 있다.

셋째, 사용자가 라이선스 서버로부터 받은 라이선스 파일 또는 포트 번호 및 서버 이름을 환경변수에 LM_LICENSE_FILE 또는 벤더 명_LICENSE_FILE이라는 이름으로 수동 설정해 주어야 한다는 점이다.

일반적으로 라이선스 서버와 캐드 툴이 설치되어 있는 컴퓨터는 같은 네트워크상의 다른 머신인 경우가 대부분이다. 그러므로 사용자는 사용하려는 캐드 툴이 설치된 컴퓨터의 환경변수를 수정해야 한다.

넷째, 사용자가 여러 가지 캐드 툴을 사용할 때, 같은 작업을 반복해야 한다는 점이다.

다섯째, 위의 전 과정은 캐드 툴에 대한 사용방법의 특성상 라이선스가 매년 1회 이상 변경되므로 앞의 과정이 반복된다는 점이다.

결론적으로, FLEXlm은 여러 사용자가 캐드 라이선스를 효율적으로 공유하기 위한 방법이라기보다, 캐드 툴을 공급하는 벤더들이 허용된 라이선스 수 이상 캐드 툴을 사용하지 못하게 하려는 관리 수단이라고 할 수 있다.

발명이 이루고자 하는 기술적 과제

본 발명은 종래 관리자에 의해 수행되던 캐드 툴 자원 공유 방법의 한계성을 극복하고, 관리자와 사용자 사이의 통신방법 및 사용자가 자신의 컴퓨터(클라이언트)에 올바른 캐드 툴 자원의 사용 환경을 설정하는 과정을 일부 자동화하여 캐드 툴의 이용률을 높이며, 사용자 및 관리자에게 편의성을 제공하는데 목적이 있다.

즉, 본 발명은 종래 수동으로 행해지던 사용자와 관리자 사이의 정보교환 부분을 자동화하고, 서버를 이용한 클라이언트/서버 방식의 종래 관리 시스템 대신 디렉토리 서비스를 이용함으로써 사용자와 관리자의 편리성을 추구하며, 캐드 툴 자원 중 캐드 툴의 원활한 사용을 위하여 환경설정이 포함된 간단한 수행 가능한 대행자(Executable Proxy)를 내려받아 아무런 설정 없이 바로 사용할 수 있도록 하는데 목적이 있는 것이다.

발명의 구성

상기 과제를 해결하기 위한 본 발명의 일 예에 따른 네트워크상에서 디렉토리 서비스를 이용한 캐드 툴 자원 공유 시스템은 A) 캐드 툴 자원 서버와 통신하고, 그 대행자를 디렉토리 서비스에 등록하는 캐드 툴 자원 서버 에이전트; B) 상기 캐드 툴 자원 서버로부터 획득한 캐드 툴 자원의 정보, 라이선스 정보/파일을 환경변수로 자동 설정하는 클라이언트 에이전트; C) 캐드 툴 자원 정보/파일이 클라이언트에게 제공되어 캐드 툴의 사용이 가능하도록, 상기 캐드 툴 자원 서버에 대한 정보와 대행자가 등록·관리되는 디렉토리 서비스; D) 서버와 클라이언트를 연결하기 위해 같은 인터페이스를 포함하는 대행자 및 캐드 툴 자원 중 캐드 툴의 경우 인터페이스뿐만 아니라, 사용자 환경을 모두 포함하는 수행 가능한 대행자; 및 E) 상기 캐드 툴 자원 서버와 디렉토리 서비스와 클라이언트 사이를 연결하는 통신망; 으로 구성될 수 있다. (청구항 1)

더욱 바람직하게는, 상기 서버 에이전트와 클라이언트 에이전트가 서로 같은 대행자의 인터페이스를 갖도록 작성하고, 대행자를 이용하여 캐드 툴 별로 1:1 통신·동작하도록 하는 것이다. (청구항 6)

또한, 본 발명의 일 예에 따른 네트워크상에서 디렉토리 서비스를 이용한 캐드 툴 자원 공유 방법은 A) 캐드 툴 자원 서버에 대한 정보/파일을 탑재한 서버 에이전트가 디렉토리 서비스에 등록을 요청한 후 등록하고, 디렉토리 서비스부터 서버 에이전트에 서비스 아이디가 부여되는 단계; B) 캐드 툴에 대한 설정 정보 또는 라이선스를 필요로 하는 클라이언트가 클라이언트 에이전트를 통해 디렉토리 서비스에 자동으로 접속하여 대행자를 검색한 후 원하는 대행자를 선택하여 내려받는 단계; 캐드 툴 자원 중 라이선스를 사용하는 경우, C) 클라이언트 에이전트가 내려받기 한 대행자를 통해 라이선스 서버와 연결을 수립하여 라이선스를 요청하는 단계; D) 클라이언트로부터 요청받은 라이선스가 있는 경우 라이선스 서버가 라이선스 정보 또는 파일을 클라이언트에 제공하고, 요청받은 라이선스가 없는 경우 라이선스 서버가 클라이언트의 라이선스 요청을 거부하는 단계; E) 클라이언트 에이전트가 라이선스 서버로부터 라이선스를 획득한 때, 클라이언트 에이전트가 상기 획득된 라이선스를 클라이언트의 환경변수에 자동으로 설정하여 캐드 툴의 사용을 가능하게 하는 단계; 및 캐드 툴 자원 중 캐드 툴을 사용하는 경우, F) 클라이언트 에이전트가 내려받기 한 대행자를 통해 캐드 툴 서버와 연결하여 설정정보를 요청하는 단계; G) 클라이언트 에이전트가 내려받은 설정정보를 이용하여 사용자 환경변수를 설정하는 단계; 및 H) 클라이언트가 내려받은 대행자가 수행 가능한 대행자일 경우, 그것을 실행하는 단계로 구성될 수 있다. (청구항 1)

이하 첨부된 도 6을 참조하여, 본 발명의 일 실시예에 따른 네트워크상에서 디렉토리 서비스를 이용한 캐드 툴 자원 공유 방법 및 시스템에 대해 설명한다.

1. 캐드 툴 자원 공유 시스템

캐드 툴 자원을 더욱 효율적으로 공유하여 사용할 수 있게 해 주는 본 발명의 공유 시스템은 캐드 툴 자원 서버 측의 서버 에이전트 부분과 클라이언트를 위한 클라이언트 에이전트 부분, 그리고 디렉토리 서비스와 캐드 툴 자원 서버 서버와 클라이언트를 연결하는 통신망으로 구성된다.

공유시스템의 서비스 부분인 서버 에이전트는 종래의 서버 관리자를 위한 프로그램으로, 관리자를 대행하여 캐드 툴 자원이 효율적으로 공유될 수 있도록 관리한다.

클라이언트를 위한 클라이언트 에이전트는 캐드 툴 사용자가 설정 정보 또는 라이선스를 얻기 위해 사용하는 프로그램으로, 자동화된 방법으로 설정 정보 및 라이선스를 얻을 수 있게 도와준다.

디렉토리 서비스는 서버 에이전트의 요청에 의해 캐드 툴 서버의 정보 또는 라이선스 서버의 정보, 그리고 각 서버의 대행자가 등록되어 관리된다.

세부적인 구성은 아래와 같다.

1) 전체 구성

도 6은 본 발명의 일 예에 따른 공유 시스템 전체 구성을 나타낸 것이다.

도 6에서 보듯이, 서버 측에는 캐드 툴 자원 서버와 캐드 툴 자원에 대한 정보를 관리하기 위한 서버 에이전트가 있다. 클라이언트 측에는 사용자의 캐드 툴 자원의 설정을 자동으로 해주는 클라이언트 에이전트가 있다. 마지막으로 서버 에이전트가 서버의 정보 및 대행자를 등록시키고, 그것을 관리하는 디렉토리 서비스가 있다.

디렉토리 서비스는 서버 에이전트의 대행자와 캐드 툴 자원의 정보를 유지하고, 클라이언트 에이전트의 요청이 있을 경우, 해당 캐드 툴 자원의 정보와 대행자를 제공해 준다.

서버 에이전트는 캐드 툴 자원 서버부터 정보를 요청하여 얻은 후, 클라이언트 에이전트에게 제공할 수 있게 된다.

또한, 클라이언트 에이전트는 서버 에이전트에게 캐드 툴 또는 라이선스 설정 정보를 요청하고, 획득한 정보를 환경변수로 자동으로 설정할 수 있다.

도 6에서 기술된 캐드 툴 자원의 자동 공유를 위한 처리과정은 다음과 같은 순서로 진행된다.

1. 캐드 툴 자원에 대한 정보를 요청한다.(①)
2. 캐드 툴 자원 대한 정보를 획득한다.(②)
3. 디렉토리 서비스를 찾아 대행자 및 서버 정보를 등록한다.(③)
4. 서비스 아이디를 받는다.(④)
5. 클라이언트 에이전트가 디렉토리 서비스를 찾아 원하는 정보 및 대행자를 요청한다.(⑤)
6. 원하는 대행자를 다운로드한다.(⑥)
7. 캐드 툴 자원에 대한 정보를 입력하고, 대행자를 통해 서비스와 연결을 수립한 뒤 캐드 툴 설정 정보 또는 라이선스를 요청한다.(⑦, ⑧)
8. 서버 에이전트는 요청받은 캐드 툴 자원 정보가 있는 경우 정보를 제공하고, 없는 경우 요청을 거부한다.(⑨)
9. 획득한 캐드 툴 자원의 정보를 환경변수에 자동으로 설정하고, 캐드 툴을 사용할 수 있게 된다.(⑩, ⑪)

첨부된 도 7은 이러한 처리과정을 시퀀스 다이어그램으로 표현한 것이다.

도 7에서 각 시퀀스는 다음과 같다.

1. 서버 에이전트가 캐드 툴 자원의 정보와 대행자를 디렉토리 서비스에 등록하기 위해, 디렉토리 서비스를 찾는 단계
2. 서버 에이전트가 디렉토리 서비스를 찾아 등록하는 단계

3. 서버 에이전트가 시스템의 신뢰성을 높이기 위해 임대를 시작하는 단계
4. 서버 에이전트가 시스템의 신뢰성을 높이기 위해 주기적으로 임대를 갱신하는 단계
5. 클라이언트 에이전트가 필요한 정보를 획득하기 위해 디렉토리 서비스를 자동으로 찾는 단계
6. 클라이언트 에이전트가 디렉토리 서비스를 찾은 후, 서비스를 검색하는 단계
7. 클라이언트 에이전트가 필요한 정보를 검색하여 찾아, 대행자를 내려받는 단계
8. 클라이언트 에이전트가 대행자를 사용할 수 있도록 설정하는 단계
9. 클라이언트 에이전트가 대행자를 통해 필요한 정보를 서버에 요청하는 단계
10. 서버가 클라이언트 에이전트의 요청에 응답하는 단계
11. 서버 에이전트가 임대를 갱신하던 중 에러가 발생한 단계
12. 에러가 발생하여, 이 정보를 클라이언트 에이전트에게 알리는 단계
13. 에러가 발생하여, 이 정보를 서버 에이전트에게 알리는 단계
14. 서버 에이전트가 캐드 툴 자원 서버의 정보를 요청하는 단계
15. 캐드 툴 자원 서버가 서버 에이전트가 요청한 정보를 제공하는 단계
16. 캐드 툴 자원 서버가 서버 에이전트의 요청에 대해 문제가 발생하는 단계
17. 문제에 대해 서버 에이전트에게 전달하는 단계
18. 서버 에이전트가 문제가 있는 해당 서버를 디렉토리 서비스로부터 제거하는 단계
19. 디렉토리 서비스의 현재 상태를 클라이언트 에이전트에게 통보하는 단계

2) 서버 에이전트

서버 에이전트는 종래 캐드 툴 또는 라이선스 서버의 관리자 역할을 하게 된다.

종래에는 사용자들에게 캐드 툴을 사용할 수 있도록 해 주기 위해, 캐드 툴 또는 라이선스 서버 관리자가 직접 사용자들에게 캐드 툴 자원 설정정보 및 설치 방법을 온·오프라인을 통해 알려줘야만 했다. 그러나 본 실시예의 서버 에이전트를 사용할 경우, 캐드 툴 자원 서버 관리자가 해야 할 일이 극히 적어지게 된다. 즉, 서버 에이전트는 종래 캐드 툴 자원 서버 관리자가 여러 사용자에게 설정 정보 및 라이선스 정보를 제공하던 작업을 대행해 주는 것이다. 즉, 서버 에이전트는 다음과 같은 일을 수행한다.

- (1) 서버 에이전트는 서버의 정보와 캐드 툴 자원 서버의 대행자를 등록하기 위해 디렉토리 서비스를 찾는다.
- (2) 캐드 툴 자원 서버는 자신의 현 상태를 특정 정보형태로 지속적으로 공표하고, 서버 에이전트는 공표된 정보를 획득하거나, 서버에 정보를 요청할 수 있다.
- (3) 서버 에이전트는 디렉토리 서비스에 자신을 등록한 후, 신뢰성을 높이기 위해 특정 시간 간격으로 임대를 성립시킨다. 즉, 임대가 성립되지 않는 경우는 문제가 발생한 경우이다.

(4) 서버 에이전트는 사용자에게 캐드 툴 자원을 사용할 수 있도록 대행자(또는 캐드 툴의 경우 수행가능한 대행자)를 제공한다.

(5) 캐드 툴 자원 서버를 후위 프로세서로 실행한다.

서버 에이전트는 다음과 같은 장점을 가진다.

첫째, 새로운 캐드 툴 자원 서버가 등록되었을 때 자동으로 그 내용을 클라이언트에게 통지함으로써, 관리자가 일일이 사용자에게 통보하는 것에 비해 편리함을 가지게 된다.

둘째, 서버 에이전트는 종래 방법 비해 높은 신뢰성을 가진다. 이것은 서버가 지속적으로 임대를 갱신함으로써 가능해진다.

셋째, 서버 에이전트는 클라이언트에게 자신의 서비스를 이용할 수 있게 하는 대행자를 제공한다. 클라이언트는 먼저 자신이 필요로 하는 정보를 디렉토리 서비스로부터 찾아 일치하는 것이 있으면, 해당 정보를 요청하기 위한 대행자를 내려받는다.

따라서 클라이언트는 이 대행자를 통해 서비스에서 제공하는 모든 기능을 이용할 수 있다. 즉, 캐드 툴 자원을 사용하기 위한 환경설정 및 설치정보 등을 얻어 올 수 있는 것이다.

또한, 캐드 툴의 공유에서 썬 클라이언트 프로그램인 수행 가능한 대행자의 경우, 해당 정보 및 환경뿐 아니라 툴 자체를 자동으로 내려받아 사용할 수 있다.

마지막으로, 서버 에이전트는 캐드 툴 자원을 후위 프로세서의 개념으로 실행함으로써 로컬 컴퓨터의 자원을 최소화하여 수행할 수 있다.

3) 클라이언트 에이전트

클라이언트 에이전트는 다음의 세 가지 기능으로 편의를 제공한다.

첫째, 클라이언트 에이전트는 설계자 자신이 속한 그룹에서 어떠한 캐드 툴 또는 라이선스 자원이 존재하고, 사용 가능한 위치는 어디인지 및 기타 정보들을 보여준다. 클라이언트 에이전트는 디렉토리 서비스를 찾은 후 서비스를 조회를 요청할 수 있다. 그 결과로서 사용 가능한 캐드 툴 자원의 목록을 사용자에게 보여주고, 캐드 툴 자원의 정보가 추가/삭제될 때마다 동적으로 그 현황을 알려준다.

둘째, 설계자를 대신하여 필요한 환경변수를 자동으로 설정해 준다.

이 경우, 이미 캐드 툴이 설치되어 있지만 설계자가 해당 캐드 툴을 사용하기 위한 환경설정을 모를 경우 설계자는 캐드 툴 자원을 올바르게 사용할 수 없게 되는데 클라이언트 에이전트는 자동으로 환경변수들을 설정하여 설계자가 캐드 툴 자원을 사용하는데 문제가 없도록 한다.

셋째, 캐드 툴 공유에서 등록된 캐드 툴 자원이 수행 가능한 대행자를 포함하는 경우 이를 자동으로 내려받아 설계자가 로컬환경에 대한 설정 없이 바로 사용할 수 있게 된다.

4) 디렉토리 서비스

디렉토리 서비스는 캐드 툴 자원 서버와 사용자 간의 중재자 역할을 한다. 클라이언트 에이전트가 캐드 툴 자원에 대한 정보를 받기 위해서는 반드시 디렉토리 서비스를 통해야만 한다.

디렉토리 서비스는 전통적인 캐드 툴 자원 관리 방법에서 설계자 중 한 사람인 관리자의 역할을 대신할 수 있다.

결과적으로, 관리자가 직접 해주던 일의 일부분이 자동화한 것이다.

만일 자원 관리자가 없는 환경에서 서비스와 클라이언트만 있다면 중재자가 없기 때문에, 현재 우리가 사용하고 있는 일반적인 클라이언트/서버 기반의 소프트웨어와 마찬가지로 사용자가 직접 필요한 환경을 설정해주어야 하며 서비스가 사용 가능한지도 매번 검사해야 한다.

서버 에이전트에 의해 캐드 툴 자원 서버의 정보 및 대행자가 등록된 디렉토리 서비스는 특정 작업그룹 또는 연구그룹에서 사용하는 캐드 툴 자원들을 일괄적으로 관리할 수 있으며, 클라이언트에게 항상 현재 사용 가능한 캐드 툴 또는 라이선스 정보를 자동으로 알려준다.

또한, 추가/삭제/에러가 발생한 자원에 대해 동적으로 이벤트를 발생시켜 클라이언트뿐 아니라 관리자에게도 알려줌으로써, 클라이언트에게는 최신의 정보를 유지·제공하고 관리자에게는 에러에 대한 발견시간이 짧아져 복구에 걸리는 시간을 단축할 수 있게 된다.

4) 대행자 및 썬 클라이언트를 위한 수행 가능한 대행자

대행자는 클라이언트 에이전트와 캐드 툴 자원 서버를 연결하기 위해 서버 측에서 서버 에이전트 및 디렉토리 서비스를 통해 제공하는 메소드의 인터페이스를 포함하는 개체이다. 클라이언트는 이 개체를 통해 서버에게 필요한 정보를 요청하여 획득한다.

캐드 툴 공유의 경우, 대부분의 클라이언트용 캐드 툴 자원 역시 사용자 또는 관리자가 직접 설치하고 원활한 사용을 위해 환경설정을 해야 한다.

그러나 클라이언트 쪽에서 실행되는 프로그램이 복잡하지 않고 크기가 작다면, 썬 클라이언트를 위한 수행 가능한 대행자를 이용하여 필요할 때에만 내려받아 사용할 수 있도록 서비스를 제공해줄 수 있다.

비슷한 개념으로, 일반적인 캐드 툴 자원공유방법에서 인터넷 캐드가 존재한다.

이러한 인터넷 캐드는 서버/클라이언트의 방식을 사용한다. 종래 서버/클라이언트 방법의 경우에는 중요한 캐드 툴 자원은 서버에 설치가 되며, 이 캐드 툴을 이용하기 위한 클라이언트 프로그램은 사용자가 자신의 로컬 컴퓨터에 설치한 후 사용을 위해 환경설정을 한다.

이 경우, 서버의 설정사항이 바뀔 때마다 매번 사용자 환경을 다시 설정해야 하며, 클라이언트를 위한 CAD 툴 자원도 재설치해야 한다.

본 실시예에서는 이와 같은 문제점을 해결하기 위해, 서버 에이전트가 특정 캐드 툴 자원에 맞게 미리 구현해 놓은 수행 가능한 대행자를 디렉토리 서비스에 등록하였다.

따라서 서비스의 사용자는 클라이언트 에이전트를 통해 이를 검색하고, 필요한 것을 내려받아 환경 설정 없이 바로 실행할 수 있다. 여기서, 상기 수행 가능한 대행자는 그래픽 사용자 인터페이스(GUI) 및 서버와 통신하기 위한 메소드의 인터페이스를 모두 포함하는 개체이므로 이것이 가능하다.

발명의 효과

이상에서 살펴본 바와 같이, 본 발명의 일 예에서는 네트워크상의 특정 그룹에서 캐드 툴 자원을 효율적으로 공유하고 사용할 수 있는 방법을 제시하였다. 예컨대, 사용자와 관리자 사이의 통신 일부를 자동화했을 뿐 아니라 원활한 캐드 툴 사용을 위한 환경설정 및 라이선스 설정이 대행자에 의해 자동적으로 이루어지도록 하였고, 캐드 툴 공유에서 작은 캐드 툴 프로그램은 바로 내려받기를 하여 특별한 설정 없이 사용할 수 있도록 한 것이다.

따라서 본 발명으로 인해 캐드 툴의 이용률은 크게 향상될 것이다.

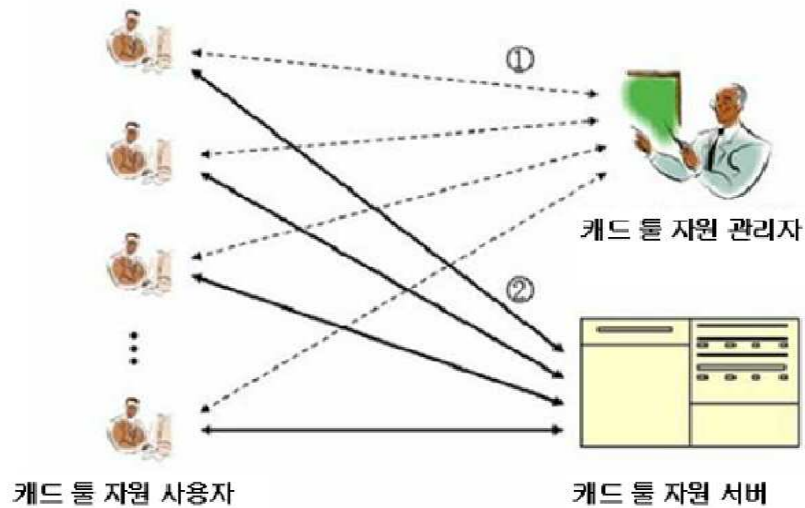
또한, FLEXlm을 사용하여 라이선스 관리를 할 경우에 발생할 수 있는 사용자와 관리자 사이의 통신 부분 및 사용자 측의 환경설정 과정을 자동화함으로써 사용자 및 관리자의 설정시간/비용이 감소하고, 캐드 툴 라이선스 관리의 편의성이 증대되어 캐드 툴의 이용률을 극대화시킬 수 있을 것이다.

도면의 간단한 설명

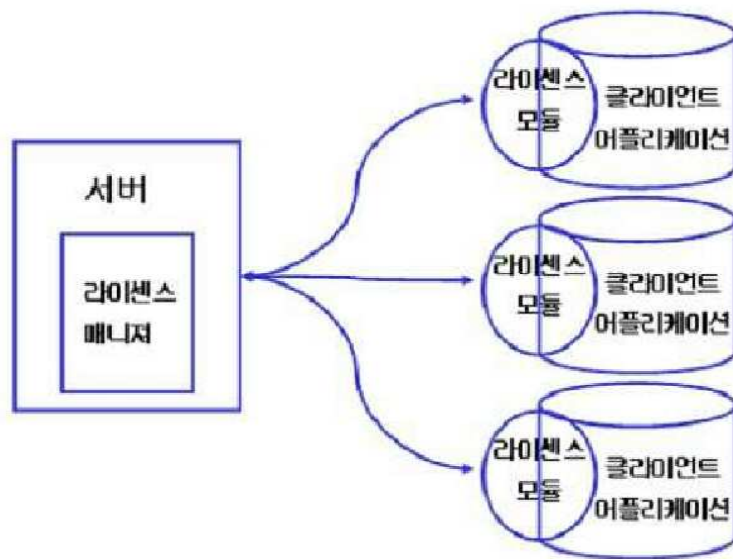
- 도 1은 종래의 캐드(CAD) 툴 자원 공유 방식을 나타낸 모형도.
 도 2는 종래의 클라이언트/서버 기반의 라이선스 공유 모델을 나타낸 모형도.
 도 3은 종래의 FLEXlm Unserved 라이선스 관리모델을 나타낸 모형도.
 도 4는 종래의 FLEXlm Served 라이선스 관리모델을 나타낸 모형도.
 도 5는 종래 FLEXlm을 통한 라이선스 요청 절차.
 도 6은 본 발명의 일 예에 따른 시스템의 전체 구성도.
 도 7은 도 6에 나타난 시스템의 시퀀스 다이어그램.

도면

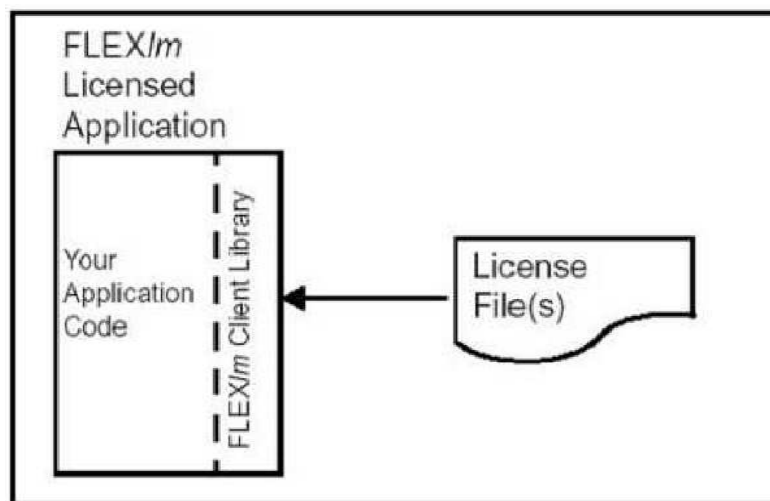
도면1



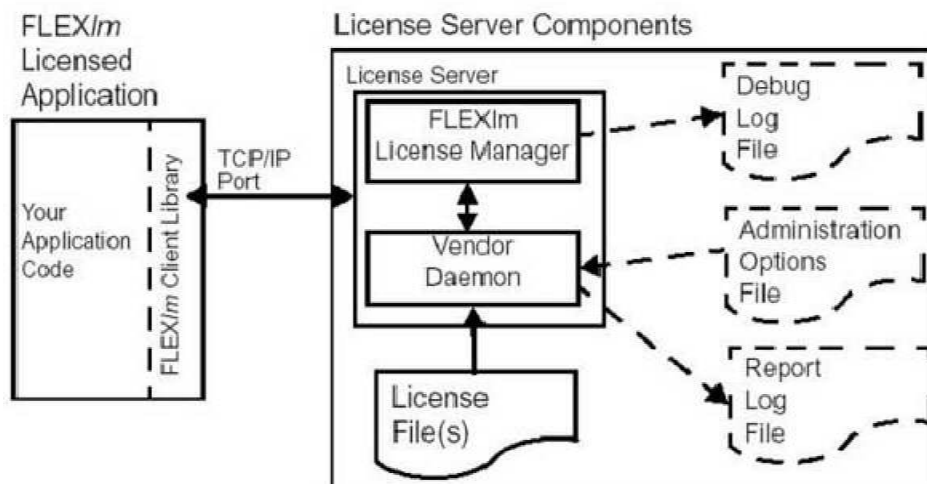
도면2



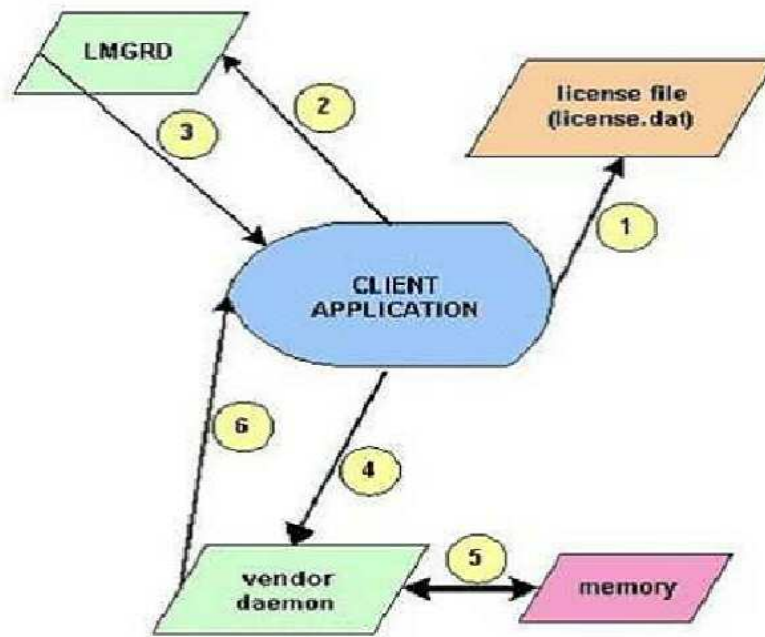
도면3



도면4



도면5



도면6

