

디렉토리 서비스에 기반한 CAD 툴 및 라이선스 자원의 자동화된 공유 방식

정 성 현^{*} · 임 인 성^{**} · 장 경 선^{***}

요 약

디자인 워킹 그룹에서 설계자가 CAD 툴을 사용하기 위해서는 현재 속한 그룹에서 사용 할 수 있는 CAD 툴 및 라이선스의 종류, 개수 및 설정 방법을 알아야 하고, 이 정보는 해당 그룹에 속한 관리자를 통해 얻는다. 기존의 CAD 툴/라이선스 공유 방법은 설계자가 CAD 툴 및 라이선스에 대한 정보를 획득하는 과정이 관리자와의 수작업을 통한 통신으로 이루어 졌으며, 해당 정보를 획득한 후에는 설계자가 직접 환경 설정을 해야 한다. 이와 같은 환경에서는 새로운 설계자가 특정 그룹에 들어왔을 때, 정보를 얻기 위한 통신과 설정 과정으로 인해 불필요한 설계시간을 소모하게 되어 결국 생산성 및 CAD 툴의 이용률을 저하시킨다. 또한, 관리자의 입장에서도 모든 설계자들에게 CAD 툴 및 라이선스에 대한 정보를 제공하기 위해서는 많은 시간 및 노력비용을 소모하게 된다. 본 논문에서는 디렉토리 서비스에 기반하여 특정 디자인 워킹 그룹에서 CAD 툴 및 라이선스를 자동화된 방법으로 공유하기 위한 방식을 제안한다. 본 논문에서 제안한 방법은 사용자와 관리자 사이의 통신 부분을 자동화하여, 관련 정보 공유를 위해 소비되던 불필요한 시간을 최소화 시킨다. 또한, 디렉토리 서비스를 이용하여 설계자와 관리자 사이의 CAD 툴 및 라이선스 정보에 대한 일관성을 항상 자동으로 유지함으로써, 설계자와 관리자 사이의 반복된 CAD 툴 및 라이선스 설정에 대한 정보 교환 작업을 제거하여 공유의 효율성을 높이고, 프락시를 사용하여 수동으로 진행되던 설계자의 환경 설정 과정을 자동화한다. 또한, 실행 가능한 프락시를 이용하여 썬 클라이언트 CAD 어플리케이션을 제공한다. 제안된 방법은 Sun Microsystems의 Jini 기술을 이용하여 구현한다. 본 논문에서 제안한 CAD 툴 및 라이선스 공유 시스템은 결과적으로 설계자와 관리자의 불필요한 설계 시간과 노력비용을 감소시키고, CAD 툴의 이용률을 향상 시킬 수 있을 것이라고 사료된다.

키워드 : 디렉토리 서비스, 공유, 캐드, 라이선스, FLEXlm

An Automated Sharing Scheme of CAD Tools and License Resources Based on Directory Service

Sung-Heon Jung^{*} · In-Sung Yim^{**} · Kyoung-Son Jhang^{***}

ABSTRACT

Designers should know CAD tool/license information such as available number of licenses or tools, types, and configuration methods to use CAD tools properly in their group. Usually, this information is provided by managers who administrate CAD tool and license servers in the specific design group. In the previous CAD tool/licenses sharing methods, designers have to get CAD tool/license information manually and setup the environments with their own hands. If a new designer comes into the design working group, the designer wastes unnecessary design time, because of these processes. As a result, designer's productivity and utilization of CAD tools will decrease. Besides, managers also waste their time and effort, since they should provide CAD tool/license information manually to each designer. In this paper, we present an automated scheme to share CAD tool/license information based on directory service. The proposed methods automate the communication processes between managers and designers that minimize unnecessary time to share CAD information. In addition, as this system maintains the consistency of CAD information automatically that remove the repeated work of managers and designers, it improves the efficiency of sharing. Besides, it automates the license configuration steps using proxy. Last, we offer Executable Proxy which is proper for thin CAD application. We believe this scheme will reduce designers and managers' time and effort as well as maximize the utilization of CAD tools.

Key Words : Directory Service, Sharing, CAD, License, FLEXlm

1. 서 론

VLSI 칩 디자인 기술의 발전과 더불어, 교육, 연구기관

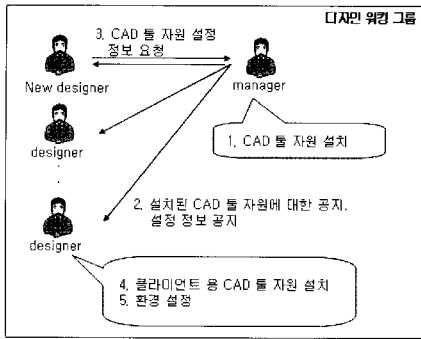
및 회사에서의 CAD 툴 수요는 점점 증가하고 있는 추세이다. 하지만 대부분의 툴 회사들은 고가의 라이선스 정책을 유지하고 있기 때문에, 툴을 사용하는 기관에서는 자신이 속한 특정 워킹 그룹이나 연구그룹을 통해서 제한된 수의 라이선스를 가지는 CAD 툴들을 지원 받아 다른 설계자들과 공유하여 사용하고 있다. 이와 같은 상황에서 설계자는

^{*} 준 회 원 : 충남대학교 대학원 컴퓨터공학과 박사과정

^{**} 정 회 원 : 한국전문연구원 책임연구원

^{***} 정 회 원 : 충남대학교 컴퓨터공학과 교수

논문접수 : 2005년 9월 14일, 심사완료 : 2005년 12월 19일

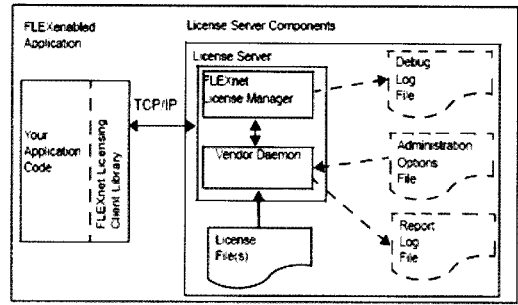


(그림 1) 전통적인 CAD 툴 자원 공유 방식

원하는 CAD 툴을 사용하기 위해, 현재 자신이 속한 워킹 그룹에 어떤 CAD 툴이 있는지, 또한 해당 CAD 툴을 사용하기 위해서는 툴을 위한 어떤 환경변수 및 라이선스 설정이 필요한지에 대한 정보를 알고 있어야 한다. 이것은 제한된 자원을 여러 설계자가 공유하므로, CAD 툴 사용에 대한 충돌을 방지하기 위해 꼭 필요한 것이다. (그림 1)은 앞선 과정을 좀 더 효율적으로 관리하기 위한 전통적인 CAD 툴 자원의 공유 방법이다.

(그림 1)에서처럼 CAD 툴을 사용하는 특정 디자인 워킹 그룹에서는 해당 CAD 툴들을 관리하는 관리자가 있으며, 관리자는 CAD 툴을 설치 및 설정 후 그 정보를 설계자들에게 통보한다. 설계자들은 통보받은 정보를 이용하여, 자신의 환경을 설정한 후 필요한 CAD 툴을 원활하게 사용할 수 있다. CAD 라이선스의 경우, (그림 1)과 같은 기존 관리 방법에서는 추가적으로 FLEXlm[1]을 이용한다. FLEXlm은 최근 라이선스 서버 관리를 위해 가장 많이 사용되는 도구로써, 특정 그룹에서 사용할 수 있는 CAD 툴 라이선스를 해당 개수만큼 허가해주는 기능을 하고, 현재 라이선스 서버의 상태를 알 수 있는 도구이다. (그림 2)는 FLEXlm을 이용한 라이선스 관리방식 중 하나이다. FLEXlm에서 라이선스를 관리하기 위한 방식으로 제공하는 Served 라이선스 모델은 클라이언트/서버에 기반을 둔 방식으로써, 사용자가 이미 라이선스에 대한 정보를 모두 알고 있을 때, 필요한 환경을 직접 설정한 후 사용가능하다. 설정에 필요한 정보 - 서버 이름, 포트 번호를 조합한 전자메일 형태의 값 - 는 관리자로부터 통보 받는다. FLEXlm은 서버/클라이언트 모델에 기반을 둔 CAD 툴 라이선스 관리 시스템이며, FLEXlm 어플리케이션은 단지 사용자가 자신의 그룹에 할당된 라이선스의 개수 이상을 사용할 수 없도록 관리한다. 이것은 사용자의 입장에서의 공유의 편의성이 아니라, CAD 툴을 공급하는 벤더들의 편의를 위한 것이라고 할 수 있고, FLEXlm을 이용한 라이선스의 관리는 설계자와 관리자 사이에 정보교환을 위한 통신에 많은 수작업이 필요하며, 관리 방법이 어렵다[2].

기존의 수작업을 통한 CAD 툴 라이선스 공유 방법에는 다음과 같은 문제점이 있다. 첫째, 새로운 설계자는 현재 자신이 속한 디자인 워킹 그룹에서 사용가능한 CAD 툴 및



(그림 2) FLEXlm을 이용한 라이선스 관리 - Served 라이선스 모델

라이선스가 무엇인지 알 수 없다. 설계자는 자신이 원하는 CAD 툴을 사용하기 전에 CAD 툴 및 라이선스 설정에 대한 정보를 미리 알고 있어야만 올바르게 사용할 수 있기 때문에 관리자가 통지한 정보를 사전에 숙지하고 있거나, 또는 설계자가 관리자에게 이에 대한 정보를 요청해야 한다. 둘째, CAD 툴 및 라이선스 서버의 설정 정보가 바뀔 때 마다 관리자 및 설계자들은 이 과정을 반복해야 한다. CAD 툴 및 라이선스는 수시로 그 환경이 변경될 수 있다. 셋째, CAD 툴 및 라이선스의 서버에 에러가 발생했을 경우, 문제에 대한 감지가 힘들고, 복구 시에도 앞서 설명한 설정 과정을 반복해야 한다. 넷째, 관리자의 입장에서는 CAD 툴 및 라이선스를 효율적으로 분배할 수 없는 문제가 발생할 수 있다. 즉, 특정 CAD 툴 및 라이선스 서버에 과부하가 걸려, 설계자는 다른 CAD 툴 및 라이선스 서버를 사용할 수 있음에도 불구하고, 해당 정보를 즉시 획득 할 수 없어 필요한 CAD 툴을 사용하지 못하게 될 수 있다. 이 네 가지 문제의 원인은 관리자와 설계자 사이의 통신이 항상 성립되어 있지 않고, 모든 과정이 수작업을 통해 이루어지기 때문이다. 마지막으로 사용자가 사전에 CAD 툴 및 라이선스 설정에 대한 정보를 알고 있다고 하더라도, 로컬 컴퓨터의 복잡한 환경 설정은 직접 해야 한다. 앞서 설명한 모든 것은 설계자와 관리자의 입장에서는 불필요한 설계시간을 소모하는 것이다.

본 논문에서는 기존 CAD 툴 및 라이선스 공유 방식에서 발생할 수 있는 문제점을 개선하여 자동화된 CAD 툴 및 라이선스 공유 방식을 제안한다. 제안한 CAD 툴 및 라이선스 공유 방식을 위해, 본 논문에서는 디렉토리 서비스라고 불리는 제 3 자 서비스를 사용한다. 디렉토리 서비스는 CAD 툴 및 라이선스에 관련된 정보를 관리하며, 통신을 위해 클라이언트와 서버를 연결할 수 있는 프락시(Proxy)를 제공하는 역할을 한다. 또한, 본 논문에서는 쉘 클라이언트 CAD 어플리케이션의 기능을 하는 실행 가능한 프락시(EP : Executable Proxy)를 제안한다.

본 논문의 다음 장에서는 제한된 CAD 툴 및 라이선스 공유 방식에 대해서 설명하고, 그 후 실제 구현된 것과 실험결과에 대해 설명하며, 마지막으로 결론 및 향후과제에 대해 언급한다.

2. 디렉토리 서비스를 이용한 CAD 자원 공유 시스템

기존 방법의 CAD 툴 및 라이선스 공유 방식은 다음과 같은 두 가지의 문제점을 가진다. 첫 째는 설계자와 관리자 사이의 통신 문제이다. 설계자는 CAD 툴의 원활한 사용을 위해, 관리자로부터 필요한 정보를 획득해야 한다. 하지만, 이와 같은 통신과정은 자동화 되지 않았기 때문에 통신과정에서 예측할 수 없는 문제가 발생 할 수 있으며, 이로 인해 설계자와 관리자 사이에 CAD 툴 및 라이선스 정보에 대한 일관성을 유지 할 수 없게 된다. 이 문제를 해결하기 위해 본 논문에서는 CAD 툴 및 라이선스에 대한 최신 정보를 항상 유지하고 관리자를 대신하여 CAD 툴 및 라이선스에 대한 정보를 설계자에게 제공하는 기능을 하는 디렉토리 서비스를 이용한다. 두 번째 문제는 환경 설정 과정이 복잡하다는 것이다. 대부분의 CAD 툴들은 라이선스 환경 변수의 설정, CAD 툴 환경 변수의 설정, 그리고 CAD 툴의 실행파일과 라이브러리에 관련된 디렉토리 설정 등과 같은 과정을 거쳐야 한다. 이와 같은 설정은 특히 UNIX와 Linux같은 운영체제에서는 사용자마다 개인이 직접 설정해야 하며, 이 설정 과정에서 문제가 발생할 수 있다. 두 번째 문제의 개선을 위해 본 논문에서는 앞서 언급한 모든 설정과정을 프락시를 이용하여 자동화 한다. 프락시는 디렉토리 서비스를 통해 CAD 툴 및 라이선스 서버로부터 제공받는다. 또한, 본

논문에서는 CAD 툴 및 라이선스 공유 시스템의 신뢰성을 높이기 위해 임대의 개념을 적용한다. 마지막으로, 썬 클라이언트 CAD 어플리케이션을 대체할 수 있는 실행 가능한 프락시를 제안한다.

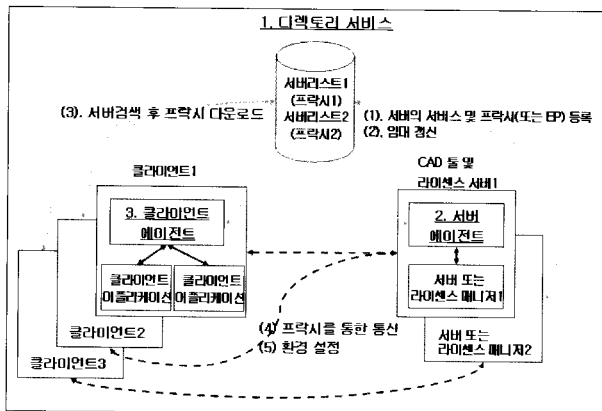
(그림 3)은 디렉토리 서비스에 기반을 둔 향상된 CAD 툴 및 라이선스 공유 시스템의 전체 구조이다. 각 부분에 대해서는 다음 절에서 자세히 설명한다.

2.1 디렉토리 서비스

본 논문에서 제안한 CAD 툴 및 라이선스 공유 방법에서 가장 중요한 역할을 하는 것은 디렉토리 서비스이다. (그림 3)에서 디렉토리 서비스의 역할은 특정 네트워크 그룹에서 CAD 툴 및 라이선스 서버의 최신 정보를 유지하는 것이다. 이 정보는 CAD 툴 및 라이선스의 속성뿐만 아니라, 서버 에이전트가 등록한 프락시도 포함한다. 디렉토리 서비스는 이 정보를 설계자에게 자동으로 지속적으로 공표하여, 관리자와 설계자 사이의 CAD 툴 및 라이선스 정보에 대한 일관성을 유지한다. 만약, 새로운 CAD 툴 및 라이선스 정보에 대해 설계자와 공유가 필요한 경우, 관리자는 해당 정보와 프락시를 디렉토리 서비스에 등록하게 되고, 설계자들은 관리자와 직접 통신하는 대신 디렉토리 서비스를 통해 그 정보를 제공 받는다.

디렉토리 서비스에 저장되는 CAD 툴 및 라이선스 정보와 프락시는 (그림 4)와 같은 구조를 가진다.

디렉토리 서비스에 저장되는 정보는 크게 두 가지로 나뉜다. 첫 번째는 CAD 툴 및 라이선스 서버에 대한 속성이다. 이 속성은 CAD 툴 및 라이선스에 관련된 모든 정보 - 예를 들어, 벤더명, 툴 이름, 타입, 프락시의 종류, 사용가능 개수, 위치, 관리자 정보 - 를 포함한다. 이 정보는 CAD 툴 및 라이선스 서버의 상태에 따라 유동적으로 변화함으로써, 설계자에게 최신의 정보를 제공한다. 두 번째는 프락시으로써, CAD 툴 및 라이선스 서버 에이전트가 등록한 프락시의 ID를 유지하고, 설계자의 요청이 있을 경우, 이것을 다운로드하여 사용할 수 있게 한다.



(그림 3) 디렉토리 서비스를 이용한 CAD 툴 및 라이선스 공유 시스템

2.2 서버 에이전트

일반적인 CAD 툴 및 라이선스 서버들은 독립적으로 수

Attributes										Proxies
Vendor	Tool name	Type	NO.	EP	Avail	Expire date	Loc.	Admin	Contact	
Synopsys	Design Analyzer	Compiler	2	N	1	2006. 1. 1	Eda1/2	s.h Jung	0164272032	proxy123@sys..
ModelTech	Modelsim	Simulator	5	N	5	2005. 9. 30	Eda1/2/3/4/5	Hwang	0428217720	proxy1model@y34..
EDA	VHDL Lint	Checker	.	Y	A	.	EDA 6	s.b Jung	0164272032	proxy1Lint@sys..
...

(그림 4) 디렉토리 서비스의 구조

행되는 어플리케이션이다. 그러므로, 이것을 디렉토리 서비스와 연동하여 동작시키기 위해서는 특별한 서버 에이전트 어플리케이션이 필요하다. (그림 3)에서 서버 에이전트는 다음과 같은 역할을 한다.

2.2.1 CAD 툴/라이선스 정보 등록 및 프락시(실행 가능한 프락시) 등록

서버 에이전트는 CAD 툴 및 라이선스 서버를 대신하여, 해당 서버의 정보를 디렉토리 서비스에 등록한다. 또한, 서버 에이전트는 클라이언트가 해당 서버에 접근할 수 있는 인터페이스인 프락시를 등록한다. 프락시는 CAD 툴 및 라이선스 서버에 접근할 수 있는 인터페이스이다. 프락시는 하나 이상의 접근 메소드를 포함하며, 해당 메소드를 통해 사용자는 CAD 툴 및 라이선스 서버에게 접근하여 필요한 정보를 요청 할 수 있다. 프락시의 인터페이스는 사용자에게 미리 공개되어, 필요할 경우 누구나 해당 인터페이스를 구현하여 이용할 수 있는 형태이다.

또한, 프락시 중에는 특별한 기능을 하는 실행 가능한 프락시가 있다. 실행 가능한 프락시는 접근 메소드들을 모두 포함하고 있을 뿐만 아니라, 사용자 인터페이스(GUI)와 일반적인 CAD 어플리케이션 기능을 모두 포함한다. 그렇기 때문에, 사용자는 실행 가능한 프락시를 다운받은 후 특별한 설정 없이 바로 사용할 수 있다. 이것은 VHDL Lint와 같은 썬 클라이언트 CAD 어플리케이션을 공유하여 사용하기 적당하다. 디렉토리 서비스에 등록되는 정보의 구조는 (그림 4)와 같다.

2.2.2 임대

임대는 분산시스템에서 네트워크상의 문제로 인해 발생하는 장애를 해결하기 위한 하나의 방법이다. 임대의 개념은 자원을 무한 공유하는 것이 아니라, 실세계의 임대의 개념처럼 자원을 특정 기간 임대해서 사용하는 것이다[3]. 임대는 CAD 툴 및 라이선스 공유 시스템에서 서버의 오동작, 정지등을 감지할 수 있게 해주며, 서비스가 불가능한 서버 자원은 임대를 갱신할 수 없게 됨으로써, 디렉토리 서비스 목록에서 제거 할 수 있도록 해준다. 이와 같이 오류로 인해 제거된 정보는 설계자에게 디렉토리 서비스를 통해 자동으로 통지함으로써, 관리자와 설계자 사이의 CAD 툴 및 라이선스 정보에 대한 일관성을 유지할 뿐만 아니라, 시스템의 신뢰성을 높인다.

2.2.3 서비스 처리

서버 에이전트는 CAD 툴/라이선스 서버를 대신하여, 프락시를 통한 설계자의 정보 요청을 대신 처리한다. 서버 에이전트는 지속적으로 서버와 통신함으로써, 서버에 대한 최근의 상태를 항상 유지하여, 프락시를 통한 설계자의 정보 요청이 왔을 때, 이것을 즉각 처리 할 수 있다.

2.3 클라이언트 에이전트

설계자들은 CAD 툴을 사용하기에 앞서, 현재 자신이 속

한 디자인 워킹 그룹에서 사용 가능한 CAD 툴 및 라이선스에 대한 정보를 알고 있어야 한다. (그림 3)에서 클라이언트 에이전트는 CAD 툴에 앞서 실행되어 현재 사용 할 수 있는 CAD 툴 및 라이선스에 대한 정보를 알려준다. 클라이언트 에이전트가 처음 실행되었을 때, 디렉토리 서비스에 자동으로 접근하여 현재 사용가능한 CAD 툴 및 라이선스 정보에 대한 목록을 받아온다. 설계자는 필요한 CAD 툴 및 라이선스 속성 - 예를 들어, 벤더명, 툴명, 타입인 시뮬레이터, 컴파일러 등 - 으로 검색하여 그 결과를 확인 할 수도 있다. 설계자는 검색된 목록 중 하나를 선택하여, 선택된 툴 및 라이선스에 대한 자세한 서비스 정보를 알 수 있으며, 필요시에는 프락시를 다운로드 받아, 프락시를 통해 필요한 설정을 자동으로 할 수 있다.

2.4 시스템의 동작 흐름

본 논문에서 제안한 CAD 툴 및 라이선스 공유 시스템의 전체적인 동작과정은 다음과 같다. (그림 3)의 첫 번째 단계에서, CAD 툴 및 라이선스 서버는 자신의 정보와 프락시를 디렉토리 서비스에게 등록한다. 서버가 디렉토리 서비스에 프락시를 등록하면, 사용자는 디렉토리 서비스로부터 해당 프락시를 다운로드 받고, 이것을 이용하여 서버에 접근하여 필요한 정보를 얻는다. 두 번째 단계로 서버 에이전트는 지속적으로 임대 갱신을 실행함으로써, 시스템의 신뢰성을 높인다. 세 번째 단계에서, 설계자들은 디렉토리 서비스로부터 현재 자신이 속한 디자인 워킹그룹에서 어떤 CAD 툴 및 라이선스가 사용가능한지를 알 수 있다. 만약 자신이 원하는 CAD 툴 및 라이선스가 있으면, 해당 프락시를 다운로드한다. 네 번째 단계에서, 설계자는 프락시를 이용하여 서버로부터 필요한 정보를 획득하고, 마지막으로 획득한 정보를 이용하여 사용자에게 필요한 환경 설정을 자동으로 한다.

3. 구현 및 실험

3.1 구현

본 논문에서 제안한 디렉토리 서비스를 이용한 CAD 툴 및 라이선스 공유 시스템을 구현할 수 있는 기술로는 동적 WEB, UPNP[4], Jini[5] 등이 있다. 본 논문은 디자인 워킹 그룹으로 구별되는 특정 네트워크 범위에서 동작하는 CAD 툴 및 라이선스 공유 방법에 대한 모델이므로, 특정 네트워크 및 모든 네트워크를 대상으로 구현 가능한 동적 WEB 보다 특정 네트워크에서만 작동하는 기술이 더 적합하다. 하지만, 특정 네트워크에 제한하지 않고, 모든 네트워크를 고려한 구현이라면 동적 WEB 기술이 적당 할 것이다. UPNP와 Jini는 PNP(Plug and Play)개념을 소프트웨어 및 하드웨어 장치에 적용시킨 기술로서 소프트웨어 및 하드웨어가 특정 네트워크에 접속되었을 때, 이것을 자동으로 인식하고, 특별한 설정 없이 바로 사용할 수 있도록 하기 위한 기술이다. 본 논문에서 제안한 방법을 구현하기 위한 기술로는 이 두 가지 모두 적합할 수 있지만, 그 중 Jini 기술

을 이용하여 구현한다.

Jini는 Sun Microsystems에서 개발하였고, 네트워크상의 장치 및 소프트웨어를 자동으로 연결하기 위한 차세대 기술이다[6, 7]. Jini는 특정 네트워크 클래스에서 플러그 앤 플레이(Plug & Play)의 개념을 가능하게 해주는 프로토콜을 제공한다[8]. Jini 기반의 서비스 소프트웨어는 네트워크에 연결되기만 하면, 해당 네트워크에 연결된 다른 Jini 기반의 소프트웨어가 즉시 이를 감지하여 서비스를 사용할 수 있게 된다[9]. Jini 서비스 소프트웨어는 네트워크를 통해 상호 연결되어 서로 서비스를 공유할 수 있는 Jini 공동체를 형성하며, 공동체를 형성한 특정 네트워크 클래스에서 Jini 기반의 소프트웨어끼리는 서로간의 서비스를 이용할 수 있는 기능을 가진다[10, 11].

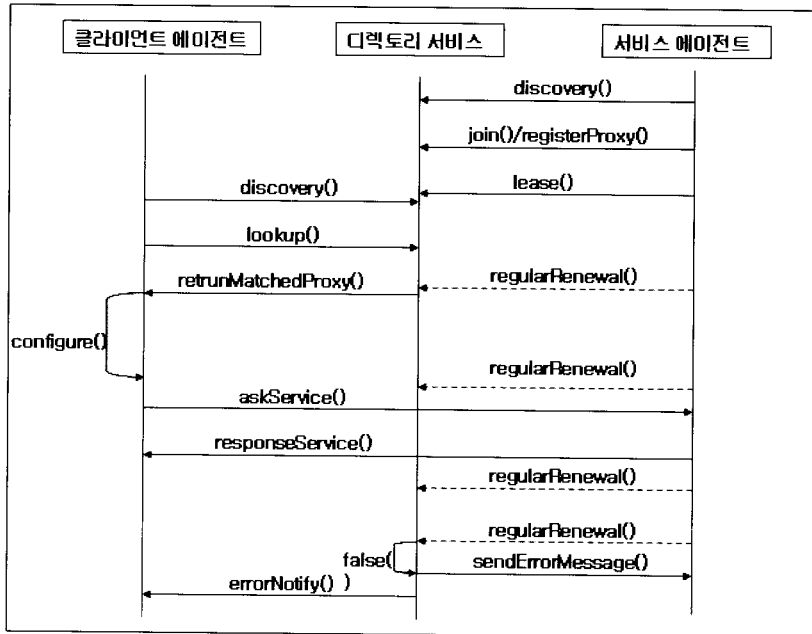
본 논문에서 제안한 시스템은 Windows XP에서 JDK 1.4.2와 Jini 2.0의 환경에서 구현되었고, 구현된 모듈은 서비스 에이전트, 프락시, 클라이언트 에이전트이다. 디렉토리 서비스는 Jini와 같이 배포되는 reggie 프로그램을 이용한다.

구현된 시스템의 동작과정은 다음과 같다.

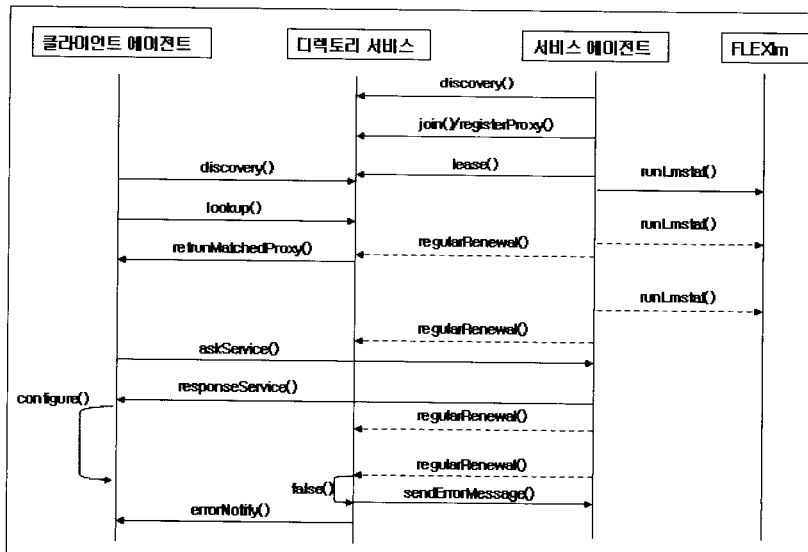
(1) CAD 툴 자원에 대한 시스템의 동작 과정

(그림 5)는 CAD 툴 자원에 대한 향상된 시스템의 상호작용 다이어그램[12]이다.

먼저 CAD 툴 서버 에이전트는 디렉토리 서비스를 찾아 자신을 CAD 툴 자원으로 등록을 한다(discovery()/join()). 이때 프락시도 같이 등록하며, 만약 수행 가능한 프락시가 있으면 같이 등록한다(registerProxy()). 등록을 마친 후



(그림 5) CAD 툴 자원에 대한 시스템의 상호작용 다이어그램



(그림 6) CAD 라이선스 자원에 대한 시스템의 상호작용 다이어그램

CAD 툴 서버 에이전트는 앞에서 설명한 신뢰성을 높이기 위해 주기적으로 임대료를 갱신한다(lease()/regularRenewal()). 임대료가 성공했을 시에는 지속적으로 서비스가 지원되는 것이며, 만약 실패하면 서버와 클라이언트 측에 그 사실을 통보한다(errorNotify()/sendMessage()). 클라이언트는 디렉토리 서비스를 발견하여 자신이 필요한 서비스를 찾고 프락시를 다운로드 한다(discovery() /lookup()/returnMatchedProxy()). 클라이언트 쪽에서는 다운로드한 프락시를 이용하여 필요한 환경이 설정되며(configure()), 만약 수행 가능한 프락시를 다운로드 하였다면, 이것을 클라이언트 프로그램으로 이용하여 CAD 서비스를 요청하고 그 결과를 얻을 수 있다(askService()/responseService()).

(2) CAD 라이선스 자원에 대한 시스템의 동작 과정

(그림 6)은 CAD 라이선스 자원에 대한 향상된 시스템의 상호작용 다이어그램이다. 먼저, 서비스 에이전트는 FLEXlm과 주기적으로 통신하며, 라이선스에 대한 최신의 정보를 유지한다(runlmstate()). 서비스 에이전트와 클라이언트 에이전트가 통신하는 방법은 (그림 5)에서 설명한 방법과 같다. 클라이언트 에이전트는 프락시를 획득 후, 라이선스 정보를 요청하고, 결과를 이용하여 환경을 자동으로 설정한다(askService()/responseService()/configure()).

(그림 7)은 구현된 시스템에서 디렉토리 서비스에 저장되는 정보가 CAD 툴/라이선스 서버의 상태에 따라 변화는 과정을 보여준다. (그림 7)의 (A)는 디렉토리 서비스의 구조의 초기 상태가 (그림 4)와 같다고 가정했을 때, 설계자가 Modelsim의 라이선스를 요구하여 사용하게 됨으로써 새롭게 갱신된 디렉토리 서비스의 구조를 보여준다. (그림 7)의

(A)에서 보듯이 Modelsim의 사용가능한 라이선스의 수가 줄어들었음을 알 수 있다. (그림 7)의 (B)는 (A)를 초기상태로 가정했을 때, 벤더명이 Synopsys인 CAD 툴 및 라이선스 서버에 문제가 발생하여 사용할 수 없게 되었을 때, 자동으로 디렉토리 서비스의 목록에서 제거되어 정보가 갱신된 디렉토리 서비스의 구조이다.

3.2 실험

본 논문에서 제안한 새로운 CAD 툴 및 라이선스 공유 방법의 효율성을 검증하기 위해, Jini로 구현된 시스템을 기존 방법과 비교한다. 현재 CAD 툴/라이선스의 공유를 자동으로 관리하기 위한 방법이 존재하지 않으므로, 기존 방법은 관리자가 설계자와의 오프라인 통신을 통해 수작업으로 진행되던 방법이다.

<표 1>은 기존 방법과 본 논문에서 제안한 방법에 대한 기술적인 비교이다.

<표 1> 기존방법과 제안된 방법의 기술적 비교

	기존 방법	본 논문에서 제안한 방법
1. CAD 툴 자원 설치 및 설정 - 관리자	수동	수동
2. CAD 툴 자원 설정 정보 통지 - 관리자	수동	자동
3. CAD 툴 자원 정보 설정 방법 획득 - 설계자	수동	자동
4. CAD 툴 자원 설정 정보 요청 - 설계자	수동	자동
5. 사용자의 CAD 툴 자원 설치 및 환경 설정 - 설계자	수동	자동
6. 디렉토리 서비스를 위한 초기 시스템 환경 설정 - 관리자	없음	수동

Attributes										Proxies
Vendor	Tool name	Type	NO.	EP	Available	Expire date	Loc.	Admin	Contact	
Synopsys	Design Analyzer	Compiler	2	N	1	2006.1.1	Eda1/2	s.h Jung	0164272032	proxy123@sys..
ModelTech	Modelsim	Simulator	5	N	4	2005.9.30	Eda1/2/3/4/5	Hwang	0428217720	proxy1model@y34..
EDA	VHDL Lint	Checker	.	Y	A	.	EDA 6	s.b Jung	0164272032	proxy1Lint@sys..
...

(A) 사용 가능한 라이선스 수 정보에 대한 갱신을 보여주는 상태

Attributes										Proxies
Vendor	Tool name	Type	NO.	EP	Available	Expire date	Loc.	Admin	Contact	
ModelTech	Modelsim	Simulator	5	N	4	2005.9.30	Eda1/2/3/4/5	Hwang	0428217720	proxy1model@y34..
EDA	VHDL Lint	Checker	.	Y	A	.	EDA 6	s.b Jung	0164272032	proxy1Lint@sys..
...

(B) CAD 툴 및 라이선스 서버의 문제로 인해 해당 서버가 제거된 상태

(그림 7) CAD 툴 및 라이선스 서버의 정보를 동적으로 반영하는 디렉토리 서비스

〈표 2〉 평균 시간 소모 비교

	기존방법	본 논문에서 제안한 방법
1. 초기 설정 - 관리자	10분 이상	30분 이상
2. 관리자로부터 필요한 정보 획득 - 설계자	15분 이상	1분 이하
3. 설계자에게 정보 전달 - 관리자	15분 이상	1분 이하
4. 환경 설정 - 설계자	15분 이상	1분 이하

앞서 설명한 것과 같이 본 논문에서 제안한 방법은 설계자와 관리자 사이의 통신 부분과 설계자의 CAD 툴 및 라이선스 환경 설정 부분을 자동화 하였다. 하지만, 본 논문에서 제안된 방법은 디렉토리 서비스에 기반한 시스템이므로 전체 시스템을 위한 초기설정이 기존 방법에 비해 한 번 더 필요하다. <표 2>는 10명의 설계자와 1명의 관리자가 있는 디자인 워킹 그룹에서 Modeltech의 시뮬레이션 툴인 Modelsim을 사용함에 있어 기존 방법과 본 논문에서 제안한 방법에 대해 평균적으로 소모되는 시간 비용을 비교한 것이다.

먼저, 초기 설정은 기존의 설정 방법에 추가적으로 에이전트를 설정하고 실행시켜야 하므로, 제안된 방법이 좀 더 많은 시간이 소모되지만, 이 과정은 전체 실험에서 1회만 발생하는 것이다. 두 번째 및 세 번째 항목은 새로운 설계자가 디자인 워킹에 들어왔다고 가정했을 때, 현재 속한 워킹 그룹에서 사용 가능한 CAD 툴 및 라이선스에 대한 정보를 공유할 때까지 걸리는 시간이다. 이것은 관리자와 설계자 사이에 이메일, 전화와 같은 직접적인 통신방법으로 수행되는 과정을 측정한 것으로 기존 방법은 평균적으로 15분이상이 걸렸지만, 본 논문에서 제안한 방법은 디렉토리 서비스에 기반하여 정보를 획득하는 과정을 자동화함으로써 설계자가 에이전트 프로그램을 수행 즉시 해당 정보를 알 수 있으므로, 월등한 시간 효율을 나타냈다. 네 번째 항목은 설계자가 필요한 환경 설정에 소비되는 시간으로서, 기존 방법은 설계자의 능력에 따라 차이는 있지만 평균적으로는 각 15분 이상이지만, 새롭게 제안된 방법은 프락시를 이용하여 자동으로 설정과정으로 수행함으로써 1분 이하의 시간이 소비되었다.

4. 결론 및 향후 과제

본 논문에서는 자동화된 CAD 툴 및 라이선스 공유 방법을 제안하였다. 제안된 방법은 디렉토리 서비스에 기반하며, 분산 환경에서 자원을 공유하기 위해 적합한 Jini 기술을 이용하여 구현하였다. 제안된 방법은 기존 방법에서 수동으로 진행되던 CAD 툴 및 라이선스 정보의 요청 및 통지, 에러 통지, 로컬 환경 설정 등의 과정을 자동화 하여 불필요하게 소비되던 시간과 노력 비용을 감소 시켰으며, 설계자의 입장에서 CAD 툴의 이용률을 최대화 할 수 있었다. 실행 가능한 프락시는 썬 클라이언트 CAD 어플리케이션을 사용

함에 있어, 설치, 설정 및 사용 과정을 간소화 시켰다. 또한, 새로운 CAD 툴 및 라이선스 공유 시스템은 임대 개념을 적용하여 네트워크 환경에서 시스템의 신뢰도를 높였다. 추후, 본 논문에서 제안한 방법을 이용하여 좀 더 많은 CAD 툴들에 적용시켜 제안된 방법의 효율성을 증명하여, 많은 CAD 툴 벤더들이 본 논문에서 적용된 시스템을 사용할 수 있도록 연구할 예정이다.

참 고 문 헌

- [1] FLEXNET "Licensing End User Guide," FLEXNET Publisher.
- [2] Dinesh R. Bettadapur, "Software Licensing Models in the EDA Industry," ASPDAC, pp.235-239, 1998.
- [3] Sun Microsystems, "Jini™ Technology Core Platform Specification," 2001.
- [4] UPNP, <http://www.upnp.org>
- [5] Sun Microsystems, Jini™, <http://www.sun.com/jini>
- [6] W.Keith Edwards, "Core Jini," Prentice-Hall, 2001.
- [7] 장익진, 권영현, 임근화, 박원배, "모바일 디바이스의 Jini 네트워킹 연동을 위한 Jini-Bridge 설계 및 구현", KNOM Review, Vol.6, No.1, 2003.
- [8] Lawrence Simth, Cameron Roe, Knud Steven Knudsen, "A JINITM Lookup Service For Resource-Constrained Devices," IEEE 4th International Workshop On Networked Appliances, pp.135-144, 2002.
- [9] Alberto Montresor, Renzo D. Ozalp B, "Enhancing Jini with Group Communication," Distributed Computing Systems Workshop, pp.69-74, 2001.
- [10] Morten Baun Moller, Bo. N. J, "Enhancing Jini's Lookup Service using XML-based Service Templates," Technology of Object-Oriented Languages and Systems, pp.19-31, 2001.
- [11] Wen-Hsien T, Hsing M, "Inter-Cluster Service Lookup Based on Jini," Advanced Information Networking and Applications, pp.84-89, 2003.
- [12] 조완수, "UML 객체지향 분석, 설계", 홍릉과학출판사, 2000.



정 성 현

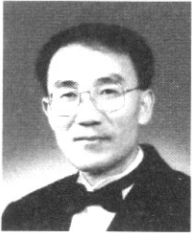
e-mail : shjung@cnu.ac.kr

2001년 한남대학교 컴퓨터공학과(학사)

2004년 충남대학교 컴퓨터공학과(석사)

2004년~현재 충남대학교 컴퓨터공학과 박사과정

관심분야: 설계자동화, 컴파일러, 컴퓨터 구조

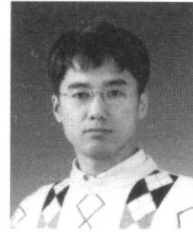


임인성

e-mail : yim@kao.re.kr
1979년 충남대학교 공과대학(공학사)
1986년 충남대학교 전자공학과(석사)
2000년 충남대학교 전자공학과(박사)
1989년 2월~1991년 1월 미국 매사추세츠
주립대학 객원연구원

2003년 3월~2004년 2월 캐나다 Herzberg Institute of
Astronomy 객원연구원

2004년 3월~현재 한국천문연구원 책임연구원
관심분야: 천문정보 및 데이터처리



장경선

e-mail : sun@cnu.ac.kr
1986년 서울대학교 전자계산기공학과
(학사)
1988년 서울대학교 컴퓨터공학과(석사)
1995년 서울대학교 컴퓨터공학과(박사)
1996년 3월~2001년 8월 한남대학교
컴퓨터공학과 교수

2001년 9월~현재 충남대학교 컴퓨터공학과 교수

관심분야: 컴퓨터 구조, 설계자동화, 하드웨어 소프트웨어 통합
설계