

# 1) Jini를 이용한 CAD 툴 라이선스의 효율적인 관리 방식

임재우<sup>o</sup>, 장경선

충남대학교

{jwlim<sup>o</sup>, ksjhang}@ce.cnu.ac.kr

## An Efficient Management Scheme of CAD Tool Licenses using Jini

Jae-Woo Lim<sup>o</sup>, Kyoung-Son Jhang

Dept. of Computer Engineering, ChungNam National University

### 요약

VLSI 설계 생산성의 향상을 위하여 여러 대학 및 연구소, 기업체들은 다양한 종류의 CAD 툴을 사용하고 있다. 각 단체에서 CAD 툴을 사용하기 위해서는 CAD 툴을 설치한 후, 사용자가 직접 라이선스 파일의 위치를 환경변수에 설정하거나, 라이선스 서비스 데몬이 실행되어 있는 컴퓨터의 포트번호와 서버의 이름을 관리자에게 요청하여 환경변수를 직접 설정해주어야 하는 번거로움이 발생한다. 또한, 라이선스의 관리 및 어떤 CAD 툴이 사용 가능한지 확인이 어렵다. 본 논문에서는 네트워크 상의 라이선스 서버로부터 필요한 라이선스 정보를 찾고, 환경변수를 설정하는 라이선스 처리과정을 보다 효율적으로 하기 위해, 네트워크 상의 서비스 발견과 이용을 쉽게 해주는 Jini를 사용하여 라이선스를 공유하는 방식을 제안한다.

### 1. 서론

최근에 많은 CAD 툴들은 FLEXlm[1]이라는 라이선스 관리 서버를 이용하여 라이선스 관련 작업을 처리하고 있다. 사용되는 라이선스 처리 방법으로는 크게 두 가지가 있는데, 첫 번째 방법은, 툴의 벤더로부터 제공받은 라이선스 파일을 로컬호스트의 특정 경로에 위치시키고, LM\_LICENSE\_FILE 환경변수를 해당 파일의 위치로 설정해주는 방법이다. 이는 이전부터 사용해오던 방법이다. 두 번째 방법은 최근 많이 사용되고 있는 방법으로써, 벤더가 제공한 라이선스 파일은 라이선스 서버라는 특정한 컴퓨터에 저장해 두고, 그 라이선스 서버에 접근할 수 있는 포트 번호와 서버 이름을 LM\_LICENSE\_FILE이라는 환경변수로 설정하는 방법이다.

이러한 라이선스 처리 방법들은 공통적으로 라이선스를 얻기 위하여 사용자가 라이선스 파일 및 서버가 어디에 위치하고 있는지, 라이선스 서버에 접근을 하기 위해서는 어떤 포트로 접근해야 하는지 등에 대한 정보를 알고 있어야만 한다. 또한, 획득한 라이선스 파일, 포트번호 및 서버이름 등에 대해 LM\_LICENSE\_FILE 환경변수 설정을 해야 하는 추가 작업이 필요하게 된다. 결국 사용자는 라이선스 사용가능 여부를 사전에 알기가 힘들고, 라이선스가 사용 가능하다는 것을 알게 되더라도, 해당 라이선스에 대한 설정을 사용자가 직접 해 주어야만 CAD

툴 사용이 가능하다.

본 논문에서는 보다 효율적으로 라이선스를 관리하기 위해서 Jini[2] 기술을 사용한다. Jini 기술은 네트워크 상에서의 플러그 앤 플레이를 가능하게 해주는 기술로, 네트워크상의 자원에 대한 공유를 쉽게 해주고, 자원의 발생시 즉시 알림이 가능하다. 이러한 Jini 기술을 사용하면으로써, 첫째, 가용한 서비스나 라이선스의 목록을 제공받을 수 있고, 둘째, 라이선스를 얻고자 하는 사용자가 보다 쉽게 라이선스 파일이나 라이선스 서버의 포트번호, 서버이름 등을 얻을 수 있게 하고, 셋째, 획득한 라이선스 정보를 자동으로 LM\_LICENSE\_FILE 환경변수에 설정되도록 함으로써 사용자에게 편리성을 제공한다. 마지막으로, 서버의 변경이나 설정변경 등을 실시간으로 확인하여 처리할 수 있다.

본 논문의 2장에서는 FLEXlm을 사용한 라이선스 처리 방법과 문제점에 대하여 알아보고, 3장에서는 Jini 기술에 대한 간단한 소개와 Jini의 어떤 부분들을 사용하여 라이선스의 관리를 효율적으로 처리할 것인가, 그리고 라이선스의 관리 및 처리 절차는 어떻게 진행되는가에 대해서 소개한다. 마지막으로 4장에서는 결론 및 향후과제에 대해서 언급한다.

### 2. FLEXlm을 통한 라이선스의 처리

FLEXlm은 Macrovision사에서 개발한 라이선스 관리 툴로써, FLEXlm Served license model, FLEXlm Unserved license model의 두 가지 라이선스 처리 모델이 있다. 본 논문에서는 두 가지 방법 중, 일반적으로 많이 사용

1) 본 논문은 한국과학재단이 지정한 지역협력연구센터(RRC)인 충남대학교 소프트웨어연구센터의 지원으로 수행된 과제의 결과입니다. 본 문에 사용된 CAD 툴은 IDEC으로부터 지원받았습니다.

되고 있는 모델인 FLEXlm served license model에 대해 알아보고, 이 모델의 비효율적인 면을 효율적으로 보완하는 방법을 제시한다.

FLEXlm served license model은 라이선스 서버에 의해 라이선스가 제공되는 방식으로써, 그림 1과 같이 FLEXlm 라이선스 응용프로그램과 라이선스 서버를 구성하는 라이선스 매니저 데몬(lmgrd) 및 벤더 데몬, 그리고 라이선스 파일등 4개의 컴포넌트들로 이루어진다.

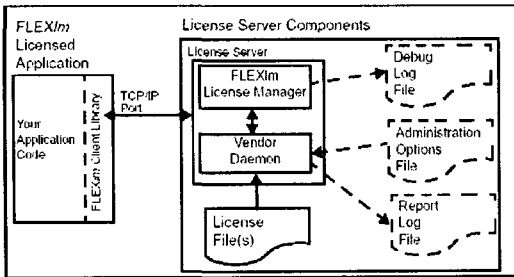


그림 1. FLEXlm served license model

FLEXlm은 그림 1과 같은 컴포넌트들을 사용하여 라이선스의 요청, 수락 및 거부를 처리한다. 라이선스 처리 절차는 다음과 같다.

1. 클라이언트 응용프로그램 내의 라이선스 모듈이 호스트 이름과 라이선스 매니저 데몬(lmgrd)의 포트번호를 포함하는 라이선스 파일을 찾는다.
2. 클라이언트는 라이선스 매니저 데몬과 연결을 수립하고, 라이선스 매니저 데몬에게 어떤 벤더 데몬과의 연결을 원하는지 알린다.
3. 라이선스 매니저 데몬은 어떤 머신과 포트가 벤더 데몬과 잘 연결되는지 결정하고, 그 정보를 클라이언트에게 보낸다.
4. 클라이언트는 벤더 데몬과의 연결을 수립하고, 라이선스를 위한 요청을 한다.
5. 벤더 데몬은 사용가능한 라이선스가 있는지 메모리를 체크하여, 승인 또는 거부를 클라이언트에게 반환한다.
6. 응용프로그램내의 라이선스 모듈에서 최종적으로 반환된 값을 통해 승인 또는 거부한다.

FLEXlm은 이러한 처리과정을 통해 동작되지만, 몇 가지 비효율적인 사항이 발생된다. 첫째, 사용자는 미리 라이선스 서버가 설치되어 있는 머신에 대한 정보를 알고 있어야만 한다. 둘째, 사용자는 라이선스 서버로부터 받은 라이선스 파일 또는 포트번호 및 서버이름을 환경 변수에 설정을 해주어야 한다. 셋째, 사용자가 몇 가지 CAD 툴을 사용하고자 한다면 같은 작업을 반복해야 한다.

이와 같이 FLEXlm을 사용할 때 발생하는 불편함은 Jini의 조회서비스를 사용함으로써 효율적으로 처리할 수 있다.

### 3. Jini를 이용한 개선 방법

Jini는 SUN 사가 추구하는 소프트웨어로 Java의 기본 개념을 사용하여 네트워크 상에서 장치 또는 소프트웨어의 접속이나 공유를 단순화 하기위해 사용되는 메커니즘이다. 일반적으로 사용자가 네트워크 상의 어떠한 장치나 소프트웨어를 사용하고자 한다면, 그 장치나 소프트웨어에 대한 드라이버의 설치를 해야만 한다. 하지만, Jini를 사용하게 되면 네트워크 상의 장치 또는 소프트웨어에 바로 접근할 수 있고, 리소스의 공유를 보다 쉽게 할 수 있다.

Jini의 가장 중요한 개념에는 발견, 조인, 조회서비스, 임대 등 4가지가 있다[3]. 조회서비스는 모든 개념들의 중심이 되는 것으로, Jini 서비스가 등록되고, Jini 클라이언트가 필요로 하는 서비스를 찾기 위해 사용되며, 임대를 처리한다. 발견은 Jini 서비스나 클라이언트가 조회서비스를 찾는 과정이고, 조인은 Jini 서비스가 조회서비스에 자신에 대한 프락시를 등록하는 과정이다. 임대는 조회서비스에 등록되어 서비스의 임대와 관련하여 서비스가 유효하지 못한 상황이나, 서비스가 등록되지 않은 상황에서, 서비스를 찾기 위한 대기시간을 위해 사용된다.

License Manager에서도 이러한 개념들을 사용하여 보다 효율적으로 라이선스를 관리하고자 한다. 조회서비스를 사용함으로써 License Manager가 프락시를 통해 가용한 서비스나 라이선스의 목록을 제공해 줄 수 있게 된다. 그리고 발견을 사용함으로써 조회서비스를 찾을 수 있게 되고, 발견한 조회서비스를 통해 사용자가 라이선스 파일이나 라이선스 서버의 포트번호, 서버이름 등을 보다 편리하게 알 수 있으며, 조인을 사용함으로써 License Manager의 서비스를 조회서비스에 등록할 수 있기 때문에, 사용자가 필요한 정보를 편리하게 제공 받을 수 있게 된다. 임대 개념을 사용함으로써, 라이선스 서버의 상태에 대한 변화를 보다 빨리 실시간으로 알 수 있게 된다.

이와 같은 Jini 기술을 사용하여 서비스를 제공하고, 제공받는 메커니즘에는 서비스의 복사본을 통한 방식과 백-엔드 프로세스를 방식이 있다[4].

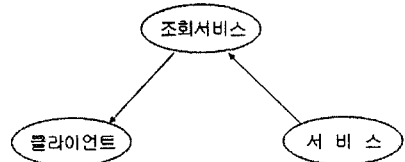


그림 2. 서비스의 복사본을 통한 방식

그림 2는 서비스의 복사본을 통한 방법으로써, Jini 서비스가 자신의 인스턴스를 프락시에 담아 조회서비스에 등록하고, 클라이언트는 조회서비스를 찾은 뒤, 자신이 원하는 서비스가 조회서비스에 등록해 놓은 인스턴스를 사용해 자신의 자바가상머신을 통해 독자적으로 실행

하는 방법이다.

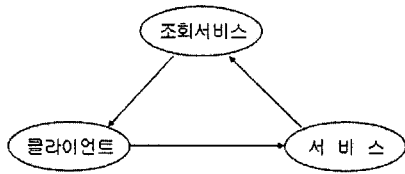


그림 3. 백-엔드 프로세스를 통한 방식

그림 3의 백-엔드 프로세스를 사용하는 방법은 서비스가 프락시를 조회서비스에 등록하면, 클라이언트는 조회서비스를 찾아 원하는 서비스의 프락시를 다운로드한다. 그리고 프락시를 통해 백-엔드 쪽의 서비스에 직접 RMI[5] 통신을 통해 연결한 뒤, 서비스를 사용하는 방법이다.

본 논문에서는 그림 2와 그림 3의 방법 중에 백-엔드 프로세스를 통해 라이센스의 처리를 보다 효율적으로 하기 위한 방법을 제안한다. 그림 4는 백-엔드 프로세스 방식을 사용하여 라이센스의 관리를 효율적으로 하기 위한 License Manager의 전체 구성도이다.

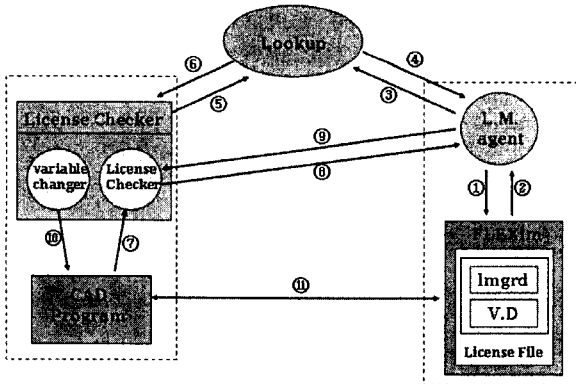


그림 4. Jini를 이용한 License Manager의 전체 구성도

그림 4에서 Jini가 이용되는 부분은 조회서비스와 License Checker, License Manager Agent 등이다. 조회서비스는 License Manager Agent의 프락시를 유지하고, License Checker에게 프락시를 제공해준다. License Manager Agent는 FLEXlm으로부터 라이센스에 대한 정보를 얻어와 License Checker에게 제공한다. License Checker는 License Manager Agent에게 라이센스를 요청하고, 획득한 라이센스를 LM\_LICENSE\_FILE에 설정해준다.

그림 4에서 기술된 라이센스의 처리 과정은 다음과 같다.

1. 라이센스 파일에 대한 정보를 요청한다.(①)

2. 라이센스 파일에 대한 정보를 획득한다.(②)
3. 조회서비스를 찾아 프락시를 등록한다.(③)
4. 서비스 아이디를 받는다.(④)
5. 클라이언트가 조회서비스를 찾아 원하는 서비스 프락시를 찾는다.(⑤)
6. 원하는 서비스의 프락시를 다운로드한다.(⑥)
7. CAD 툴 정보를 입력하고, 프락시를 통해 서비스와 연결을 수립한 뒤, 라이센스를 요청한다.(⑦, ⑧)
8. 서비스는 요청받은 라이센스가 있다면 정보를 제공하고, 없을 경우에는 거부를 반환한다.(⑨)
9. 획득한 라이센스를 환경변수에 설정하고, CAD 툴을 사용한다.(⑩, ⑪)

Jini를 이용하여 라이센스를 처리하면, 사용자는 라이센스 파일에 대한 정보나 라이센스 서버에 대한 정보 등을 미리 알고 있을 필요가 없기 때문에 라이센스를 보다 쉽고 효율적으로 처리할 수 있다. 또한, License Checker의 Variable Changer를 사용하면 사용자가 직접 환경변수를 설정해주지 않아도 되므로, 사용자에게 편리함을 제공해 준다.

#### 4. 결론 및 향후과제

본 논문에서는 라이센스의 효율적인 처리와 관리를 위해 Jini를 이용한 방법을 제안하였다. Jini를 이용한 라이센스 처리 방법을 사용하여 라이센스를 처리할 경우, 사용자가 라이센스에 대한 정보를 미리 알고 있지 않아도 라이센스의 획득을 가능하게 해준다. 또한 Jini를 이용한 라이센스 처리 방법은 라이센스 서버로부터 받은 라이센스를 얻는 과정을 보다 편리하게 해준다. 또한, 로컬머신의 환경변수 설정에 있어서 발생하던 복잡함을 없애준다. 향후에는 제안한 방법을 통하여 서비스와 클라이언트, 그리고 서비스와 클라이언트의 연결을 위해 Jini를 사용하여 실행 프로그램의 구현에 대한 연구가 요구된다.

#### 참고문헌

- [1] FLEXlm End User guide, Version 9.2 2003  
[http://www.macrovision.com/products/legacy\\_products/flexlm/index.shtml](http://www.macrovision.com/products/legacy_products/flexlm/index.shtml)
- [2] Jini specification and API Archive,  
<http://java.sun.com/products/jini/>
- [3] W. Keith Edwards, Core JINI, 영한출판사, 2002
- [4] 강신동, Jini 기술과 실제, KOMS 기고원고, 2000  
[http://www.javastudy.co.kr/docs/lec\\_j2me/jini/jini.htm](http://www.javastudy.co.kr/docs/lec_j2me/jini/jini.htm)
- [5] Java RMI Specification,  
<http://java.sun.com/j2se/1.4.2/docs/guide/rmi/spec/rmiTOC.html>