

임베디드시스템용 소프트웨어플랫폼을 위한 동적 재구성 관리자 설계

박충범⁰, 유용덕, 최훈
충남대학교 컴퓨터공학과
{cbpark⁰, ydyu}@ce.cnu.ac.kr, hc@cnu.ac.kr

The Design of Dynamic Reconfiguration Manager of Software Platform for Embedded Systems

Choong-Bum Park⁰, Yong-Duck You, Hoon Choi
Dept. of Computer Engineering, Chungnam National University

요 약

현재 대부분의 컴포넌트기반개발 기법은 J2EE나 .NET과 같은 플랫폼상에서 적용되고 있다. 하지만 제한된 자원을 갖는 임베디드시스템에 J2EE나 .NET 플랫폼을 탑재하기에는 부담이 크다. 본 논문에서는 소형 디바이스 환경에 적합한 컴포넌트 기반의 임베디드시스템용 소프트웨어플랫폼을 설계하고, 동적 재구성 관리자를 통해 컴포넌트를 관리하는 기법을 제안한다.

1. 서 론

하드웨어의 성능이 향상됨에 따라 소프트웨어의 기능도 향상되게 되고 그 크기와 복잡성이 증가하고 있다. 크고 복잡해지는 시스템을 효과적으로 개발하고 유지 보수하기 위한 방안으로서 J2EE나 .NET 플랫폼상에서의 컴포넌트기반개발(CBD: Component Based Development) 기법이 부각되고 있다. 하지만 낮은 컴퓨팅파워와 제한된 리소스를 가지는 소형 디바이스를 위한 임베디드시스템 환경에서 J2EE나 .NET 플랫폼을 적용하기에는 큰 부담이 된다[1].

본 논문에서는 CBD 기법을 이용하여 소형 디바이스 환경에서 동작 가능한 컴포넌트 기반의 임베디드시스템용 소프트웨어플랫폼을 설계하였고 컴포넌트기반 플랫폼 관리에 필요한 동적 재구성 관리자(Dynamic Reconfiguration Manger)의 동작과정을 기술한다[2][3].

2장에서는 J2EE와 .NET환경에서의 CBD 기법에 대해서 소개하고, 3장에서는 본 논문에서 설계한 임베디드시스템용 소프트웨어플랫폼의 특징과, 동적 재구성 관리자의 구조와 동작과정에 대해서 기술한다. 마지막으로 4장에서는 결론 및 향후 연구 방향

에 대해서 기술한다.

2. CBD 기법

CBD 기법은 시스템을 구성하는 컴포넌트를 조립하는 방식으로서 시스템을 구축하는 개발 방법을 말하며 컴포넌트와 인터페이스를 이용하여 구현한다.

2-1. CBD 기법의 이점

컴포넌트는 시스템을 구성하는 하나의 요소로서 각 컴포넌트는 전체 시스템 기능의 일부분을 제공한다. 전체 시스템은 컴포넌트들을 조립함으로써 완성되고 동작한다.

컴포넌트는 기존의 함수, 클래스 등과 비하여 보다 큰 규모이므로 재사용을 하는 경우 재사용 효과가 크며 강한 수준의 정보 은닉 개념을 지원한다. 또한, 기존 컴포넌트를 수정하는 대신에 기존 컴포넌트를 새로운 컴포넌트로 대체하는 기능을 제공하여 시스템 운영에 효율성을 증가시킨다.

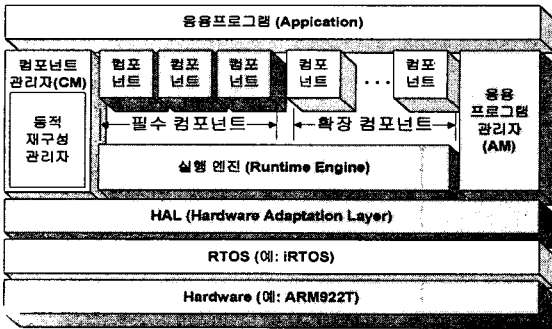
2-2. 일반적인 CBD 기법의 한계

현재 CBD 기법은 J2EE 플랫폼의 EJB 컴포넌트와 .NET 플랫

* 본 논문은 국방과학연구소의 실시간 운영체제 인터페이스용 미들웨어 연구과제의 수행결과임

품의 COM+ 컴포넌트와 같이 데스크탑 이상의 컴퓨팅 파워를 갖는 머신을 대상으로 개발되어 있다. 하지만 소형 디바이스를 위한 임베디드시스템의 제한된 환경에 위와 같은 플랫폼을 탑재하기에는 많은 비용이 필요하다. 소형 디바이스에 탑재될 임베디드시스템용 플랫폼은 작고 가벼우면서도 CBD 기법의 장점을 갖기 위해 새로운 소프트웨어 프레임워크의 개발이 필요하다.

3. 임베디드시스템용 소프트웨어플랫폼



[그림 1] 임베디드시스템용 소프트웨어플랫폼 구조

임베디드시스템용 소프트웨어플랫폼은 DLL(Dynamic Linking Library) 형태의 컴포넌트 단위로 설계되었다. 실행 엔진은 플랫폼 동작에 필요한 핵심 역할을 담당하며 동적 재구성 관리자에게 다이내믹 링커와 로더를 제공한다[4][5]. 실행 엔진 상 위에는 각각의 컴포넌트들이 위치하며 컴포넌트 관리를 위해 컴포넌트 관리자와 응용프로그램 관리자가 존재한다. 컴포넌트는 역할에 따라 필수 컴포넌트와 확장 컴포넌트로 구분된다. 필수 컴포넌트는 커널이나 네트워크 기능 등과 같이 플랫폼 운영에 필수적인 컴포넌트로서 반드시 탑재되어야 한다. 확장 컴포넌트 그래픽 처리나 보안통신기능 등과 같이 디바이스 용도에 따라 추가적으로 탑재 가능하다.

3-1. 임베디드시스템용 소프트웨어플랫폼 특징

① 소형 디바이스에 적합한 CBD 기법을 적용하였다.

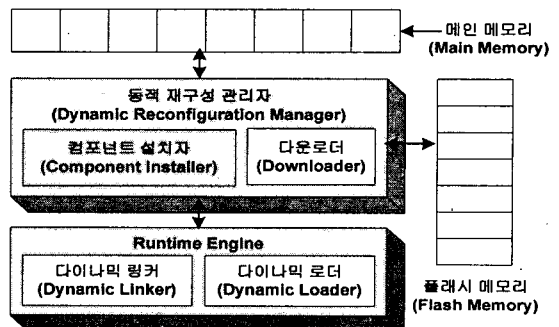
J2EE나 .NET 플랫폼이 바이트코드 실행환경환경을 가지는 것과는 달리 임베디드시스템용 소프트웨어플랫폼은 바이너리 실행 환경에서 동작하므로 낮은 컴퓨팅파워에서도 높은 실행속도를 제공한다. 또한 디바이스의 용도에 따라 필요한 컴포넌트들만을 조립하여 플랫폼을 구성하므로 리소스를 절약할 수 있고, 동적 재구성 관리자를 통해 플랫폼의 구성을 변경할 수 있다.

② 운영체제에 독립적이다.

임베디드시스템용 소프트웨어플랫폼에서는 플랫폼 하단에 HAL(Hardware Adaptation Layer)을 정의함으로써 운영체제에 독립적이다. 결과적으로 디바이스의 운영체제가 바뀌게 되면 HAL의 교체만으로 동일한 동일한 플랫폼을 운영할 수가 있다.

3-2. 동적 재구성 관리자

임베디드시스템용 소프트웨어플랫폼에서는 운영에 있어서 예기치 못한 오류에 대한 대응이나 버전관리, 기능확장 등을 목적으로 컴포넌트의 설치/삭제/교체 및 컴포넌트의 동적 링크 설정 기능을 지원하며 이는 동적 재구성 관리자가 담당한다.



[그림 2] 동적 재구성 관리자 구조

컴포넌트는 다운로더에 의해 플래시 메모리로 다운로드 되고 컴포넌트 설치자의 동작에 의해 컴포넌트의 재구성과 관리가 이루어진다. 컴포넌트는 다이내믹 링커와 로더에 의해 메모리에 적재되고 응용에 의해 사용된다.

① 동적 재구성 관리자의 컴포넌트 관리

가) CDF(Component Description File)

```

Authentication_key = a1bb434
Name = 3d_library
FileName = 3D.dll
Major = 2
Minor = 1
Vendor = CNU distsys
Security_level = CP
Description = This contains API set related to 3D
    
```

[그림 3] CDF 형식

각각의 컴포넌트들은 자신의 이름, 버전, 보안수준 및 특성에 대한 설명 등의 정보를 명세화한 CDF를 가지며 동적 재구성 관리자는 CDF로부터 추출한 컴포넌트 정보를 이용하여 해당 컴포

넌트를 플랫폼에 설치하게 된다.

나) LCL(Loaded Component List)

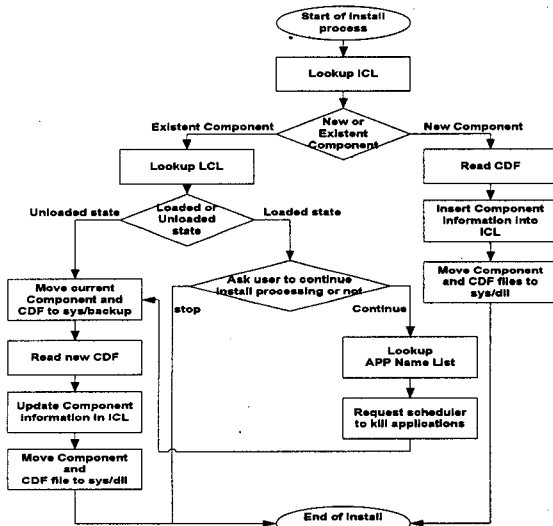
동적 재구성 관리자는 설치된 컴포넌트 정보를 이용하여 설치된 컴포넌트들을 메모리에 로드하고 그 정보를 LCL에 추가한다. LCL에는 메모리에 로드된 컴포넌트들의 정보를 유지한다.

<표 1> Load_Component_Info 구조체

필드명	설명
Component_name	메모리에 적재된 컴포넌트의 심볼 이름
Handle	컴포넌트에 대한 핸들
Count	컴포넌트의 참조 카운트
*APP_list	컴포넌트를 참조하고 있는 응용프로그램의 목록 리스트
*Interface_list	컴포넌트의 인터페이스 정보 리스트

② 컴포넌트 설치자 동작

동적 재구성 관리자의 컴포넌트 설치자는 ICL의 정보를 참조하여 컴포넌트의 설치/삭제/교체하게 되고, 다이나믹 링커와 로더를 이용하여 해당 컴포넌트를 메인 메모리에 동적으로 링크하고 로드하게 되며 그 결과를 LCL에 반영한다. [그림 4]는 컴포넌트 설치 과정을 나타낸다.



[그림 4] 컴포넌트 설치 순서도

가) 컴포넌트 설치

새로운 컴포넌트를 설치하거나 기존의 컴포넌트를 업데이트한다. 기존의 컴포넌트는 백업되어 컴포넌트의 복구기능을 제공하여야 한다.

나) 컴포넌트 삭제 및 복구

필요성이 없어진 컴포넌트를 삭제하거나 백업하였던 기존의 컴포넌트를 복원한다.

다) 컴포넌트의 메모리 적재

플랫폼에 설치된 컴포넌트를 메모리에 적재하여 응용프로그램이 사용 가능하게 한다. 컴포넌트의 보안 수준과 응용프로그램의 보안 수준을 비교하여 컴포넌트에 대한 접근을 관리한다.

라) 컴포넌트의 메모리 해제

메모리로부터 적재된 컴포넌트를 해제하여 컴포넌트를 설치/삭제/복구 한다.

4. 결론

본 논문에서 제안하는 임베디드시스템용 소프트웨어플랫폼은 DLL 형태의 컴포넌트 단위로 설계되었다. 컴포넌트들은 동적 재구성 관리자에 의해 조립되어 하나의 플랫폼으로 구성되고 필요에 따라 재조정 가능하다. 결과적으로 필요한 컴포넌트만으로 플랫폼을 구성하여 리소스가 절약되고 필요에 따라 플랫폼을 확장 및 축소가 자유롭다. 또한 바이너리 실행환경에서 하드웨어에 독립적인 구조로 설계되어 낮은 컴퓨팅파워에서도 빠른 실행속도와 높은 호환성을 제공한다.

현재의 동적 재구성 관리자는 응용프로그램이나 사용자의 요구에 의해서만 컴포넌트를 재구성한다. 앞으로 애기치 못한 오류에 대응하기 위한 자가치료(self-healing)나 시스템 환경변화에 따른 적응(adaptation) 기능을 제공할 수 있도록 확장할 계획이다.

5. 참고문헌

- [1] 채흥석, "객체지향 CBD 개발 Bible", 한빛미디어, 2003. 8. 5
- [2] 유용덕, 김연수, 최훈, "동적 Upgrade 기능을 구비한 WPI 개발환경 설계", 한국통신학회 추계 종합 학술발표회 논문초록집, vol. 28, p.343, 2003.12
- [3] 임형택, 장준, 최훈, "우선인터넷 플랫폼에서의 API 관리자 설계 및 구현", 한국정보처리학회 2004 추계학술발표논문집, 제 11권 2호 pp. 1763-1766, 2004.11.13.
- [4] 김연수, 강민철, 유용덕, 최훈, "우선인터넷 플랫폼에서 다중 응용프로그램 수행을 위한 스케줄러 설계", 한국정보처리학회 2004 추계학술발표논문집, 제 11권 2호 pp. 1759-1762, 2004.11.13.
- [5] 유용덕, 김연수, 임형택, 강민구, 최훈, "우선인터넷 플랫폼을 위한 메모리 관리 모듈 설계 및 구현", 한국정보처리학회 2004 추계학술발표회 논문초록집, 제 11권 2호, pp. 1783-1786, 2004.11.13.