

ISSN 1598-5164



한국정보과학회  
Korea Information Science Society

# 2006 한국컴퓨터종합학술대회 논문집(C)



한국정보과학회

KOREA INFORMATION SCIENCE SOCIETY

2006년 6월 21일~23일 · 용평리조트



6. RFID 미들웨어에 적합한 부하 분산 기법 연구 .....	박재걸 · 임종호 · 이광민 · 황경수 · 채홍석	145
7. BORE 프로세스를 적용한 정보 시스템 개발 .....	한광신 · 김상수 · 인호	148
8. ECA규칙을 이용한 동적 확장가능 로봇 아키텍처 .....	우희정 · 정우성 · 이은주 · 이종석 · 우치수	151
9. 웹 응용을 제품 계열로 재공학하기 위한 도메인 요구사항 추출 방법 .....	이광우 · 김진한 · 이창호 · 이병정	154
10. 커뮤니티 컴퓨팅 어플리케이션 개발을 지원하기 위한 이클립스 기반의 통합개발환경 .....	김동욱 · 이정태 · 류기열	157
11. 임베디드 소프트웨어를 위한 과거 이력 기반 테스트 케이스 순위화 기법 .....	백창현 · 대상원 · 신승훈 · 박승규	160
12. 상호 영향 계층분석방법을 통한 Win-Win 협의 모델 .....	김도훈 · 이택 · 인호	163
13. 소프트웨어 설계 모듈의 재사용을 위한 Statemate 일반화 차트의 확장 .....	김창진 · 최진영	166
14. 시맨틱 웹 서비스의 기술을 위한 모델지향접근방식 .....	김일웅 · 김연석 · 이경호	169
15. TFM 기반 무기체계 소프트웨어의 시스템 테스트 모델 설계 .....	김재환 · 윤희병	172
16. 적용형 소프트웨어에 대한 웹 서비스 발견 기반 기법 .....	김진한 · 이창호 · 이광우 · 이병정 · 김희천	175
17. WinWin 합의 모델을 적용한 HW/SW 분할 방법론 .....	박지용 · 김상수 · 채정욱 · 인호	178
18. OCL을 이용한 UML 2.0 행위 모델의 시간 일관성 검사 .....	한아람 · 전상욱 · 홍장의 · 배두환	181
19. 유닛 테스트 자동화 테스트 도구를 위한 프레임워크 설계 .....	김영상 · 백창현 · 박승규	184
20. 방대성과 복잡성에 대응하는 요구사항분석방법 .....	임상원 · 이택 · 김도훈 · 인호	187
21. SMV를 이용한 Structural Decision Table 명세의 정형검증 .....	전승재 · 지은경 · 차성덕	190
22. 사용자 평등성을 제공하는 실시간 분산 웹 환경의 폭주하는 트랜잭션 관리 기법 .....	조홍래	193
23. 병렬 소프트웨어 프러덕트 라인을 위한 컴포넌트 생성기 .....	장정아 · 최승훈	196
24. 웹 서비스 컴포넌트를 이용한 소프트웨어 개발에서의 명세 확장 방법과 그 효과 분석 .....	서평익 · 이동근 · 최은만	199
25. 모바일 환경을 이용한 연구과제 관리 시스템 개발 .....	최성만 · 유철중 · 장옥배	202
26. RFID 비즈니스 이벤트 생성을 위한 모델링 프로세스 및 참고관리 시스템의 응용 .....	정민선 · 문미경 · 염근혁	205
27. 서비스 지향 아키텍처를 적용한 e-Engineering 프레임워크 구축 .....	오일노 · 국승학 · 김현수 · 이재경 · 박성환	208
28. Acme를 이용한 Hyper-torus Architecture 원격의료시스템의 성능평가 .....	최인화 · 조민주 · 방혜미 · 김명주 · 이병걸	211
29. 액세스그리드 프레임워크 .....	백종권 · 이태동 · 정창성	214
30. 센서노드상의 응용모듈 동적재구성을 위한 버전동기화 도구 .....	정선우 · 길아라 · 정기원	217
31. 설계명세서를 이용한 안전등급 PLC 운영체제 컴포넌트 시험방법 .....	이영준 · 성아영 · 최병주 · 손한성	220



32. AOP를 위한 동적 결합 메커니즘	김태웅 · 김경민 · 김태공	223	14. C
33. AspectJ를 지원하는 AOP 개발 프레임워크	박옥자 · 김정옥 · 김재웅 · 유철중 · 장옥배	226	
34. 사례 연구 : 속성기반설계(ADD)를 적용한 하나로 연구로 방사선감시시스템(RMS) 개발	서용석 · 홍석봉 · 김성진 · 김종명 · 김현수	229	15. C
35. 공동운용환경에서의 내장형 S/W 개발을 위한 구성요소 설계	류동국 · 김우열 · 김영철	232	16. C
36. 아키텍처 기반의 자가 성장 로봇을 위한 컴포넌트 선택 메커니즘	박유식 · 고인영	235	17. C
37. 기업 환경에 최적화된 닷넷 기반 솔루션 구현	왕정휘	238	18. C
38. 관점 지향 프로그래밍을 이용한 컴포넌트의 테스트 프로시저 모듈화 방안	허승현 · 최은만	241	19. C
39. 레거시 컴포넌트의 유지보수를 위한 AOSD 기반의 Component Connector	김경민 · 김태웅 · 김태공	244	20. C
40. 서비스 지향 아키텍처 기반 P2P 비즈니스 프로세스 모델링	장영원 · 이명희 · 유철중 · 장옥배	247	21. C
41. 패킷 필터링 보안 정책을 테스트하기 위한 테스트베드 구축	국승학 · 김현수	250	23. C

### 3. 정보보호

1. 정보보호 가상망 모델링 및 시뮬레이션	윤호상 · 장희진 · 김상수 · 박제근 · 김철호	253	24. S
2. 부요 리눅스 상에서 패치 업데이트를 위한 GUI 프로그램 구현	오정현 · 장혜영 · 최종찬 · 조성제	256	25. C
3. 프라이버시 보장을 위한 RBAC 기반의 접근 제어 프레임워크	홍성호 · 조은애 · 문창주 · 백두권	259	26. R
4. 일괄 보안정책 관리 시스템의 설계 및 구현	김이곤 · 신영선 · 유성훈 · 안소진 · 박진섭	262	27. C
5. 사용목적 분류화를 통한 프라이버시 보호를 위한 보안 접근제어 모델	나석현 · 박석	265	28. R
6. 분산 기업 환경을 위한 위임 관리 모델	변창우 · 박석	268	
7. 유비쿼터스 업무공간의 협업 RBAC 모델	이수정	271	
8. e-seal 보안 프로토콜을 위한 효율적인 Pseudo Random Function	민정기 · 강석훈 · 정상화 · 김동규	274	
9. 센서네트워크에서 인-네트워크 프로세싱을 위한 경량 키 관리 프로토콜	김경태 · 김형찬 · R.S.Ramakrishna	277	
10. RFID 애플리케이션을 위한 상황 인식 보안 아키텍처	권중규 · 정목동	280	
11. 고가 물품에 적합한 강력한 RFID 프라이버시 보호 기법	조정환 · 여상수 · 김성권	283	
12. 스마트 더미들 이용한 위치기반서비스(LBS)의 익명성 보장 방법	임현수 · 조은애 · 문창주 · 박대하 · 백두권	286	
13. 임베디드 보드를 이용한 암호화 성능측정	정윤호 · 최태영	289	



# 공동운영환경에서의 내장형 S/W 개발을 위한 구성요소

## 설계

류동국<sup>o</sup>

김우열 김영철

홍익대학교 컴퓨터정보통신 소프트웨어공학연구소  
{ rdk<sup>o</sup>, john, bob }@selab.hongik.ac.kr

### Designing Construction Elements for Embedded S/W Development on Common Operation Environment (COE)

Dong-Kuk Ryu<sup>o</sup>

Woo-yeol Kim, R. Young-Chul Kim  
Hong-ik University

#### 요약

항공기, 핸드폰, 자동차 등 여러 분야에서 내장형 S/W가 사용되고 있다. 내장형 S/W는 일반 S/W와는 달리 운용 및 개발 환경이 특화되고 다양하여 개발비용이 많이 들고 관리에 어려움이 있다. 이러한 문제점들을 해결하기 위하여 미군에서는 공동운영환경(Common Operating System)을 도입하여 시스템들의 통합을 용이하게 하고 상호운용성을 증진하고 있다. 공동운영환경은 운영체계 및 시스템 S/W는 표준화하고 공통으로 재사용가능한 부분을 일종의 컴포넌트라 할 수 있는 세그먼트로 개발하여 재사용하는 개발 및 운용환경을 말한다. 본 논문에서는 공동운영환경의 컴포넌트 기반 개발 방법을 소개하였다. 그리고 내장형 S/W 개발 및 운용에 필요한 내장형 공동운영환경 구조를 정의하고 세그먼트를 설계하였다.

#### 1. 서론

공동운영환경은 현재 미군에서 운용중인 세그먼트라는 일종의 컴포넌트 뱅크 개념의 운용환경이다[1,2]. (그림 1)과 같이 하위 수준에서는 상위의 시스템 소프트웨어로 표준화한다. 그리고 상위 수준에서는 공통으로 중복 개발되는 기능들에 대하여 세그먼트로 동축한 추재사용하여 시스템을 개발한다. 공동운영환경은 동축된 세그먼트를 재사용함으로써 개발비용 절감, 개발의 적시성 확보, 상호운용성 증진 그리고 체계통합이 용이한 장점이 있다.

정보기술이 변화되고 많은 내장형 소프트웨어가 개발되면서 자동차, 정보통신기기, 통신 위성 및 항공기와 같은 임무 중심적(mission-critical)이고 안전성이 요구되는 제품이 디지털화되고 있다. 이러한 내장형 시스템은 제품의 개발목적에 따라 다양한 운용환경 및 개발환경이 존재하여 많은 개발 비용이 요구되며 개발된 소프트웨어의 재사용 또한 매우 어려운 현실이다.

내장형 시스템의 개발에 있어서 공동운영환경 개념을 도입하면 일반 정보 시스템의 개발에서와 같이 개발비용의 절감은 물론, 개발 기간을 단축할 수 있고 개발된 시스템간의 통합성도 손쉽게 확보할 수 있을 것이다. 따라서 본 논문에서는 현재 다양한 형태로 존재하고 있는 내장형 시스템의 운용환경 및 개발환경을 표준화 할 수 있는 내장형 공동운영환경 개발을 위하여 구성요소를 정의하고 핵심 구성요소인 내장형 공동운영환경 세그먼트를 설계하였다.

내장형 공동운영환경 개발을 위해서 본 논문에서는 내장형 공동운영환경 개발에 필요한 구성요소들을 도출하여 정의하였다. 내장형 시스템 개발의 특성을 반영하여 공동운영환경을 설계하였다. 그리고 내장형 공동운영환경의 핵심 구성요소인 세그먼트의 구조를 정의하였다. 재사용 컴포넌트들은 세그먼트의 형태로 정의한다. 공동운영환경 개발에 활용할 수 있고 내장형 시스템의 특성을 반영한 세그먼트의 구조를 설계하였다.

#### 2. 공동운영환경

본 장에서는 공동운영환경에 대하여 설명한다. 공동운영환경은 미군에서 제정하여 사용 중인 시스템 개발 프레임워크이다. 공동운영환경은 좁은 의미에서는 컴포넌트의 재사용을 위한 운용환경으로 정의할 수 있다. 반면 넓은 의미로는 정보 시스템 개발을 위한 표준 구조(Architecture)를 포함한 소프트웨어 재사용을 통한 시스템 개발 방법론이라고 할 수 있다 [2].

#### 2.1 공동운영환경의 개발 배경

공동운영환경은 미군이 개발하였다. 미군은 여러 정보 시스템을 개발하여 운용하여 왔다. 그러나 (그림 1)과 같이 시스템의 이질성, 중복개발, 독립적인 시스템 운영의 문제점이 발생하였다. 정보 시스템의 개발이 독립적으로 수행되어 정보시스템 간에 동일한 기능을 중복적으로 개발되어 예산이 낭비되었다. 운용환경 또한 시스템

마다  
우 어  
하고  
이  
으로  
컴포  
개발  
제, 마  
표준  
하게  
세그  
출차  
라는

공  
동  
용  
성  
해  
결  
하  
는  
사  
의  
전

2.1 공

공  
동  
공  
동  
지  
경  
커  
사  
용  
자  
갈  
수  
로  
먼  
저  
한  
세  
개  
발  
을  
세  
우  
음  
을  
수  
행  
하



마다 상이하여 전체 시스템의 통합이나 상호운용은 매우 어려웠다. 이러한 결과 시스템의 개발비용이 증가하고 운용의 비효율적인 측면이 발생하였다.

이 국방부는 이러한 문제점을 해결하기 위하여 중복적으로 개발되는 기능들을 세그먼트(segment)라는 일종의 컴포넌트로 개발하고 관리하여 재사용이 가능하게 하여 개발비용 및 시간을 절감하고자 하였다. 그리고 운영체제, DBMS 그리고 통신 인프라 등은 상용제품을 이용하여 표준화된 운용환경을 정의하여 시스템간의 통합을 용이하게 하고 상호운용 가능하도록 하였다. 미군은 이러한 세그먼트의 재사용 및 시스템 운용환경 및 제반 제도 및 절차를 공동운용환경(Common Operating Environment)이라는 이름으로 운용하고 있다.

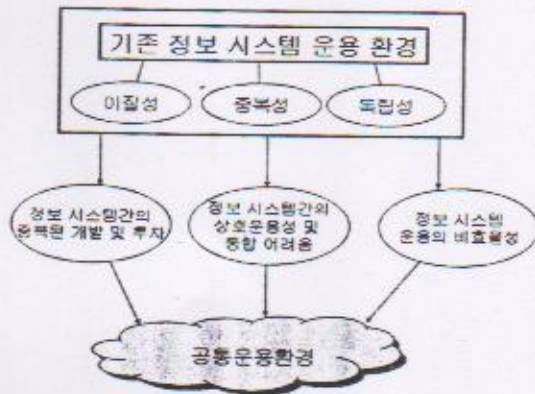


그림 1. 공동운용환경 등장 배경

공동운용환경은 이러한 정보체계의 중복개발, 상호운용성 및 통합의 문제점, 비효율적인 운영의 문제점들을 해결하기 위하여 개발되었다. 공동운용환경은 일종의 컴포넌트 개념의 세그먼트(segment)를 개발하여 후발 정보 시스템 개발에 재사용한다. 그리고 운영체제, DBMS 등의 시스템 소프트웨어 등은 상용제품을 표준화하는 것이다. 또한 공동운용환경은 세그먼트 개발을 위한 개발 방법론, 시스템의 구조, 데이터 운용환경 등의 시스템 개발의 전반적인 요소들을 포함하는 운용환경이다.

### 2.1 공동운용환경의 구조

공동운용환경의 구조는 (그림 2)와 같다. 임무응용, 공동자원응용, 공동 기반 구조 서비스, 국방 공동운용환경 커널 계층으로 구분된다. 상단계층으로 갈수록 사용자가 개발한 기능들이 추가되는 반면 하단 계층으로 갈수록 상용제품을 표준화하여 정의된다.

먼저 임무응용계층은 실제 사용하는 응용 시스템을 위한 세그먼트들이 존재한다. 이러한 세그먼트는 시스템 개발 과정에서 앞으로 중복 개발 가능성이 있는 부분 중 세그먼트로 등록된 것이다. 공동자원응용 계층은 임무응용 계층의 기능을 구현하기 위한 라이브러리 기능을 수행하는 함수들의 집합이다. 응용체계의 기능 요소의

개발에 필요한 시스템 기능을 지원하는 함수들로 구성된다. 공통 기반 구조 서비스는 일종의 시스템 서비스로써 DBMS나 통신 인프라의 기능을 제공한다. 주로 운영체제에서 제공하는 기능들이다. 국방 공동운용환경 커널 계층은 운영체제의 커널 서비스와 동일하다. 공통 기반 구조 서비스와 국방 공동운용환경 커널은 상용 제품을 표준화하여 결정된다.

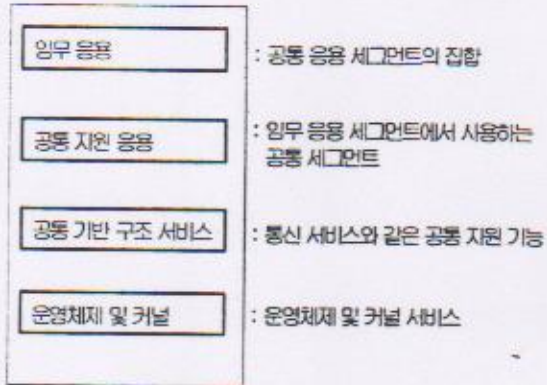


그림 2. 공동운용환경 계층 구조

### 3. 내장형 공동운용환경

#### 3.1 내장형 공동운용환경의 필요성

많은 정보기기들이 내장형 소프트웨어를 탑재하여 개발되고 있다. IT 기술이 발달되면서 각종 가전제품에서부터 인공위성을 발사하는 미사일에 이르기까지 많은 제품들이 내장형 소프트웨어를 개발하여 사용하고 있다. 일반 소프트웨어와는 달리 내장형 소프트웨어의 개발환경 및 운용환경은 시스템에 특화되어 개발되고 있다. 따라서 개발 비용이 일반 소프트웨어와 비교하여 많이 들고 있다. 그리고 운용환경도 다양하여 한번 개발된 소프트웨어의 재사용은 매우 어려운 현실이다.

이러한 문제점을 해결하기 위하여 내장형 소프트웨어를 컴포넌트 단위로 재사용하고자 하는 연구가 수행된바 있다[3]. 본 논문에서는 이러한 내장형 컴포넌트 재사용 기술을 내장형 공동운용환경에 적용하였다. 내장형 소프트웨어의 높은 개발 비용과 다양한 운용환경에서 발생하는 문제점을 해결하기 위해서는 내장형 시스템을 위한 공동운용환경의 개발이 필요하다. 내장형 공동운용환경을 개발하여 내장형 시스템 개발에 적용하면 기 개발 세그먼트를 재사용이 가능함에 따라 개발 비용이 절감되고 신속한 개발이 가능하게 될 것이다. 또한 시스템 운용환경이 표준화되기 때문에 여러 내장형 시스템이 유기적으로 운용되는 복합 시스템의 상호운용 능력의 향상에도 크게 기여할 수 있을 것이다.

#### 3.2 내장형 공동운용환경 구성요소 정의

내장형 공동운용환경의 구성 요소는 내장형 소프트웨어의 구성 요소를 반영하여야 한다. 일반적으로 내장형



소프트웨어의 구성요소는 하드웨어에서 동작하는 소프트웨어는 먼저 장치를 구동하기 위한 드라이버와 운영체제, 미들웨어 그리고 어플리케이션으로 구성된다. 미들웨어는 내장형 소프트웨어의 다양한 운용환경과 응용어플리케이션을 연결하는 역할을 수행한다. (그림 3)은 이러한 내장형 소프트웨어의 운용환경을 고려한 내장형 공동운용환경의 구조이다. 내장형 공동운용과는 달리 내장형 시스템은 하드웨어와 소프트웨어가 결합된 형태이므로 임무 응용 시스템 계층이 추가된다. 이 계층은 하드웨어와 소프트웨어가 결합된 시스템 수준에서의 재사용을 지원한다. 그리고 내장형 소프트웨어의 다양한 운용환경을 지원하기 위하여 미들웨어 계층이 포함된다. 미들웨어 계층에서는 시스템의 특성과 독립적으로 임무응용 세그먼트가 개발될 수 있도록 지원하는 역할을 수행한다. 내장형 소프트웨어는 일반 정보 시스템과 비교하여 규모가 작기 때문에 공동된 지원 라이브러리의 사용이 적어지는 경향이 있다. 따라서 내장형 공동운용환경에서는 공통지원기능 계층은 삭제되었다. 공통기반구조 서비스와 운영체제 및 커널 계층은 동일하다. 내장형 공동운용환경에서는 하드웨어의 사용이 일반화되어 있어 시스템에 적합한 장치 드라이버의 개발이 필요하게 된다. 따라서 장치 드라이버 계층을 추가하였다.

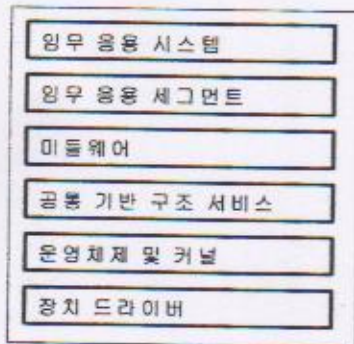


그림 3. 내장형 공동운용환경 구조

### 3.2 내장형 세그먼트 구조 설계

내장형 공동운용환경 개발에 있어서 가장 중요한 요소는 재사용 가능한 세그먼트를 사용하는 것이다. 이러한 내장형 세그먼트를 재사용하기 위해서는 내장형 세그먼트의 구조를 정의하여야 한다. 본 논문에서는 내장형 공동운용환경 세그먼트를 (그림 4)와 같이 정의하였다.

- Event Handler : 세그먼트 외부 및 내부의 이벤트를 처리한다.
- Driver : 세그먼트 내부의 비즈니스 로직을 제어한다.
- Business Logic : 세그먼트 내부의 기능을 수행한다.

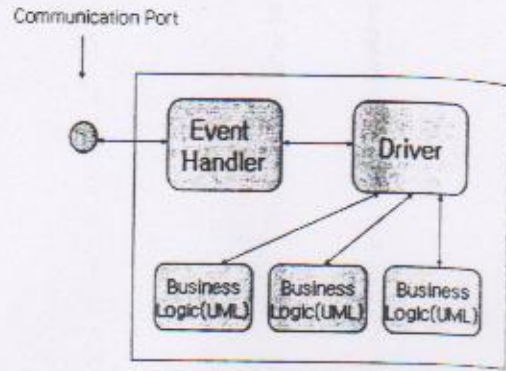


그림 4. 내장형 공동운용환경 세그먼트 구조

내장형 세그먼트는 executable UML을 사용한 모델 기반의 언어로 개발한다[4]. 모델 기반의 언어는 설계단계에서 사용자 검증이 용이하며 운용환경에 독립적인 개발이 가능하여 재사용이 용이한 장점이 있다.

### 4. 결 론

여러 정보기술을 활용하여 로봇이나 각종 센서 등의 장비에 내장형 소프트웨어를 탑재한 내장형 시스템들이 많이 개발되고 있다. 이러한 내장형 시스템들은 아직 표준화된 운용환경이 정립되지 못하여 각각 독립적인 운용환경 및 개발환경에서 개발 및 운용되고 있다. 따라서 이러한 방식으로 개발된 시스템은 개발 비용 및 시간이 많이 소요되며 개발의 성공도 보장하지 못한다. 또한 개발된 시스템이라 할지라도 타 시스템과의 상호운용이나 시스템 통합에 있어서 많은 문제점을 발생하고 있다. 본 논문에서는 이러한 문제점을 해결하기 위하여 내장형 공동운용환경을 제안한다. 내장형 공동운용환경은 내장형 시스템의 다양한 운용환경을 표준화하고 기 개발된 세그먼트를 재사용하여 성공적인 내장형 시스템 개발을 가능하게 지원할 것이다.

### 참고문헌

1. George, T. H. and William, T. C., "Component based Software Engineering", Addison-Wesley, 2001.
2. "Defense Information Infrastructure Common Operating Environment(DII COE) : Integration and Runtime Specification 4.0 (I&RTS)", US DoD, 1999.
3. S. Wang and K. G. Shin H, "An Architecture for Embedded Software Integration Using Reusable Components", International Conference on Compiler, Architecture, and Synthesis for Embedded Systems(CASES 2000)
4. 김인기, UML 설계와 응용: 클래스 모델 만들기, 정보문화사, 2003.