

컴포넌트 기반 모바일 임베디드 컨버전스 소프트웨어 개발 프레임워크

Framework for Developing Mobile Embedded Convergence Software using CBD

김 행 곤*
Haeng-Kon Kim

요 약

최근의 컴퓨팅 시스템은 모바일을 사용한 비즈니스와 다양한 컨버전스 분산 업무 처리로 확대되면서 모바일 임베디드 소프트웨어 개발 방법론에 대해 모바일 비즈니스에서 많은 관심을 가지고 있다. 아울러 최근 재사용성과 독립성 그리고 이식성을 가진 컴포넌트를 기반으로 한 모바일 임베디드 소프트웨어 개발에 또한 많은 초점이 집중되고 있다. 컴포넌트 기반 임베디드 응용 시스템 개발은 제품의 생산성과 유지보수성 그리고 신뢰성을 보장한다. 컴포넌트 각 요소들 간의 계층적, 수평적 서비스 지원 및 협력을 위한 명확한 인터페이스 정의를 통한 컴포넌트의 원활한 조립이 컴포넌트 기반의 임베디드 소프트웨어 개발 성공을 위한 필수적 요소이다. 즉, 관련 아키텍처 정의와 이를 기반으로 한 생성 프로세스 및 컴포넌트의 명세화, 그리고 컴포넌트 프레임워크의 효과적 적용 단계를 통해 모바일 임베디드 소프트웨어 개발의 성공을 달성할 수 있다.

따라서 본 논문에서는 사용자의 요구사항에 최대한 접근하고 모바일 임베디드 도메인을 기반으로 둔 소규모 단위 모바일 컴포넌트(MEC: Mobile Embedded Component)를 대상으로 조립을 위한 인터페이스 명세 제공을 서술한다. 모바일에 확장적 컴포넌트 계층화와 모바일 비즈니스 로직 확보를 위해 재구성 가능한 설계 패턴 및 컴포넌트 군(비즈니스 도메인 카테고리)을 형성하고 제공한다. 제안 하는 모바일 임베디드 컴포넌트 프로세스는 기존 프로세스가 가지는 비 일치성을 보완하여 컴포넌트의 개발과 사용에 실제적으로 활용할 수 있도록 정의한다. 모바일 비즈니스 프로세스를 위한 의미 지향적이며 모델링 기반 원칙에 따라 명확하고 풍부한 프로세스 정보를 포함한다. 또한 기능의 모듈성과 독립성이 보장되고 조립 가능한 컴포넌트를 기반으로 동적이고 복잡한 모바일 비즈니스 영역에 적용 가능한 개발 모델을 제시하고 작성된 모델을 기반으로 하는 모바일 임베디드 개발 사례를 제시한다. 본 연구에서 제시하는 컴포넌트 기반 모바일 임베디드 소프트웨어 개발 프레임워크는 효율성, 생산성 및 신뢰성과 유지보수성을 증대할 수 있는 이점을 가진다.

Abstract

Computing systems in the modern era are expanding rapidly to include mobile-based businesses that make us of the various convergence distributed business process. This has lead to growing interest in the field of mobile embedded software development methodology, which has in turn lead to the proliferation of the embedded mobility. The use of CBD (Component Based Development) provides reusability, maintainability and portability, all of which are very important and focus issues to the business process. It also comes with the inherent productivity, quality and reliability of CBD. To make efficient use of CBD, though, clarified interface definitions for component integration are necessary. These definitions should be made up of collaborative hierarchical and horizontal architecture layers. Successful definitions should apply an effective framework made up of the architecture and process.

In this paper, we describe an interface specification for small grained mobile embedded components(MEC) for the mobile embedded domain to meet maximum user requirements. We build and deploy the reconfigurable design patterns and components (in business domain categories) to make a component hierarchy and business logics for mobile embedded software.

Proposed components specification plays a major role in development of the software for handling inconsistency in existing specification. It also includes plenty of specification information, using semantics and modeling based mechanisms to support business processes. We propose a development model of mobile embedded software using CBD for very complex and dynamic mobile business. We can apply it in a plug and play manner to develop the software. We verify that our framework supports very good productivity, quality and maintainability to meet the user's requirements in mobile business.

☞ keyword : Component Based Development, Mobile Convergence, Frameworks, Embedded Software, Component Architecture

* 정희원 : 대구가톨릭대학교 컴퓨터정보통신공학부 정교수
hangkon@cu.ac.kr

[2008/05/23 투고 - 2008/05/25 심사 - 2008/06/16 심사완료]

1. 서론

최근 IT산업에서는 휴대성과 무선 통신의 개념의 중요성이 강조되면서 다양한 모바일 단말 장치와 주변 환경에 대한 연구가 활발하게 진행되고 있다. 4세대 네트워크 환경은 어떤 장비를 가지고 원하는 콘텐츠를 시간이나 장소에 상관없이 접근할 수 있는 유비쿼터스 및 컨버전스 상용화 기술 연구에 박차를 가하고 있다[1,2]. 또한, 기존 웹 환경이 무선 장비들의 대중적으로 보급되면서 무선 인터넷 환경으로 도입하기 위한 서비스 기술이 요구된다. 모바일 장치에서 Post-PC의 개념으로 스마트 모바일 관련 임베디드 소프트웨어 개발에 많은 관심을 가지고 모바일 사용자의 능동적, 지능적인 서비스 요구사항을 지원하며 차세대 지능형 유무선 연동 서비스 시스템 설계를 위해 모바일 임베디드 영역에 대한 다양한 개발 방법론 및 관련 기술이 요구된다[3,4,5].

모바일 임베디드 비즈니스 애플리케이션 시스템은 융통성, 적용성, 확장성, 그리고 자율성 등을 요구하고 있으며, 이러한 요구사항은 새로운 소프트웨어 개발 방법론의 적용을 통해 해결하는 것이 중요하며 가속화되고 있다.

컴포넌트 기반의 모바일 임베디드 소프트웨어 개발은 소프트웨어 개발의 최상의 경쟁적인 전략으로서 빠르게 확산되고 있다. 잘 정의된 인터페이스에 기반 한 블랙박스 소프트웨어 부품들의 조립에 의해 응용 개발의 생산성과 유지보수성을 보장하며, 웹과 자바 등의 인터넷 활용 기술의 지원을 통해 컴포넌트 보급과 획득의 대중화는 기반 시스템 소프트웨어에서 미들웨어, 그리고 도메인 응용에 이르기까지 보증되어진 체계로서 적용되고 있다[6,7]. 이미 컴포넌트 시장이 형성되어 컴포넌트의 개발과 유통, 그리고 획득과 시스템으로의 통합을 위한 컴포넌트 비즈니스가 운영되고 있는 실정이다[8]. 그러나 컴포넌트 기반 모바일 임베디드 소프트웨어 개발자들은 소프트웨어 재사용의

초기 과정에서 겪어야만 했던 것과 같이 도메인 컴포넌트의 식별과 발견, 적용의 프로세스에서 비용적, 품질적 위험 부담을 고려해야만 한다. 이는 응용 시스템 개발자가 필요로 하는 컴포넌트를 식별하기 위한 명세가 개략적 수준만 제공함으로써 시스템으로의 적용시 개발자의 자의적인 해석에 의한 개발 의도와 비일치가 발생하며, 적절한 행위적인 인터페이스의 의미적 확보가 미흡함에 따라 조립을 위한 쉬운 플러그 지점을 가질 수 없기 때문이다. 따라서 각 인터페이스 요소의 의미 및 이들 간의 관련성, 컴포넌트 사용을 위해 보증된 시나리오 등을 비롯해 평가를 위한 품질 요소들이 명세에 포함되어야 한다[9].

본 논문에서는 사용자의 요구사항에 최대한 접근하고 모바일 임베디드 도메인을 기반을 둔 소규모 단위 분산 컴포넌트를 대상으로 인터페이스 프로세스 제공을 서술한다. 모바일에 확장적 컴포넌트 계층화와 비즈니스 로직 확보를 위해 재구성 가능한 설계 패턴 및 컴포넌트 군(비즈니스 도메인 카테고리)을 형성하고 제공한다. 제안 하는 모바일 임베디드 컴포넌트 프로세스는 기존 프로세스가 가지는 비 일치성을 보완하여 컴포넌트의 개발과 사용에 실제적으로 활용할 수 있도록 정의한다. 모바일 비즈니스 프로세스를 위한 의미 지향적이며 모델링 기반 원칙에 따라 명확하고 풍부한 프로세스 정보를 포함한다. 또한 기능의 모듈성과 독립성이 보장되고 조립 가능한 컴포넌트를 기반으로 동적이고 복잡한 모바일 비즈니스 영역에 적용 가능한 개발 모델을 제시하고 작성된 모델을 기반으로 하는 모바일 임베디드 개발 사례를 제시한다[10]. 따라서 먼저, 표준 하부 구조를 정의하고 소프트웨어 전개 모델을 제공함으로써 컴포넌트 생성과 사용, 평가를 위한 근거를 확보하기 위해 모바일 임베디드 적용 MEC-ABCD 컴포넌트 아키텍처 모델을 정의한다. 이를 바탕으로 모바일 임베디드 컴포넌트(Mobile Embedded Component: 이하 MEC)의 기능 및 서비스 구현시의 행위적인

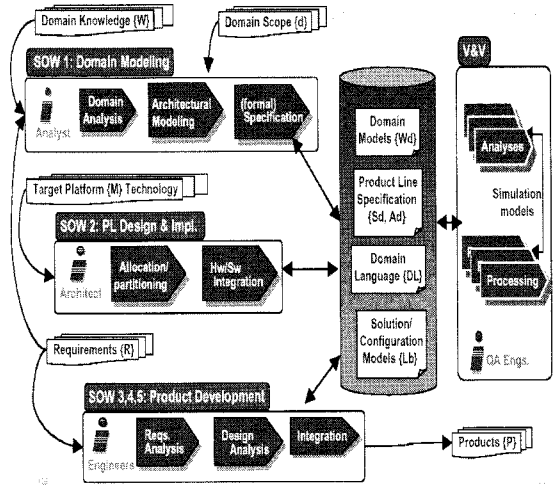
정규 서술과 조립을 위한 일반적인 메소드를 제공하는 명세 방법을 연구, 제시한다. 또한 동일 카테고리 내의 MEC 간의 명세 조립을 위한 프레임워크를 제시한다.

2장에서는 관련 연구로 모바일 임베디드 시스템 개발 및 CBD에 관련된 현재의 연구 현황과 컴포넌트 아키텍처, 명세에 관한 전반적인 기술에 대해 서술하고 3장에서는 MEC-ABCD 아키텍처와 명세 방법을 정의한다. 그리고 4장에서는 MEC 명세 예를 제시하고 명세 조립을 위한 프레임워크 예를 제시하며 5장에서 결론짓는다.

2. 관련연구

2.1 모바일 임베디드 소프트웨어 개발 방법론

(YES : Yamacraw Embedded System)는 최근 S/W 개발 기술들을 임베디드 S/W 개발 영역에 도입하기 위한 효율적인 통합환경을 구축하고 기존 자산들의 재사용 접근과 컴포넌트 기반 개발 방법을 적용함으로써 지속적인 형상 변환이 가능하며 갱신이 예상 가능한, 높은 품질의 임베디드 S/W 제품 생산을 목표로하는 시스템이다. 이를 위해 다중 제품의 생산 비용이, 개별 단일 제품을 개발하는 비용의 합보다 적을 경우, 스코프의 경제성 (economic of scope)을 실현시킬 수 있는 다중 제품들의 생산을 위한 임베디드 S/W 제품 계열 체제를 개발하였다. SOC(System-On-a-Chip) 재사용 프로세스를 확립하여, 임베디드 제품 계열을 통한 임베디드 S/W 생산에 적용하였다. 그림 1에서와 같이 프로세스는 도메인 모델링, 제품 계열 설계 및 구현, 제품 개발을 위한 3단계로 구성되며 Reusable IP(Intellectual Property)를 통해 임베디드 S/W 개발을 위한 공통성을 위한 설계 (Design-for-Commonality)와 다형성의 제어(Control-of-Variability)를 실현하였다. 재사용 기반의 임베디드 S/W의 제품 계열 프로세스와 재사용 IP 라이



(그림 1) SoC와 IP가 통합된 YES 임베디드 프레임워크 예

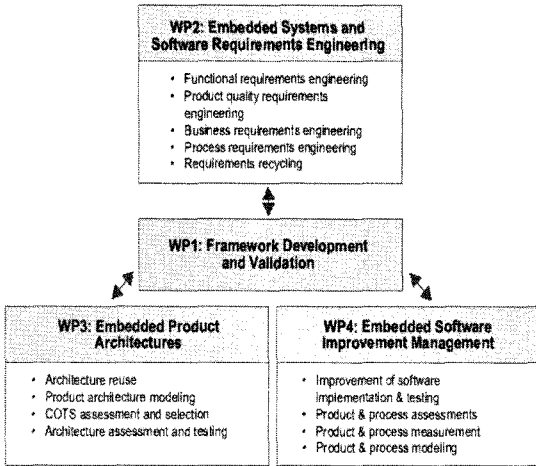
브러리의 통합을 통해, 미래의 다양한 임베디드 요구를 수용할 수 있는 자동화화된 임베디드 프레임워크이다[11].

그림 2에서와 같이 MOOSE 는 지네릭 임베디드 시스템 개발 프레임워크를 만들고 이를 다양한 임베디드 소프트웨어 프로젝트에 적용해 검증한 시스템이다[11]. 시스템 및 소프트웨어 공학, 요구 공학, 제품 아키텍처 설계 및 분석, 소프트웨어 개발 및 테스트, 제품 품질 및 소프트웨어 프로세스 개선 방법론을 임베디드 도메인을 위한 프레임워크 및 지원도구에 통합. 이 프레임워크가 임베디드 소프트웨어 프로젝트의 구조화, 관리 및 평가에 유용하게 활용될 수 있도록 하는 특징을 가진다.

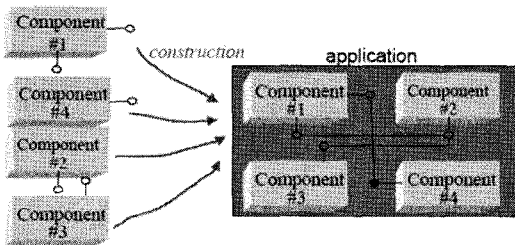
2.2 컴포넌트 기반 모바일 응용 개발 방법론

컴포넌트 기반 방법론은 일반적으로 모델 기반으로 소프트웨어 컴포넌트를 설계하고 구현하는데 솔루션을 제공할 수 있다. 컴포넌트 기반 개발은 그림 3에서와 같이 컴포넌트를 개발하는 방법뿐만 아니라, 최적의 컴포넌트를 선택하여 이를 조립함으로써 새로운 애플리케이션을 생성하는 새로운 소프트웨어 개발 방법론으로 높은 융통성과 유지

보수성을 제공하므로 소프트웨어 개발 생산성을 향상시키고 소프트웨어의 품질 또한 보증할 수 있다.



(그림 2) MOOSE 작업 구조



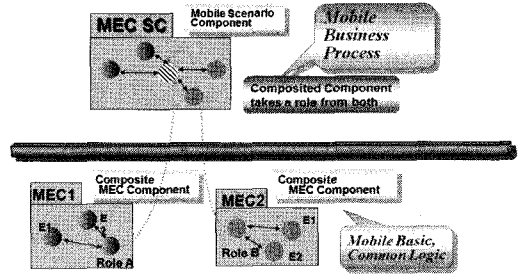
(그림 3) 컴포넌트 기반 모바일 응용 개발 방법론

이러한 프로세스는 컴포넌트 저장소를 중심으로 컴포넌트와 관련된 기본적인 행위들 즉, 컴포넌트 아키텍처, 명세, 검색, 적용 및 조립에 의한 시스템 전개라는 CBD 행위들을 말한다.

2.3 MEC (모바일 임베디드 컴포넌트)

MEC 컴포넌트는 그림 4에서와 같이 실행 가능하며 다른 컴포넌트와의 인터페이스를 제공하는

가장 작은 단위의 모바일 응용 실행 객체 클래스들(E1 and E2)의 조합으로 모바일에 적극 재사용 가능한 임베디드된 컴포넌트(MEC1 and MEC2)로 정의한다.



(그림 4) 모바일 임베디드 컴포넌트(MEC) 정의 및 기능

MEC는 효과적인 개발 환경 지원을 통해 모바일 임베디드 개발 전 단계 동안 사용된다. 이러한 MEC 접근을 통해 재구축 또는 변경 없이 또 다른 컴포넌트를 서브 클래스 함으로써 상세화 할 수 있으며, 기술적인 바인딩을 통해 다른 플랫폼으로 이식이 이루어지며, 메시징내의 메타데이터를 사용하여 클라이언트의 변화 가능한 요구사항과는 독립적으로 인터페이스가 개발되어지므로 최소화된 의존성을 가진다[12,1,3,14]. 따라서, 모바일 비즈니스 도메인의 MEC 측면에서 가장 기본적인 면서 모든 비즈니스 컴포넌트의 주요 요소인 MEC 를 정확히 명세함으로써 명확하고 용이하게 모바일 비즈니스 컴포넌트 시스템을 구축할 수 있으며, 기존의 컴포넌트의 재사용성이 용이하고 비용절감과 모바일 비즈니스 업무 처리 서비스를 폭 넓게 확장시킬 수 있다[15].

2.4 MEC 명세

MEC 명세는 모바일 임베디드 컴포넌트 아키텍처를 기반으로 컴포넌트의 개발, 관리, 사용을 위한 기초를 제공한다. 즉, 컴포넌트의 기능 및 서비스 구현시의 행위적인 정규 서술로서 조립을 위한

일반적인 메소드를 제공하고 수평적, 수직적 위상을 확장한다. 또한 CBD로의 전개를 위한 기초적인 세부 작업을 확장한다. 또한 MEC개발과 사용을 위한 유일한 식별 수단이다. 따라서 의미적으로 정규화된 명세를 갖는 MEC 만이 의미를 가진다.

현재 제시된 MEC 명세 방법은 크게 인터페이스 기술 방법에 따라 표 1과 같이 구분된다 [10,13].

(표 1) MEC 컴포넌트 인터페이스 서술 방법

기법	설명
Contract	<ul style="list-style-type: none"> · Server_Broker_Vender 간의 약속 명시 ①기본약정 : OOP, IDL로 Operation, I/O 인자, 예외 명시 ②행위약정 : 전/후/불변조건 assertion으로 행위 검증 ③동기화약정 : 메소드 사이의 동기화 고려한 행위 기술 <ul style="list-style-type: none"> - 컴포넌트 사이의 순차성, 병행성, 혼합성 관계 서술 ④서비스 품질 약정 : 행위들의 정량화
Modeling	<ul style="list-style-type: none"> · UML 기반의 컴포넌트 정의 - 기존 모델링 기술에 스테레오 타입, 인터페이스클래스, 실현 관계 추가하여 표현 - 패키지, 클래스, 시퀀스, 컴포넌트 다이어그램
Formal Spec.	<ul style="list-style-type: none"> · 아키텍처 정의 언어, 프레임워크 정의 언어로 작성 - Wright, COM, CORBA, Z 언어 - 시스템 통합의 불일치 제거, 명세 검증

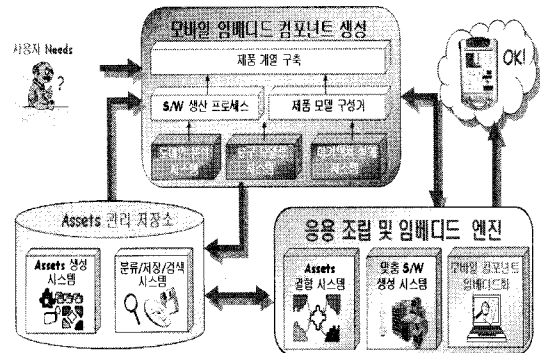
2.5 MEC 생성

MEC 생성과 확보는 모바일 비즈니스 도메인에 적합한 CBD 구축을 위한 가장 선행되어야 할 작업이다. MEC 생성을 위한 접근은 1)특정 영역 전문가에 의한 기능적 컴포넌트 추출과 2)도메인 분석을 위해 객체지향 방식의 도입에 의한 MEC 식별, 그리고 3)기존 시스템의 역 공학을 통한 식별 세 가지로 구분된다. 어떤 접근 방법에 의한 MEC 생성이든지 MEC의 pre-, post-condition에 의해 액세스 가능한 오퍼레이션의 집합인 인터페이스가 명확히 정의된 컴포넌트 명세의 정규화를 통한 컴포넌트의 공개와 보급이 필수적이다.

3. MEC 프레임워크 개발

3.1 개요

본 논문에서는 MEC 기반 모바일 임베디드 소프트웨어 개발을 위해 MEC 프레임워크 구축을 통한 모바일 임베디드 시스템 생성의 자동화로 정의한다. 모바일 관련 표준 하부 구조를 정의하고 소프트웨어 전개 모델을 제공함으로써 MEC 생성과 사용, 평가를 위한 근거를 확보하기 위해 MEC 아키텍처 모델을 정의와 컴포넌트의 기능 및 서비스 구현시의 행위적인 정규 서술과 조립을 위한 일반적인 메소드를 제공하는 MEC 명세 방법을 제시한다. 그리고 최종 MEC 프레임워크 구축을 위한 체계적인 접근을 시도한다.



(그림 5) 제안 프레임워크 시스템 구성도

그림 5는 본 논문에서 제안하는 MEC 프레임워크의 전체 시스템 구조이다. 본 프레임워크는 크게 모바일 임베디드 컴포넌트(MEC) 생성, 모바일 임베디드 자산 관리 저장소, 그리고 모바일 컴포넌트 응용 조립 및 임베디드 실행 엔진으로 구성된다.

3.2 MEC-ABCD 컴포넌트 아키텍처

(1) 아키텍처 생성을 위한 요구 사항

모바일 임베디드 지원 컴포넌트 아키텍처는 관련된 다른 종류의 컴포넌트들을 연관시키기 위한 표준 계층으로 MEC의 획득, 이해, 조립을 위한 레이아웃을 제시함으로써 사용자들이 필요로 하는 MEC 컴포넌트들을 식별하고, 검색하며 커스터마이징할 수 있는 가이드라인을 제공해야한다. 따라서 현재의 시스템 개발 환경과 관련하여 모바일 컴퓨팅을 위한 멀티 벤더의 멀티 솔루션 통합에 초점을 둔다. 또한 기본 인터페이스를 공유하고 하위 계층 컴포넌트 조합에 의한 상위 계층의 독립적 응용/그룹화된 컴포넌트 형성하여 scope와 Abstraction에 따른 수평적 컴포넌트 계층과 동일 영역의 Granularity에 따른 수직적 컴포넌트의 범주를 형성한다. 동일 계층의 컴포넌트 조립 메커니즘 제공하여 다음과 같은 3개 계층으로 구성한다.

- 하부 계층은 미들웨어 역할을 하는 API 및 확장 가능하며 응용의 하부구조, 분산 객체 서비스를 위한 기본 포맷을 제공
- 중간 계층은 많은 어플리케이션에서 필요로 하는 객체(기능) 들을 지원하기 위한 독립적인 컴포넌트들로 다중 도메인 상의 일반화된 메커니즘을 지원
- 최상위 계층은 Vertical 도메인을 위한 특정 응용 프로세스 컴포넌트로 구성

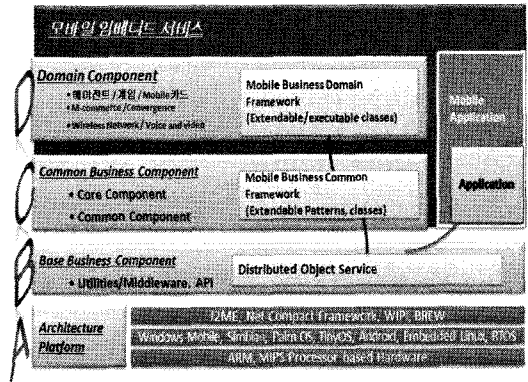
(2) MEC-ABCD 컴포넌트 아키텍처의 특징

본 논문에서는 모바일 컴퓨팅 환경 하에서 임베디드 비즈니스 솔루션을 위한 표준 모델로서 MEC-ABCD 컴포넌트 아키텍처를 제시한다. 본 아키텍처는 멀티 벤더/멀티 솔루션의 통합을 위해 컴포넌트의 스코프와 추상성, 입자성을 기준으로 계층적 분류를 시도했다.

그림 6은 MEC-ABCD 아키텍처이다. 전체 4개의 계층으로 구성되며 각 계층의 특징은 표 2와 같이 정의된다[10].

A 계층은 모바일 아키텍처 플랫폼과 OS 지원 층이 B 계층은 미들웨어 역할의 API와 분산 객체

서비스를 위한 기본 포맷을 제공한다. 따라서 계층 B에는 분산 MEC 객체 컴포넌트들이 포함된다. 계층 C는 많은 모바일 응용에서 요구되는 기



(그림 6) MEC-ABCD 아키텍처 모델

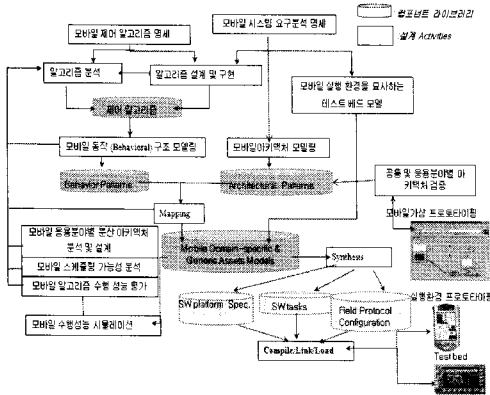
능들을 지원하기 위해 다중 도메인 상의 일반화된 메커니즘을 지원한다. 이 계층은 비즈니스 응용의 핵심 MEC 컴포넌트로서 공통 컴포넌트 영역을 두 개 계층으로 구분한다. 여기서 CBD를 위한 풍부한 조립 자원을 제공하고 비즈니스 프레임워크 구축을 위해 커스터마이징 가능한 패턴을 포함하여 컴포넌트의 재사용 스코프를 확장한다. 그리고 계층 D는 모바일 수직적 도메인을 위한 특정 응용 컴포넌트이다. 특히, 비즈니스 영역에 내에서 카테고리 되어진 컴포넌트 집합을 제공함으로써 조립시의 부가 비용을 축소화한다.

3.3 MEC 개발 프로세스

모바일 임베디드 응용 시스템을 구축하기 위한 상세 프로세스를 본 논문에서는 그림 7과 같이 제안한다.

모바일 시스템요구명세를 통해 아키텍처 모델링을 수행하며 제어 알고리즘 명세에서 행위 (behavior) 구조 모델링을 수행한다. 이들 모델링 정보는 모바일 도메인 특징과 지네릭(generic) 자

산을 포함하고 있는 저장소를 통해 모바일 플랫폼 명세와 잘 정의된 인터페이스를 통해 조립된다. 조립된 모바일 응용은 가상 프로토타이핑과 실 환경 프로토타이핑으로 분리되어 실행하여 개발자가 검증하도록 한다. 이와 병행해서 실행환경 테스트 베드도 같이 확인 검증된다.



(그림 7) MEC 응용 개발을 위한 프로세스

3.3.1 MEC 모바일 시스템 요구사항 식별

MEC 도메인과 사용자, 시스템, 기반환경 등을 고려하여 문제 영역을 분석하고 요구사항 시나리오를 작성함으로써, 개발 목표를 명확히 정의한다. 사용자의 관점에서 시스템이 어떻게 동작하는지 결정하여 전체 시스템 예측 활동은 스토리보딩 기법을 사용하여 요구사항 시나리오를 작성한다. 작성된 요구사항 시나리오를 기반으로 사용사례를 식별하고, UML의 Use case Diagram을 작성한다. 이는 목표 시스템의 영역을 정의하고 전체 업무 영역 중에 어느 업무 기능을 책임질지 명확히 하는 것이며, 행위자와 기능간의 관계 및 부가적인 설명이 된다. 요구사항 식별단계에서는 후보가 될 수 있는 MEC에 대한 이해와 MEC를 기반으로 구성되는 기능적인 요소나 시스템 전체의 구성 파악이 주 목적이다. 여기서 MEC는 단일의 MEC이거나 조합 가능한 집합적인 MEC의 대표일 수 있다.

3.3.2 MEC 명세 개발

식별된 요구사항을 기반으로 개발될 MEC에 대한 식별을 통해 MEC 아키텍처 모델과 각각의 MEC에 대한 명세 작성을 주요 단계로 구성된다. 이 단계를 통해 얻은 부산물은 구성될 컴포넌트의 식별 및 새로운 컴포넌트의 개발에 주요한 자원이 된다.

MEC 명세 개발은 컴포넌트 참조 아키텍처를 기반으로 한다. 컴포넌트 참조 아키텍처에서 MEC의 영역을 식별하고 식별된 참조 아키텍처 부분에 해당하는 MEC간의 연관성을 정의한다. 정의된 MEC는 각각의 기능 및 특징을 중심으로 명세를 하게 된다. 이 명세 과정을 통해 MEC를 개발하기 위한 적절한 컴포넌트의 구성이 가능하다. 사용자의 요구에 기반한 MEC 명세는 기본 명세와 4-3개의 모델을 생성한다. 이는 명세단계에 획득되며, 새로운 컴포넌트를 식별하고 개발하기 위한 주요 자원으로 사용된다. MEC 모델은 더 높은 수준의 표현을 제공하기 위해 UML 메타모델로 확장된다.

① MEC 식별 및 역할 모델 생성

각각의 MEC 역할에 초점을 두고 특징을 표현할 수 있도록 다이어그램을 생성한다. 행위규칙에 따라 동작하는 작업이나 자원의 제어, 어떤 이벤트에 반응하는지 등을 중심으로 식별하게 된다. 요구사항을 통해 얻어진 MEC의 기본적인 요구사항을 통해 MEC의 CBD 참조 모델에 어느 영역에 위치하는지를 식별한다. 또한, MEC의 이름을 MEC의 사용 사례 기술서를 참조하여 명명하며, 이는 MEC 명세 아키텍처 모델과 스펙 생성에서 목표 MEC로 정의한다.

② MEC 목표 모델 생성

본 모델은 MEC간의 목표, 태스크, 상태와 의존성들을 표현한다. 목표와 태스크는 상태와 함께 연관되며 논리적인 의존성에 의해 연결될 수 있다. 상위의 목표는 하위의 서브목표의 집합이 달

성됨으로써 충족될 수 있으며, 이러한 하위 서브 목표를 식별하고 모델로 제시함으로써 MEC 시스템을 이해하고 분석 가능하다. 또한, 태스크는 목표를 달성하기 위해 수행되며, UML 액티비티 다이어그램으로 표현 가능하다.

③ MEC 상호작용 모델 생성

설계단계에서 어떻게 통신하는지를 상세하게 모두 나타내는 것으로 어느 MEC와 어떻게, 언제 통신이 이루어지는지를 MEC와 역할을 중심으로 표현한다. 상호작용 모델은 새로운 상호작용이 발견되기 전까지 몇몇의 상호작용을 통해서 정련화된다. 이 모델에서는 상호작용을 중심으로 각각의 참여로 달성되고 제공되는 트리거 조건과 정보처럼 선택적인 정보에 상호작용 가능한 initiator, responder, motivator를 표현한다.

④ MEC 아키텍처 모델 생성

식별된 MEC는 각각의 기능에 따라 MEC 타입에 대응되며, 구별될 수 있는 특징을 가진다. 동일한 MEC 타입은 목적에 따라 다른 MEC와 협동이나 협상 등의 기능을 수행한다. 이러한 다른 MEC와의 관계 및 상호 작용하는 동적인 성질을 명세하기 위한 부분이 MEC 아키텍처 모델을 생성하는 것이다. 본 논문에서는 비즈니스 사용자와의 관계 및 식별된 MEC를 MEC 영역에 정의하고 각 MEC간의 관계를 MEC 아키텍처 모델로 제시한다. MEC 영역은 두 형태로 나뉘는데, 다른 MEC와 통신하거나 가용성을 알리기 위한 영역과 사용자와 상호작용하기 위한 사용자 영역으로 나뉜다.

3.3.3 MEC 기본 명세

본 논문에서는 기존에 제시되어진 컴포넌트 명세 기법에 새롭게 요구되는 명세 특성들을 포함하여 모바일 컴포넌트 명세를 작성한다. 이는 컴포넌트 개발을 위한 상세 명세로서 응용으로의 전개 시 조립을 위한 명확한 의미적인 플러깅 지점을

확보하고 비즈니스 프로세스의 계층적 실현을 위해 명세 항목을 정의한다. 명세 항목은 컴포넌트 구현을 위해 명세된 것이며 식별될 수 있는 컴포넌트의 카테고리, 이름, 식별 코드 등을 작성한다. 명세는 컴포넌트 개발과 컴포넌트의 획득을 위해 작성되는 가이드라인의 역할을 한다. 구현단계에서는 Component Diagram 항목에 Invariant와 Variant, Exception으로 인터페이스를 명확히 한다. 또한, Usage Scenario 항목은 카테고리 내의 컴포넌트들의 이용 절차를 예시한 것으로 조립을 위한 확신된 가이드라인으로서 이용한다. Qualities Attribute에는 타입, 언어, 컨테이너, 데이터베이스, 미들웨어 등의 구현적 선택과 성능, 보안, 플랫폼 관점에서 준수해야 하는 요소들이 나열된다.

기존에 제시되어진 컴포넌트 명세 기법에 새롭게 요구되는 명세 특성들을 포함하고 ABCD 아키텍처에 기반한 특징을 포함하여 컴포넌트 명세를 작성한다. 이는 컴포넌트 개발을 위한 상세 명세로서 전개 시 조립을 위한 명확한 의미적인 플러깅 지점을 확보하고 비즈니스 프로세스의 계층적 실현을 위해 표 2와 같은 명세 항목으로 정의한다. 제시된 명세 항목은 컴포넌트 구현을 위해 명세된 것이지만, 기본 명세 단계에서는 음영으로 된 부분을 작성한다. 이는 식별될 수 있는 컴포넌트의 카테고리, 이름, 식별 코드 등을 작성한다.

명세는 컴포넌트 개발과 컴포넌트의 획득을 위해 작성되는 가이드라인의 역할을 한다. 구현단계에서는 컴포넌트 다이어그램 항목에 Invariant와 Variant, Exception으로 인터페이스를 명확히 한다. 또한, Usage Scenario 항목은 카테고리 내의 컴포넌트들의 이용 절차를 예시한 것으로 조립을 위한 확신된 가이드라인으로서 이용한다.

3.3.4 MEC 적용

명세된 컴포넌트가 실행 환경에서 동작할 수 있도록 기반 환경에 기능을 포함한 프로그래밍이 요구된다. 즉, MEC 컴포넌트와 애플리케이션이

(표 2) MEC 컴포넌트 명세 목록

Items	Description
M.Category	모바일 컴포넌트가 속한 비즈니스 도메인의 세부 기능적인 분류에 따른 모바일 컴포넌트 군
M.Component Diagram	인터페이스를 포함한 모바일 컴포넌트의 정의
M.Component Name	식별된 모바일 컴포넌트 이름
M.Classification Code	ABCD Architecture에 기반한 모바일 컴포넌트의 식별 코드
M.Short Description	모바일 컴포넌트 전반에 대한 기능, 동기, 동작 과정, 제약사항 등에 대한 서술
M.Glossary	모바일 컴포넌트 명세에 사용된 용어의 의미 설명
M.Component Context Diagram	모바일 컴포넌트의 주요 기능을 서술
M.Component Interaction Diagram	모바일 컴포넌트 실행시 관련된 모바일 컴포넌트들 간의 관계성을 표현
M.Component Sequence Diagram	모바일 컴포넌트 자체의 행위적 순서 표현
M.Component Relation Diagram	모바일 컴포넌트가 제공하고 제공받는 인터페이스와 다른 모바일 컴포넌트간의 관계를 표현
M.Interface Description	공급하고 요구하는 인터페이스의 의미적 서술, pre/post-condition, 입력/출력 결과 명시
M.Usage Scenario	모바일 컴포넌트 사용을 위한 인증된 시나리오

표준 분산 환경에서 컴포넌트 명세를 실체화해야 하며, 대응되는 MEC 인터페이스는 인터페이스 타입을 실체화해야 한다. 적용 단계의 첫 번째인 준비과정에는 명세의 실체화 과정에서 분산 객체 환경에 컴포넌트를 등록하는 과정이 포함된다. 표준 컴포넌트 환경으로는 마이크로소프트의 COM+나 Enterprise Java Bean, CORBA 등이 있으며, 컴포넌트의 통합 환경 및 컨테이너의 역할을 하게 된다. 등록된 컴포넌트는 목표시스템의 일부 혹은 전체로 구성하기 위해 조립을 하게 된다. 조립은 컴포넌트와 기존 소프트웨어와의 결합과 컴포넌트간의 결합을 통해 실행 시스템을 구성하고 사용자 인터페이스를 설계하여 애플리케이션을 만들어내는 과정이다. 또한, 사용자 인터페이스와 식별된 MEC 응용시스템에서의 MEC 기능을 확인하고 비즈니스 로직을 정의해야 한다. 조립은 프로그래밍 레벨에

서 표준 환경에서 제공하는 도구를 통해 쉽게 구축 가능하며, 부분적인 기능 추가나 애플리케이션 형태로 출력된다. 조립을 통해 구성된 애플리케이션은 테스트 후 배포된다.

4. MEC 명세사례

4.1 모바일 메시지 송/수신 MEC

본 논문에서는 모바일 메시징용 시스템을 개발을 위해 필요한 MEC의 명세를 작성하였다. 모바일 메시지 관리 영역은 모든 비즈니스 영역의 필수적인 영역으로 그 적용 범위가 광대함에 따라 공통 MEC로서 큰 가치를 가진다. MEC 식별을 위한 도메인 분석에 의한 순공학적 접근을 시도하였다. 표 3은 모바일 메시지 관리 카테고리 중 E-Mail 프로세스 수행을 위해 식별된 “모바일 메시지 송/수신 관리” MEC의 명세이다.

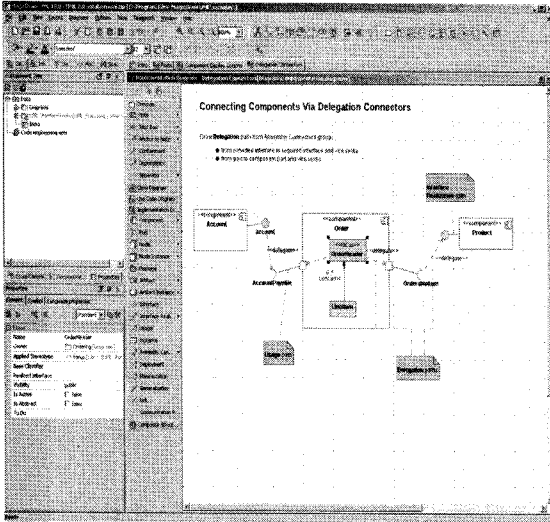
4.2 MEC 개발 지원 프레임워크

모바일 임베디드 소프트웨어 개발은 MEC 인터페이스를 통해 이루어진다. 이는 복수 개의 클래스에 의해 구현되는 MEC 제공 서비스로 MEC 내부 상태에 변화를 줄 수 있는 속성 인터페이스와 제공되는 서비스를 호출하는 함수 인터페이스 그리고 이벤트 인터페이스로 구성되며 인터페이스의 수를 MEC 입자성(granularity) 구분의 기준으로 사용하였다. 만약 요청 인터페이스만이 존재한다면 연관된 상위 MEC로 통합하며, 제공하는 인터페이스가 많을 경우에는 동일 계층의 다른 컴포넌트와 비슷한 수준의 인터페이스를 갖도록 컴포넌트를 분할 명세화한다 [10].

(표 3) 모바일 메시지 송/수신 MEC의 명세 예

(1) 모바일 메시지 관리(Mobile Message Management) MEC 카테고리(MEC-CO)	
High Component	Element Component
계정 관리(02)	접속 관리(01), 인증(02)
공기 사항(03)	공지사항 권한검사(01), 공지사항 관리(02)
모바일 메시지 관리(04)	모바일 메시지 송/수신 관리(01), 모바일 메시지 컴포저(02)
주소록 (05)	주소록 관리(01), 주소록 퍼실리티(02), 주소록 인증(03)
음성 동영상 자료실 관리(06)	파일송/수신(01), 데이터관리(02), 자료실퍼실리티(02), 무결성 검사(04)
프로토콜 관리(07)	SNMP 기능(01), CGI 기능(02), FTP 기능(03)
기타(01)	메뉴 관리(01), 시나리오(02), 데이터 관리(03)
"비고"	데이터 관리 : 데이터를 저장, 갱신, 수정, 삭제 시나리오 : 카테고리 MEC의 동작 시나리오 정의
(2) 모바일 메시지 관리 MEC 상관도	
(3) 이름 : 모바일 메시지 송/수신 관리	(4) 분류 코드 : MEC-CO-04-02
(5) 개요 : 모바일 메시지 전송 관리 MEC는 사용자가 보내는 모바일 메시지 내용을 전자모바일 메시지의 헤더를 보고 지정된 호스트의 메시지_processing에 메시지를 전달하는 client_메시지_processing과 Receiver-SMTP를 통해 받아들여진 모바일 사용자 메시지 box에 보내는 server_메시지_processing로 구성된다. SMTP는 송신측의 Sender-SMTP와 수신측의 Receiver-SMTP로 이루어져 있으며 사용자 전송요구가 client_메시지_processing에 도착하면 모바일 메시지는 목적지 주소를 번역하고 메시지 Address를 전송프로그램 포맷으로 전환한다. Sender-SMTP는 Receiver-SMTP와 양방향 전송 채널을 설립하고 선택된 전송프로그램은 모바일 메시지를 전달한다.Receiver-SMTP는 OK 또는 reject로 응답하게 되는데 OK응답이 오면 설립된 전송채널을 통해 메시지를 전송하며 주소를 해석하여 사용자 모바일 메시지박스나 다른 시스템에 포워드 한다.	
(6) 용어사전	
message_Queue : 사용자 메시지의 송신을 위한 임시 저장 공간 message_Processing : 메시지 송수신 관련 일련의 작업수행 프로세스 SMTP : 모바일 메시지 전송을 위해 사용되는 프로토콜 User 메시지 Box : 사용자 메시지 수신을 위한 저장 공간	
(7)Component Context Diagram	(8)Component Interaction Diagram
"비고" : 메시지 전송자는 모바일 메시지 수신자가 될 수 있으며 모바일 메시지 전송 프로세스는 양방향 모바일 메시지 송수신 프로세스 담당	

(9) MEC Component Diagram	(10)Component Sequence Diagram
"비고" : Invariant 메시지 주소 형식 signaling과 전송 · 수신 순서 Variability : 전송 채널 설정	
(11) 인터페이스 명세	
.모바일 메시지 정보 조회 서비스	
Provide Interface 모바일 메시지 정보 조회 [to] 사용자 Description : 모바일 메시지 송수신 관리가 사용자에게 모바일 메시지 정보를 제공 [Preconditions : 모바일 메시지 관리가 message Box에 저장되어야 한다.] [Postcondition : 새로운 정보가 있음을 표시(확인된 정보임을 표시)]	
사용자의 특정 메타데이터 저장	
Require Interface 사용자의 특정 메타데이터 저장 [to] 데이터 관리 Description : 사용자에게 관한 메타데이터를 저장하고 삭제 [Preconditions : 사용자에게 관한 메타데이터를 추출해야 한다.] [Postconditions : 저장된 메타데이터를 사용자 정보 조회에 활용.] Input : 사용자 ID. /Output : 사용자 ID에 따른 메타데이터.	
.모바일 메시지 사용자 인증 호출	
Require Interface 모바일 메시지 사용자 인증 호출 [to] 인증 Description : 사용자의 ID와 패스워드를 통해 인증을 확인 [Preconditions : ID와 패스워드가 등록되어져 있어야 한다.] [Postconditions : 모바일 메시지 작성이 가능하다.] Input : 사용자 ID와 패스워드. /Output : 인증됨, 비인증됨	
.모바일 메시지 컴포저 서비스 호출	
Require Interface 모바일 메시지 컴포저 서비스 호출 [to] 모바일 메시지 컴포저 Description : 모바일 메시지 작성과 File 컴부등 모바일 메시지에 관련된 서비스를 이용 [Postconditions : 모바일 메시지가 Local Server에 저장되어야 한다.] Input : 문서 형식의 모바일 메시지. /Output : 정규화된 모바일 메시지 형식.	
.모바일 메시지 전송 프로토콜 호출	
Require Interface 모바일 메시지 전송 프로토콜 호출 [to] SMTP 기능 Description : Mail Processing간 메시지를 Send/Receive 할때 사용 [Preconditions : 모바일 메시지 Queue에 모바일 메시지가 등록되어야 한다.] [Postconditions : 사용자의 수신 모바일 메시지 BOX나 다른 시스템에 포워드 Input : 모바일 메시지 형식으로 저장된 mail, signal. Output : signal, mail형식으로 전송된 수신 mail.	
(13) Usage Scenario	(14) Quality Attribute
모바일 메시지 전송의 경우 - 모바일 메시지 컴포저 MEC의 모바일 메시지 컴포저 서비스를 획득한다. - 작성한 모바일 메시지의 전송 전에 이 모바일 메시지에 관한 사용자의 특정 메타 데이터를 데이터 관리 컴포넌트를 이용, 저장 - 실제 Destination 주소를 보내기 위해 SMTP기능 컴포넌트로부터 모바일 메시지 전송 프로토콜을 획득 - 사용자 모바일 메시지가 전송 모바일 메시지 송신의 경우 - 인증 컴포넌트의 사용자 인증 기능의 획득과 모바일 메시지 관리 컴포넌트의 모바일 메시지 정보 조회기능으로 수신된 모바일 메시지를 인증됨 또는 비인증된 처리를 결정 - 인증된 경우 모바일 메시지 관리 MEC가 사용자의 모바일 메시지를 관리 - 비인증된 경우 모바일 메시지를 전송자로 되돌려보낸다.	Environment ; Implementation Choice - Type : JavaBeans, WFC - Language : Java - Tool : MS Visual Studio - Container : MS Office/Repository - Company/Database - Orbx, Visibroker Technology ; Architecture platform - Fundamental Component Model : EJB - Technical Architecture : Client/Server, 3-rier Performance 모바일 메시지 전송시 시간이 30초이상의 초과시에 타임 아웃이 되어 모바일 메시지를 리턴 모바일 메시지가 리턴될 시에 2시간 간격으로 모바일 메시지를 재전송(3 번)하고 취소 인증이 확인되지 않을 시에는 에러 메시지를 출력

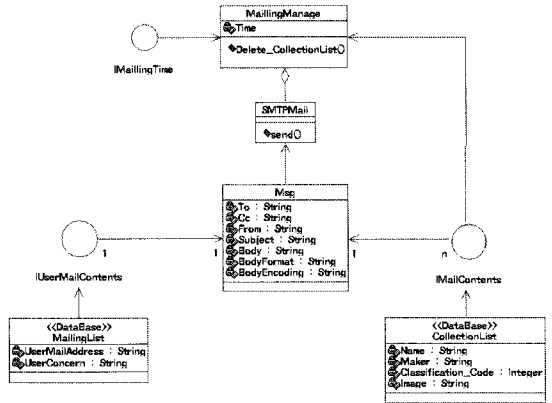


(그림 7) 본 논문에서 제안한 MEC 개발 지원 프레임워크 screen

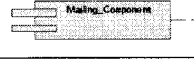
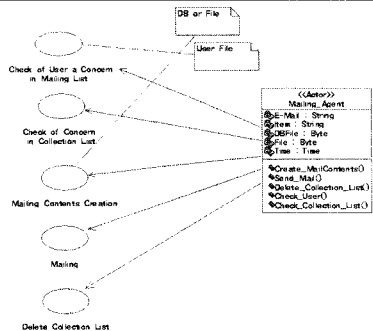
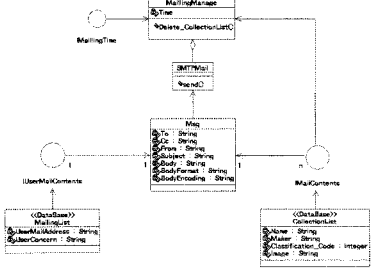
모바일 임베디드 서비스를 위한 환경을 지원하고 개발이 용이하도록 MEC 개발을 고려한 사용자 인터페이스를 그림 7과 같이 정의, 구현 하였다. 이는 사용자 측면과 MEC의 기능을 구현하기 위한 측면 그리고 사용자 서비스를 지원하기 위한 데이터 접근 측면을 고려하였다. 특히 컴포넌트와 모바일 MEC를 위해 컴포넌트 계층, MEC 계층을 각각 분리하여 MEC의 기능은 컴포넌트 계층에서 구현되며, MEC의 관리는 MEC 계층에서 이루어진다.

4.3 프레임워크를 적용한 MEC 조립 예

생성된 MEC 명세를 실체화하기 위해 MEC 기반 프레임워크 상에서 실행 가능한 모바일 메시지 MEC 컴포넌트를 구현하였다. 그림 8은 모바일 메시지 MEC 간의 인터페이스를 식별하는 UML 그림이며 그림 9는 모바일 메시지 처리를 위한 명세 작성 예 이다.



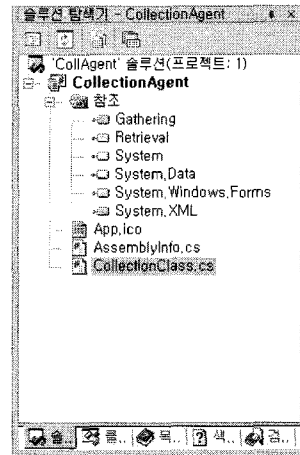
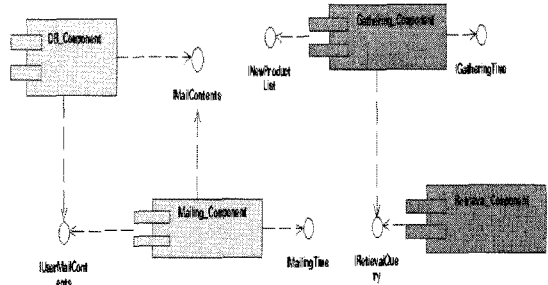
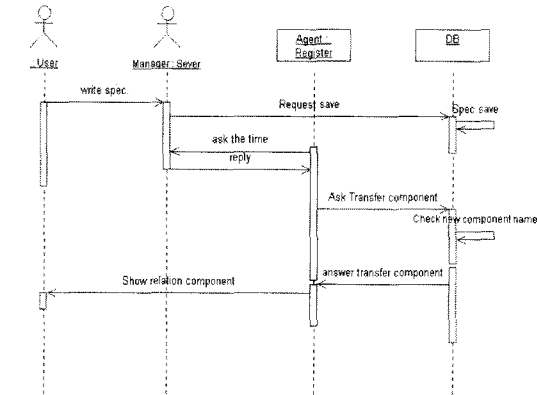
(그림 8) 모바일 메시지 MEC 인터페이스 식별

Item	Description
Category	E-Business Agent
Component Diagram	
Component Name	Mailing Component
Classification Code	EBA.00.01.01
Short Description	해당되는 정보를 DB에서 가져와서 사용자에게 메일을 전송한다.
Component Context Diagram	
Component Interaction Diagram	

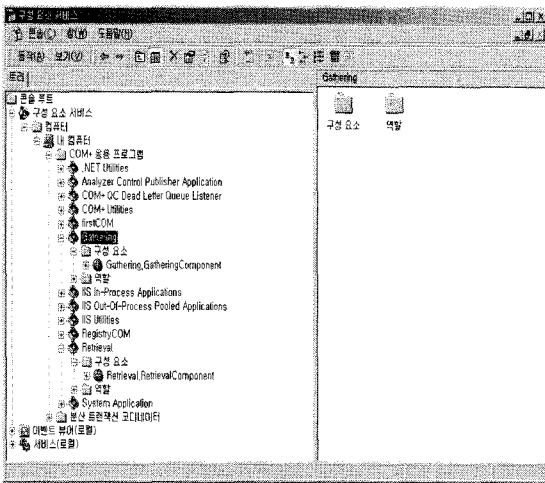
(그림 8) 모바일 메시지 처리 MEC 명세

그림 10은 모바일 메시지 처리를 위한 프레임 환경에 등록된 MEC 이며 모바일 메시지를 구성

하기 위해 수집 및 검색된 MEC를 그림 11과 같이 조립한다.



(그림 11) 모바일 메시지 수집 및 검색 MEC 조립



(그림 10) 프레임워크 환경에 등록된 메시지 수집 MEC

4.4. 제안 MEC 프레임워크 특징 및 평가

현재, MEC 기반 임베디드 소프트웨어 개발을 위한 MEC 식별에서 표현 기호를 이용한 명세화, 저장소에서의 관리 및 응용으로의 전개를 지원하는 도구 등의 모바일 임베디드 소프트웨어 개발 방법론에 대한 총체적인 평가 기준이 정립되지 않은 상태이다[10,11,13]. 따라서 어떤 명세가 명확하며 어떤 MEC의 도메인 비즈니스에 유용한지는

시스템 개발자의 주관적 판단에 의존해야 한다. 본 논문 역시 정량적인 논리적 평가 기준의 적용이 어려운 까닭에 MEC 개발과 사용 모두의 측면에서 상세하고 체계적 수준의 MEC 정보 제공을 원칙으로 직관적인 인식이 가능하도록 MEC 아키텍처의 기준을 정립하고 명세 항목을 결정했다.

현재 모바일 임베디드 컴포넌트 명세 작성과 저장 및 검색을 위한 체계적 연구는 없으며 본 연구에서 제안하는 MEC 프레임워크는 모바일 응용 시스템 개발에 용이하기 위해 MEC 컴포넌트를 사용자 측면과 관리 측면, 그리고 저장 측면으로 나누는 명세를 통해 MEC를 개발하여, 모바일 비즈니스 도메인에 적합하며 모바일 임베디드 컴포넌트로의 확장이 가능하다. 또한 사용자도 손쉽게 원하는 컴포넌트를 등록 및 수정할 수 있으며, 검

색을 통해 원하는 컴포넌트를 찾을 수 있고, 이러한 모든 기능은 MEC를 통해 이루어지므로, 사용자에게 좀 더 빠른 컴포넌트 정보 제공과 함께, 자동화된 기능을 통해 많은 노력과 시간을 절약할 수 있으며, 컴포넌트의 목적인 재 사용성을 극대화할 수 있다. 또한 멀티 MEC의 기능으로 타 시스템간의 정보 전달을 통해 모바일 컴포넌트 유통에도 용이하다.

5. 결론 및 향후연구

본 논문에서는 모바일 비즈니스 도메인에서 MEC (mobile Embedded Component)를 생성하기 위한 방법으로 모바일 임베디드 컴포넌트 구축을 지원 하는 프레임워크를 제시하였다. 사용자의 요구 사항에 최대한 접근하고 모바일 임베디드 도메인을 기반으로 둔 소규모 단위 분산 컴포넌트를 대상으로 모바일에 확장적 컴포넌트 계층화와 비즈니스 로직 확보를 위해 재구성 가능한 설계 패턴 및 컴포넌트 군(비즈니스 도메인 카테고리)을 형성하고 제공한다. 제안 하는 모바일 임베디드 컴포넌트 명세는 기존 명세가 가지는 비 일치성을 보완하여 컴포넌트의 개발과 사용에 실제적으로 활용할 수 있도록 정의한다. 모바일 비즈니스 프로세스를 위한 의미 지향적이며 모델링 기반 원칙에 따라 명확하고 풍부한 명세 정보를 포함한다. 본 논문에서 제안한 MEC 명세를 통해 사용자가 분석/설계된 모바일 임베디드 MEC 명세를 통해 비즈니스 측면의 컴포넌트 유통과 분산 환경을 통한 빠른 인터페이스, 그리고 컴포넌트 관리와 많은 사용자에게 쉽게 배포할 수 있는 MEC 등록과 검색을 관리를 위해 MEC를 도입함으로써, 자동적인 서비스 제공과 함께 사용자에게 빠르고 정확한 서비스도 사용자가 원하는 정보를 얻을 수 있는 통합형 시스템으로 제공되는 이점을 가진다.

향후연구로는 본 MEC 프레임워크의 명세 내용을 확대하고, 추가적인 MEC 명세를 통해 일관성

있고 완벽한 모바일 컴포넌트 조립으로 모바일 응용 애플리케이션 개발을 할 수 있는 통합 환경으로의 시스템으로 개발하며, MEC를 사용자 측면에서 좀 더 편리한 기능을 추가 할 것이다. 또한 본 제안 프레임워크를 통해 다양한 최근의 임베디드 소프트웨어 개발 기법등과 연계하여 모바일 컴포넌트 개발 생산성을 높이며 모바일 임베디드 컴포넌트 저장소에 대한 완벽한 설계구현을 필요로 한다.

<Acknowledgement>

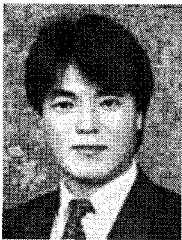
“이 논문은 2007-2008년 대구가톨릭대학교 교비해외파견 연구지원금에 의한 것임”

참고 문헌

- [1] Chairman's Report, 'Toward the Realization of a Ubiquitous Network Society', Ubiquitous Network Conference at Tokyo, 2005. <http://www.soumu.go.jp/s-news/2005/pdf/>
- [2] IT Strategic Headquarters, 'New IT Reform Strategy', 2006. <http://www.kantei.go.jp/jp/singi/it2/kettei/>
- [3] Katsumasa Shinozuka, 'Mobile Interactive Services: Risks and Solutions', Oki Technical Review, Vol.67, No.1, pp.3-8, 2007. <http://www.oki.com/jp/Home/JIS/Books/KENKAI/>
- [4] Katsumasa Shinozuka, 'Ubiquitous Security - Towards Realization of a Safe and Secure Digital World', Oki Technical Review Vol.74, No.2, pp.76-83, 2007.
- [5] I. Richardson, 'Mobile Technosoma: Some Phenomenological Reflections on Itinerant Media Devices', <http://journal.fibreulture.org/issue6/>
- [6] Columbus Project, 'Design of Embedded Controllers for Safety Critical Systems', <http://www.columbus.gr/innovation.htm>
- [7] R. Reussner, H. Schmidt, I. Poernomo,

- 'Reliability Prediction for Component-Based Software Architectures', Journal of Systems and Software, Vol.66, No.3, pp.241-252, 2003.
- [8] H.W.Schmidt, I.Peake,J.Xie, and etc., 'Modelling Predictable Component-Based Distributed Control Architectures', In Proceedings of the Ninth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, 2004.
- [9] H W Schmidt, B J Krämer, I Poernomo, and R Reussner, 'Predictable Component Architectures Using Dependent Finite State Machines', Lecture Notes in Computer Science, Vol.2941, pp.310-324, 2004.
- [10] 차정은, 컴포넌트 기반 개발 프로세스 지원을 위한 컴포넌트 저장소의 설계 및 구현, 대구 가톨릭대학교 대학원 전산통계학 전공 박사 학위청구논문, 2001년.
- [11] 임채덕 등, '임베디드 소프트웨어 기술동향 및 산업발전 전망', 정보통신연구진흥 제 4권 제3호, 정보통신연구진흥원.
- [12] Dayang N. A. Jawawi, Rosbi Mamat and Safaai Deris, 'A Component-Oriented Programming for Embedded Mobile Robot Software', International Journal of Advanced Robotic Systems, Vol.4, No.3, pp.371-380, 2007.
- [13] Haeng Kon Kim, Lee Kyeong Son, 'Product Line Development Process for Mobile Software based on Product Line', Journal of KIPS, Vol.12-D, No.3, pp.395-408, 2005.

● 저 자 소 개 ●



김 행 곤(Haeng-Kon Kim)

1985년 중앙대학교 전자계산학과 (공학사)

1987년 중앙대학교 대학원 전자계산학과 (공학석사)

1991년 중앙대학교 대학원 전자계산학과 (공학박사)

1978년~1979년 미 항공우주국 객원 연구원

1987년~1989년 한국전기통신공사 전임연구원

1988년~1989년 AT&T 객원 연구원

2000년~2001년 Central Michigan University 교환교수

2007년~2008년 미 SEITI 연구소 연구원

1990년~현재 대구가톨릭대학교 컴퓨터공학과 교수

관심분야 : 임베디드 소프트웨어공학, 모바일 컨버전스 소프트웨어 개발 방법론, CBSE, 소프트웨어 재공학, CASE, 유지보수 자동화 툴, 사용자 인터페이스 개발 공학, 요구공학 및 도메인 공학

E-mail : hangkon@cu.ac.kr