

병원 전사적 정보시스템 개발을 위한 Component 지원 도구

박 화 규*

요약

본 논문에서는 진화적 병원정보시스템 개발을 위한 컴포넌트 지향 방법론을 제안하였다. 또한 내외부의 환경 및 커뮤니티 목표에 따라 기회적이며 동적으로 변화하며 진화하는 병원정보의 특성을 수용하기 위해, 저장소 메타 스키마 구조를 제시하고 이에 기반한 컴포넌트 개발 및 유지보수 지원도구(C-Handler)를 제안하였다.

이를 통해 순차적으로 논리적인 단계에서 역할의 정보와 사상을 컴포넌트(Component)로 변환하여, 진화적 병원정보시스템 요구사항과 이에 따르는 유지보수에 컴포넌트 재사용성(Reusability)을 극대화 하여 대응함으로써, 개발 싸이클 기간을 줄이고, 시장성(市場性)을 향상시킬 수 있다. 구현 단계에서는 제안된 산출물 컴포넌트 명세를 기반으로 H-Handler에서 3가지 형태(Visual, Logical 및 Data Component)의 컴포넌트를 구현하고, 이들을 상호 결합함으로써, 병원정보시스템을 개발한다.

이는 의료조직의 소프트웨어 컴포넌트를 개발 및 관리하는데 필요한 메타 스키마를 기반으로 구현되었다. C-Handler는 웹 서버로써 개발되어 다수의 사용자가 상용화된 인터넷 브라우저를 통해 실시간 협업적 구현을 가능하게 하였으며, 주 사용자인 시스템 개발자 및 관리자의 병원정보관리 및 유지보수 활동을 효율적으로 지원 할 수 있다.

핵심주제어 : 병원정보시스템, 소프트웨어 컴포넌트, 보건의료조직

* 순천향대학교 의료과대학 보건행정경영학과, E-mail: hkpark1@sch.ac.kr

〈논문 투고일〉 2014. 09. 02 〈논문 수정일〉 2014. 09. 12 〈게재 확정일〉 2014. 09. 22

I. 서 론

최근 대형의료기관의 병원정보시스템(HIS : Hospital Information System)은 Management Service Organization(MSO), Medical Tourism 중심 Network 병의원연계, Tele-medicine, u-healthcare 및 의료조직의 해외진출 등 보건의료조직의 내외부의 요구사항과 관련제도에 따라 동적으로 변화특성이 있으므로(Agarwal et al., 2010) 이에 따른 Functional Requirements 수용에 대한 Dynamic Re-customizing이 필수적이고, 이를 위해서는 합리적 여러 형태의 모델링 표현력을 갖고, 해당 Domain에 빠르게 Modification이 가능한 Component 방법론을 활용하여야 한다. 이러한 HIS Component 지원 방법론은 의료기관별 전사적 병원자원관리의 강건한 구축을 위한 중요한 도구가 되며 다양한 보건의료 업무영역에 대하여 높은 모델링가능성을 갖고 업무중심의 접근에 대한 적용원리가 합리적이고 유연하며, Computer-Aided System Engineering 도구와 체계적 문서화를 위한 지원도구가 지원됨으로써 급변하는 요구사항 변화에 효율적이다. 대형병원에서의 HIS를 구성하는 가장 기본적 구성요소는 다음과 같이 분류되 모델링되어질 수 있어야 한다.

1. EMR(Electronic Medical Record)

다양한 의학 지식에 기초한 기억보조와 의사결정보조도구 등을 이용 한 Expert System으로 User지원하는 체계 위에 세워진 Electronic 의무기록으로, 디지털화를 통하여 Paperless 환경을 구축할 수 있으며, Multi-media 기반의 음성, 영상, Text 등 Digital Data를 Control.

2. OCS(Order Communication System)

OCS 의사의 처방정보를 전산망을 이용하고 활용함으로써 보건의료조직내부의 Communications와 Information Sharing을 Real-time화함으로 Medical Service Organization의 Efficiency와 Service Enhancement를 제고.

3. PACS(Picture Archiving and Communication System)

PACS는 각종 의료 영상 Modality로 촬영한 의료영상들을 디지털화하여 하드디스크와 같은 저장 매체에 저장, Network를 통해 각 단말기로 전송하여 진찰실, 병동 등에서 환자의 영상을 조회할 수 있는 Sub-System 기능(Wu et al, 2011).

4. PRM(Patient Relationship Management)

의료조직의 Customer 특성에 기반을 둔 Marketing 활동을 계획하고, 지원하며, 평가하며, 전략수립과 경쟁력 강화를 위해, 여러 Channel을 통해 발생하는 고객 정보의 종합적이고 신속한 분석을 바탕으로 한 Marketing 체계 일부로, Datawarehouse 기반으로 진단 및 치료중심에서 개인의 유전정보와 건강행태를 기반으로 하여 개인의 질환 발생을 예측 한 후 고위험도 집단 예방 기능을 포함. 예측 및 관리모형으로 변화함에 따라 환자들의 유전정보, 건강행태 정보 등을 활용하여 질병 발생 확률을 예측하여주고 개인들이 스스로 건강관리를 할 수 있도록 해주는 새로운 형태의 기능 지원.

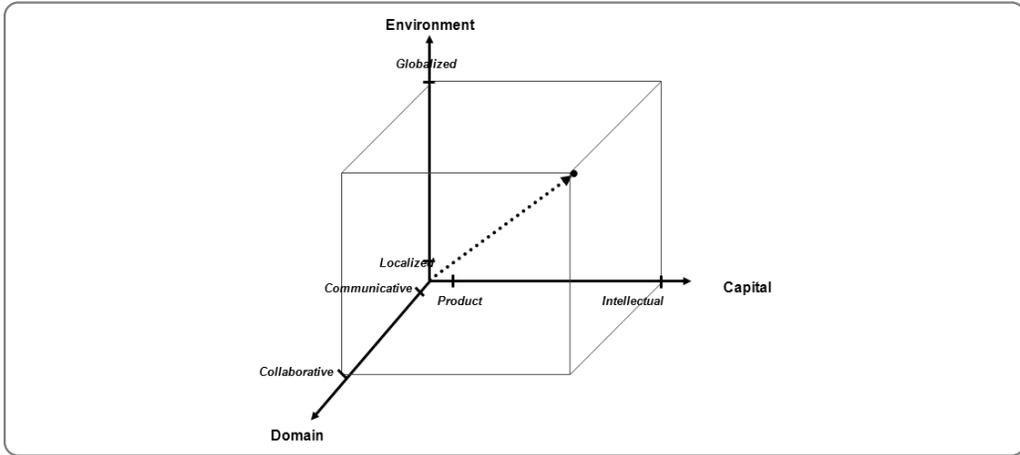
5. MMIS(Medical Management Information System)

의료조직의 자재, 구매, 물류, 원무, 수납, 인적자원관리, 회계, 경영전략 제고등 병원행정에 관련된 자원관리 업무를 최적화하는 시스템으로, 다양한 경영Index 생성, Decision Making 기반이 되는 여러 통계와 추계자료를 요약하여 향후 조직의 방향을 제시하는 기능 포함. 유기적으로 EMR, OCS, PACS와 유기적으로 연동전제(De Vries et al., 2008).

이외에도 Groupware, 의무기록 광파일과 전자서명지원, Point of Care, Datawarehouse와 Mining, Medical Knowledge Management System등 시스템이 추가되면서 진화성 수용의 컴포넌트 병원정보 시스템 중요도가 더해가고 있다. 이를 3개의 축으로 진화성을 도식화 할 수 있다(그림 1). 끊임없이 변화하는 자원관리 환경 속에서 의료조직이 경쟁력을 유지하기 위해서는 내외부 환경을 사전에 예측하고 이에 신속하게 대응할 수 있는 능력을 요구한다. 이를 위해 의료조직은 기존 단위업무 및 부서별로 적용하던 정보시스템 구축 차원을 벗어나 모든 비즈니스 업무 영역을 통합적으로 지원할 수 있는 전사적 정보시스템으로써의 HIS (Hospital Information System)이 요구되고 있다(Wu et al, 2013).

그 핵심은 경영 정보의 통합과 정보 흐름의 자동화이다. HIS 구축을 통해 의료조직은 치열한 경쟁체제하에서 정보 우위를 통한 경쟁력을 향상시킬 수 있으며 급변하는 새로운 의료 환경에도 유연하고 신속하게 대처할 수 있다. 한편, 클라이언트/서버, 분산객체 컴퓨팅, 인터넷/웹 등 계속되는 소프트웨어 신기술 등장은 HIS 소프트웨어에 대해서도 지속적으로 새로운 컴퓨팅 모델을 요구하고 있다(Wu et al, 2013).

[그림 1] 진화하는 HIS시스템의 방향



하지만, 의료기관 내부의 제한된 비용, 개발 인력과 기술력으로는 효율적인 의료 조직 내 HIS 구축 작업도 쉽지 않은 상황이며 지속적인 HIS 소프트웨어의 보완과 확장을 통한 최신의 컴퓨팅 환경을 수용하는 것은 현실적으로 많은 어려움이 존재한다. 최근 적극 추진되는 Outsourcing 방안 또한 다양한 업종과 패턴을 갖는 의료조직 요구사항에 대한 맞춤형태의 구축 방법으로는 지속적인 유지보수 및 확장 등에 많은 어려움이 있다. 또, 고도화되고 복잡해져만 가는 소프트웨어 기술 환경은 서비스 공급자나 개발자들에게도 사용자의 요구 시기에 맞춰 최신의 어플리케이션을 개발하여야하는 어려움을 주고 있다.

이에 대한 최선의 방안이 바로 HIS Component 패키지 시스템의 활용이다. HIS Component 패키지를 사용함으로써 의료조직은 HIS 초기 구축시간과 비용을 줄일 수 있을 뿐만 아니라 소수의 개발 인력만으로 지속적인 유지관리 및 확장이 가능하다는 이점이 있다. 즉, 사용자는 HIS시스템 패키 지중에서 자신의 의료조직환경에 적합한 모듈만 부분적으로 선택하여 시스템을 유연하게 구축하거나 기존 시스템을 확장할 수 있다. 또한, 최신의 소프트웨어 기술을 반영한 패키지 시스템을 지속적으로 공급받음으로써 최신 컴퓨팅 환경에 기반한 정보시스템을 사용할 수 있다. 이러한 시스템 확장성, 통합성 및 유연성 등의 HIS 패키지 시스템 이점으로 인해 의료조직은 정보시스템을 자체 개발해야 하는 부담을 줄일 수 있으며 손쉽게 HIS를 의료기관 정보인프라로 구축, 활용할 수 있다.

Component 프로그래밍 기법은 하드웨어 IC칩과 같이 다양한 어플리케이션에 사용될 수 있는 단위 소프트웨어를 Component로 개발, 공유함으로써 바이너리 수준에서의 소프트웨어 재 사용 지원하는 개념으로 Component Based Software Development를 활용한다(Palmer 2012).

HIS 시스템 운영환경 또한 Host에서 유무선 기반 Client Server Computing Environment로 빠르게 변화하고 있고, Component 기술과 맞물려 새로운 HIS 시스템 기술을 요구하고 있다 (Wu, 2013).

요구되는 Component들의 구성, 검증, 연계가능성을 Simulation하고 그 재사용 가능성을 제고하고 Component간의 연계가능성을 다양화하여 진료, 진료지원 및 원무행정 조건에 맞게 변경하고 활용하게 하기 위해 본 논문에서는 Software 재사용을 위한 Component 공학 기법으로 의료기관의 전사적 정보시스템을 위한 Hospital Information System(HIS) 컴포넌트 C-Handler를 개발 제시한다. HIS 컴포넌트들을 관리하고 검증하는 도구를 개발하여 개발기간 단축과 검증된 공통 Component 기능을 적용하여 요구사항 변경 시에도 HIS 산출물 품질의 신뢰성 예측하고 서비스 품질 향상을 꾀하고자 하였다.

II. Software Component

1. Component-Driven Modeling

Component 모델링의 메타 모델은 Component 모델링자체의 분석/설계 모델을 의미한다. 즉, 개발하고자 하는 HIS 시스템을 위해 모델링을 할 때, 여기서 HIS시스템이 Component 모델링에 해당되며 모델링이 된 결과물이 바로 Component 모델링 메타모델이 되는 것이다. Component 모델링 메타 모델의 모델링 구성 요소는 다음과 같다(CMU, 2014, SUN, 2014).

- Class : 공통된 구조와 행동을 가진 객체들의 개념적인 집합체
- Use Case : 상호 연관된 객체에 의해 시스템의 기능을 완성하는 단위
- Pattern : 업무 처리에 사용되는 일반적인 형태 혹은 폼
- State : 한 객체가 시간과 공간 그리고 외부 사건에 대해 정적이면서 동적의 상태의 임의의 값을 갖기 위한 조건
- Interface : 한 객체의 공용부분
- Component : 논리적이면서 물리적인 재사용요소
- Node : 완성된 소프트웨어 시스템이 운영되는 하드웨어 시스템
- Package : 모델링 구성요소의 논리적/물리적 집합체
- Note : 부연 설명을 위한 요소

- Association Relation : 두 구성 요소간의 개념적인 관계
- Generalization : 일반화된 모델링 구성 요소와 구체화된 구성 요소간의 분류(taxonomic)를 나타낸 관계
- Dependency Relation: 한 모델링 구성요소(Client)가 다른 구성요소(Server)를 이용하는 관계
- Refinement Relation : 점차로 추상화 수준을 구체화하는 관계
- Component 모델링 자체를 사용자가 확장할 수 있도록 하는 Mechanism : 명세(Specification), 스테레오 타입(Stereotype), 제한(Constraint), 태그 값(Tag Value)

위의 모델링 구성요소들은 상세한 설명을 위한 명세부분, 추상화 수준에서 편리한 분류(Classification)를 위한 스테레오 타입, 모델링의 범위를 구체화하기 위한 제한 그리고 부연 설명을 위한 태그 값들로 구성된다.

2. Reusability

소프트웨어 개발 과정에서 재사용의 체계적인 행위는 두 가지 범주로 나눌 수 있다(박화규 외 2000). 첫번째, 재사용을 위한 개발(develop for reuse)은 고도의 재사용 가능성을 갖는 소프트웨어 모듈을 생산하는 것이다. 두번째, 재사용에 의한 개발(develop by reuse)은 기존 소프트웨어 부품의 사용을 극대화하는 것으로 다시 소프트웨어 부품 그대로를 다시 재사용(reuse as-is) 하는 경우와 수정(modification), 생성(generation), 조합(composition) 등의 변경 과정을 통한 재사용(reuse with change)으로 구분된다(Laforcade, 2010).

Component기반 방법론은 객체지향 방법론에 비하여 소스 수준이 아닌 바이너리 수준에서의 재사용을 지원함으로써 새로운 형태의 소프트웨어 부품화를 지원하고, 신규 소프트웨어 개발 시 표준화되고 신뢰성 있는 소프트웨어 Component 조립에 의한 개발 시간과 비용 감소 뿐 아니라 유지보수에도 능동적으로 대처할 수 있도록 한다(OMG, 2013).

3. Component

소프트웨어 Component는 프로그램 로직을 포함하고 간단한 기능을 수행하는 소프트웨어로서 자신의 인터페이스를 외부에 알려주는 기능을 가지고 있는 재사용 가능한 소프트웨어 Component 부품을 의미한다. 따라서 이에 대한 Component 유형으로는 프로시저(procedure),

클래스(class), 모듈(module), 라이브러리(library), 어플리케이션(entire application) 등이 모두 포함된다. 그러나, 최근에는 고유한 기능을 지원하며 독립적인 개발, 획득 및 적용이 가능한 바이너리 수준에서의 재사용 단위를 의미하는 개념으로 사용되고 있다(박화규 외 2000).

한편, 객체지향 시스템에서도 기존 객체들을 이용하여 자신의 목적에 맞는 기능을 얻고자 하는 의도에서 Component 합성에 대한 관심이 시작되었다. Zheng, Helm 등은 Component 들의 합성이 보다 높은 재사용성을 보장할 수 있다는 이론을 제시하고 객체들 간의 행위적 합성을 제안하기도 하였다. 그러나, Component는 기존의 객체지향 언어에서 표현하는 객체와 유사하지만 유일한 식별자와 고유한 State를 갖는 객체와는 달리 상태를 갖지 않으며 실행코드 형태로 소프트웨어 조합의 단위로 사용된다는 점에서 다르다. 또한, 인터페이스뿐만 아니라 자신의 구현을 포함하고 있어 자체적으로 동작하며 Customizing 방법을 제공하는 점 등이 객체와 주요한 차이점이다.

또한, Abstraction 측면에서 비교할 경우, 객체가 어떤 서비스를 Encapsulation하여 상태와 행위, 그리고 식별자를 갖는 동적인 Runtime entity임에 반하여 Component는 Plug가 있는 추상화된 형태로 시스템을 구성하는 Entity 로서 자신에게 주어진 연산과 외부환경과의 통신을 통하여 특정한 일을 수행하는 동적인 Entity로 정의하고 있다(Laforcade, 2010).

현재 객체지향 프레임워크에서는 Component를 객체의 의미로 사용하지만 소프트웨어 구조에서는 객체와 Component를 구별하여 사용하고 있다. Component와 객체의 가장 큰 차이점은 Component에서는 잘 정의된 플러그 또는 인터페이스를 통하여 다른 Component와 결합이 가능하다는 것이다. 물론, 여기서 말하는 결합이란 물리적인 결합이 아니라 Component들 간의 통신을 통한 논리적인 결합을 의미한다. 그리고, 이러한 논리적인 결합은 Component들 간의 상호작용을 정의하는 커넥터 또는 Component 프레임워크를 통해서 가능하다(Palmer, 2012).

Component들의 개방적인 특성은 어플리케이션 개발자들로 하여금 통합 개발 환경에서 마치 레고 블록을 조립하듯 Component들을 조합하여 원하는 어플리케이션을 손쉽게 만들 수 있도록 한다. 즉, 다양한 어플리케이션에서 공동으로 사용되는 기능은 공용 Component로서 중복하여 재사용하고, 만약 개발하려는 어플리케이션의 기능을 수행할 Component가 없다면 개발자는 그 부분만 코딩하는 것이다(Laforcade, 2010).

Component에 기반을 둔 어플리케이션 개발 방법이 개발자 사이에서 보편적으로 활용되기 위해서는 강력하고 통일된 Component 프레임워크와 Component기반 어플리케이션 개발 환경이 보급되어야한다(Kos et al, 2012).

소프트웨어 재사용을 위한 주된 이슈는 시스템을 구성하고 있는 Component들이 갖는 기능이 아니라 이들이 얼마나 호환성 있게 결합하여 원하는 목적을 달성하느냐에 있다. 이는

Component가 높은 수준의 추상화를 제공할 뿐 아니라 이를 이용한 시스템 개발을 위해서는 Component들 간의 합성이 가능해야 하며, 그 합성에 대한 효율적이면서 정형화된 프레임워크가 요구됨을 의미한다. 객체설계의 가장 큰 특징 중 하나도 단독으로 고립된 객체는 존재하지 않는다(Palmer, 2012). 어떠한 형태의 시스템이든 반드시 객체들 간의 상호작용 즉, 프레임워크가 요구됨을 의미한다.

동일한 이유로 Component 기반 개발 환경에서도 Component들 간의 상호작용 및 결합 구조, 즉, Component 프레임워크가 주된 이슈가 되고 있다(Pfaffenbichler et al., 2010).

일반적으로 소프트웨어 프레임워크는 Component들의 집합과 이들 Component들 간의 상호작용을 나타내는 커넥터(connector)들의 집합으로 정의하고 있으며, 프레임워크에 대한 일반적인 두 가지 측면은 구조(structure)적 측면과 행위(behaviour)적 측면이 있다. 구조적 측면은 하나의 시스템을 그 구성원들의 Component들로 분해(decomposition), 이들 간의 결합 관계(composition relationship), 그리고 Component들 간의 통신 통로(path)들을 의미하며, 행위적 측면은 시스템에 의해 수행될 활동(activity)들 간의 시간적 순서와 목적을 성취하기 위해 필요한 통신 패턴(communication pattern)들을 의미한다.

한편, 소프트웨어 프레임워크는 특정 요구 사항에 국한하여 설계하는 것이 아니라 다양한 도메인 상에서 호환성을 갖는 Component들을 만들어낼 수 있는 환경을 제공하는 것이지만 현재 Component 프레임워크는 크게 분산객체로 대변되는 DCOM, CORBA 및 JavaBeans 세 종류의 표준이 제시되어 사용되고 있다(Cosmin, 2013; Palmer, 2012).

DCOM과 CORBA는 모두 RPC(Remote Procedure Call)를 기반으로 한 클라이언트/서버 모델을 사용한다. 일반적으로 클라이언트는 동일 지역 또는 원격 머신에 있는 객체의 메소드를 호출(involve)함으로써 서비스를 요청한다. 그 객체가 제공하는 서비스는 IDL(Interface Definition Language)로 작성된 인터페이스 정의를 사용하여 접근할 수 있으며, 또한 IDL로 정의한 인터페이스는 서버와 클라이언트간의 통신의 지침으로 사용되기도 한다. 그 하위 레벨의 통신은 User Transparency 지원으로 객체 위치에 무관하게 호출이 가능하다(박화규 외 2000).

본 연구에서 고려 될 수 있는 기존 방법으로는 Catalysis, RUP, Innovator가 있다(Cosmin, 2013; Palmer, 2012).

Catalysis는 모델의 일관성과 균형을 유지하며, 프로젝트 전반에 대한 설계를 지원하고, 개방적 분산 시스템의 모델링과 구축을 전제로 Component와 Framework의 사용·재사용을 지원한다. RUP는 독립적 대형병원서는 활용하기 힘든 대형 Software 시스템, 혹은 다양한 조직, 다양한 프로젝트 크기에 전문화 될 수 있는 포괄적인 프로세스 Framework을 특징으로

하고 있다. Innovator는 요구사항 정의 및 분석, 설계, 개발, 테스트, 구현 단계로 구성되며, 프로젝트 착수 및 요구사항이전의 플랜수립, Solution 및 정의, 반복적인 분석설계 및 전개, Project Management로 나누어 진다. 컴포넌트 개발과 컴포넌트 기반 개발로 분리하며, 점진적 개발을 특징으로 갖는다. 이는 개발 프로세스를 CDM 방법과 CBD개발로 분리하였으며 UML Use Case를 반복적 개발하고 아키텍처 중심적 개발을 하며, EJB 명세에 기반을 두고 있다.

기존 방법들은 데이터에서 재사용을 위한 contents를 추출하기 위한 목적으로 개발 되었으나, 그들이 지원하는 범위는 서로 다르다. HIS관점에서, 기존 연구의 적용 한계점은 이들은 모두 방법절차가 제공하는 지침에 따라 영역별 개발자들이 Component 분석과 설계후, 타 Software를 개발할시에 Component를 재사용하는 것이다. 그러나 Component 요구사항정의, 분석, 설계 프로세스에서 다양한 웹 환경에서 Reusablilty는 고려되지 않았고, 무엇보다 의료 기관에서의 진료, 진료지원, 행정 부문별에서 재사용되는 Component 보다 그 사용자가 활용할 수 있는 데이터의 수준의 세분화 수준인 Granularity 크기가 커서 재사용에 어려움이 있다. 따라서, HIS를 위한 단위별 기능을 하는 Visual Interface, Data Interface, Logical Program별 Component 들을 개발하여 재사용성을 꾀할 필요가 있다.

3. HIS지원 C-Handler

C-handler는 Component기반 HIS Component 지원 시스템이다. 개발한 C-handler는 의료 조직의 표준 업무 처리를 위한 병원정보 어플리케이션 Component 개념을 지원하고 독자적인 Development Frame과 실행브라우저 등의 다양한 Tool Set을 통한 어플리케이션 개발과 실행이 가능한 HIS 시스템으로 DCOM 분산객체 미들웨어에 기반하고 있다(Eclipse, 2014; Park et al, 2004).

특히, C-handler는 다양한 업종 및 업무와 관련한 표준적인 비즈니스 Component들을 제공함으로써 커스터마이징을 통해 특정 의료조직에 맞는 HIS를 효과적으로 구축할 수 있는 솔루션을 제공한다.

1. System Configuration

[그림 2]에서 보는 것처럼 C-handler는 Component Repository를 토대로 하여 다양한 개발,

Repository를 C-handler와 같은 패키지형 개발도구에 활용함으로써, 다양한 개발 도구들을 Component 개념에서 상호 유기적으로 연결함으로써 통합적인 개발접근이 가능하다.

Repository는 전체 HIS의 구현 및 관리의 모든 분야를 포괄하는 개념이기 때문에 개발과 관련한 자원관리 부분뿐만 아니라 Component 개체와 속성 그리고 Component간의 관계, 비즈니스 규칙, Component 타입, Component 정의, Component 소스코드, 개발관련 서류, 도움말, 아이콘 등 시스템 개발에 필요한 모든 Component 및 메타정보가 포함되어 있다. 업무 분석을 통해 설계된 어플리케이션은 일관되고 통합된 개발환경을 통해 개발, 유지, 관리되어야 한다. 특히, 개발된 표준업무 어플리케이션은 다른 시스템 및 운영환경에서 재사용될 수 있도록 Component 형태로 저장, 관리될 필요가 있으므로 이를 효과적으로 지원할 수 있는 최적화된 저장 기술 및 도구가 필요하다.

여기에는 스크립트 프로그램 편집기, 다양한 화면 컨트롤을 지원하는 화면 편집기, 데이터베이스 테이블 구조를 정의하고 스키마를 생성할 수 있는 데이터베이스 편집기, 편리한 사용자 환경을 구성할 수 있도록 지원하는 메뉴 편집기 및 에러 분석을 위한 디버거 그리고, 편리한 사용자인터페이스를 지원하는 GUI 형태의 전용브라우저 등이 포함된다(Park et al, 2004).

비즈니스 Component에 대한 서비스 기능을 지원하는 서비스 Component를 DCOM과 같은 분산객체 Component로 구축함으로써 위치 투명성, 개발 언어 및 플랫폼 독립성, 클라이언트와 서버의 효율적인 연결 관리, 버전 및 보안 관리 등 많은 기능들을 지원받을 수 있다.

C-handler의 장점중 하나는 유용한 Component 정보들을 전체 소프트웨어 개발 과정에 일관되게 적용함으로써 각 Component들 간의 관련 및 파생 결과를 예측할 수 있어 기존 CASE 도구들과는 달리 소프트웨어 재사용 체계를 제공한다는 점이다.

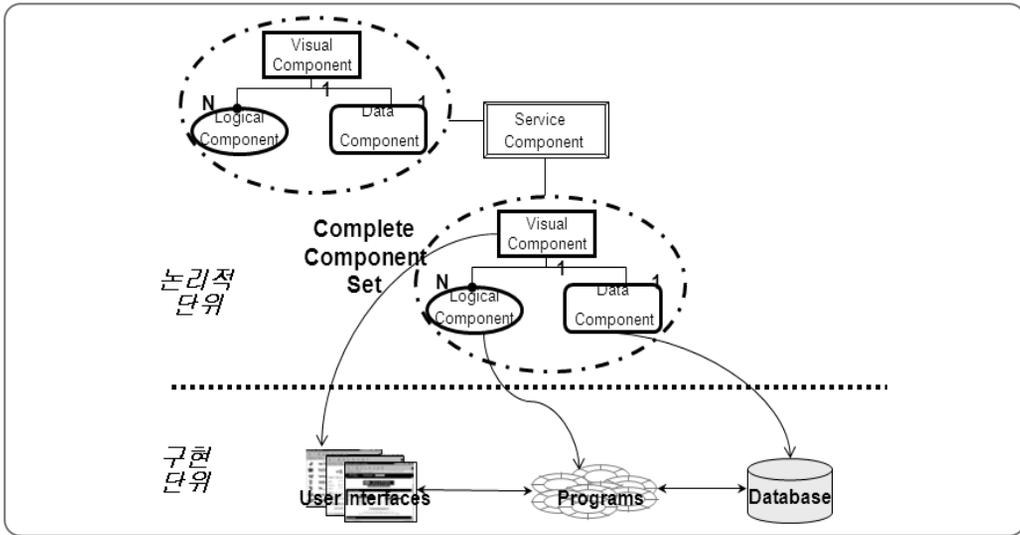
2. C-handler Component

C-handler에서는 이미 개발된 비즈니스 Component에 관한 다양한 정보를 커스터마이징 도구 및 개발 도구를 통해 제공함으로써 개발자가 적절한 Component를 재사용 또는 확장함으로써 새로운 비즈니스 Component를 개발할 수 있도록 지원한다. 따라서, C-handler 사용자는 이미 구축된 많은 비즈니스 Component들을 활용할 수 있기 때문에 보다 빠른 시간 내에 고품질의 HIS 시스템을 새로이 구축할 수 있다.

C-handler는 소형 Component 중심의 조합(composition) 및 진화(evolution)를 지원하기 때문에 새로운 HIS 시스템 구축뿐만 아니라 기존 HIS 시스템의 유지보수 및 확장 등에도 강점을 갖는다.

C-handler 시스템에서 Component는 [그림 3]와 같이 다음 4개의 Component 타입으로 분류된다.

[그림 3] Component Type에 연계도



<표 1> Component Type 정의

정의	Hospital Information System 통합 환경에서는 이미 개발된 비즈니스 Component에 관한 다양한 정보를 커스터마이징 도구 및 개발 도구를 위한 Component 구조를 통해 이를 재사용 또는 확장함으로써 새로운 비즈니스 Component를 개발할 수 있도록 지원한다
종류 및 역할	<p>비즈니스 Component(BCO : Business Component) : 좁은 의미의 비즈니스 Component에는 VCO, LCO, DCO가, 넓은 의미의 비즈니스 Component에는 SCO, 비즈니스 어플리케이션 Component, 비즈니스 트랜잭션 Component 포함.</p> <ul style="list-style-type: none"> - VCO(Visual Component) : VCO에는 비주얼한 사용자 인터페이스와 관련된 Screen Screen-Process, Control, Dialog 등의 Component 타입이 포함된다. ScreenProcess는 다시 ASP(After ScreenProcess), BSP(Before ScreenProcess), CSP(Common ScreenProcess)로 세분화 - LCO(Logic Component) : LCO는 어플리케이션 로직을 수행하는 Program, 상위 Method 등의 Component 타입이 포함 - DCO(Data Component) : DCO는 어플리케이션 수행 대상이 되는 관계형 테이블 Component 와 Element Component가 포함 <p>서비스 Component SCO(Service Component) SCO는 런타임 오브젝트, 저장 오브젝트, 세션 오브젝트, 트랜잭션 오브젝트, 워크플로우 오브젝트, 서버통제/운영 오브젝트 등의 여러 DCOM 오브젝트들로 구성되어있다. 앞서 언급한 비즈니스 Component들은 SCO를 통해서만 상호 관련성을 맺거나 호출</p>

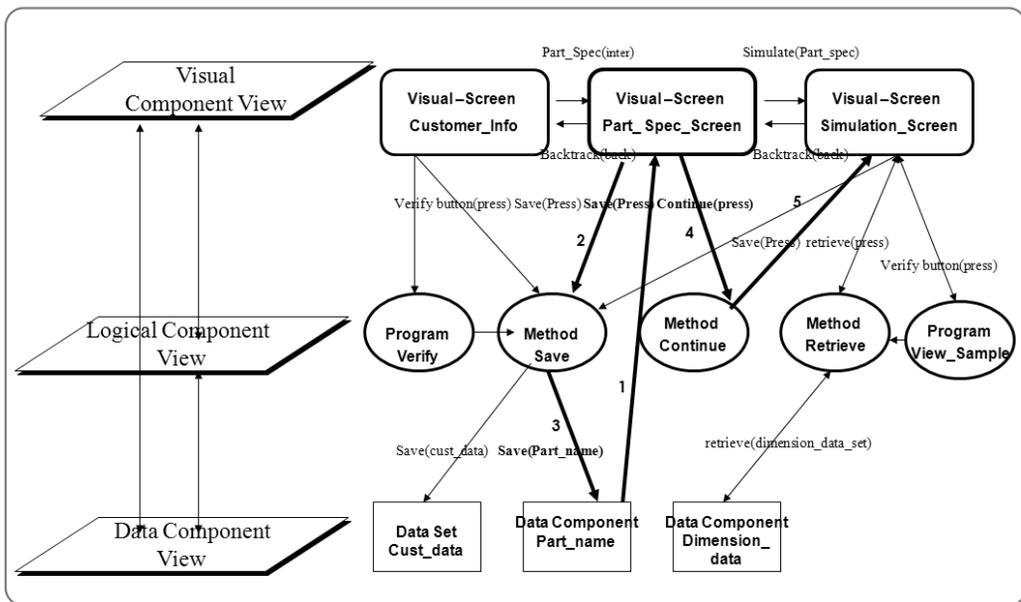
C-handler 시스템은 <표 1>에서 언급된바와 같이 SCO를 통해 COM Component 프레임워크를 따르고 있으므로 다른 COM Component 시스템간의 인터페이스가 존재하며, 서비스의 제공에서도 신뢰성을 얻을 수 있다. 또한, In-Process Server, Local Server, Remote Server 등의 다양한 형태로 구성하면서도 일관된 Location transparency를 제공함으로써 모든 종류의 클라이언트와 서버간 커뮤니케이션 문제를 해결한다(OMG, 2013).

서비스 Component 역할을 수행하는 DCOM 오브젝트들은 MTS(Microsoft Transaction Server)라고 하는 또다른 미들웨어에 의해 패키지단위로 내포되어 MTS가 제공하는 분산 트랜잭션, 쓰레드 관리, 자원 공유 및 Resource pooling 기능 등을 지원 받는다(Cosmin, 2013).

3. C-handler Component Framework

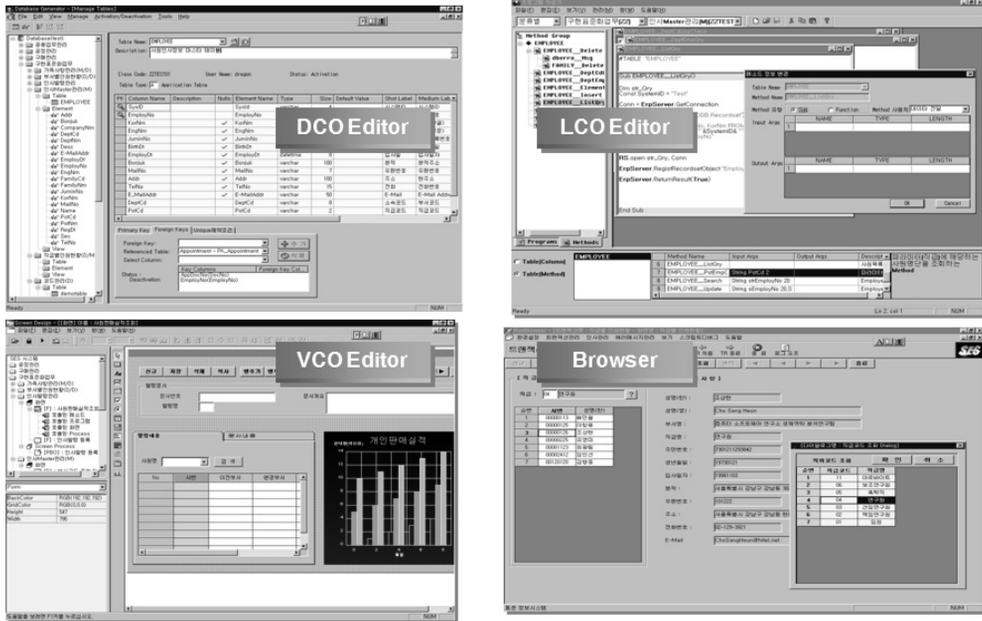
C-handler Component 프레임워크는 응용 프로그램간의 통합뿐만 아니라 Component의 생성, 소멸에서부터 저장, 그리고 트랜잭션 기능에 이르기까지 Component 개발 및 실행 환경에서 필요한 모든 서비스를 총괄한다. 비즈니스 서비스 Component는 마치 컴퓨터의 하드웨어 버스처럼, 소프트웨어 버스로 불리어지는 DCOM을 통해 Component의 위치와 상관없이 접근할 수 있는 구조를 제공한다.

[그림 4] Three Layers 구조



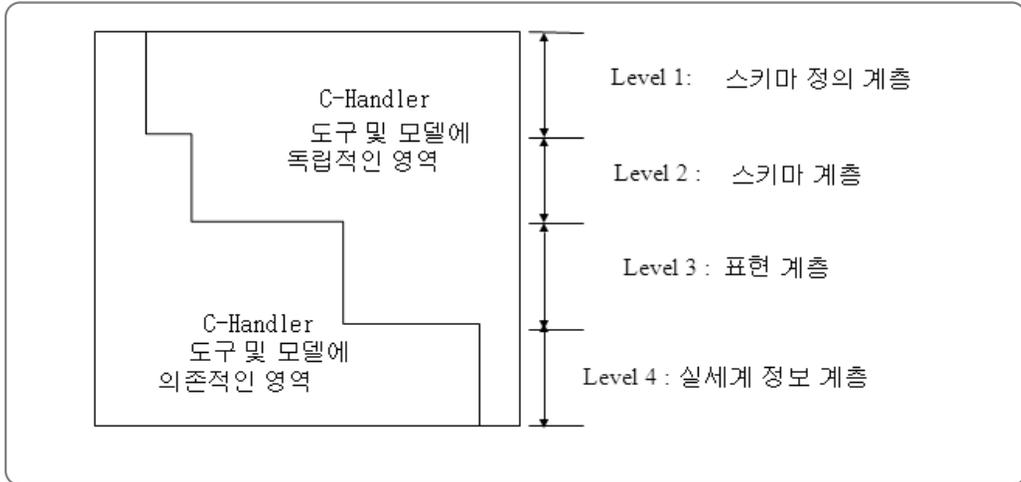
[그림 4]는 C-handler Component들 간의 상호작용을 정의한 Component 프레임워크로 다양한 Component들 간의 3층 관계성을 보여준다. 프레임워크에서 특정 업무에 대응되는 비즈니스 어플리케이션 Component는 여러 개의 비즈니스 트랜잭션 Component로 구성되며, 다시 하나의 비즈니스 트랜잭션은 여러 개의 비즈니스 Component 셋으로 구성된다. [그림 4]의 Component 프레임워크는 특정 Screen Component로부터 다양한 비즈니스 Component들 간의 관계성이 어떻게 이루어지고 관리되는지를 보여주며 이러한 프레임워크는 비즈니스 서비스 Component에 의해서, 프레임워크에 관한 메타모델정보는 리파리토리에 의해 지원된다.

[그림 5] C-handler Component 프레임워크



C-handler 시스템의 Component 계층을 보여주며 그림에서 보는바와 같이 C-handler는 크게 Component 계층, 프레임워크 계층, 인프라구조 계층의 3계층으로 구성된다. Component 계층에는 비즈니스 어플리케이션, 비즈니스 트랜잭션, BCO, SCO Component들이 포함되며 프레임워크 계층에는 SCO Component와 DCOM Component 버스, Component Repository가 해당된다.

[그림 6] C-handler Component 계층



특히, 서비스 Component(SCO)는 Component Repository와 쌍(pair)으로 동작하며, Component 개발 및 실행시에 미리 메타 모델에 의해 정의된 Component Repository를 통해 해당 비즈니스 Component들을 저장하거나 호출하여 실행한다.

IV. 결 언

논문에서는 소프트웨어 효율적 전사적 병원자원관리 측면에서 HIS 소프트웨어와 소프트웨어 Component 개발 기술에 대해 고찰하고 Medical Tourism, Management Service Organization (MSO) 기반의 Network 병원연계, Tele-medicine, u-healthcare 및 의료조직의 해외진출등 급변하는 의료조직환경의 요구사항을 수용하기 위하여 Component 기반 HIS Component Workbench 시스템인 C-Handler를 제안하였다. 이를 위해 현 개발 중인 C-handler 패키지 시스템과 관련한 비즈니스 Component 개념을 정의하고 이를 지원하는 비즈니스 Component 프레임워크 구조를 제시하였다.

C-Handler를 통해 개발사용자는 요구되는 Component들의 구성, 검증, 연계가능성을 Simulation하고 그 재사용 가능성을 확인 할 수 있다. 더불어, Component간의 연계가능성을 다양화하여 진료, 진료지원 및 원무행정 조건에 맞게 변경하여 활용하는 장점이 있다. 이를 통한 기대효과는 개발기간 단축과 검증된 공통 Component 기능을 적용하여 요구사항 변경

시에도 HIS 산출물 품질의 신뢰성 예측하고 서비스 품질 향상 된다는 점이다. 이는 다양한 의료기관을 위한 HIS 표준화 수준 향상으로 표준을 따르는 개발방식과 Pattern 등을 산출 제공하여 그 수준 향상을 꾀할 수 있다.

향후 의료기관에서 요구되는 컴포넌트 기반의 정보시스템의 효율적으로 관리 운용을 위해 진료, 진료지원, 원무행정프로세스 관점에서의 각 단위별 컴포넌트와 이를 상호연동하기위한 Framework 표준화를 확립하고, Repository 확장 연구가 요구된다.

References

- Agarwal R, Sands D.Z, Schneider J.D.(2010), Quantifying the economic impact of communication inefficiencies in U.S. hospitals, *Journal of Healthcare Management*, Vol. 55, No. 4, pp.265-281.
- CMU.(2014), Component-Based Software Development/COTS Integration, *Software Technology Review*, http://www.sei.cmu.edu/str/descriptions/CBD_body.html.
- Cosmin Arad(April 2013), "Programming Model and Protocols for Reconfigurable Distributed Systems". Doctoral Dissertation, Stockholm, Sweden: KTH Royal Institute of Technology.
- E.N. De Vries, M.A. Ramrattan, S.M. Smorenburg, D.J. Gouma and M.A. Boormeester, The incidence and nature of in-hospital adverse events: a systematic review, *Qual Saf Health Care* 17(2008), pp.216-223.
- Eclipse. Model Development Tools (MDT)/UML2 Tools, 2014.
<http://www.eclipse.org/modeling/mdt/?project=uml2>.
- Kos, T. Kosar, T. Mernik, M.(2012), Development of data acquisition systems by using a domainspecific modeling language, *Computers in Industry*, vol. 63, pp.181-192.
- Laforcade P.(2010), A Domain-Specific Modeling approach for supporting the specification of OVisual Instructional Design Languages and the building of dedicated editors," *Journal of Visual Languages & Computing*, Vol. 21, No. 6, pp.347-358.
- OMG.(2013), Object Management Group. Systems Modeling Language, SyML v. 1.3.
- Palmer J.W.(2012), Web site usability, design, and performance metrics, *Information Systems Research*, Vol. 13, No. 2, pp.151-167.
- Park H, Suh W, Lee H. A Role-Driven Component-Oriented Methodology for Developing Collaborative Commerce Systems. *Information and Software Technology*, 2004, 46(12): 819-837.
- Pfaffenbichler P, Günter E, Simon S.(2010), A System Dynamics Approach to Land Use Transport Interaction Modelling: The Strategic Model MARS and Its Application, *System Dynamics Review*, Vol. 26, No. 3, pp.262-282.
- Rausch, A. and Reussner, R. The Common Component Modeling Example, volume 5153

- of Lecture Notes in Computer Science. Springer, 2008.SUN. The Reection API, 2014. <http://java.sun.com/docs/books/tutorial/reect/TOC.html>.
- World Health Organization a.: (2013). Private Health Insurance in OECD Countries: OECD Health Project.
- Wu RC, Rossos P, Quan S, Reeves S, Lo V, Wong B, et al. An evaluation of the use of smartphones to communicate between clinicians: a mixed-methods study. *J Med Internet Res* 2011, 3(3): 51-63.
- Wu RC, Morra D, Quan S, Lai S, Zanjani S, Abrams H, et al. The use of smartphones for clinical communication on internal medicine wards. *J Hosp Med* 2010, 5(9): 553-559.
- Z. Liu, C. Morisset, and V. Stolz. rCOS: theory and tools for component-based modeldriven development. Technical Report 406, UNU-IIST, February 2009, Lecture Notes in Computer Science.
- 박화규, 박성진, “컴포넌트기반 ERP 패키지시스템 개발 기술,” 대한설비관리학회지 5권 4호, pp.91-103, 2000.

C-Handler Driven Hospital Information System Developing Technology

Park, Hwa-Gyoo*

ABSTRACT

Component-based development (CBD) is to emphasize the separation of concerns in respect of the wide-ranging functionality available throughout a given hospital information system. The concept is useful for analyzing the evolving medical organization requirements; it can specify the complex behavior of hospital information system requirements with multiple healthcare stakeholders while addressing its dynamic interactions. It can help control the roles of multiple stakeholders so that they cooperate with each other effectively. The component-oriented approach also shows several strengths such as reusability, rapid development, cost effectiveness and dynamics. It is becoming less possible for developers of integrated software solutions to quickly, efficiently, and inexpensively develop all of the functions and modules desired by actual users in the industries. The requirements can be achieved by reusing existing software components. The principle of component software is already changing the software development concept. In addition a component handling tool (C-Handler) with meta-data scheme is developed for implementing and managing the component artifacts. C-handler provides a programming environment for the information system development in 3-tier solution and enable the production of reusable artifacts in a systematic fashion.

Key Words : Hospital Information System, Software Component, Healthcare Organization

* Professor, Department of Health Administration and Management, Soonchunhyang University,
E-mail : hkpark1@sch.ac.kr

