

경영정보학연구
제10권 제1호
2000년 3월

컨텐츠 지향적 하이퍼미디어 설계 방법론

서우종*, 이희석*

A Methodology for Designing Contents-Oriented Hypermedia

Suh, Woo-Jong, Lee, Hee-Seok

Recently there have been many efforts to develop hypermedia systems. Hypermedia applications can be categorized into process-oriented or contents-oriented. However, it is not a trivial task to build contents-oriented hypermedia in a systematic fashion. The contents-oriented hypermedia systems are developed primarily for the applications of marketing or information services. This paper proposes a methodology for a contents-oriented hypermedia system. Hypermedia contents are typically structured in the form of hierarchies. An index node can be used in order to give an effective access to this hierarchical structure. The proposed methodology employs three types of nodes (contents, index and function) and various navigation mechanisms, and helps design contents-oriented hypermedia systematically. Furthermore, various design techniques are proposed for maintaining contents-oriented hypermedia systems that have an evolutionary feature. Two real-life cases are illustrated to demonstrate the applicability of the methodology.

* 한국과학기술원 테크노경영대학원

I. 서론

최근, 인터넷 기술의 급속한 발달과 더불어 하이퍼미디어 기술은 사회 각 분야의 조직에서 폭넓게 응용되고 있으며 하이퍼미디어 구축 사례가 급속한 속도로 증가하고 있다. 하이퍼미디어의 응용은 (i) 조직의 프로세스를 지원하기 위한 정보시스템 분야와 (ii) 컨텐츠 지향적 (contents-oriented) 응용 분야로 대별할 수 있다. 프로세스 지향적인 하이퍼미디어는 조직의 업무 흐름에 따른 정보처리를 지원하기 위한 그룹웨어 기능에 중점을 두고 주로 인트라넷 (intranet) 을 기반으로 구축된다. 이는 하이퍼미디어 기술이 네트워크 기반의 운영에 있어 많은 장점들을 가지고 있기 때문에 그 중요성이 대두된다. 한편, 상대적으로 보편적인 하이퍼미디어의 응용은 컨텐츠 지향적인 분야에서 이루어져 왔는데, 이는 하이퍼미디어가 멀티미디어의 한 유형으로써 다양한 형태로 컨텐츠를 표현할 수 있을 뿐만 아니라, 하이퍼링크 (hyperlink)에 의한 네비게이션 원리는 하이퍼미디어의 가장 중요한 특징으로 컨텐츠들 간의 관계를 인지적 과정에 효과적인 구조로 구현할 수 있기 때문이다. 이 분야의 하이퍼미디어 시스템은 주로 광고, 정보서비스 및 커뮤니케이션을 위한 수단으로 활용되고 있다. 즉, 인터넷을 사용하는 불특정 다수의 네티즌들을 대상으로 구축되거나, CD-ROM 타이틀을 통해 멀티미디어북 (multimedia book)으로 구현되기도 한다.

이러한 하이퍼미디어의 응용분야에 대한 특성을 고려해 볼 때, 구축방법도 크게 두 가지 관점에서 구분될 수 있다. 프로세스 지향적인 시스템은 과업 주도형 접근방법 (task-driven approach)에 의한 방법론이 필요한데, 이러한 방법론으로는 SOHDM (Scenario-Based Object-Oriented Hypermedia Design Methodology) [Lee et al., 1998, 1999b], EORM (Enhanced Object-Relationship Model) [Lange, 1994, 1996], 그리고 WHDM

(Workflow-Based Hypermedia Development Methodology) [Lee and Suh, 1999]이 있다. 이 방법론들은 프로세스 분석을 고려한 하이퍼미디어 구축 방법론으로, 특히, WHDM은 프로세스 분석을 문서기반의 워크플로우 모형 (document-based workflow model)을 이용하고, 이에 따라 자연스럽게 하이퍼미디어 문서를 도출한다. 한편, 컨텐츠 지향적인 하이퍼미디어 시스템 개발에 있어서는, 컨텐츠에 대한 구성을 사용자의 인지과정에 효과적인 방식으로 구현하는 것이 중요하다. 그러나, RMM (Relationship Management Methodology) [Isakowitz et al., 1995, 1997], OOHDM (Object-Oriented Hypermedia Design Methodology) [Schwabe and Barbosa, 1994; Schwabe and Rossi, 1994, 1995a, 1995b, 1998], 그리고 VHDM (View Based Hypermedia Design Methodology) [Lee et al., 1996, 1999a]과 같은 기존 방법론들은 컨텐츠의 구성을 효과적으로 반영하는 하이퍼미디어의 전체 구성에 대한 지원이 미흡하다. 그러므로, 이러한 요구를 해결하기 위해서 본 연구에서는 계층화된 인덱스 (index)를 효과적으로 활용하는 방안을 채택한다. 인덱스에 따라 원하는 컨텐츠에 대한 접근을 용이하게 하도록 하는 방식은 사용자들의 하이퍼미디어 전체 구성에 대한 이해와 적응을 효과적으로 유도할 수 있다. 특히, 컨텐츠 지향적인 하이퍼미디어는 지속적으로 진화하는 특징을 갖는다. 따라서, 새로운 컨텐츠의 전달을 위한 지속적인 컨텐츠의 갱신과 효과적인 컨텐츠 전달을 위한 지속적인 인터페이스의 개선 노력이 필요하다. 이를 위해서는 개발 과정 뿐만이 아니라, 효과적인 유지보수까지도 지원함으로써, 이러한 진화적 특성에 유연하게 대처할 수 있는 방법론이 필요하다.

이와 같은 동기를 기반으로, 본 연구에서는 컨텐츠 지향적인 하이퍼미디어 개발 뿐만이 아니라, 유지보수 과정도 실용적이고 효과적으로 수행할 수 있는 인덱스 주도의 하이퍼미디어 설계 방법론 (Index-driven Hypermedia Design Metho-

dology: IHDM)을 제시하고자 한다. 또한, IHDM의 적용 가능성을 예시하기 위해, 개발과 유지보수에 관한 두 사례를 소개하고자 한다.

우선, 2절에서는 IHDM의 아키텍처에 대한 설명을 하고, 3절에서는 구축 사례와 더불어 방법론의 각 단계별 내용을 설명 한다. 4절에서는 IHDM을 기존의 하이퍼미디어 설계 및 구축 방법론들과 비교 분석을 한다. 끝으로 5절에서는 결론 및 향후 연구방향에 대해서 언급하며, 부록에서는 유지보수 사례를 설명한다.

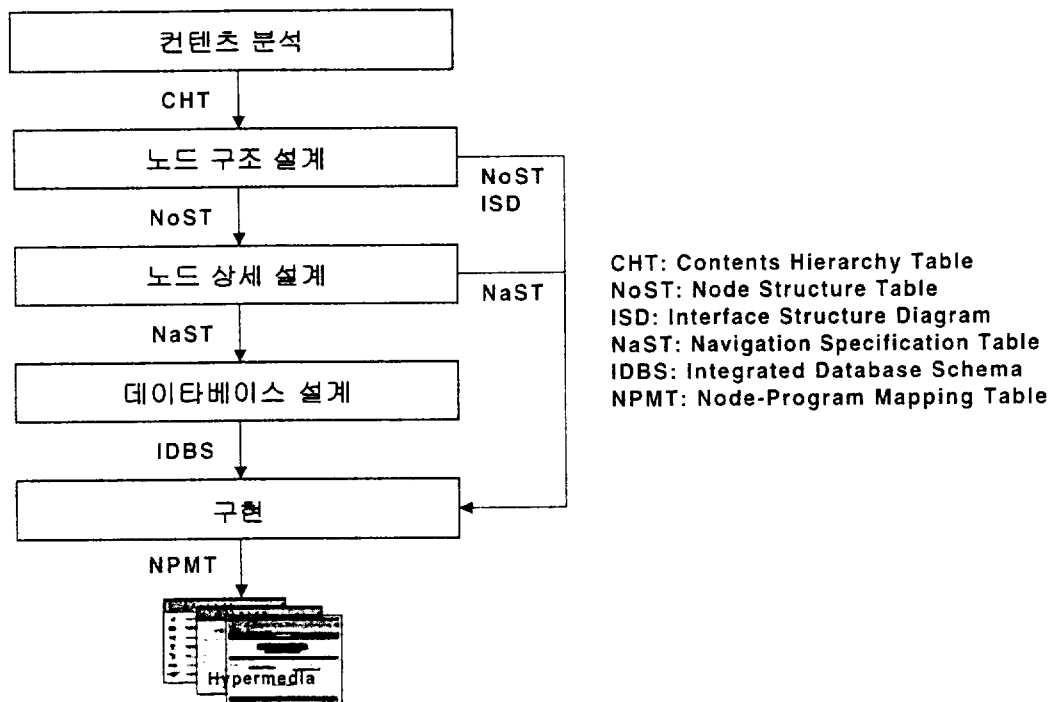
II. 방법론 아키텍처

본 방법론은 컨텐츠 중심의 하이퍼미디어 구축을 위해 제안된다. 컨텐츠는 사용자의 관점 (view)에 따라 적절한 단위로 세분화할 수 있는데, 이러한 컨텐츠의 분할 구조에 대한 사용자의 인지를 효과적으로 반영하기 위해서는 계층적 구조로 구성하는 것이 효과적이다. 이러한 계층적 컨

텐츠 구조를 하이퍼미디어 시스템에서 효과적으로 구현하기 위해 인덱스 개념을 중심으로 개발 방법론 (IHDM)을 제안한다. 이러한 접근법은 하이퍼링크 기능의 유용성을 최대한 활용할 수 있도록 한다는데 의의가 있다. IHDM의 개발 단계를 보여주는 아키텍처는 <그림 1>과 같다. 연관된 단계 간의 피드백 관계는 개략적 표현을 위하여 생략하였다.

컨텐츠 분석 단계 (단계1)에서는 개발하고자 하는 컨텐츠를 정의한다. 일차적으로, 사용자의 요구사항에 따라 컨텐츠 목차를 정하고 이 목차를 세분화 한다. 나아가서, 컨텐츠에 대한 사용자 그룹을 구분하여 접근권한제어를 정의하고, 각각의 컨텐츠 목차에 대하여 담길 내용 및 역할을 기술한다. 이러한 과정은 Contents Hierarchy Table (CHT)을 이용하여 수행한다.

노드 구조 설계 단계 (단계2)에서는 앞 단계에서 정의된 컨텐츠를 역할의 관점에서 세분화된 노드 (인덱스 노드, 컨텐츠 노드, 기능 노드)를



<그림 1> IHDM 아키텍처

기반으로 단위를 정의하고 이들간의 구성 링크 (structural link)를 결정한다. 구성 링크는 노드 간의 연관 구조에 대한 관점에서 설계되는데, 인덱스 노드를 기반으로 하이퍼미디어 전체의 구성에 대한 접근의 효율성을 높이는데 중점을 둔다. 설계 과정에서 Node Structure Table (NoST)이 이용된다. 또한, 설계된 노드를 인터페이스에 표현하기 위한 인터페이스 구조를 설계하는데, 결과로서 Interface Structure Diagram (ISD)를 완성한다.

노드 상세 설계 단계 (단계3)에서는 NoST를 기반으로, 각 노드의 세부적인 요소들을 정의하고, 이 요소들을 이용하여 컨텐츠 간의 상호 관련성에 의거한 참조 링크 (referential link)를 설계한다. 이 과정에서 네비게이션 메커니즘을 설계 요소에 포함시킴으로써, 체계적인 네비게이션 설계를 수행할 수 있다. 이와 같은 설계 결과는 Navigation Specification Table (NaST)로 정리된다.

데이터베이스 설계 단계 (단계4)에서는 노드 상세 설계 단계에서 작성된 NaST를 기반으로 데이터베이스 구축을 위한 설계를 수행한다. 하나의 NaST는 하나의 하이퍼문서에 관한 뷰 (view) 단위로서, 여기에 포함된 요소들 중 데이터베이스로 관리할 필요가 있는 요소들을 추출하여 데이터 뷰로 활용한다. 데이터 뷰 설계가 끝나면, 속성 (attributes)에 대한 이름 상충해소 (naming conflict resolution) 과정을 통해 통합 데이터베이스 스키마 (Integrated Database Schema, IDBS)를 생성한다.

마지막으로, 구현 단계 (단계5)에서는 앞 단계들에서 생성된 NoST, ISD, NaST, 그리고 IDBS를 기반으로 하이퍼미디어 시스템이 구현된다. 인터페이스의 구현, 데이터베이스 구축, 그리고, 인터페이스상에서 노드와 데이터베이스의 연동 과정을 수행하기 위한 프로그래밍이 주요 업무이다. 이 방법론의 각 단계에서 산출된 결과물은 시스템의 유지보수 과정에서 유용하게 사용될 수

있으며, 이러한 맥락에서, 구현 단계에서는 각 노드와 구현된 프로그램 파일간의 복잡한 관계를 기록한 Node-Program Mapping Table (NPMT)가 생성된다. NPMT는 소프트웨어와 프로그래밍 방식에 따라 달라질 수 있는 노드와 프로그램 파일간의 관계에 대한 정보를 제공하여 유지보수를 효과적으로 수행할 수 있도록 한다.

Ⅲ. 방법론 단계

본 절에서는 IHDM의 각 단계를 HyDoMiS (Hyperdocument Meta-information System)에 대한 개발 사례를 예로서 사용하여 구체적으로 설명한다. 추가적으로 IHDM 방법론은 개발 뿐 아니라 하이퍼미디어 시스템의 유지 및 보수 시에도 유효하게 사용 가능하며 이에 대한 사례로서 인터넷 시스템 구축 사례를 부록에서 예시한다.

HyDoMiS 는 메타-정보시스템 [서우종과 이희석, 1999]의 하나로써, 조직의 프로세스를 지원하기 위한 하이퍼미디어 문서에 대한 유지보수를 목적으로 개발되었다. 하이퍼미디어 문서에 관한 메타데이터 기반으로 조직의 프로세스와 연관된 요인들에 초점을 맞춘 조직적 관점과 시스템 요인들과 관련된 기술적 관점의 메타정보들을 제공하고 관리한다. 즉, HyDoMiS는 기업의 하이퍼미디어와 연관된 워크플로우 관련된 메타정보 및 처리 데이터, 네비게이션, 인터페이스 소스, 데이터베이스 등에 관련된 메타정보들을 중심으로 검색 및 리포팅과 같은 기능을 제공한다. 이와 같이 HyDoMiS는 기업의 업무 흐름과는 상관 없이 하이퍼미디어 문서를 관리하는데 필요한 컨텐츠로 구성되어 있는 컨텐츠 지향적 시스템이다.

3.1 단계1: 컨텐츠 분석

하이퍼미디어 시스템을 개발하기 위해서는, 우선 구현하고자 하는 컨텐츠를 개발하고 이를

컨텐츠 지향적 하이퍼미디어 설계 방법론

Ver.	1.1	Date	1999.1.26	
User Group (UG)	MG: System Management Group (System Analysts, Developers, Administrators)			
Contents Structure		Information-related Description [Function-related Description]		
Metadata Management		메타데이터의 저장, 메타정보의 생성에 관한 기능을 제공함. [생성, 수정, 삭제]		
	Workflow Analysis		워크플로우에 관련된 메타데이터를 분석함.	
	Domain Analysis	도메인 수준에서의 워크플로우 분석을 기반으로 한 메타데이터를 관리함. (워크플로우 도메인, 에이전트 도메인, 태스크 도메인, 문서 도메인)		
		Specification Analysis		세부적인 수준에서의 워크플로우 분석을 기반으로 한 메타데이터를 관리함. (문서, 에이전트 객체, 태스크)
	Document Analysis		기업 하이퍼문서에 관한 구체적인 사양들에 대한 메타데이터를 관리함.	
	System Specification	관련 파일과 그것의 위치, 하드웨어, 소프트웨어에 관한 메타정보를 관리함.		
		Document Attribute		문서에서 처리되는 데이터의 개념적 속성과 연관되는 데이터베이스 관련 정보 제공.
	Navigation		해당 문서를 네비게이션 시킬 수 있는 소스 노드에 대한 메타정보 및 해당 문서가 네비게이션 시킬 수 있는 목적 노드에 대한 메타정보를 관리함. 목적 노드는 문서노드와 데이터 노드, 두 가지로 구분하여 각각에 대한 메타정보를 관리함.	
	Interface Source		하이퍼미디어 문서에 나타나는 멀티미디어 요소들에 대한 메타정보를 관리함.	
	Log		해당 문서의 유지보수에 필요한 메타정보들을 관리함. 버전, 갱신 날짜, 시스템에 올린 날짜, 시스템에서 철수 시킨 날짜, 감독자, 설명등에 관한 정보를 관리함.	
	Database Analysis		기업 하이퍼문서와 연동되는 데이터베이스에 대한 메타정보를 관리함.	
	Database Information	데이터베이스의 DBMS, 하드웨어, 저장위치 등에 관한 정보를 관리함.		
		Table Information		데이터베이스의 테이블, 애트리뷰트 등에 관한 상세 정보들을 관리함.
	* Search		필요한 하이퍼문서를 찾을 수 있는 기능을 제공함. [검색 결과들 중 특정 문서를 선택하여, 선택된 문서 번호가 자동으로 리포트 생성 모듈로 전달되도록 함.]	
		Drill-Down Search		쿼리를 통해 범위를 좁혀나가면서, 범위에 해당하는 문서를 검색함.
Keyword Search		키워드를 이용한 쿼리 결과를 제공함.		
Report		문서 번호를 이용하여 문서 관련한 메타정보들을 리포트 형식으로 제공함. 리포트 내용의 범위를 선택할 수 있도록 함.		
Maintenance Notes				
[*] search 결과를 report 모듈로 자동으로 제공할 수 있는 기능을 추가함.				

<그림 2> HyDoMiS 개발을 위한 Contents Hierarchy Table

체계적으로 정리할 필요가 있다. 컨텐츠 분석을 위해서 시스템 분석가는 사용자의 요구분석을 기반으로 컨텐츠의 단위를 결정하고 이 단위를 보다 세분화 해나가는 절차를 수행한다. 일반적으로 컨텐츠는 계층적 구조화가 가능한데, 아래 <그림 2>와 같은 Contents Hierarchy Table (CHT) 을 이용하여 효과적으로 표현할 수 있다. <그림 2>의 CHT는 HyDoMiS의 컨텐츠를 계층적 구조로 표현하고 이에 대한 상세한 설명을 보여준다.

컨텐츠에 대한 개발은 정보와 기능 두 가지 측면에서 접근할 수 있다. 유용한 정보 제공 및 교환은 하이퍼미디어 시스템의 부가가치 창출을 좌우하므로, 구축 목적의 효용을 극대화할 수 있는 아이디어 개발이 중요하다. 기능에 대한 고려는 시스템에서 다루는 정보의 활용성을 제고하는 측면에서 필수적이다. 따라서, CHT는 컨텐츠에 대한 설명을 정보-관련된 것과 기능-관련된 것으로 구분하고 있다. <그림 2>에서 "Metadata Management"에 관한 컨텐츠 단위에 기술된 생성, 수정, 삭제와 같은 기능은 하위 컨텐츠 단위에 계승된다. 상위 컨텐츠 단위에서 계승되지 않는 기능은 해당 컨텐츠 단위에만 기술하면 된다. 직접적으로 전자상거래를 지원하기 위한 시스템의 경우라면, 상품 조회, 주문 및 결제에 필요한 기능들이 기술될 수 있다.

CHT의 컨텐츠 계층 구조는 다음 단계인 노드 구조 설계과정에서 인덱스 노드로 변환된다. 이 인덱스 노드는 응용 시스템의 기본 메뉴로 구현되는데, CHT의 컨텐츠 계층 구조는 이러한 관점에서 세분화 수준을 결정하면 된다. <그림 2>에서 컨텐츠 단위의 세분화 수준은 2단계의 깊이로 되어 있다. 수준의 정도는 구현할 시스템이 대상으로 하는 컨텐츠의 복잡성 또는 깊이에 따라 달라질 수 있는데, 사용자 인터페이스의 효율성을 고려할 때, 통상적으로 2단계 또는 3단계의 깊이로 구성하는 것이 적절하다.

프로세스 지향적인 하이퍼미디어 시스템은 조직의 업무 흐름에 따라 수행되어야 할 과업이

정형화되어 있어 이에 대한 요구분석 범위를 정의하기가 용이하다. 그러나, 주로 정보서비스 및 광고 분야에 응용되는 컨텐츠 지향적 하이퍼미디어 시스템은 시스템 효과를 높이기 위한 다양한 아이디어에 의존하는 경향이 강하다. 따라서, 개발 과정에서 일반적으로 컨텐츠에 대한 요구사항이 지속적으로 변화되고 구체화되는 과정에서 컨텐츠 단위에 대한 수정이 반복된다. 이러한 이유로 컨텐츠 지향적 하이퍼미디어 시스템은 프로토타입 기반의 (prototype-based) 접근 방법에 의해 개발되는 것이 일반적이다. 시스템 개발이 완료된 이후에도 컨텐츠 확장 또는 통합 등의 변화가 끊임없이 일어날 수 있다. 따라서, CHT는 이러한 변화에 대한 버전을 관리할 수 있도록 고안 되었다. 개선 과정에서 피드백에 대한 내용을 CHT의 관련 컨텐츠 단위에 표시를 하고, 자세한 내용을 "Maintenance Notes" 란에 기술할 수 있다. 이와 같은 개선 작업에 대한 정보가 축적되면, 지속적인 유지보수 과정에서 개선 방향성에 대한 유용한 정보로 활용할 수 있다.

한편, HyDoMiS 개발 사례에서는, 주요 사용자는 시스템 관리자 또는 개발자로서, 시스템 컨텐츠에 대한 사용자 구분이 필요 없는 경우이지만, 일반적으로 인트라넷 시스템 또는 전자상거래 시스템은 종종 사용자 구분에 따라 컨텐츠 제공 범위를 달리한다. 이러한 경우, 컨텐츠 앞에 사용자 도메인을 표시하면 된다. 이에 대한 구체적인 예는 4절의 유지보수 사례에 소개된 CHT에서 볼 수 있다.

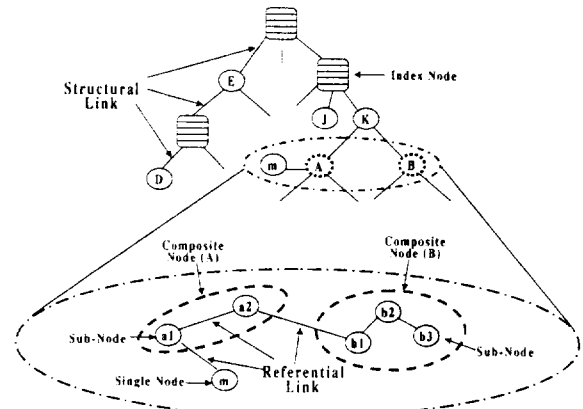
3.2 단계2: 노드 구조 설계

하이퍼미디어의 가장 기본적인 구성 요소는 노드와 링크로써 이는 하이퍼미디어 설계의 핵심적인 대상이다 [Mayfield and Nicholas, 1993, DeRose, 1996]. 링크는 하이퍼링크 (hyperlink)라고 불리기도 하는데 [Mabrouk et al., 1991], 노

드간의 접근 경로 역할에 한다. 링크를 통해 다른 노드로 접근(access)하는 방식을 네비게이션 (navigation)이라 한다. 네비게이션은 하이퍼미디어의 가장 핵심적인 특징이므로 하이퍼미디어 시스템을 네비게이션 기반의 시스템 (navigation-based system)이라 부르기도 한다 [Isakowitz and Bieber, 1995]. 기존의 문헌에서는, 노드에 대해서, 하이퍼텍스트의 정보 단위 (information unit) [Nielsen, 1993], 하이퍼텍스트 조각들 (fragments) [Graham, 1994], 또는 정보의 조각들 (fragments) [Fluckiger, 1995]과 같이 기술 하고 있다. 그러나 본 연구에서는 보다 구체적으로, 노드를 "하이퍼미디어에 연결되어 네비게이션이 가능한 구조적 단위 (structured units to be navigated in a hypermedia)"로 정의하고자 한다. "단위"의 의미는 객체 (object) 개념으로도 해석 가능한데, 하이퍼미디어 설계 핵심 요소인 노드의 단위에 대한 관점을 세분화 하여, 보다 정밀한 설계를 할 수 있다. 또한, "구조적"이란 의미는 하나의 노드는 보다 작은 단위의 부속 노드 (sub-node)들로 구성될 수 있다는 것이다. 이러한 복합 구조를 가지는 노드를 복합 노드 (composite node)라 정의하고, 이와는 대조적으로, 단일한 속성으로만 이루어진 노드를 단일 노드 (single node)라 정의하고자 한다.

노드에 대한 개념은 링크 방향 또는 속성 등의 여러 가지 관점에 따라 세분화 될 수 있는데 [Lee and Suh, 1999], 본 연구에서는 역할 관점에 따라 인덱스 노드와 콘텐츠 노드, 그리고 기능 노드 개념을 이용한다. 인덱스 노드는 콘텐츠에 대한 분류 사항 및 세부적인 요소들에 대한 인덱스를 포함하며, 콘텐츠 노드는 콘텐츠 자체를 표현하는데 이용된다. 그리고 기능 노드는 콘텐츠 노드의 관리 및 활용에 필요한 기능적 역할을 수행한다.

단계2에서는, 이와 같은 세 가지의 노드 개념과 CHT를 기반으로 구현할 노드를 정의함과 동시에 이들간의 관계를 설정한다. 이러한 작업은



<그림 3> 구성 링크와 참조 링크에 대한 개념

인덱스 노드를 근간으로 콘텐츠 노드 전체를 대상으로 접근 효율성을 높이는데 초점을 맞춘다. 이와 같은 관점에서 설정된 링크를 구성 링크 (structural link)라 부를 수 있다. 반면, 단계3에서도 링크에 대한 고려를 하는데, 이는 콘텐츠 노드의 세부적인 요소들을 정의하는 과정에서, 이들간의 연관관계를 설정하는 관점에서, 참조 링크 (referential link)라 부를 수 있다. 구성 링크와 참조 링크에 대한 개념은 <그림 3>과 같이 정리할 수 있다.

구성링크를 설계하는 과정에서 인덱스 노드는 콘텐츠에 대한 효율적인 접근을 위한 매우 유용한 수단으로 활용될 수 있다. 인덱스 노드는 콘텐츠에 대한 적용 범위에 따라, (i) 콘텐츠의 전체 범위를 대상으로 하는 광역 인덱스 (global index) 노드와 (ii) 부분적인 콘텐츠 범위에만 적용되는 지역 인덱스 (regional index) 노드로 대별된다. 광역 인덱스 노드는 콘텐츠 노드들에 대한 초기 접근 구조를 형성하는데, 일반적으로 하이퍼미디어 시스템의 모든 인터페이스에서 볼 수 있도록 인터페이스 일부분에 위치하도록 구현한다. 광역 인덱스 노드가 너무 자세한 인덱스 수준까지 포함하면 사용자의 인지적 과정에 부담을 줄 수 있으므로 하이퍼미디어의 전체적인 구성 내용 및 역할을 파악할 수 있는 수준에서 인덱스 항목이 결정되는 것이 바람직하다. 지역

컨텐츠 지향적 하이퍼미디어 설계 방법론

Ver.	3.1	Date	1999.1.21
Notation	Node	Index Node : (1) Global Index [GI: Seq.], (2) Regional Index [RI: Seq.] Contents Node : [C: Seq.], Function Node : [F: Seq.] (Seq. : A, B, ...)	
	Index Item	Global Index Item : (GSeq.)	
Node Structure			
[GI] Global Index			
(G1) Go Home			
(G2) Metadata Management			
(G21) Workflow Analysis			
(G211) Domain Analysis			
[RI: 1] Domain Analysis			
[C: 1-1] Workflow Domain Information Management			
[F: 1-1-A] Creating Workflow Domain			
[F: 1-1-B] Editing Workflow Domain			
● ● ●			
(G22) Document Analysis			
(G221) System Specification			
[C: 3] System Specification			
[F: 3-A] Creating System Specification			
[F: 3-B] Editing System Specification			
(G222) Document Attribute			
[RI: 4] Document List			
[F:4-1(a)] Editing Document Basic Information			
[C:4-1(b)] Attribute Information Management			
[F: 4-1(b)-A] Creating Attribute Information			
[F: 4-1(b)-B] Editing Attribute Information			
(G223) Navigation			
[RI: 5] Document List			
[RI: 5-1] Destination Node Type Selection			
[C: 5-1-1] Document Node Information Management			
[F: 5-1-1-A] Creating Doc. Node Specification			
[F: 5-1-1-B] Editing Doc. Node Specification			
[C: 5-1-2] Data Node Information Management			
[F: 5-1-2-A] Creating Data Node Specification			
[F: 5-1-2-B] Editing Data Node Specification			
[C:5-2] Source Node Information			
● ● ●			
(G3) Search			
[RI: 9] Search			
[F: 9-1] Drill-Down Search			
[C: 9-1-1] Drill-Down Search Result			
* [F: 10] Report Generation			
[F: 9-2] Keyword Search			
[C: 9-2-1] Keyword Search Result			
* [F: 10] Report Generation			
(G4) Report			
[F: 10] Report Generation			
[C: 10-1] Generated Report			
Maintenance Notes			
[*] Drill-down search result와 Keyword search result 노드에서 결과값 중 선택된 값이 report generation 노드로 직접 자동 전달될 수 있도록 함.			

<그림 4> HyDoMIS의 NoST

인덱스 노드는 광역 인덱스 노드에 의해 접근할 수 있는 수준이하의 계층 구조를 대상으로 하는데, 컨텐츠에 대한 세부적인 부분에 접근이 이루어진 후 나타나도록 구현되는 것이 일반적이다. 구성 링크의 거점 단위 역할을 하는 인덱스 노드에 대한 설계는 하이퍼미디어 운영 방식에 직접적인 영향을 미친다. 따라서, 시스템 효율성과 사용자 만족도를 위해 인덱스 노드들의 효과적인 구성 및 배치는 중요한 설계 초점이다.

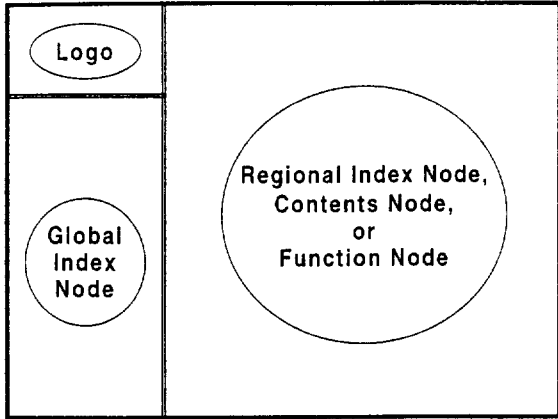
노드 구조 설계 단계에서는 지금까지 설명한 노드와 링크에 대한 개념을 활용하여 Node Structure Table (NoST)를 작성한다. NoST는 CHT를 기반으로 작성되는데, 인덱스 노드, 컨텐츠 노드, 그리고 기능 노드에 대한 제목과 이들에 대한 접근 구조를 표현한다. CHT에서 정보-관련 기술 (information-related description)을 근거로 컨텐츠 노드로 설계되며, 기능-관련 기술 (function-related description)를 기반으로 기능 노드가 설계된다. 그리고, CHT의 컨텐츠 구조 (contents structure)에 따라 광역 인덱스 노드가 설계된다. NoST에서 인덱스 항목 (index item)에 대한 표현은 광역 인덱스에 적용한다. 이 항목들은 노드들의 구조를 이해할 수 있는 직접적인 단서가 되므로, NoST에서 이를 표현하여 시스템 구성에 대한 이해를 용이하게 한다. 이와 같은 방법에 따라서 HyDoMiS의 NoST는 <그림 4>와 같이 생성되었다. 여기서, 광역 인덱스의 깊이는, 예를 들어, (G221)과 같이 2단계까지 허용하고 있다. 광역 인덱스에 모든 노드로 접근할 수 있는 항목들을 포함시키면 노드의 계층 구조 때문에 인덱스의 깊이가 매우 깊어지게 된다. 이러한 경우, 광역 인덱스 노드 자체가 너무 커지고 복잡해져서 시스템 효율성에 부정적인 효과를 가져올 수 있다.

<그림 4>에서 볼 수 있듯이, 특정 하이퍼미디어 문서와 관련된 문서 노드들에 대한 네비게이션 정보를 관리하기 위한 노드, [C:5-1-1]로 접근하기 위한 절차는 다음과 같다. 우선, 광역 인덱스

노드의 (G2) Metadata Management, (G22) Document Analysis, (G223) Navigation의 순서를 거쳐 지역 인덱스 노드 [RI:5]에 도달하고, 이 노드에서 특정 문서를 선택하면 또 하나의 인덱스 노드 [RI: 5-1]에 도달하게 된다. 여기서 선택한 특정 문서에 관련된 노드 종류가 인덱스 항목으로 제시되므로 여기서 문서 노드를 선택하면 [C:5-1-1]에 도달할 수 있다. 이 과정에서, [RI:5]는 [RI: 5-1]와 [C:5-2]를 동시에 네비게이션 시키도록 설계되었기 때문에 구현 단계에서, [RI: 5-1]와 [C: 5-2]는 복합 노드로 구현되었다.

노드 구조 설계단계의 두번째 작업은 NoST를 기반으로 인덱스 노드, 컨텐츠 노드, 그리고 기능 노드에 대한 인터페이스 구성을 설계하는 것이다. 하이퍼미디어 시스템은 일반적으로 몇 가지의 인터페이스 유형으로 구축되어 인터페이스의 일관성을 유지한다. 이로써, 사용자의 하이퍼미디어 운영에 대한 이해를 증진시키는 효과가 얻어진다. 우선적으로 고려할 것은 인덱스 노드를 어떤 방식으로 구현할 것인지에 대한 전략인데, 이는 크게 세 가지 경우로 나누어 볼 수 있다. 첫째는, 인덱스 노드를 하나의 인터페이스로 구성하는 것이다. 둘째는, 인터페이스 일부분에 배치하는 것이다. 셋째는, 이 두 가지의 경우를 혼합적으로 적용하는 경우이다. 최근에는 하나의 인터페이스를 여러 개의 하이퍼문서에 의해 복합적으로 구성되며, 인터페이스의 일부분에 항상 인덱스가 보이도록 구현하는 방식이 일반적인 유형으로 활용되고 있다. 이 경우, HTML의 프레임 (frame) 태그를 이용하면, 손쉽게 구현할 수 있다.

단계2에서는 이러한 인터페이스 유형을 결정하고, 이것을 그림으로 작성한다. 하나의 시스템에서는 인터페이스 유형을 몇 가지 사용하는 경우가 종종 있기 때문에, 이 유형에 대한 설계를 함으로써, 개발자로 하여금 일관성 있는 인터페이스를 구현할 수 있도록 할 수 있다. HyDoMiS 개발 과정에서는, 광역 인덱스 노드는 분할된 인



<그림 5> HyDoMiS의 Interface Structure Diagram

터페이스의 한 부분에 고정시키고 지역 인덱스는 이와 다른 영역에 표현되도록 설계하였다. 그 결과, HyDoMiS의 Interface Structure Diagram (ISD)는 <그림 5>과 같이 작성되었다. 인터페이스 구조를 설계하는 과정은 인터페이스의 분할과 노드 위치 설정이 주된 관심 내용이지만, 인터페이스에 항상 표현되는 요소에 대한 고려도 필요하다. 아래의 그림에서는, 이러한 요소로써, HyDoMiS 로고가 나타나 있다.

3.3 단계3: 노드 상세 설계

단계2에서는, 개념적 수준에서, 역할 구분에 따른 인덱스 노드, 콘텐츠 노드, 그리고 기능 노드의 개념을 적용한 반면, 단계 3에서는, 물리적 수준에서, 문서 노드와 데이터 노드의 개념을 적용하여 설계를 수행한다. 하나의 문서 노드는 하나의 하이퍼미디어 문서에 대응되는 단위이다. 데이터 노드는 하나의 문서 노드에서 네비게이션 시킬 수 있는 데이터들의 집합 단위라 할 수 있다. 데이터 노드는 멀티미디어 속성을 가질 수 있으며, 데이터베이스에 저장되고 사용자 뷰에 따라 집합적으로 네비게이션 될 수 있다. 데이터 노드가 문서 노드와 항상 일정한 관계를 가지고 있는 경우, 고정 노드 (fixed node) 또는 정적 노드 (static node)라 할 수 있다. 반면, 문

서 노드에서 데이터 노드에 대한 조건을 임의로 바꿀 수 있어서 네비게이션 되는 데이터 노드의 내용이 가변적인 경우, 이러한 데이터 노드를 동적 노드 (dynamic node)라 한다.

이와 같은 문서 노드와 데이터 노드에 대한 개념을 기반으로, 단계 3에서는 다음과 같은 두 가지 초점을 가지고 설계 과정을 수행한다. (i) 첫째는 하나의 문서 노드에 포함되는 내용들에 대한 구체적인 콘텐츠 항목과 관련된 데이터 노드에 대한 요소들을 결정하는 것이고, (ii) 둘째는 네비게이션 메커니즘을 설계하는 것이다. 네비게이션 메커니즘은 노드들간의 복잡한 하이퍼링크를 구현하기 위한 방법 또는 원리로 정의할 수 있는데, 하이퍼미디어 기술의 발달과 더불어 메커니즘의 유형도 다양화되고 있다. 따라서, 보다 효과적인 기능을 제공하는 하이퍼미디어를 구축하기 위해서는 네비게이션 메커니즘을 설계 단계에서 고려하는 것이 중요하다.

네비게이션의 유형은 크게 노드간 네비게이션 (inter-node navigation)과 노드내 네비게이션 (intra-node navigation)으로 구분할 수 있다. 단계 3에서는, 문서 노드와 데이터 노드 개념을 기반으로 설계를 수행하는 입장이므로, 이 두 가지 노드 개념에 메커니즘을 적용하여 사용한다. 노드간 네비게이션은 하나의 노드에서 다른 노드를 네비게이션 시키는 경우이다. 노드내 네비게이션은 (i) 노드 내에서 특정 부분을 가리키기 위하여 특정 부분을 강조 되는 위치 보일 수 있도록 하거나 (ii) 노드간 네비게이션을 통해서 네비게이션 된 데이터 노드가 하나의 인터페이스에 다 보여질 수 없을 때, 이 노드의 내용을 분할하여 2차적으로 네비게이션 시키는 것이다. 이러한 네비게이션 유형에 따라 세부적인 네비게이션 메커니즘이 적용 노드에 대한 내용은 <표 1>과 같이 정리할 수 있다. 네비게이션 유형에 따라 네비게이션 메커니즘들은 문서 노드와 데이터 노드에 대해 적용 경우가 각기 차이가 있음을 알 수 있다.

<표 1> 네비게이션 메커니즘

네비게이션 메커니즘	노드간 (Inter-Node)		노드내 (Intra-Node)	
	문서 노드	데이터 노드	문서 노드	데이터 노드
단순 (Simple)				
의미적 관계 (Semantic Relationship)	Y	Y	Y	Y
슬라이드 (Slide)				
전호 (Back-and-Forth)	Y	Y	N	Y
일련번호 (Serial Number)	Y	Y	N	Y
기능적 (Operational)				
쿼리 (Query)				
키워드 (Keyword)	N	Y	N	Y
선택 (Selection)	N	Y	N	Y
실행 (Executable)	Y	Y	N	Y

네비게이션 메커니즘은 크게 단순 네비게이션과 기능적 네비게이션으로 구분할 수 있는데, 단순 네비게이션 메커니즘은 앵커가 직접적으로 연결된 노드를 네비게이션 시키는 것을 의미한다. 반면, 기능적 네비게이션 메커니즘은 앵커를 클릭했을 때, 어떤 기능 수행 과정을 거쳐서, 그 결과 생성된 노드를 네비게이션 시키는 메커니즘을 의미한다. 단순 네비게이션은 목적 노드에 대한 의미를 가지는 앵커를 이용하는 방식과 의미적 관계와는 상관없는 전후 관계 또는 일련번호를 이용한 슬라이드 방식으로 구분할 수 있다. 기능적 네비게이션 메커니즘은 쿼리를 이용한 방식과 연산 또는 시스템적 로직을 통해 네비게이션 시키는 방식으로 구분할 수 있다.

의미적 관계를 이용한 메커니즘은 가장 전통적인 네비게이션 방식으로서, 노드간 네비게이션의 경우, 어떤 단어를 클릭했을 때, 그 단어에 대한 자세한 내용을 담고 있는 문서 노드 또는 데이터 노드가 네비게이션 된다. 노드내 네비게이션의 경우에는, 하나의 문서 노드 안에서 앵커와 관련된 내용으로 위치 변경하는 효과를 보이게 하는 경우 이용된다. 슬라이드 메커니즘은 노드간 네비게이션의 경우, 전후 관계, 또는 "1장,

2장, 3장" 등과 같은 단순 일련번호를 이용하여 해당 문서 노드를 네비게이션 시킬 수 있다. 이러한 방식은 이미지와 같은 데이터 노드를 네비게이션 시키는데도 사용된다. 노드내 네비게이션의 경우, 슬라이드 네비게이션은 "처음(first), 앞(back), 뒤(forth or next), 끝(end)"과 같은 앵커를 이용하여 하나의 문서 노드 내에서 데이터 노드를 반복적으로 네비게이션 시킨다. 일련번호 방식은 이미 불러진 데이터 노드의 양이 많아서 화면에 다 표현될 수 없는 경우, "1,2,3,n"과 같은 할당 앵커를 이용하여 원하는 데이터 노드를 선택하여 불러낼 수 있다. 이 두가지 방식이 혼합되어 사용되는 경우가 종종 있다.

기능적 메커니즘 중 쿼리 방식은 사용자가 입력한 키워드를 이용하는 방식과 쿼리 범위에 대한 제약조건을 선택하여 데이터 노드를 네비게이션 시키는 방식으로 세분된다. 선택 방식은 이미 제시되어 있는 범위들 또는 "가, 나, 다" 등과 같은 단순 항목들을 대상으로 쿼리를 수행하는 방식이다. 이러한 두 가지의 쿼리 네비게이션 메커니즘은 종종 검색 엔진에서 복합적으로 활용된다. 노드간 네비게이션의 경우, 실행 네비게이션 메커니즘은 문서 노드 및 데이터 노드 모두에 적용가능한데, 데이터 노드에 적용 예로는 수식 연산과 같은 로직을 수행하여 그 결과를 네비게이션 시키는 경우를 들 수 있다. 또한, 문서 노드에 적용하는 경우는, 접근 통제 또는 데이터 처리와 같은 로직을 수행했을 때, 그 결과에 따라 여러 가지 고지(notification)에 대한 문서를 네비게이션 시키는 경우를 예로 들 수 있다. 한편, 노드내 네비게이션의 경우, 기능적 네비게이션 메커니즘은 노드간 네비게이션 메커니즘에서와 설명한 원리와 동일하다. 네비게이션된 데이터 노드의 양이 많을 때, 이 노드를 대상으로 2차적인 쿼리를 수행하여 원하는 결과를 집약시켜나가는 방식은 하이퍼미디어 시스템의 기능 수준을 한 층 더 높일 수 있는 방안이 될 것이다.

네비게이션 메커니즘은 하이퍼미디어 기술 발달

<표 2> NaST 작성을 위한 표기법

Category		Description/Representation Format		Notation
Operator				
	AND	Can be used for Properties and Roles		+
	OR	Can be used for Properties and Roles		
Property				
	Text	A Word, Phrase, or Sentences		T
	Image	Image Data		I
	Animation	Animation Data		A
	Video	Video Data		V
	Sound	Sound Data such as Music, Voice		S
	Button	HTML Button		Bu
	Box	Text Box, Combo Box, List Box, Check Box		Bo
Role				
Simple		S(Type)		
	Type	Show information		S
		Get a typed information		G
Anchor		A		
Event		E(Related Unit#:Operation)		
	Related Unit	Data Node		DN
		Data Element		DE
	Operation	Data	Save	Sa
			Deletion	De
			Query	Qu
		Window	Functional Window	FW
			Browser Window	BW
		Others	Download	Bo
Destination Unit				
	Data Node	DN(#)		
	Data Element	DE(#)		
	Notification	N(#)		
Navigation Mechanism				
	Semantic Relationship	SR		
	Back-and-Forth	B&F		
	Serial Number	SN		
	Keyword	Q(K)		
	Selection	Q(S)		
	Executable	Ex		

에 직접적인 영향을 받으므로, 향후, 보다 진보된 하이퍼미디어 기술이 개발되면, 새로운 네비게이션 메커니즘 또한 개발될 수 있을 것이다.

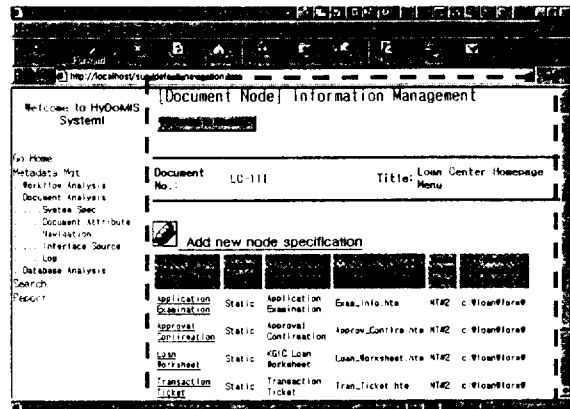
단계3에서는 지금까지 설명한 네비게이션 메커니즘에 대한 적용한 문서 노드의 내역에 대해 NaST (Navigation Specification Table)를 이용하여 설계한다. NaST는 단계 2에서 작성된 NoST를 기반으로 작성된다. NaST를 작성하는데 필요한 표기법은 <표 2>와 같다. NoST의 표기에 대한 범주는 연산자 (operator), 속성 (property), 역할 (role), 목적 단위 (destination unit), 그리고 네비게이션 메커니즘 (navigation mechanism)으로 구분된다. 속성은 하이퍼미디어 정보의 속성 및 인터페이스 요소에 대한 것이고, 역할은 단순 (simple), 앵커 (anchor), 그리고 이벤트 (event)로 구분되어 설계 요소들의 역할을 규정하는데 이용된다. 연산자는 속성과 역할에 적용된다. 목적 단위는 데이터 노드 (data node), 데이터 요소 (data element), 그리고 고지 (notification)으로 구분되는데, 데이터 요소의 집합이 데이터 노드이며, 고지는 사용자의 시스템에 대한 어떤 행위에 대하여 필요한 메시지를 알리는 단위이다.

NaST는 <그림 6>에서 볼 수 있듯이, 버전 번호, 날짜, 노드 인덱스, 제목등을 포함하는 머리 부분, HTML 부분, 그리고 데이터 노드 부분으로 구성된다. HyDoMiS 개발 사례에서, 문서 노드 [C:5-1-1]은 하나의 하이퍼미디어 문서에 관련된 내역들을 관리하기 위한 것이다. HTML 부분에서, "Creation"은 기능 노드 [F:5-1-1-A]를 네비게이션 시키는 이미지로 된 앵커이며, "Return to Menu"는 지역 인덱스 노드 [RI:5-1]를 네비게이션 시키는 버튼 형태의 앵커임을 알 수 있다. [C: 5-1-1] 노드를 실제로 구현한 인터페이스는 <그림 7>에서 점선의 범위에 해당한다.

기능 노드 [F:1-1-B]의 경우는 데이터 노드를 포함하는 복합 노드로서, <그림 8>와 같이 설계되었다. [F:1-1-B]에 대한 NoST에서, Destination Unit에 "N(8)" 과 "N(9)"을 볼 수 있는데, 이는

Ver.	2.1	Date	1998.12.07	
Node Index	C: 5-1-1			
Title	Document Node Information Management			
HTML Part				
Item	Property	Role	Destination Unit	Navigation Mechanism
Title	T	S		
Creation	I	A	[F:5-1-1-A]	SR
Return to Menu	Bu	A	[I: 5-1]	B&F
Data Node (1)				
Data Element	Property	Role	Destination Unit	Navigation Mechanism
1 Document No. (Current)	T	S		
2 Title	T	S		
3 Anchor Name	T	A	[F:5-1-1-B]	QIS
4 Anchor Type	T	S		
5 Document No. (Destination)	T	S		
6 Node Name	T	S		
7 Node File Name	T	S		
8 Node H/W Name	T	S		
9 Node Location Path	T	S		
Maintenance Notes				

<그림 6> HyDoMiS 의 [C:5-1-1]에대한 NaST



<그림 7> [C:5-1-1]의 구현 인터페이스

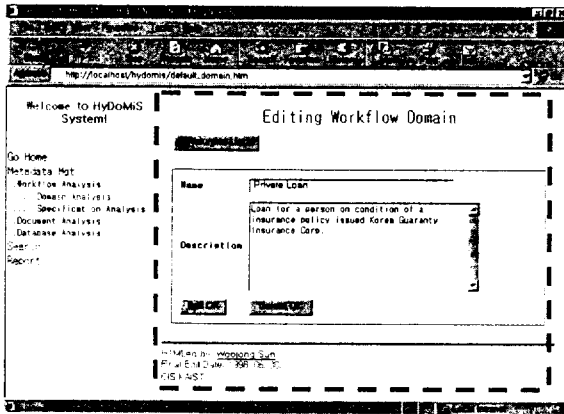
Ver.	2.1	Date	1998.12.07	
Node Index	F: 1-1-B			
Title	Editing Workflow Domain			
HTML Part				
Item	Property	Role	Destination Unit	Navigation Mechanism
Title	T	S		
Return to List	T	A	[C: 1-1]	B&F
Edit OK	Bu	E(DN(1)::Sa)	N(8)	Ex
Delete OK	Bu	E(DN(1)::De)	N(9)	Ex
Log Description	T	S		
Data Node (1)				
Data Element	Property	Role	Destination Unit	Navigation Mechanism
1 Workflow Domain Name	Bo	SIG		
2 Workflow Domain Description	Bo	SIG		
Maintenance Notes				

<그림 8> HyDoMiS의 [F:1-1-B]에대한 NaST

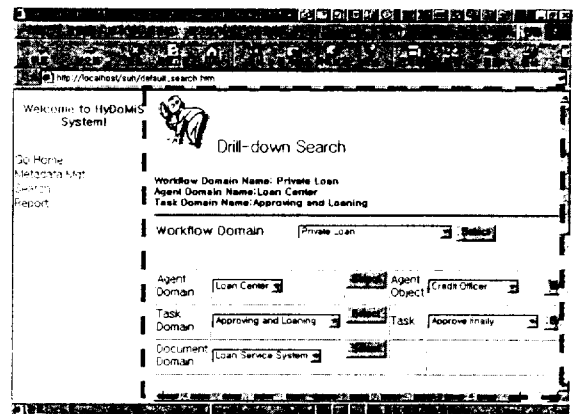
"Edit OK" 와 "Delete OK"라는 이벤트 발생 결과를 알려 주는 역할을 한다. 즉, "Edit OK"라는 버튼을 누르면 이는 데이터 노드 (DN1)을 저장하는 이벤트가 수행되고, 이 이벤트의 수행결과에 따라서 필요한 고지 N(8)을 제공한다. 하나의 고지 단위는 데이터에 대한 처리 결과에 따라 제공될 수 있는 여러 가지의 메시지를 포함하고 있다. 고지 내용은 문서 노드 또는 고지 창 (window)을 통해 제공되도록 구현할 수 있다. 문서 노드 또는 고지 창에 대한 선택은 개발자의 결정 영역

으로 돌린다. HyDoMiS에서 고지 내용은 문서 노드로 구현되었다.

HyDoMiS에서 하이퍼문서들에 대한 검색 모듈 중 드릴-다운 검색을 수행하는 노드에 대한 설계 내용은 <그림 10>과 같다. "Workflow Domain Select" 버튼은 "Data Node (1)" 즉, "DN(1)"에 속한 첫번째 데이터 요소, "DE(1)"에 대해서 쿼리를 수행하고, 이 결과로 네비게이션 되는 데이터 요소는 2, 4, 6 번 요소임을 나타낸다. 노드 [F:9-1]을 구현한 인터페이스는 <그림 11>과 같다.



<그림 9> [F:1-1-B]의 구현 인터페이스



<그림 11> [F:9-1]의 구현 인터페이스

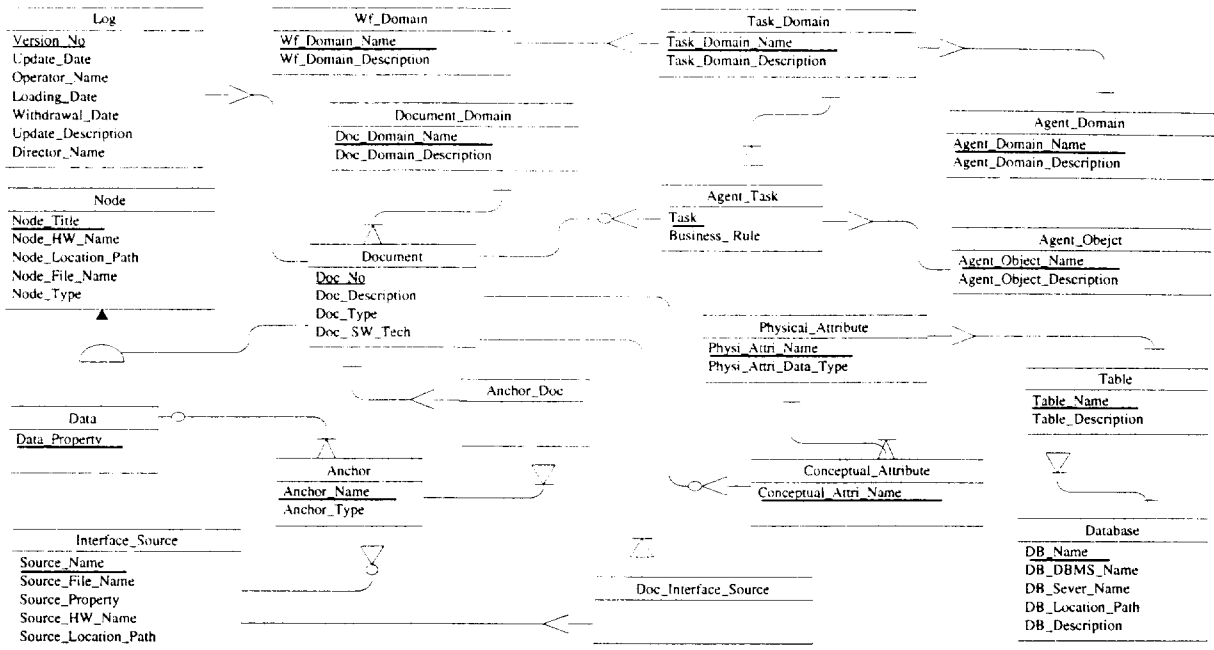
Ver.	2.1	Date	1998.12.23	
Node Index	F:9-1			
Title	Drill-Down Search			
HTML Part				
Item	Property	Role	Destination Unit	Navigation Mechanism
Search Logo	I	S		
Workflow Domain Select	Bu	E:DE(1):Qu	DE(2,4,6)	Q(S)
Agent Domain Select	Bu	E:DE(2):Qu	DE(1)	Q(S)
Agent Object Select	Bu	G		
Task Domain Select	Bu	E:DE(4):Qu	DE(5)	Q(S)
Task	Bu	G		
Document Domain Select	Bu	G		
...
Interface Source Name Select	Bu	G		
Search	Bu	E:DE(1)-DE(1):Qu	[C:9-1-1]	Q(S)
Log Description	T	S		
Data Node (1)				
Data Element	Property	Role	Destination Node	Navigation Mechanism
1 Workflow Domain Name	T	Bo		
2 Agent Domain Name	T	Bo		
3 Agent Object	T	Bo		
4 Task Domain	T	Bo		
5 Task	T	Bo		
6 Document Domain	T	Bo		
...
11 Interface Source Name	T	Bo		
Maintenance Notes				

<그림 10> HyDoMiS의 [F:9-1]에 대한 NaST

3.4 단계4: 데이터베이스 설계

하이퍼미디어 구현은 데이터베이스를 기반으로 구축되는 경우가 대다수이며 전자상거래를 위한 하이퍼미디어 시스템 구축에 있어서 데이터베이스의 활용은 필수적이다. 따라서, 하이퍼미디어 기술 발전의 주요 관심 사항 중 하나가 데이터베이스와의 연계이다.

그러므로 하이퍼미디어를 효과적으로 구축하기 위해 제시된 대부분의 기존 하이퍼미디어 설계 방법론들은 하이퍼미디어를 통해 구현되는 정보에 대한 설계를 중심으로 하는 접근법을 사용하고 있다. 이들 방법론이 기반을 두고 있는 데이터 모형은 개체관계 (Entity-Relationship) 모형과 객체지향 (Object-Oriented) 모형으로 구분할



<그림 12> HyDoMiS의 데이터베이스 스키마

수 있다. 개체관계 모형에 기반한 방법론으로는 RMM [Isakowitz et al., 1995, 1997]과 VHDM [Lee et al., 1996, 1999a] 이 있고, 객체지향 모형에 기반한 접근법을 취하는 방법론으로는 OOHDM [Schwabe and Barbosa, 1994; Schwabe and Rossi, 1994, 1995a, 1995b, 1998], EORM [Lange, 1994, 1996], SOHDM [Lee et al., 1998, 1999b] 이 있다. 한편, 이러한 데이터 모형 중심의 방법론 외에, 워크플로우 개념을 이용한 프로세스 지향적인 접근법에 기반한 WHDM [Lee and Suh, 1999]에서는 데이터 모형으로 이 두 가지 모두가 선택 가능하다.

단계4에서는, 단계3에서 작성된 NaST를 기반으로 데이터베이스 설계를 수행한다. 하나의 NaST는 개념적 수준에서 정해진 하나의 노드 단위로써, 데이터베이스 설계에서 하나의 뷰 (view)로 간주될 수 있다. 따라서, NaST에 나타나 있는 데이터 노드의 요소들은 뷰를 구성하는 속성으로 간주되며 데이터베이스의 뷰 설계에 직접 이용한다. 뷰설계를 수행한 후, 뷰 통합 과정을 거쳐, 통합 데이터베이스 스키마 (IDBS)가 완성된다.

뷰통합 과정에서는 이음동의어 (synonym)과 동음이의어 (homonyms)에 대한 이름 상충해소 (naming conflict resolution) 과정 [Joobineh et al., 1992; Lee et al., 1995]이 필요하다. 이음동의어는 동일한 대상을 각기 다른 이름으로 표현할 때 발생하며, 동음이의어는 상이한 대상을 동일한 속성 이름으로 표현할 때 발생한다. 이러한 상충들을 해소함으로써, 효과적인 데이터 관리를 할 수 있는 시스템을 구축할 수 있다. 이와 같은 과정을 통해 완성된 HyDoMiS의 데이터베이스 스키마가 <그림 12>이다.

3.5 단계5: 구현

구현 단계는 앞 단계들에서 결과물로 산출된 NoST, ISD, NaST, 그리고 IDBS를 기반으로 수행되는데, 인터페이스 구현, 데이터베이스 구축, 그리고, 인터페이스를 통한 노드들과 데이터베이스의 연동 프로그래밍이 주요 업무이다. 효과적인 인터페이스 표현 방식과 효과적인 하이퍼미디어 기능의 구현을 위해서는 하이퍼미디어

관련 프로그램 기술의 선택이 중요하므로 최신 기술의 신중한 비교검토가 선행되어야 한다. 특히, 멀티미디어 요소의 활용과 데이터베이스와의 연동기술이 시스템 효과성에 직접적인 영향을 미칠 수 있다. 이미 설계된 노드가 인터페이스로 표현되는 과정은 사용하는 프로그램과 개발자의 프로그래밍 방식에 따라 매우 복잡한 관계로 구현되고 있다. 그러므로 인터페이스만으로는 이와 관련된 구현 프로그램 파일들과의 관계를 파악하기 어렵기 때문에, 효과적인 유지보수를 위해서 개발자는 각 노드와 구현된 프로그

램 파일과의 복잡한 관계를 정리해 놓는 것이 바람직하다. 이를 위해 Node-Program Mapping Table (NPMT) 사용하면 효과적으로 이 문제를 해결할 수 있다. HyDoMiS 개발 사례에서, 구현 결과 <그림 13>과 같은 NPMT가 생성되었다. NPMT는 노드에 대한 정보 ("Node Index"와 "Node Title")와 노드를 구현한 프로그램 파일 이름 및 위치에 대한 정보를 제공한다. 또한, "Ver."은 구현 시스템 버전을 나타내며, 버전과 관련된 유지보수 내역은 NPMT 하단의 "Maintenance Notes" 란에 기록할 수 있다.

Ver.	3.1		Date	1999.5.14
Node Index	Node Title	Related Program Files	Location	
[GI]	Root Index	Default.htm Index0.htm Mm_index.htm Doc_index.htm Search_index.htm Report_index.htm	c:\inetpub\wwwroot\HyDoMiS\RI\	
[RI: 1]	Domain Analysis	Default_domain.htm Domain_analysis.asp	C:\inetpub\wwwroot\HyDoMiS\Wf\	
[C: 1-1]	Workflow Domain Information Management	Da_menu.asp	C:\inetpub\wwwroot\HyDoMiS\Wf\	
[F: 1-1-A]	Creating Workflow Domain	Da_menu.asp Wf_domain_create.asp	C:\inetpub\wwwroot\HyDoMiS\Wf\	
[F: 1-1-B]	Editing Workflow Domain	Da_menu.asp Wf_domain_edit.asp	C:\inetpub\wwwroot\HyDoMiS\Wf\	
● ● ●	● ● ●	● ● ●	● ● ●	
[RI: 9]	Search	Search.htm Default_search.htm	C:\inetpub\wwwroot\HyDoMiS\Search\	
[F: 9-1]	Drill-Down Search	Domain_search.asp Search_menu.asp	C:\inetpub\wwwroot\HyDoMiS\Search\	
*[C: 9-1-1]	Drill-Down Search Result	Domain_search_result.asp	C:\inetpub\wwwroot\HyDoMiS\Search\	
[F: 9-2]	Keyword Search	Keywor_search.asp Search_menu.asp	C:\inetpub\wwwroot\HyDoMiS\search\	
*[C:9-2-1]	Keyword Search Result	Domain_search_result.asp	C:\inetpub\wwwroot\HyDoMiS\Search\	
[F: 10]	Report Generation	Default_report.htm	C:\inetpub\wwwroot\HyDoMiS\Report\	
[C: 10-1]	Generated Report	Report_result.asp	C:\inetpub\wwwroot\HyDoMiS\Report\	
Maintenance Notes				
[*] 이전에는 Search result 중 리포트 생성을 원하는 것을 리포트 생성 노드를 불러서 수동 입력시켰었는데, search result중 하나를 선택하면, 자동으로 하이퍼문서 번호가 리포트 생성 노드로 전달되어 바로 리포트 생성이 가능토록 자동화 함.				

<그림 13> HyDoMiS의 NPMT

IV. 방법론 비교

현재까지 하이퍼미디어 개발을 위한 방법론이 다양하게 제시되어 왔으며 나름대로의 장점을 가지고 있다. 그러나, 하이퍼미디어 응용 분야의 특징이 매우 다양하므로 하나의 방법론이 모든 분야에 효과적으로 적용되기는 어렵다 [Nanard and Nanard, 1995].

하이퍼미디어 응용 분야는 크게 컨텐츠 지향적인 것과 프로세스 지향적인 것으로 구분할 수 있다. 이러한 관점에서 주요한 기존 방법론 중, RMM (Relationship Management Methodology) [Isakowitz et al., 1995, 1997], OOHDM (Object-Oriented Hypermedia Design Methodology) [Schwabe and Barbosa, 1994; Schwabe and Rossi, 1994, 1995a, 1995b, 1998], 그리고 VHDM (View Based Hypermedia Design Methodology) [Lee et al., 1996, 1999a]는 컨텐츠 지향적인 응용 분야에 보다 적합하다. 이들 방법론은 하이퍼미디어 컨텐츠에 대한 분석과 설계에 초점을 맞추고 있다. 초기 분석 단계의 접근 방법에 대한 관점은 주로 개체-관계 패러다임과 객체 지향 패러다임에 의존하고 있다. IHDM의 적용 영역에 대한 분석 관점은 컨텐츠의 계층적 구조를 파악하는데 초점을 맞추고 있다.

한편, SOHDM (Scenario-Based Object-Oriented Hypermedia Design Methodology) [Lee et al., 1998, 1999b], EORM (Enhanced Object-Relationship Model) [Lange, 1994, 1996], 그리고 WHDM (Workflow-Based Hypermedia Development Methodology) [Lee and Suh, 1999]과 같은 방법론들은 앞서 언급한 방법론들과는 달리, 응용 영역의 프로세스에 대한 분석 과정을 하나의 방법론 단계로서 포함하고 있다. EORM와 SOHDM의 경우 시나리오를 이용한 시스템의 기능 분석에 초점을 맞추는 반면, 정보와 기능에 대한 분석 과정이 별도로 진행된다. 그러나, WHDM은 워크플로우 모형을 이용하여, 시스템을 통한 과업

(task)의 흐름과 더불어 과업 수행과정에서 처리되는 데이터를 동시에 파악하는데 초점을 맞추고 있다. 따라서, WHDM은 정보시스템의 정보와 프로세스에 대한 요구분석을 통합적으로 할 수 있는 장점이 있다. 그리고, 모든 방법론들이와 같은 주요 모델링 기법을 적용한 결과로부터 네비게이션 설계에 필요한 기본적인 정보 얻고 있다.

WHDM과 IHDM은 초기 분석과정에서 파악한 데이터 뷰를 기반으로 통합 스키마를 생성하는 상향 (bottom-up) 접근 방식을 따른다. 반면, 다른 나머지 방법론들은 초기 영역 분석 과정에서 전체적인 컨텐츠에 대한 스키마를 결정하고, 이것을 이용하여 차후에 사용자 뷰를 정의하는 하향 (top-down) 접근 방식을 따른다. 이와 같은 접근 방법은 개발 대상 영역이 매우 클 때에는 적용하기가 쉽지 않다. 수많은 개체 (entity) 또는 클래스 (class)와 이들간의 관계를 정의하는 일은 쉽지 않으며, 완성된 영역 스키마를 가지고 네비게이션을 위한 노드를 별도로 정의하기 위한 작업은 부담스러운 일이다. 따라서, 광범위한 영역에 대한 개발을 할 때에는 사용자 뷰를 먼저 분석함으로써 보다 체계적인 요구분석을 수행할 수 있고, 이것을 직접 네비게이션 단위로 사용하면 개발과정의 복잡성을 감소시키기에 유리한 장점이 있다. 특히, IHDM은 사용자 뷰를 정의하기 위한 접근 단위로, 컨텐츠 노드, 기능 노드, 그리고 인덱스 노드와 같이 역할 관점에서 세분화된 노드 개념을 적용하는 것이 특징이다.

한편, EORM, SOHDM, WHDM에서도 하이퍼미디어를 통한 데이터 처리를 강조하나, IHDM에서는 노드에 대한 개념을 문서 노드와 데이터 노드로 세분화하여 데이터 처리에 대한 설계를 보다 정교하게 수행할 수 있도록 고안되었다. 문서 노드와 데이터 노드에 적용될 수 있는 네비게이션 메커니즘 유형들을 명시화하여 이를 노드에 대한 세부 설계 단계에서 설계 요소로 활용함으로써 데이터베이스 기반의 데이터 처리를

<표 3> 하이퍼미디어 설계/개발 방법론 비교

비교기준 \ 방법론	FMM	COHDM	EOFM	VHDM	SOHDM	WHDM	IHDM
주요 응용 영역	Contents-oriented	Contents-oriented	Process-oriented	Contents-oriented	Process-oriented	Process-oriented	Contents-oriented
영역 분석에 대한 접근 개념	E-R	O-O	O-O	E-R	O-O	Workflow	Contents Hierarchy
주요 분석 대상	Contents	Contents	Contents, Process	Contents	Contents, Process	Contents, Process	Contents
노드 정의를 위한 접근 방법	Top-down	Top-down	Top-down	Top-down	Top-down	Bottom-up	Bottom-up
주요 네비게이션 원천	E-R Relationship	O-O Relationship	O-O Relationship	View Relationship	Scenario & O-O View Relationship	Task Relationship	View Relationship
보안에 대한 접근	N/A	N/A	N/A	N/A	N/A	Navigation Domain, Access Level	Navigation Domain
구현결과에 대한 고려	N/A	N/A	N/A	N/A	N/A	N/A	N/PMI
단계	<ol style="list-style-type: none"> 1. E-R Design 2. Slice Design 3. Navigational Design 4. Conversion Protocol Design 5. UI Screen Design 6. Run-time Behavior Design 7. Construction 	<ol style="list-style-type: none"> 1. Conceptual Design 2. Navigational Design 3. Abstract Interface Design 4. Implementation 	<ol style="list-style-type: none"> 1. Information Analysis 2. Functional Analysis 3. Object Modeling (1) Class Framework (2) Composition Framework (3) GUI Framework 4. Hypermedia Mapping 	<ol style="list-style-type: none"> 1. Requirement Analysis 2. E-R Design 3. View Design 4. Navigational Design 5. Mapping 6. Implementation 	<ol style="list-style-type: none"> 1. Domain Analysis 2. O-O Modeling 3. View Design 4. Navigation Design 5. Implementation Design (1) Page Design (2) User Interface Design (3) Logical Database Design 6. Construction 	<ol style="list-style-type: none"> 1. Workflow Analysis (1) Domain Analysis (2) Specification Analysis 2. Document Analysis (1) Contents Analysis (2) Data Analysis (3) Navigation Analysis 3. Hypermedia Design (1) Navigation Design (2) Database Design (3) Implementation Design (3-1) Interface Design (3-2) Technical Design 4. Implementation 	<ol style="list-style-type: none"> 1. Contents Analysis 2. Node Structure Design 3. Node Specification Design 4. Database Design 5. Implementation

위한 기능들을 고급화할 수 있는 기반을 제공하고 있다.

IHDM과 WHDM은 대부분의 기존 방법론들과는 달리 보안 문제를 설계 과정에서 고려하고 있는 것이 특징이다. 보안 문제는 구현상의 문제로만 여겨져 온 경향이 있다. 그러나, 보안의 문

제도 응용 영역의 특성에 따라 사용자 요구분석을 통해 설계에 반영하여 기술적인 대안 마련 이전에 고려되어야 할 중요한 요소이다. IHDM과 WHDM은 보안 문제에 대한 설계 개념으로써 네비게이션 영역을 제한 하는 방법을 사용한다. 즉, 하이퍼 정보의 네비게이션 영역을 사용자에

따라 제한적으로 적용하는 것이다. WHDM은 이러한 기법 외에도 접근된 정보에 대한 처리 권한 (access authority) 까지도 고려하고 있다. WHDM은 기업의 정보시스템 구축을 위한 방법론이므로, 정보 처리에 대한 권한에 대한 고려를 하는 것은 바람직하다. 그러나, IHDM은 주로 정보서비스 및 광고 목적으로 구축되는 컨텐츠 지향적 시스템을 위한 방법론이므로 회원 또는 비회원 과 같은 사용자 그룹에 따른 정보 접근 통제만으로도 효과적인 시스템을 구축할 수 있을 것으로 판단된다.

한편, IHDM은 각 단계에서 활용되는 테이블 모형의 도구들이 유지보수를 위한 문서로 활용될 수 있도록 버전 또는 유지보수 내역에 관한 정보들을 포함하도록 고안되었다. 특히, 구현 단계의 산출물인 NPMT는 유지보수의 기초적인 정보를 제공함으로써 유지보수 효율성을 제고시키는데 기여할 것으로 기대된다. 또한, 대부분의 방법론들은 네비게이션 설계를 하나의 단계로 포함하고 있는데, IHDM은 네비게이션 설계를 노드 구조 설계 단계 (단계2)와 노드 상세 설계 단계 (단계3)로 구분하고 있다. IHDM에서 네비게이션 설계의 관점이 구성링크와 참조링크를 중심으로 구분되기 때문이다. 지금까지 기술한 IHDM과 기존의 하이퍼미디어 개발 방법론들과의 비교 사항은 다음 <표 3>과 같이 요약할 수 있다.

V. 결 론

최근 하이퍼미디어 시스템들의 구축 사례가 급속한 속도로 증가하고 있으나, 대부분의 경우, 하이퍼미디어 특성을 효과적으로 반영하는 체계적인 방법론의 혜택을 받지 못하고 있다. 기존에 소개된 방법론들이 있기는 하지만 다이어그램 등의 방법론 수행 도구를 자동적으로 지원할 수 있는 시스템의 미비로 적용하는데 한계가 있다. 또한, 기존의 방법론들은 데이터 설계 중심적인

경향을 보이는 반면 하이퍼미디어에 대한 접근 방식에 대한 사용자의 요구를 반영하는 점에는 소홀한 측면이 있었다. 특히, 컨텐츠 지향적인 하이퍼미디어 시스템 개발에 있어서 컨텐츠에 대한 구성이 사용자의 인지과정에 효과적인 방식으로 구현되는 것이 중요한데, 이러한 요구를 해결하기 위한 고려가 미흡하다.

따라서, 본 연구에서는 컨텐츠 지향적인 하이퍼미디어를 효과적으로 구축하기 위하여 인덱스 주도의 접근방법에 기반한 방법론 (Index-driven Hypermedia Design Methodology: IHDM)을 제시하였다. 일반적으로, 컨텐츠는 계층적 구조화가 가능한데, 이러한 구조에 대한 접근을 효율적 구현 하기 위해서, 인덱스 개념은 유용한 수단으로 활용될 수 있다. 따라서 IHDM은 인덱스 노드를 근간으로 하이퍼미디어 노드에 대한 구성에 초점을 맞춘 설계 단계를 포함하고 있는데 이점이 특징이라 할 수 있다. 방법론의 각 단계는 실제 방법론 구축 사례인 HyDoMiS에 대한 개발 과정과 더불어 설명하였다.

IHDM은 대부분의 기존 방법론들과는 달리, 정보 분석 및 설계 과정을 상향 (bottom-up) 접근 방식으로 수행하는 점이 특징이다. 사용자 뷰에 따라 노드를 정의하고 이것으로부터 데이터 베이스를 설계한다. 나아가서, 이것을 네비게이션 단위로 이용함으로써 개발 과정을 단순화시킬 수 있고 광범위한 응용 영역에 대한 시스템 개발을 신속하게 진행시킬 수 있다.

하이퍼미디어의 가장 큰 특징은 하이퍼링크를 통한 네비게이션이라 할 수 있는데, IHDM은 구성 링크와 참조 링크 개념을 중심으로 네비게이션 설계에 대한 접근을 하고 있다. 구성 링크는 인덱스 노드를 기반으로 하이퍼미디어 전체의 구성에 대한 접근의 효율성을 높이는데 중점을 두는 반면 참조 링크는 노드에 포함된 세부 요소를 중심으로 컨텐츠간의 연관성에 따라 설계된다. IHDM은 이와 같은 두 가지 수준의 링크에 대한 관점에서 네비게이션 설계를 체계적인

수행할 수 있도록 고안되었다. 또한, 참조 링크 설계 과정에서는 문서 노드와 데이터 노드 개념을 활용하고 이에 다양한 네비게이션 메커니즘을 적용함으로써 하이퍼미디어의 기능을 고급화시킬 수 있는 설계가 가능하다. 이와 같은 특징은 특히 데이터 노드를 구성하는 멀티미디어 요소 및 데이터베이스를 기반으로 한 데이터에 대한 응용 기능의 효과적인 구현을 가능케 한다.

IHDM은 각 수행 단계에서 테이블 형태의 다양한 도구들을 이용하는데 이러한 도구들은 계층 구조에 대한 표현을 쉽게 할 수 있고 인지적 측면에서도, 다이어그램과 같은 효과를 가질 수 있다. 또한, 자동화된 도구 대신 손쉽게 사용 가능한 워드 프로세싱 소프트웨어의 테이블 기능을 이용할 수 있어서 별도의 지원 도구 없이도 방법론을 실용적으로 적용할 수 있게 한다. 한편, IHDM은 컨텐츠 분석 단계에서 사용자 그룹에 대한 구분을 고려함으로써 인트라넷 또는 엑스트라넷의 구축시 고려해야 하는 보안문제에 대한 현실적인 요구를 설계과정에서 반영할 수 있는 장점이 있다.

컨텐츠 중심의 하이퍼미디어는 빈번한 정보의 갱신과 빈번한 인터페이스의 개선 과정을 통해 지속적인 진화과정을 거치는 특성이 있는데 IHDM의 각 수행 단계에서 산출된 결과물은 유지보수

과정에서 필요한 정보들을 제공하고 있어서 유지보수를 수행하는데 유용하다. 본 연구의 부록에 예시된 인트라넷 시스템에 적용한 사례와 같이, 이미 개발된 하이퍼미디어 시스템에도 유지보수를 위한 도구로 방법론의 도구들을 활용할 수 있다.

향후, IHDM의 개선을 위해, 크게 다음과 같은 두 가지 과제를 수행하고자 한다. 첫째는, 현재의 계층 구조에 의한 컨텐츠 구성 접근 이외의 다른 구조에 기반한 접근 방식에 대해서 추가적인 연구를 하고자 한다. 노드에 대한 구성은 계층적 구성 이외에도 선형, 그리드, 또는 네트워크 구조를 따르는 구성이 가능하다 [Lynch and Horton, 1999]. 따라서 기본 메뉴로 구현되는 광역 인덱스를 이와 같은 다양한 구조로 구성함으로써, 보다 다양한 형태의 다중 메뉴체계 구현을 위한 설계 기법을 포함할 수 있도록 개선하고자 한다. 둘째는, IHDM을 보다 효율적으로 적용하기 위하여 자동화된 지원 도구에 대한 연구이다. 특히, 방법론 수행 과정에서 발생하는 하이퍼미디어에 대한 정보들을 효율적으로 관리할 수 있는 리퍼지토리를 개발하고, 이것이 방법론 지원 도구와의 연계를 자동화하면, 방법론의 효율적 지원 뿐만 아니라 효율적인 유지보수 효과도 얻을 수 있을 것으로 기대된다.

부록 : 유지보수 사례: C 연구실 인트라넷 시스템

최근, 많은 조직들에서 인터넷 기술을 활용하여 비즈니스를 확장시키는 노력을 경주하고 있다. 이와 같은 환경에서, 조직 하이퍼문서는 조직의 중요한 자원으로 인식되고 있으며, 이러한 문서에 대한 효과적인 관리의 필요성이 대두되고 있다. 이러한 맥락에서, IHDM은 하이퍼미디어 시스템 개발 과정에서 산출되는 결과물들을 활용하여 효과적인 유지보수를 지속적으로 수행할 수 있도록 고안되었다. 개발 과정에서 이용되는 도구들은 하이퍼미디어에 대한 유용한 정보들을 포함하고 있는데, 이러한 정보들은 유지보수 과정에서도 유용하게 활용될 수 있다. 또한, IHDM에 의해 개발된 시스템이 아니더라도, 이러한 도구들을 역 적용함으로써, 기존 하이퍼미디어의 정보들의 관리를 통해 유지보수를 효과적으로 수행할 수 있다. 하이퍼미디어의 유지보수를 위한 시스템 또는 방법론에 관한 연구가 미흡한 상황에서, IHDM의 유지보수 기능은 하나의 대안으로써 그 역할이 기대된다.

본 부록에서는 IHDM를 유지보수에 활용한 사례로 CIS 인트라넷 시스템을 소개하고자 한다. CIS 인트라넷 시스템은 한국과학기술원 테크노경영대학원 소속의 기업정보시스템 (CIS: Corporate Information Systems) 연구실에서 사용하고 있다. CIS 인트라넷 시스템은 외부인에 대한 연구실에 관련된 다양한 정보를 제공할 뿐만 아니라, 연구원 간의 의사소통 도구로써 연구 활동 및 연구원간의 협업을 위해 효과적으로 활용되고 있다.

컨텐츠 분석 (단계1)의 산출물인 CHT는 사용자 도메인에 대한 고려를 포함하도록 고안되었는데, HyDoMiS 개발 사례에서는 주요 사용자는 시스템 관리자 또는 개발자로서, 시스템 컨텐츠

에 대한 구분이 필요 없었다. 그러나, 대부분의 인트라넷 시스템은 정보의 보안관리를 위해 사용자의 직급이나 역할에 따라 접속 권한을 통제하며, 엑스트라넷 시스템의 경우에는 회원과 비회원을 구분하여 정보 획득에 대한 권한을 통제하는 경우가 흔히 있다. 따라서, 컨텐츠 분석 시 컨텐츠의 공유 범위를 사용자 그룹 관점에서 지정하는 것은 매우 현실적인 요구 사항으로 인식되고 있다. 기업정보시스템 연구실의 인트라넷에 고려하고 있는 사용자 그룹은 <그림 14>에서 볼 수 있듯이, 세 부류로 구분된다. C는 연구실원을 의미하며, A는 연구실원과 외부의 방문자를 의미한다. 그리고, M은 인트라넷 시스템 관리자를 의미한다. Description 부분은 각 컨텐츠 단위에 대한 설명을 기술하는 부분으로 컨텐츠에 개발 결과에 대한 의사소통에 많은 도움이 될 수 있다.

노드 구조 설계단계 (단계2)에서 산출되는 NoST로부터 하이퍼미디어 시스템의 노드의 내용과 구조 및 조직 링크에 대한 관리를 할 수 있다. 또한, 노드 상세 설계단계 (단계3)에서 산출된 NaST를 이용하여 노드의 세부적인 변동 사항에 대한 관리를 할 수 있는데, 예를 들어, 노드 [RI:6-3]의 인터페이스는 <그림 15>와 같다.

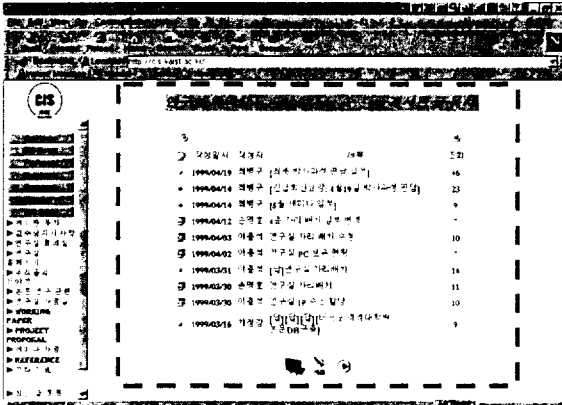
노드 [C:6-3]은 "연구실 회의실"의 "계시판 글목차"를 나타내는데, 이는 연구실에서 공식적으로 논의할 주제들에 대해 올린 글들에 대한 인덱스 노드 역할을 하는 것으로써, 이에 대한 NaST는 <그림 16>과 같이 작성할 수 있다. NaST의 내용 중 "다음" 버튼은 노드내 네비게이션 메커니즘의 슬라이드 방식에 속하는 전후 (Back-and-Forth) 방식으로 구현된 것임을 알 수 있다.

또 하나의 예로써, CIS 인트라넷 시스템의 기능 노드 [F:6-3-3]는 연구실회의실에 올릴 글을 작성하는 역할을 수행한다. 이 노드에 대한 인터페이스는 <그림 17>와 같이 구현되었는데, 이에

컨텐츠 지향적 하이퍼미디어 설계 방법론

Ver.	3.0	Date	1998.12.23
User Group (UG)		C: CISers A: All = CISers + Visitors M: System Manager	
Contents Structure		Information-related Description [Function-related Description]	
C	CIS New	[생성, 수정, 삭제, 접근통제]	
	공지사항	연구실원들이 주지해야할 공지사항 게재	
	교수님 동정	교수님에 관한 동정 게재	
	연구실 동정	연구실에 관한 동정 게재	
	졸업생 동정	졸업생에 관한 동정 게재	
M	News 관리	시스템 관리자가 New의 생성 및 삭제에 대한 관리	
A	Professor	교수님에 대한 소개 및 경력사항에 대한 내용 게재	
	Introduction	교수님의 약력 소개	
	Project	프로젝트 경력을 소개	
	Research	국내외 학술지 및 학술대회 게재 논문 리스트를 소개함.	
	Lecture	강의 내용을 소개	
A	Members	연구실원들에 대한 소개	
A	Master Course	석사과정학생의 이름, 연구분야, e-mail 주소	
A	Doctor Course	박사과정학생의 이름, 연구분야, e-mail 주소	
C	Address	전화번호만 제공 (Home, Office) [접근통제]	
C	Address2	년차, 주소, 전호(Home, Office), 과거전공, 생일, 가족관계 [접근통제]	
C	사용자 정보열람	사용자별 접속정보 (순위, 이름, 최종접속일, 접속수, 누적접속수, 접속비중, 누적접속비중) [접근통제]	
C	일별사용량조회	날짜, 접속수 [접근통제]	
A	Research	연구실 Research 분야에 대한 안내: BPR, Knowledge Management, Internet Information Systems, Telecommunication Management	
A	Alumni	졸업생에 대한 신상정보 (이름, 논문제목, 졸업시기, e-mail 주소)	
	Master	석사과정 졸업생에 대한 신상정보	
	Doctor	박사과정 졸업생에 대한 신상정보	
C	CIS BBS	연구실원들간의 정보공유와 의사소통 기능 제공 [접근통제]	
	게시판 목차	BBS의 이하 목차들과 이와 관련된 글수, 최종등록일에 관한 정보 제공	
	교수님 지시사항	교수님의 지시사항 게재	
	연구실 회의실	연구실 공동에 관련된 내용 게재	
	우리들의 이야기	사적인 얘기들 게재	
	논문 연구 관련	논문 관련 의견 및 정보 교환	
	*연구실 자료실	연구실에서 소장하고 있는 자료 게재 (소프트웨어포함)	
	*Working Paper	학술지 및 학술대회에 게재된 논문 및 Working Paper를 올림.	
	*Project Proposal	프로젝트 제안서 올림.	
	*세미나 자료	세미나 발표 자료 및 결과 사항 올림.	
	Reference	연구에 참고가 될 만한 참고서적 또는 참고 사이트 게재	
	*기타자료	연구실 운영과 관련된 기타 자료들 게재	
	최신글 목록	BBS에 올린 글들을 가장 최근의 것부터 순서대로 정렬	
	글 검색	BBS에 올린 글들을 검색	
A	Guestbook	방문객이 글을 올리수 있도록 함.	
Maintenance Notes			
[*] Ver.2.1에서 Resource Center에 있던 항목들을 BBS로 통합함.			

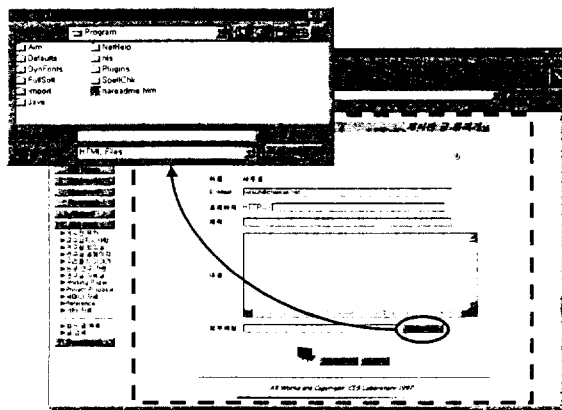
<그림 14> CIS 인트라넷 시스템에 적용된 Contents Hierarchy Table



<그림 15> CIS 인트라넷 시스템의 [RI:6-3]노드에 대한 구현 인터페이스

Ver.	2.1	Date	1998.10.24	
Node Index	RI: 6-3			
Title	연구실회의실 -- 게시판 글 목록			
HTML Part				
Item	Property	Role	Destination Unit	Navigation Mechanism
Title	T	S		
항목	T	S		
* 게시판 리스트	I	A	[RI:6-3-2]	SR
* 새 글	I	A	[F:6-3-3]	SR
* 다음	I	A	DN(1)	B&F
Data Node (1)				
Data Element	Property	Role	Destination Unit	Navigation Mechanism
1 첨부파일	I	E		
2 작성일시	T	S		
3 작성자	T	S		
4 제목	T	A	[C:6-3-1]	SR
5 조회	T	S		
Maintenance Notes				
[*] 텍스트 유형의 경로를 이미지 유형의 경로로 대체시킴.				

<그림 16> CIS 인트라넷 시스템의 [RI:6-3]노드에 대한 NaST



<그림 17> CIS 인트라넷 시스템의 [F:6-3-3]노드에 대한 구현 인터페이스

대한 NaST는 <그림 18>와 같이 작성할 수 있다. NaST에서 "Browse" 요소는 데이터 노드의 6번 요소 (DE6)의 이벤트를 발생시키는 버튼이다. <그림 17>에서 볼 수 있듯이 이 이벤트의 수행 결과로 첨부파일을 선택할 수 있는 Functional Window (FW)를 생성함을 알 수 있다.

IHDM을 적용하여 개발한 HyDoMiS 사례에서 설명한 바와 같이, 데이터베이스 설계 단계 (단계4)에서 생성된 데이터베이스 스키마와 구현 단계 (단계5)의 결과물로 생성된 NPMT는 유지보수를 지원하는데 필수적인 정보를 제공하고 있다. 하이퍼미디어 시스템 구축에 사용되는 소프트웨어의 특성 및 개발자의 구현 방식에 따라, 소스코드를 포함하고 있는 프로그램들의 관계는 매우 복잡하다. 그러나, NPMT로부터 노드와 프로그램간의 연관 정보를 제공 받음으로써 갱신된 노드와 연관된 프로그램들을 빠뜨리는 실수를 방지할 수 있었으며, 연관된 프로그램을 찾는 과정에서 소요되는 시간을 단축할 수 있는 효과를 얻을 수 있었다.

Ver.	2.1	Date	1998.10.22	
Node Index	F: 6-3-3			
Title	새 글 작성			
HTML Part				
Item	Property	Role	Destination Unit	Navigation Mechanism
Title	T	S		
제목 입력	T	S		
게시판 리스트	I	A	[C:6-3-2]	SR
글 올리기	Bu	E(DN(1), S(2), S(3))		SR
지우기	Bu	E(DN(1), De)	S(4)	SR
Browse	Bu	E(DE:6:FW)		
*Copyright	I	S		
Data Node (1)				
Data Element	Property	Role	Destination Unit	Navigation Mechanism
1 이름	T	S		
2 E-mail 주소	Bu	SIG		
3 Homepage	Bu	SIG		
4 제목	Bu	G		
5 내용	Bu	G		
6 첨부파일	Bu	G		
Maintenance Notes				
[*] Copyright 에 관한 이미지 삽입.				

<그림 18> CIS 인트라넷 시스템의 [F:6-3-3] 노드에 대한 NaST

〈참 고 문 헌〉

- [1] 이희석, 김희진, 김영삼, "A form-based distribution design methodology for distributed databases," *경영정보연구*, 5(2), 1995, pp. 101-129.
- [2] 서우중, 이희석, "Developing a meta-information system for hyperdocuments," *한국 경영정보학회 '99 춘계 학술대회 논문집*, 1999, pp. 713-721.
- [3] Choobineh, J., M.V. Mannino, and V.P. Tseng, "A form-based approach for database analysis and design," *Communications of the ACM*, 35(2), Feb. 1992, pp. 108-120.
- [4] De Rose, S. J., "Expanding the notion of links," *Proceedings of Hypertext '89*, Nov. 1989, pp. 249-257.
- [5] Fluckiger, F., *Understanding Networked Multimedia: Applications and Technology*, Prentice Hall, 1995.
- [6] Graham, I., *Object Oriented Methods*, Addison-Wesley, 1994.
- [7] Isakowitz, T., and M. Bieber, "Special section: navigation in information-intensive environments," *Journal of Management Information Systems*, 11(4), 1995, pp. 5-8.
- [8] Isakowitz, T., E. Stohr, and P. Balasubramanian, "RMM: A methodology for structured hypermedia design," *Communication of the ACM*, 38(8), Aug. 1995, pp. 34-44.
- [9] Isakowitz, T., A. Kamis, and M. Koufaris, "Extending RMM: Russian dolls and hypertext," *Proceedings of the 30th Hawaii International Conference on System Science*, 6, 1997, pp. 177-186.
- [10] Lange, D.B., "An object-oriented design method for hypermedia information systems," *Proceedings of the 27th Hawaii International Conference on System Sciences*, 3, 1994, pp. 366-375.
- [11] Lange, D.B., "An object-oriented design approach for developing hypermedia information systems," *Journal of Organizational Computing and Electronic Commerce*, 6(3), 1996.
- [12] Lee, H. C. Lee, and C. Yoo, "A scenario-based object-oriented methodology for developing hypermedia information systems," *Proceedings of the 31st Hawaii Int. Conference on System Sciences*, 1998, pp. 45-57.
- [13] Lee, H., C. Lee, and C. Yoo, "A scenario-based object-oriented methodology for developing hypermedia information systems," *Information and Management*, 36, 1999b, pp. 121-138.
- [14] Lee, H., J. Kim, Y. Kim, and S. Cho, "A view based methodology for designing hypermedia applications," *Proceedings of the Pacific Workshop on Distributed Multimedia Systems*, 1996, pp. 101-110.
- [15] Lee, H., J. Kim, Y. Kim, and S. Cho, "A view based hypermedia design methodology," *Journal of Database Management*, 10(2), April-Jun. 1999a, pp. 3-13.
- [16] Lee, H. and W. Suh, "A workflow-based methodology for developing hypermedia information systems," *Proceedings of the 5th International Conference of the Decision Sciences Institute*, Athens, Greece, Jul. 1999.
- [17] Lynch, P.J. and S. Horton, *Web Style Guide*, Yale University Press, March 1999. (<http://info.med.yale.edu/caim/manual>)
- [18] Mabrouk, M., R. Dykiel, J. Henry, and J.M. Pinon, "A hyperdocument model based

- on the ODA standard," *Proceedings of a Conference on Intelligent Text and Image Handling, "RIA091"*, Barcelona, Spain, Apr. 1991, pp. 245-263.
- [19] Mayfield, J. and C. Nicholas, "SNITCH: Augmenting hypertext documents with a semantic net," *International Journal of Intelligent and Cooperative Information Systems*, 2(3), 1993, pp. 335-351.
- [20] Nielsen, J., *Hypertext and Hypermedia*, Academic Press Professional, 1993.
- [21] Nanard, J. and M. Nanard, "Hypertext design environments and the hypertext design process," *Communication of the ACM*, 38(8), 1995, pp. 49-56.
- [22] Schwabe, D. and G. Rossi, "From domain models to hypermedia applications: An object-oriented approach," Technical Report MCC 30/94, Dept. of Information, PUC-Rio, Expanded version of position paper presented at International Workshop on Methodologies for Designing and Developing Hypermedia Applications, Edinburgh, Sep., 1994.
- [23] Schwabe, D. and S.D.J. Barbosa, "Navigation modeling of hypermedia applications," Technical Report MCC 42/94, Dept. of Information, PUC-Rio, 1994.
- [24] Schwabe, D. and G. Rossi, "Building hypermedia applications as navigational views of information models," *Proceedings of the 28th Hawaii International Conference on System Science*, 3, 1995a, pp. 231-240.
- [25] Schwabe, D. and G. Rossi, "The object-oriented hypermedia design model," *Communication of the ACM*, 38(8), Aug. 1995b, pp. 45-46.
- [26] Schwabe, D. and G. Rossi, "Developing Hypermedia Applications using OOHDM," *Proceedings of Workshop on Hypermedia Development Processes, Methods and Models, Hypertext'98*, 1998.

◆ 이 논문은 1999년 8월 6일 접수하여 1차 수정을 거쳐 1999년 11월 23일 게재 확정 되었습니다.

◆ 저자소개 ◆



서우종 (Suh, Woojong)

공동저자 서우종은 연세대 응용통계학과에서 학사, 석사 학위를 취득하였으며, 현재 KAIST 테크노경영대학원에서 박사과정을 이수중이다. 주요 관심분야는 하이퍼미디어 시스템, 전자상거래, 지식경영, ERP, 리파지토리, 워크플로우 분석이다.



이희석 (Lee, Heeseok)

공동저자 이희석은 서울대 산업공학 학사, KAIST 산업공학 석사, 그리고 University of Arizona at Tucson에서 경영정보시스템을 전공하여 경영학박사학위를 취득하였으며 University of Nebraska at Omaha에서 MIS를 강의하였다. 현재는 KAIST 테크노경영대학원에서 경영정보분야를 연구하고 있으며 최고정보경영자과정(AIM)의 책임교수로 재직하고 있다. 주요 관심분야는 지식경영, 정보전략, ERP, 데이터베이스이다.