제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

SiteHelper를 이용한 디지털 라이브러리 시스템 설계

최선희* 이계성**
* ** 단국대학교 전자컴퓨터학부
e-mail:c8843@hotmail.com

Design of Digital Library System with Localized Agent-SiteHelper

Sun-Hee Choi*, Gye-Sung Lee**
* ** Dept of Computer Science and Electronics, Dan-Kook University

요 약

지식 객체를 이용하여 멀티미디어 문서를 모델링하고, 학습과 추론 설비를 사용하여 사용자의 검색을 도와주는 Localized 에이전트(SiteHelper)를 비롯한 여러 가지 에이전트들을 이용하여 진보된 서비스를 제공할 수 있는 디지털 라이브러리 시스템을 디자인한다.

1. 서론

이 문서는 기존의 라이브러리 시스템에 지식 객체 개념과 데이터 마이닝 기술을 이용한 지능형 에이전트를 가미하여 진보된 서비스를 제공할 수 있는 새로운 방법론을 소개하고자 한다. 우리의 디지털 라이브러리 시스템은 다음과 같은 특징을 가진다.

- 웹에 물리적으로 위치한 데이터 저장소(HTML 문서, 데이터베이스, 이미지, 저널 논문)로 웹 링크를 집중화할 것이다.
- 멀티미디어 문서는 지식 객체의 형태로 모델링되어진다.
- 데이터 마이닝 기술을 이용하여 진보적인 사용자 중심의 서비스를 지원하는 지능형 에이전트(SiteHelper)를 제공할 것이다.

이 접근법은 구축되어져있는 종래의 라이브러리를 Web 링크를 이용하여 집중화하는 것으로부터 시작한다. 그리고 WWW의 급속한 발전과 유연성, 분산의 이점을 유지하면서 WWW를 통해 디지털 라이브러리에 대한 액세스를 제공한다. 우리 시스템의 핵심은 멀티미디어 문서의 지식 객체 모델링과 데이터 마이닝 수단을 가미함으로써 진보된 서비스를 제공하도록 하는 에이전트를 구성하는 것이 될 것이다. 정보 검색과 선택을 위한 지능형 에이전트는 디지털 라이브러리 시스템에 서비스를 요청하는 각각의 개별적인 사용자가 관심을 가지고 있는 것이 무엇인지를 학습하여 사용자에게 최적의 정보를 제공해준다.

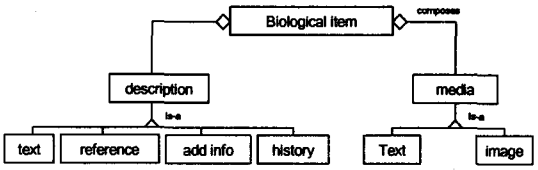
앞으로 소개되어질 내용을 간략히 살펴보면, 2장에서는 멀티미디어 문서를 지식 객체의 형태로 모델링하는 방법을 소개한다. 그리고 3장에서는 우리의 디지털 라이브러리 시스템 구조를 보여주고, 여기에 사용되어진 에이전트들에 대해 설명할 것이다. 특히 우리의 디지털 라이브러리 시스템에서만 볼 수 있는 SiteHelper에 포커스를 맞추어 설명한다.

2. 멀티미디어 문서를 위한 지식 객체 모델링

우리의 디지털 라이브러리 시스템에서는 추론 규칙과 객체 지향 개념을 통합한 지식 객체의 형태로 멀티미디어 문서를 표현한다. 즉 멀티미디어 문서의 모든 컴포넌트들을 객체로 분해하고, 컴포넌트들 사이에 존재하는 의미상의 링크는 규칙의 형태로 표현한다. 이렇게 함으로써 문서 컴포넌트들의 재사용성을 강화시켜주고, 하이퍼 기반에서의 검색을 좀 더 용이하게 해준다.

문서 모델링 과정에서 우리는 클래스의 수와 그

들의 속성, method를 최소로 유지하면서, 모든 멀티미디어 문서 타입을 커버할 수 있도록 시도할 것이다.



<그림 1>지식객체를 이용한 멀티미디어 문서 모델

<그림 1>은 페이퍼 기반 문서의 객체 지향 모델링에 대한 우리의 사전 작업을 기반으로, 객체 모델링 기술 표기법을 사용하여 우선적으로 디자인해본 멀티미디어 문서 모델의 객체 다이어그램을 보여준다.

이 모델에 따라, 라이브러리의 모든 문서는 지식객체(클래스), holding 내부 상태, method들로 분해되어진다. 객체의 내부 상태는 속성들과 제약조건, 그리고 규칙에 의해 정의되어지는 객체들의 method에 의해 정의되어진다.

document 객체는 시스템 컨텐트에 대한 virtual container의 역할을 수행하고, media 객체는 실제적인 컨텐트들을 정의하기 위해 사용되어진다. 즉 document 객체는 복합 문서를 나타내는 것이다. document 객체 내에 삽입되어져야 하는 media 객체의 offstring 클래스들을 참조한 후, assemble() method를 사용하여 실제 컨텐트를 나타낸다. 예를 들어 이미지가 중간에 삽입되어져있는 textbook의 chapter를 생각해보자. 이 때 chapter가 document 객체로 정의되어진다면, 이 객체는 세 개의 분산된 media 객체들로 구성되어질 것이다.- 두 개의 분리된 텍스트 객체와 하나의 이미지 객체

이렇게 media와 document 객체를 분리하는 이유는 문서에 포함되어지는 컨텐트를 완전히 분산시킴으로써 미디어 객체로 정의되어지는 컨텐트들의 재사용성을 강화하고자 하는 것이다. 또한 사용자가 요구하는 컨텐트가 summary일수도 있고, 어떤 topic에 대한 세부항목이 될 수도 있으므로, 사용자의 요구에 커스터마이즈한 응답을 해주기 위해서도 이 두 객체의 분리는 필요하다.

structure 객체는 문서에 포함되어 있는 컴포넌트들의 구조를 나타내는 클래스이다. 이 객체는 text chunks, tables, figures, reference entries, articles 등의 다양한 다른 형태로 나타난다. text chunks는 연속되는 하나 또는 그 이상의 ASCII 문자들이다. tables, figures, reference entries, articles는 모두 기존의 라이브러리에서 사용되어지는 방법 그대로 사용되어질 것이다. text chunks는 structure나 document 객체를 위한 기본적인 프리젠테이션 단위가 될 것이다.

publication은 출판된 자료의 모든 타입을 위한 일반 클래스로써 보통 출판물을 식별할 수 있는 국제적 인증번호 ISBN이나 ISSN을 사용한다. 이 클래스로부터 상속되어지는 하위 클래스들은 다른 문서 타입들의 특정한 특징들을 처리하기 위해 디자인되어진다.

<그림 1>에서 제시되어진 모델을 기반으로 라이브러리의 문서를 쉽게 구축, 확장, reordering, assemble, disassemble, truncate할 수 있는 문서 관리 시스템을 디자인한다.

3. SiteHelper를 이용한 디지털 라이브러리의 시스템 구조

3.1 SiteHelper

World Wide Web이 기하급수적으로 성장을 계속함에 따라 방대한 정보들이 급속하게 생겨남으로써 사용자들이 원하는 정보를 발견하는 것을 어렵게 하고 있다. 이런 문제를 도와주기 위해 검색 엔진이나 기계 학습을 이용하는 지능형 에이전트인 로봇 기술들이 나오게 되었다.

검색 엔진은 검색 가능한 데이터베이스들에 대한 인덱싱 설비들이다. 그러나 웹의 성장이 계속되어짐에 따라 하나의 검색 요청에 대해 리턴해야 하는 페이지의 수가 너무 많아지게 됨으로써 사용자가 원하는 정보를 발견하기 위해 걸리는 시간이 길어지게 되었다.

로봇들도 검색 엔진들과 유사하지만, 그들은 검색 엔진처럼 웹 페이지를 인덱싱하기 보다는 웹을 오가며 관련 문서를 분석하고 저장한다. 이런 로봇들의 가장 중요한 단점은 네트워크 자원에 대해 많은 요구를 한다는 것이다. 즉 로봇이 작동되기 위해서는 많은 bandwidth가 요구되어지기 때문에 네트워크 오버로드, bandwidth 부족, 유지 비용의 증가와 같은 결과를 초래하게 된다.

위에서 언급한 문제를 개선할 수 있는 방법으로 SiteHelper라 불리우는 지능형 소프트웨어 에이전트가 제안되어졌다.[1] 이 에이전트는 local web server를 상대로 작동한다는 점에서 기존의 검색 엔

진이나 로봇들과는 다르다. SiteHelper는 각각의 local 웹 서버를 위한 housekeeper로 작동하고, 사용자가 관심 있어하는 분야에 대한 관련 정보를 발견하는 것을 도와주도록 디자인되어진다. SiteHelper는 각 사용자의 특징이나 관심 분야에 대해 incremental하게 학습하고, 그것을 기반으로 논리적 규칙을 생성, 변경시킴으로써 성과를 향상시킬 것이다.

local site에 대한 정보는 SiteHelper의 효율성을 증명하기 위해 키워드 dictionary의 형태로 주어진다.

3.1.1 interactive incremental learning

SiteHelper는 사용자가 그들의 관심 영역을 키워드의 집합으로 입력하는 것을 허용하는 GUI를 제공함으로써 사용자와 interaction한다. 그 때 키워드들은 dictionary의 키워드들과 매치하는데 사용되어지고, 이것을 기반으로 로컬 사이트에서 웹 페이지를 검색한다. 예를 들어 만약 사용자가 "Artificial Intelligence"라는 키워드를 입력했다면, SiteHelper는 "Artificial Intelligence"를 포함하는 웹 페이지와 "Artificial Intelligence"와 관련된 토픽을 검색할 것이다. SiteHelper는 사용자에게 검색된 페이지의 리스트를 리턴한 후, 그것에 대한 피드백을 요구할 것이다. 사용자가 어떤 페이지는 승인하고 어떤 페이지를 거부하는 작업이 이루어진 후에 사용자의 관심 영역과 관련된 실제 키워드를 식별하기 위한 학습이 시작되어진다.

HCV(Heurictic Covering Vector)[4]는 사용자가 승인한 페이지를 positive case로 하고, 그렇지 않은 페이지는 negative case로 하여 사용자의 관심 영역을 나타내기 위한 규칙들의 집합을 생성한다.

SiteHelper는 자신이 리턴한 페이지들에 사용자가 만족할 때까지 위 사이클을 계속해서 반복 수행한다. 그러므로 SiteHelper는 사용자의 피드백에 따라 HCV 규칙들을 변경하면서 웹 페이지들을 검색, 선택하기 위한 incremental learning을 실행한다.

3.1.2 silent incremental learning

많은 웹 서버들은 사용자가 그들의 사이트에서 액세스한 레코드에 대한 정보를 로그 파일 시스템에서 제공한다. 로그 파일은 일반적으로 computer machine name, numerical address, 액세스 타임, 액세스한 페이지로 구성되어진다. SiteHelper의 silent incremental learning은 log 파일의 정보를 사용하는 것에서 출발할 것이다.

SiteHelper는 각 사용자르 위한 로그 파일을 추출할 것이다. 로그 파일로터 사용자가 액세스한 페이지에 대한 정보를 얻은 후, 이런 액세스되어진 웹 페이지들을 HCV를 실행하기 위한 positive case로 하여 사용자의 관심 분야에 대한 논리적 규칙을 생성할 것이다. 예를 들어 만약 사용자가 규칙적으로 "Artificial Intelligence" 그룹의 웹 페이지들을 액세스한다면, SiteHelper는 사용자 관심 영역의 하나로 AI를 선택한다.

3.1.3 Using learning results to assist the user

두 개의 학습 프로세서는 사용자의 관심 영역을 학습하는데 사용되어지고 이 학습의 결과는 logic rule의 형태로 주어진다.

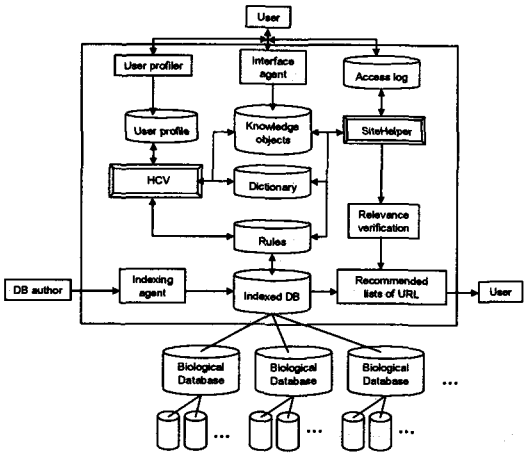
새로운 웹 페이지가 생성되어지거나 변경되어질 때, SiteHelper는 그것들을 인덱스하는 키워드 집합을 식별하기 위해 인덱싱 에이전트를 실행시키고, 이를 통해 인덱싱되어진 데이터는 indexed database에 저장되어진다.

indexed database에 저장되어지는 키워드들은 그것이 사용자의 관심 영역과 관련된 키워드인지를 체크해보기 위해 HCV 규칙에 매치되어진다. 이 결과로 선택된 페이지를 사용자에게 리턴하고, 만약 사용자가 이 페이지를 거부한다면 학습 프로세서를 다시 수행하여 HCV 규칙을 변경함으로써 디지털 라이브러리의 성능을 개선시켜준다.

3.1.4 Agent Struture with SiteHelper

SiteHelper는 사용자의 관심 분야를 학습한 후, HCV 추론을 통해 규칙을 생성하여 그것을 User Profiles에 저장한다. 사용자가 복합 문서(document 객체)를 요청하는 경우 이것이 참조되어진다. 사용자의 관심 분야에 포함되어지는 키워드들과 매칭되어지는 Indexed Database에 있는 문서들을 assemble() method를 사용하여 사용자에게 리턴해준다. 예를 들어 사용자가 회의록을 액세스하고자 한다면, SiteHelper는 Indexed Database에 있는 회의록 페이퍼들을 모두 모은다. 그 후 사용자의 관심 분야에 매치되어지는 회보의 페이퍼들을 모아서 우선적으로 리턴해주고, 나머지 페이퍼들은 사용자의 요청이 있을 경우를 대비하여 테이블에 기입해둔다.

<그림 2>는 기존의 프로토타입 [Ngu & Wu



<그림 2> SiteHelper와 에이전트를 이용하여 만들어지는 디지털 라이브러리 시스템 구조

1997]을 기반으로 SiteHelper를 이용하여 만들어지는 디지털 라이브러리 시스템 구조의 사전 디자인을 보여준다. 여기에 사용되어지는 주요 에이전트들은 다음과 같은 기능을 수행한다.

Dictionary는 디지털 라이브러리의 범위를 정의하는 계층에 있는 키워드들의 집합이다. 관심 영역이나 디테일한 토픽과 관련된 키워드들은 계층적으로 중첩되어질 수 있다.

Dictionary에 포함되어지는 키워드들은 전문가를 통해 인터랙티브하게 입력되어지거나 Interface Agent에 사용자가 다루기 쉬운 툴(예 : C++)을 제공하여 구축한다.

SiteHelper는 에이전트들 사이의 주요 커뮤니케이션과 디지털 라이브러리 시스템의 기술들을 총체적으로 관리한다.

물론 사용자가 이전의 방문과는 다른 목적을 가지고 방문한 경우에는 사용자의 관심 분야에 대해 새롭고 업데이트된 문서를 제공하는 SiteHelper는 불필요하므로, 이것을 멈추고 우선권을 설정할 수 있도록 디자인되어진다.

HCV 유도 엔진은 학습을 통해 식별되어진 사용자의 관심 영역과 dictionary에 있는 키워드들과의 결합 형태로 규칙을 생성한다.

Indexed DataBase는 디지털 라이브러리의 각 문서를 위한 개체/인덱스를 가지고 있다.

Relevance Verification Agent(RA)는 디지털 라이브러리 시스템의 dictionary를 사용하는 사용자로부터 추천되어진 문서의 적합성을 확인한다.

Document Update Agent(DA)는 디지털 라이브러리 시스템에 링크되어져 있는 웹 문서의 업데이트 여부와 인가된 사용자들이나 RA로부터 보고되어진 개체를 체크한다. 여기에서 업데이트는 개정, 삽입, 삭제를 포함한다.

Indexing Agent는 디지털 라이브러리를 검토한 후, dictionary에 따라 모든 관련 문서를 인덱스하여 Indexed DataBase에 그 결과를 저장한다. 이 결과는 Document Update Agent에 의해 체크되어지는 문서들의 최신 정보에 따라 자동으로 업그레이드되어진다.

User Profiles은 사용자의 account(계정), 관심 분야, 액세스 히스토리와 HCV에 의해 생성되어진 규칙들로 구성되어진다.

4. 결론

이 문서에서는 지식 객체를 이용하여 멀티미디어 문서를 모델링하고, SiteHelper라는 지능형 소프트웨어 에이전트를 이용하여 사용자가 관련 정보를 발견하는 것을 도와주도록 설계되어진 디지털 라이브러리 시스템을 소개하였다. 우리는 이 시스템의 성능을 객관적으로 평가해보기 위해 텍스트와 이미지, 오디오, 애니메이션 등을 포함하는 멀티미디어 복합 문서를 가지는 디지털 라이브러리를 구축해볼 필요가 있다.

참고문헌

[1] Daniel Siaw Weng Ngu and Xindong Wu "SiteHelper : A Localized Agent that Helps Incremental Exploration of the WWW", proc. of the 6th Inter. WWW Conf. Santa Clara, CA, 1998

[2] [Ngu & Wu 1997] D.S.W. Ngu and X. Wu " SiteHelper : A Localized Agent that Helps Incremental Exploration of the World Wide Web", Proceedings of the 6th International World Wide Web Conference, Santa Clara, California, USA, 1999, 691-700.

[3] X. Wu and K. Cai, Knowledge Object Modeling, IEEE Transactions on Systems, Man, and Cybernetics Part A Systems and Humans, 30(2000), 2: 96-107

[4] X.Wu, Rule Induction with Extension Matrices, Journal of the American Society for information Science 49(1998), 5 : 435-454