객체기반 다중 시스러스 시스템의 설계 및 구현

정규상감, 김원중, 양재동
전북대학교

Design and Implementation of an Object-Based Multi-Thesaurus System
Kyusang Chong, Wonjung Kim, Jaedong Yang
Chonbuk National University

요약
본 논문은 하나의 도메인에 대해 서로 연관된 여러 개의 시스러스를 효율적으로 구축, 관리, 검색, 브라우징, 향상하기 위한 확장된 객체기반 시스러스 시스템을 제시한다. 이 시스템은 1) 대규모의 시스러스를 효율적으로 분산하여 구축할 수 있는 기능을 제공하고, 2) 구축된 시스러스간의 일관성 있는 검색, 브라우징, 향상을 제공한다. 이러한 기능은 동시에 사용자들은 대규모 시스러스를 하나의 시스러스처럼 이용할 수 있다.

1. 서론
현재 인터넷의 발전과 더불어, 대량의 정보를 검색하기가 백미에 따라 정보검색 시스템의 중요성이 더욱 증가하고 있다. 정보검색 시스템은 일반적으로 사용자가 제공한 정보와, 문서의 상세적인 핵심 성분을 분류해주고 있다. 이러한 분류의 복잡성과 향상에 의해 사용자의 관심을 끌어내며, 그외에도의 중요한 요소로, 인터넷의 발전과 함께 사용자의 관심도와 검색에 의한 어휘의 확장으로 인해 정확도(precision)와 재현율(recall)이 높아질 수 있는 시스러스가 도입되었다.[2] 이에 따라 시스러스가 높고 내적 이용적으로 구축되었으며, 대표적인 예로, NASA 시스러스, INSPEC Thesaurus, 그리고 Roget Thesaurus 등이 있다.

이러한 시스러스는 대부분 도메인에 축적되어 있으며, 하나의 도메인 당각 하나의 시스러스로 구축되어 있다. 이런 한계 하나의 도메인을 하나의 시스러스로 표현하는 경우, 산업이 저해되어 구축될 시스러스가 방대해 정책으로 관리 및 검색 등의 어려움이 발생될 수 있다. 따라서 시스러스의 효과적인 관리 및 검색을 위해서 시스러스를 적당한 규모의 시스러스로 분류하여 구축할 필요가 있다. 그러나 하나의 시스러스를 여러 개의 분류된 시스러스로 구축하여 유탱 관리한 경우, 일반적으로 결과나 구축, 관리 비용 중当地政府의 문제를 해결해야 한다.

본 논문에서는 도메인 방식의 시스러스를 능리적으로 분산 구축하여 효율적 관리할 수 있고, 분산된 시스러스를 원활한 관리에 의해 일관성이 보장되는 적절한 객체기반 시스러스를 협동 검색, 브라우징, 향상할 수 있는 시스템을 설계하고 구현하였다.

2. 관리구조
2.1 객체기반 시스러스[2]
본 논문에서 제시한 시스러스는 객체기반 시스러스를 제시하여, 객체의 지식을 보다 효과적으로 모집할 수 있도록 하였다. 객체기반 시스러스는 기존의 시스러스와 객체기반 시스러스의 특성에 특성을 적합한 시스러스이다. 즉, 모든 시스러스 내의 개념을 객체로 간주하고, 이러한 객체와 관계시스러스를 표현하는 복잡한 시스러스 시스템이다. 객체는 개념을 추상화하는 개념 객제와 인스턴스를 추상화하는 인스턴스 객제로 구성된다. 객제와 시스러스 객제와 관계는 기본적인 시스러스 개념을 식별하는 상위어(Broader Term), 하위어(Narrower Term), 관계어(Related Term), 관계어(Owner/Instance-of), 공유어(Whole/Part-of) 등이 있다. 관리자들이 시스러스의 의미에 따라 재정의

하의 결과로, 1종 "Switching System" 객체기반 시스러스의 예이다.

2.2 다중 시스러스 시스템

본 논문에서는 다중 시스러스를 하나의 도메인에 대해 같은 구축을 가진 동음의 복수 시스러스로 그 의미를 제한하고, 여러 시스러스가 제공하는 일관성과 통합의 효과성을 가진 시스템을 제공한다. 여러 시스러스나 메타 시스러스는 구축된 시스러스들 간의 통합을 동시에 받고 본 논문에서 제시한 객체기반 다중 시스러스 시스템은 시스러스 객제와 관계는 객제기반 시스러스에서 이미 일관성이 있는 다중 시스러스로 구축할 수 있다는 점에서 다른 관점이 가능하다.

3. 객체기반 다중 시스러스 시스템

그래프 1. 객체기반 시스러스의 예

구현 결과.
본 논문에서 제안한 객체기반 다층 시스템의 핵심이 되는 여러 개의 시스템으로 분할 구조가 가능하다. 또한 본 논문의 동등이 시스템의 동일한 작업을 수행하게 되며, 외부로 보았을 때 하나의 시스템으로 볼 수 있는 유기적으로 작동하는 기능을 제공한다. 이를 위해 본 논문은 시스템에서 갈 Credits의 유일성을 보장하기 위한 GUIOD (Global Unique Object Identifier), 주로 객체 그래프로 다층 시스템을 관리하기 위한 시스템 리스코가 존재한다.

3.1 GUIOD와 시스템 리스코
본 시스템에서 GUIOD를 3요소로 분류하여 서버 식별자, 시스템 식별자, 객체 식별자로 이루어진다. 각 서버 식별자는 구축된 시스템들이 존재하는 서버 ID값, 시스템 식별자는 시스템 서버에 존재하는 시스템의 유일한 ID값, 객체 식별자는 시스템을 구축하는 객체들의 ID값 각각 생성한다. 예를 들어, GUIOD를 구조화하여, GUIOD는 객체 식별자가 '1'이고 서버 식별자는 '2', 그리고 객체 식별자는 '3'의 GUOD를 생성하고 있다. 시스템의 구조는 시스템 내에 존재하는 모든 객체 식별자를 관리하기 위한 객체이며, 일반적으로 시스템의 구성에 의해 reference-of라는 추가적인 속성을 가지게 된다. reference-of 속성은 본문에서 설명한 본문의 정보를 가진다.

시스템 통합 데이터베이스 (그림 4 참조)는 4요소로 구성되어 '서버 주소, 서버 식별자, 시스템 식별자, 시스템 이름'으로 이루어진 시스템의 리스코를 가 진다. 여기서 서버 주소는 시스템 서버의 존재는 메타데이터의 주요 구조, 서버 식별자는 구축된 시스템들이 존재하는 서버의 ID값, 시스템 식별자는 시스템 서버에 존재하는 시스템의 유일한 ID값, 시스템 이름은 시스템의 이름을 나타낸다. 그림 식별자는 같 은 구문 시스템의 내부 ID값 각각 생성한다. 예를 들어, GUIOD는 '21.11.17.21', '2.2. Switching System' 등이 시스템 식별을 생성한다.

3.2 참조 객체를 통한 시스템 구축
본 시스템에서 새로운 객체를 추가하거나, 객체의 관계를 설정할 경우, 먼저 모든 시스템 내에 해당 객체가 존재하지 않는지 여부를 확인해야 한다.

먼저, 원소를 추가하기를 허용하기로 한다. 일반 시스템을 포함한 모든 시스템들로부터 각각의 원소에 대한 객체를 검색한다. 검색이 실패한 각각의 결과는 reference-of 속성을 이용하여 통합한다. 통합된 각 객체들의 시스템 정보를 이용하여, 추가하기로 하는 객체와 동일한 객체가 존재할 경우, 객체의 객체가 시스템 내에서 이미 존재하는 객체에 추가되는 위치이다. 만일 그 객체가 다른 시스템 내의 객체이면 현재 시스템에 그 객체의 참조 객체가 생성된다.

응용된 관계에 설정할 경우에는, 일단 외부 객체 추가자의 단말에 객체가 생성된 객체(원본 객체나 참조 객체) 관계를 설정하면 된다.

그림 3 참조 객체를 통한 시스템 구축

3.3 참조 객체를 통한 시스템 구축

3.4 참조 객체를 통한 시스템 구축
본 시스템에서는 분산되어 있는 각각의 시스템으로서 참조 객체를 통해 각 시스템의 객체에 연결할 수 있다. 이렇게 하여 하는 객체가 참조 객체의 경우, reference-of 속성에 위치한 원본 객체의 key와 GUIOD를 이용하여 원본 객체의 참조가 이루어진다. 예를 들어, 그림 2의 "Circuit Switching" 시스템에서 'Public Exchange' 객체를 참조 객체의 형태로 참조할 때, reference-of 속성의 <2, 2, 3>의 GUIOD에 이용하여 서버 식별자가 '2', 그리고 시스템 식별자가 '3'인 시스템을 시스템 리스코에서 파악할 수 있다. 그리고 이 시스템의 서버정보와 시스템 식별자를 통해 원본 객체의 형태를 할당할 수 있다.

4. 참조 객체를 통한 시스템의 구현
4.1 시스템 구조
그림 4는 본 논문에서 제안한 객체기반 다층 시스템으로
3-계층 구조를 가진다.

Application 계층은 도메인 전문가들에 의해 객체기반 다중 시스템의 관리자와 협업하는 인터페이스로 구성된다. 이 관리자는 도메인 전문가가 분산된 객체기반 시스템을 마치 통합된 하나의 시스템처럼 구축할 수 있는 기능을 제공한다. 협업 관리자와 협업한 시스템의 확장 관리에 제공한다.

Mediator 계층은 다양한 어플리케이션 요구 사항을 받아, 시스템 통합 데이터베이스(Thesaurus Integration DB)와 각각의 독립적인 시스템의 데이터베이스에는 관리할 수 있도록 하기 위한 계층이다. 시스템 통합 서버(Thesaurus Management Server)는 시스템 통합 데이터베이스를 관리하는 기능과 각각의 시스템 데이터베이스에 시스템을 구축, 관리하는 기능을 제공한다. 시스템 통합 서버는 시스템의 상호 관계를 설정하고, 각각의 시스템 데이터베이스로부터 비주얼화 정보를 리턴 받은 뒤, 각 객체의 reference-of 속성에 의해 하나의 완전한 정보로 방향을 수행한다.

Database 계층은 시스템 서버, 시스템의 각 서브시스템의 시스템의 리소스를 관리하는 시스템 통합 데이터베이스와 각 시스템 데이터베이스에 구성된다.

그림 4 객체기반 다중 시스템 시스템 구조

4.2 관리자 사용자 인터페이스

본 시스템에서 객체를 추가하거나, 객체간의 관계를 설정할 경우, 시스템 관리자는 시스템을 통합 관리하려면 데이터를 정의하고, 시스템 관리자가 시스템의 데이터베이스에 대해 스크립트를 생성되며, 관리기로부터 전달받은 배울률을 이용하여, 병렬적으로 관리한다. 이 과정을 통해 각 스크립트의 각이 DML에 의해 실행되며, 각각의 결과를 반영하여 관리할 데이터베이스의 관계를 정의하고, 관리자에 의해 관리할 데이터베이스의 관계를 관리한다. 관리자 관계는 관리자용 뷰에서 관리자용 뷰에 배운 관계를 관리하여 관리한다.

그림 5 객체 검색 및 통합검색 결과 리스트

예를 들어, 그림 6에서 사용자의 "Electronic Company" 시스템의 "Switching Company" 객체와 "Circuit Switching" 객체를 선택하여 검색하기 위해서, 모든 시스템의 데이터베이스에 "Circuit Switching" 객체와 연관된 리소스를 얻을 수 있다. 사용자는 이 리소스와 객체들을 상세정보를 통한 관계의 상호 관계 관리에 필요한 정보를 얻을 수 있다. 이러한 상세정보를 얻은 사용자는 데이터베이스에, 데이터베이스에 "Association of" 객체, 통합 관계 관리기로 사용된 "Circuit Switching" 객체를 얻을 수 있다. 또한 관리자의 관계의 정보를 얻기 위해서, 통합적인 시스템 정보를 얻을 수 있다.

4.3. 협업 인터페이스

사용자가 협업 인터페이스를 통해 시스템을 사용한 확장 검색을 할 경우, 관리기 사용자 인터페이스의 경우처럼, 시스템의 각 서버를 통해, 통합된 확장 검색 결과를 얻는다.

그림 6 시스템을 이용한 확장 검색

5. 결론 및 향후 연구 과제

본 논문에서는 분산 환경에서 하나의 도메인을 여러 개의 시스템으로 분산 관리할 수 있는 확장된 객체기반 시스템을 개발하였다. 이 시스템은 창조를 통해 통합화하여 하나의 시스템으로 통합화하면서 사용자의 관리에 대응한 정보를 제공할 수 있다. 향후 연구로는 다국어 시스템과 배치 등의 특성을 할당하여 외부의 하나의 시스템으로 통합화하면서 사용자에게 효율적으로 정보를 제공할 수 있다.

참고문헌


183