

# 전문 문서 개념을 사용한 지식 미디어 시스템에 관한 연구

손영수\*, 조정래\*\*, 배상현\*\*

A Study on Knowledge Media System  
using the Concept of Special Document

Young-Su Son\*, Jeong-Lae Cho\*\*, Sang-Hyun Bae\*\*

## Abstract

The knowledge of the specialized fields has been changed rapidly in the both side of quantity and quality. A hyper media as a knowledge based system is so fixed in the linked media in mutual that we couldn't tell it information provision with the view of users. In this paper, we propose the way of offering intellectual and flexible information which is the same with demand in the side of user, selecting, searching and composing the hypertext at the point of user's view through the design of knowledge media system. Three concepts are used in order to challenge the knowledge media system : special document, agent system and ontology. The special document is a knowledge using special fields. The agent system is a homogeneous controller that operates on an application to ensure that its activities are coordinated with those of the others within the community, providing a uniform control mechanism. Finally, ontology is a language for exchanging knowledgement, which is a message exchanged among agents to ensure the proper interaction among them. The combination of these three concepts is used to design the prototype of knowledge media system.

## 1. 서 론

전문분야의 지식정보를 관리하는 것은 중요하다. 전문분야의 지식정보 관리에 요구되고 있는 관점은 과거와

현재, 미래에도 변화한다. 지금까지 전문분야의 지식은 보존되고 있는 그 자체가 중요시되어왔다. 지식의 증가량은 빠르지 않아 장기간 보존하는 관점에서 지식을 관리하여 왔다. 이 관점을 만족하는 지식의 대표적인 것으로 전문문서가 있다.

\* 한려산업대학교  
\*\* 조선대학교

전문분야의 지식 관리의 관점으로는 이용자의 요구와 시점에 맞는 지식의 제공을 행하는 것이 중요하다. 이용자의 의사를 이해하여 그것에 맞는 정보의 선택과 제시라는 지적인 처리가 가능하여야 한다.

이와 같이 필요한 관점을 만족하는 전문분야의 지식 정보 관리시스템은 현재 존재하지 않는다. 현 시대를 대표하는 지식 제공 시스템인 하이퍼미디어도 미디어간의 링크가 고정적이고 이용자의 시점을 파악한 정보제공은 아니다. 본 논문에서는 요구되는 정보를 이용자의 시점에서 검색, 구성하여, 요구에 맞는 전문문서를 선별, 그것들을 이용자의 시점에서 지적인 처리를 행하는 유연한 정보제공 실현방법을 제안하고 있다.

또한, 전문문서의 지식 미디어화의 시도를 통하여 이용자의 요구나 시점에 일치한 문서를 구조적으로 제시하는 설계 개발 방법을 수행한다. 특히 지적인 정보의 제공을 수행하는 시스템으로 대상이 되는 세계를 기술하는 지식의 구성과 그것을 시스템이 이용하는 방법의 실현에 초점을 맞추고 있다.

## 2. 지식 ontology의 구성과 표현

### 2.1 지식의 재이용을 목적으로 한 ontology

문제해결의 지식에는 대상분야에 의존하여 기술된 부분과 문제 해결 방법을 기술한 부분이 있다. 예를 들면 ESPRIT project의 하나인 KADS<sup>1)</sup>에서는 문제해결의 지식베이스 개발로서 전문가 모델을 포함한 개발 프로세스 전체의 모델화를 행하였다. 전문가 모델은 개념모델로서 전문가의 문제해결 방법을 추상적, 계층적으로 기술한 것이다. 개념모델은 구체적인 것으로부터 도메인층, 추론층, 타스크층, 전략층으로 나누어 각 계층은 상위계층의 지식에 의해 메타 수준으로 이용된다. 이 계층화에 의해 분야에 의존하는 지식(도메인층)과 추론의 종류를 기술한 지식(추론층), 추론지식을 어떻게 조합하면 문제 해결이 가능한가를 기술한 지식(타스크층), 이용자의 요구와 문제의 성질에 맞는 타스크를 선정하는 지식(전략층)과 같이 문제 해결 지식이 분류된다. 특히 도메

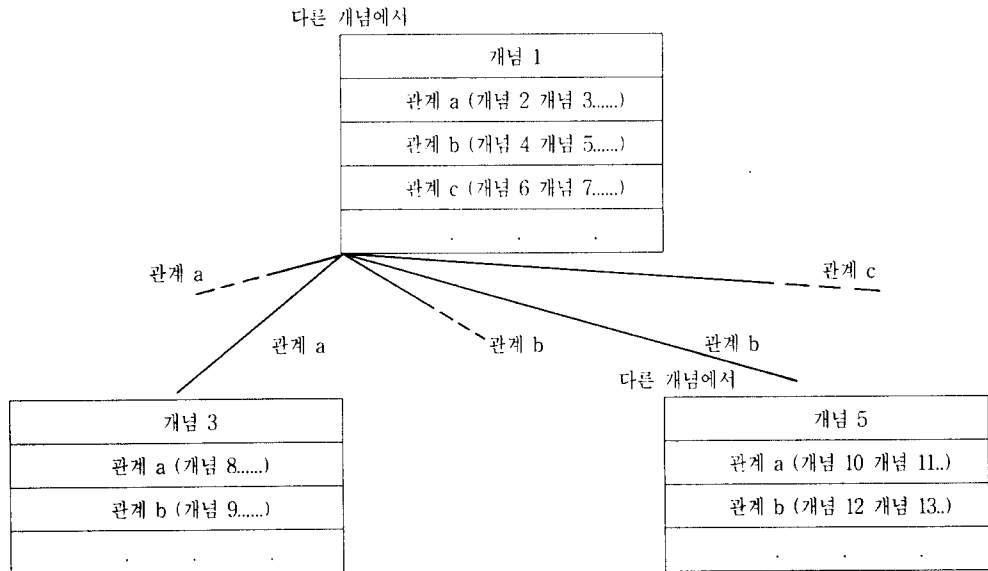


Fig. 1 The expression form of ontology.

인층의 지식을 다른 메타 수준의 지식으로부터 분리하는 것이 중요하고, 추론층 이상의 지식 재이용이 별도의 분야에서도 가능하게 된다.<sup>2)</sup>

본 연구에서는 대상분야의 지식을 ontology화 하고 이것을 도메인의 지식에 대응하는 도메인 ontology라 할 수 있다.

본 연구의 ontology 표현에서 이용하는 것은 의미네트워크지만, 개념과 개념과의 관계를 기술하는 점에서 KL-ONE의 기본적인 방법과 일치하고 있다 KL-ONE 언어<sup>3)</sup>에 속한 것은 개념 클래스와 클래스간의 관계를 정하는 슬롯을 정의하여 개념간의 포함관계를 추론하는 능력을 가지고 있다. 개념과 관계의 표현에는 and, or, all, some 등의 논리표현을 이용할 수 있다.

## 2.2 배경지식과 전문지식

전문분야의 지식은 배경지식(상식)과 전문지식으로 나누어진다. 전문서적에는 그 분야의 전문적인 지식에 대해서만 쓰여져 있지 않다. 전문서적을 이해하기 위해서는 전문적인 지식의 기초가 되는 배경지식이 필요하다.

## 2.3 배경지식 ontology

배경지식 ontology는 전문서적에 포함되어 있지 않은 지식으로, 전문서적을 읽을 때 필요한 상식적 지식의 체계이다. 예를 들면 「점포 시설의 지식」의 분야에는 전문서적에서는 당연히 생략되고 있는 「점포의 물리구조」에 관한 지식이라 할 수 있다. 전문서적에서는 국소적인 부분에 관한 구조 등은 게재되어 있어도 「건물에는 계단과 화장실이 있고....」라는 당연한 지식은 명시되어 있지 않다. 여기에는 대상의 속성을 나타내는 개념, 실체가 없는 추상적인 것을 나타내는 개념, 속성을 나타내는 관계, 계승을 나타내는 관계 등 여러 가지 표현이 고려되고 있다.

## 2.4 전문지식 ontology

전문지식은 전문문서편의 집합과 전문지식 ontology로부터 이루어진다. 지식의 상세한 기술은 각각 전문문서편에 의뢰하고, 전문 ontology는 각각 전문문서편이 포함된 전문지식이 상호간에 어떻게 연결되어 있는 가를

나타내고 있다. 전문지식의 ontology는 전문서적의 각 장별로 대응하여 구성한다. 전문지식 ontology의 표현형식은 배경지식 ontology와 같은 네트워크로서 표현한다.

### 2.4.1 ontology의 데이터 표현형식

ontology는 배경지식 ontology, 전문지식 ontology 모두 공통형식으로 표현한다. 표현형식은 개념을 프레임 구조로, 관계를 슬롯(slot)으로 표현한다. 슬롯값으로는 개념명을 지정하는 것에 의해 개념간의 관계를 확실하게 한다. 또 슬롯의 종류를 여러개 설치하여, 복수종류의 관계를 표현한다.(Fig. 1)

### 2.4.2 전문문서편과 전문지식 ontology

전문문서편을 전문지식 ontology에 관련되는 개념에 대응하는 것을 맵핑(mapping)이라 한다. 전문문서편을 정의하는 데이터 구조는 Fig. 2와 같이 나타낸다. title, file, image는 각각 전문문서의 타이틀, 저장되고 있는 파일명, 화상 파일명을 나타낸다.

맵핑에 관한 정보는 맵에서 나타내는 슬롯에 저장된다. 이 슬롯에 들어오는 값은 개념명의 리스트이다. 외부 리스트의 구조에 의해 하나의 전문문서편이 복수의 개념에 맵핑되는 것도 표현할 수 있다. 내부리스트 구조에서는 맵핑하는 구조를 나타낸다. 맵핑되는 개념명이 개념A와 개념B인 경우는 전문문서 데이터 구조 중 맵 슬롯은 ((개념 A)(개념 B))와 같이 기술된다.

### 2.4.3 배경지식 ontology와 전문지식 ontology의 연결

배경지식 ontology를 구성하는 개념과 전문지식 ontology를 구성하는 개념으로 같은 것이 있으면,

전문문서의 정보구조단위

|                           |        |
|---------------------------|--------|
| document                  | 문서명    |
| title                     | 타이틀명   |
| file                      | 저장 파일명 |
| image                     | 화상 파일명 |
| map ((개념 1 ..... 맵핑하는 개념) |        |

Fig. 2 Data structure presenting a part of special document.

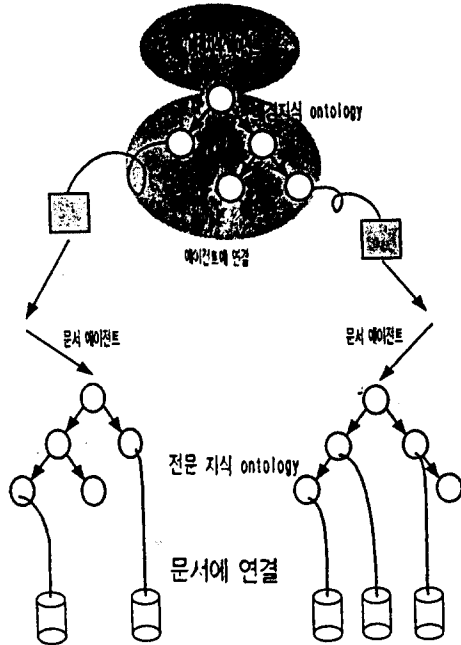


Fig.3 Linking a background knowledge ontology and a special knowledge ontology

그것을 배경지식 ontology와 전문지식 ontology의 접속점으로 한다.(Fig. 3) ontology는 에이전트에 의해 관리된다. 에이전트는 데이터와 그것을 조작하는 순서, 다른 에이전트와 통신 수단을 가진 소프트웨어 프로세서를 말한다.

실제로는 배경지식 ontology와 전문지식 ontology는 담당 에이전트를 정하여 각각 관리 하지만, 배경지식 ontology의 각 개념에는 같은 개념의 전문지식 ontology를 가진 에이전트의 이름이 개념의 프레임 구조의 1슬롯값으로 저장된다.

## 2.5 ontology 지식의 이용

배경지식 ontology를 어떻게 이용할 것인가에 대해서 나타낸다. 특히 응용시스템 구축에 관해서는 질문에 응답하기 위한 전문문서편을 추출하여 나열하는 것이 아

니고, 그것들이 어떠한 관계를 가지고 인출되었는가를 나타내는 구조표시를 행할 필요가 있다. 그러기 위해서는 질문의 정확한 파악과 ontology의 일부를 질문에 맞추어 인출해 오는 순서가 필요하다. 여기에서는 우선 어떠한 질문의 패턴이 있는가를 실제문제를 가지고 분석한다. 다음에 각각의 질문패턴에 대응한 방법, ontology를 인출해오는 순서를 도입한다.

### 2.5.1 예상되는 질문의 분류

본 연구에서는 다음 5종류의 질문패턴으로 분류하였다.

- 『...의 ~』
- 『~에 대해서』
- 『~과 ...의 관계』
- 『~이라는 관점에서 본다』
- 『... 별로 ~』

다음에는 그 분류 근거를 나타낸다.

전문서적<sup>4)</sup>에 있는 check list를 질문 예로 든다. 본 연구의 응용시스템에는 이것들의 질문에 직접 답을 나타내는 대신에, 답을 포함한 전문문서편의 위치를 나타내는 정보와 함께 제시한다. 그와 같은 경우에도 질문의도를 정확히 파악할 필요가 있다. 질문은 표현을 조금 변화시켜도 질문의도가 변화하지 않는다고 가정한다.

### 2.5.2 방법의 도입

#### 1) 『...의 ~』 : of 방법

『...의 ~』은 ...의 ~속성에 대해서 알고 싶다고 해석한다. 이것에 대응하는 방법을 of 방법이라고 하고 ...의 ~속성 개념을 중심으로 거기에서 1링크로 연결되는 개념까지 범위를 추출하는 순서를 행한다. of 방법은 하나의 개념에 착안하여 정보를 추출하는 방법이다.

#### 2) 『~에 대해서』 : about 방법

이 질문의 의도는 ~이라는 개념 그 자체와, 그것에 무엇인가의 관계가 있는 개념에 대해서 알고 싶다고 해석한다. 이것에 대응하는 방법을 about 방법이라고 하고 ~이라는 개념을 중심으로 1링크로 연결된 개념까지의 범위를 추출하는 순서를 가지고 있다. about 방법은

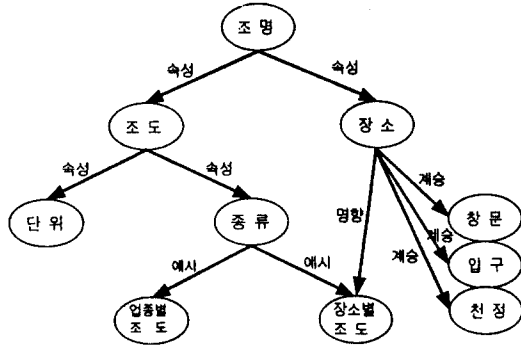


Fig. 4 Ontology of 'Lighting of shop'

하나의 개념에 착안하여 정보를 추출하는 방법이다. 이 질문의 예로 『주차장에 대해서』, 『부대시설에 대해서』 등 of 방법과 같다. Fig. 4에 있어서 『조도에 대해서』라는 질문을 하며 『조도』를 중심으로 『조명』, 『환경』, 『밝기』, 『계산방법』, 『종류』...라는 개념 그리고 그것에 따르는 문서가 추출된다.

3) 『~과 ...의 관계』 : relationship

relationship 방법은 동시에 2가지 개념에 착안하는 방법이다. 2가지의 개념이 미리 결정된 관계에 있는가를 조사한다. 구체적으로는 다음에 연결되는 3가지 중 어디에 해당되는가를 조사한다.

① 하나의 링크로 연결되어 있는 경우

상호간 개념이 직접하나의 링크로 관계를 유지하고 있는 경우도 있다.

② 두개의 링크로 연결되어 있는 경우

이 경우 중요한 관계는 『형제관계』, 『속성·예시 관계』가 있다.

형제관계 : 어느 개념의 속성 구성요소 등은 인간 가족형태의 형제의 예와 같다. 따라서 어떠한 공통개념에 관계하고 있는가라는 비교적 중요한 정보가 얻어진다.

속성·예시관계 : 예시관계는 어느 개념의 속성 개념에서의 관계이고, 원래 개념의 구체적 예(계승개념)를 나타내는 관계이다.

③ 세 개의 링크로 연결되어 있는 경우

일련의 순서관계에서 연결되어 있는 복수개념 등이 해당한다.

『점포의 조도』에 관한 ontology Fig. 4에 있어서, 『환경과 조도의 관계』라는 질문을 하면 『환경』, 『조도』라는 개념이 따르는 문서와 함께 『영향』 관계에 연결되어 추출된다

4) 『~이라는 관점에서 본~』 : viewpoint 방법

『~이라는 관점에서 본~』은, 어느 『~』사항에 대해 ...이라는 착안점을 가지고, ~에 대해서 분석하여 나타냄이라는 입장을 나타내고 있다고 해석한다. 『B라는 관점에서 본 A』라는 경우의 방법에 대해서는 A의 구성요소와 그 계승개념 B라는 속성에 관한 ontology 구조에 포함되어 있는 개념에 따르는 전문문서편을 추출하였다. 예를 들면, 『재료라는 관점에서 본 건물』의 의도를 생각해 보자. 당연히 건물에 사용되고 있는 재료라는 것은 건물의 구성요소 별로 다르기 때문에 『건물의 구성요소별 재료에 대해서 알고 싶다.』라는 것이 가장 정확하다고 생각되어 진다. 이 예에서 보면 『~』에 있는 개념명이 오고 『...』에는 그 개념의 속성개념이 온다고 생각되어 진다. 또 건물과 구성요소에 몇 가지 종류가 있는 경우는 그것들의 개념에 대해서도 같은 해석을 행한다.

5) 『...별로 ~』 : by 방법

『...별로 ~』이라는 해석은 『~이라는 점에 대해서 ~의 변수를 알고 싶다』라 한다.

이 문장의 뉘앙스에 대해서는, viewpoint 방법과 비슷하다. ontology 기술 방법으로 보면 복잡한 처리가 필요하다.

구체적인 예로 『업종별 조명방법』이 있다. 『업종』이라는 것은 『점포』의 『종류』개념으로 『조명방법』은 『조명』개념의 『방법』개념이라 생각하면, 이것을 표현하는 ontology는 복잡하게 되어 Fig. 5와 같은 구조가 된다.

가장 좋은 정보는 Fig. 5의 점선과 같이 직접적으로 ontology중에 나타나지 않는 것이 있다. 점선부분의 ontology존재는 시스템의 추론에 의해 채워진다. 특히 각 업종별로 조명방법에 대해서 전문문서편의 맵핑은



ARPA 지식공유활동에 관련하여 일리노이 대학에서 개발된 톨로서 Mbus와 Mbus · Common Lisp 사이의 기본 인터페이스인 KAPI<sup>5)</sup>가 있다.

각 에이전트의 기술은 Common Lisp에서 행한다. 메시지 통신에 관한 것은 KAPI의 제공하는 함수를 불러오고 에이전트 내부에서의 데이터 처리 등에 관한 부분은 Common Lisp언어를 이용하여 기술한다.

각 에이전트의 데이터와 ontology 저장방법으로서 Frolic<sup>2)</sup>(Common Lisp상에서 구축된 prolog 처리계), Frob<sup>3)</sup>(Common Lisp 상에서 구축된 Frame 언어)라는 언어를 이용한다.

일부 에이전트에 관해서는 범용 에이전트 기술언어 proto Lisp<sup>6)</sup>을 이용하여 기술할 수 있다. proto Lisp에서는 통신에 관한 기구를 쉽게 기술할 수 있다. Fig. 7에 에이전트 개발환경의 구조를 나타낸다.

### 3.3 시스템에서의 프로토타입 처리

프로토타입 시스템에 있어서 『업종별 조명방법』이라는 질문을 한 경우 시스템 내부 처리를 나타낸다.

#### 3.3.1 사용자 인터페이스의 처리과정

사용자로부터의 질문을 인터페이스를 통하여 입력한다. 질문은 인터페이스 에이전트를 통하여 배경지식 에이전트가 받아들이고, 배경지식 에이전트가 배경지식 ontology 정보를 이용하여 증개하므로써 관계되는 문서 에이전트에 질문이 전달된다. (Fig. 6의 (1) - (7))

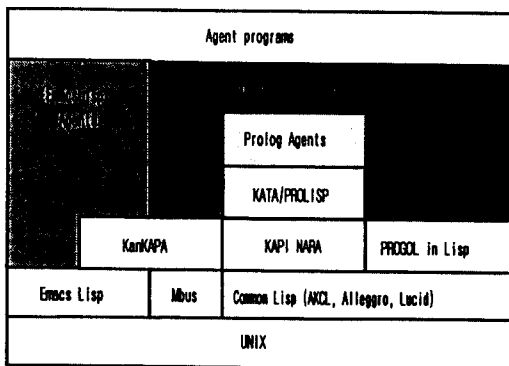


Fig. 7 Software structure of the agent.

#### 1) 입력

입력 사용자는 2종류의 항목에 대해서 입력한다. 첫째, 무엇에 대하여 조사하고 싶은가(what)와 둘째, 그것에 대해 어떠한 조사를 할 것인가(how)이다. what는 배경지식 ontology와 전문지식 ontology 중 각 개념에 대응하고 있다. what에 대해서는 2가지 항목에 대해서 입력하도록 하고, 각각 2가지 개념을 취하는 방법(『과 ...의 관계』, 『...이라는 관점에서 본-』, 『...별로-』)을 이용하는 경우에 대응하고 있다. how는 방법의 종류에 대응하고 있다.

여기서 『『업종별 조명방법』에 대해서 알고 싶다』라는 질문을 할 경우 what으로 『조명방법』과 『업종』을 입력하고, how는 『...별로-』를 선택한다.

#### 2) CGI-bin 프로그램

입력이 끝나면, 그것에 대응한 CGI-bin 프로그램이 서버에서 자동적으로 기동된다.(2) Mosaic에서 입력된 내용은 CGI-bin 프로그램에 표준 입력으로 주어진다.

CGI-bin 프로그램에서는 우선 입력 내용을 입력변수와 값으로 인식한다. 이 경우 what에 해당되는 입력 변수명이 what-to-do-1, what-to-do-2, how에 해당되는 입력변수명이 how-to-do 이다. 값은 각각 『조명 방법』, 『업종』, 『...별로-』이다.

변수명과 값을 파일 spool-file에 출력한다.(3) spool-file의 내용은 Fig. 8과 같이 나타낸다.

```
how_to_do_0 by
what_to_do_1 lighting-method
what_to_do_2 store-kind
```

Fig. 8 Contents of spool-file.

```
(ask-about
: content (by/lighting-method store-kind) ? tree)
: reply-with QO)
```

Fig. 9 Message from the interface agent to the background knowledge agent

```
(def-class lighting-agent agent ;; 상점의 조명
  : slots(
    (nic-name "lighting")
  )
)
```

Fig. 10 Data expression of the document agent.

CGI-bin 프로그램에서 인터페이스 에이전트를 기동한다.<sup>(4)</sup>

spool-file은 인터페이스 에이전트가 기동된 후 인터페이스 에이전트가 읽어들이는 것을 행한다. 즉, CGI-bin 프로그램에서 인터페이스 에이전트로 입력정보의 전달은 파일을 통해서 행한다.

### 3) 인터페이스 에이전트

인터페이스 에이전트 CGI-bin 프로그램이 만든 spool-file의 내용<sup>(5)</sup>으로부터 Fig. 9와 같은 메시지를 생성하여, 배경지식 에이전트에 송신한다.<sup>(6)</sup> 이 메시지는 KQML 형식으로 쓰여져 있다. ask-about은 질문의 되를 나타낸다. 또, : content로 나타내는 S식은 실제의 메시지 내용을 나타낸다.

### 4) 배경지식 에이전트

배경지식 에이전트는 앞에 나타냈던 메시지를 인터페이스 에이전트로부터 받으면, 우선, 『조명방법』과 『업종』의 양쪽을 알고 있는 문서 에이전트가 어느 에이전트인가를 조사한다. 이것은 배경지식 에이전트가 가진 배경지식 ontology의 『조명방법』과 『업종』이라는 개념을 참조하는 것에 의해 알 수 있다.

문서 에이전트에 관한 정보는 Fig. 10과 같이 데이터화 되어있다. 이것들에 의해 실제 문서 에이전트 이름이 "lighting"이라는 것을 알 수 있다. 여기에서 배경지식 에이전트는 "lighting" 에이전트에 Fig. 11과 같은 질문 메시지를 보낸다.<sup>(7)</sup>

### 3.3.2 필요한 정보와 문서 수집

질문을 받은 각 문서 에이전트는 각각의 전문지식

ontology로부터 정해진 순서에 따라 필요한 정보(필요한 ontology의 일부와 그것에 따르는 전문문서편)를 추출하여 배경지식 에이전트에 보낸다.

```
(ask-about
  : content (by (lighting-method store-kind)
    ?tree)
  : reply-with Q2)
```

Fig. 11 Message from the background knowledge agent to the document agent.

#### 1) 문서 에이전트

배경지식 에이전트로부터 질문을 받은 "lighting" 문서 에이전트는 Fig. 4에 나타낸 전문지식 ontology를 가지고 있다. 문서 에이전트에서는 들어온 질문에 따라 『업종별 조명방법』에서 by 방법이 적용된다. 그 결과 Fig. 4에서 강조된 부분과 점선 부분의 ontology 및 그 개념에 부수되는 문서가 추출된다. 점선 부분의 ontology는 시스템에 따라 존재가 추론되는 부분이다. 여기서 추출되는 개념의 정의를 Fig. 12에 나타낸다. 또, 『조명방법』 및 『종합상점의 조명방법』이라는 개념에 따라 문서의 데이터 표현을 Fig. 13으로 나타낸다.

문서 에이전트는 이와 같이 수집한 정보를 ontology 구조를 잘 표현할 수 있도록 관계를 나타내는 링크에 정리하여 회답을 구성한다. 구체적으로는 Fig. 14에 나타낸 것 같이 정리하여 배경지식 에이전트로부터의 질문에 대한 답을 나타내는 메시지로서 보낸다.<sup>(8)</sup>

### 3.3.3 수집한 문서의 재구성 제시

배경지식 에이전트는 각 문서 에이전트로부터 받은 회답을 통합하여 일단 인터페이스 에이전트에 보낸다. 다음으로 인터페이스 에이전트는 그 정보를 조합 에이전트에 보내어 사용자 인터페이스 표시용으로 조합을 행하도록 의뢰한다. 조합 에이전트는 받은 정보에 의해, 표시용의 파일을 작성하여 그 내용이 다시 인터페이스 에



이전트를 통하여 사용자 인터페이스에 도착하여 출력된다.(Fig. 6의 (9)-(17))

1) 배경지식 에이전트

배경지식 에이전트는 질문을 제시한 문서 에이전트로부터 순차적으로 받아들여 축적한다. 질문을 보낸 문서 에이전트로부터 회답이 되돌아 올 때, 배경지식 ontology로부터 얻어진 정보와 문서 에이전트로부터 보내온 정보를 통합하여 인터페이스에 보낸다.(9) 이 경우에는 문서 에이전트로부터 Fig. 14에 나타난 응답을 받아들인 후, 배경지식 에이전트 자체와 자기가 가지고 있는 배경지식 ontology를 참조하여 lighting-method, store-kind이라는 개념에 대해 by방법을 적용한 정보를 얻는다.

2) 인터페이스 에이전트

인터페이스 에이전트는 배경지식 에이전트로부터의 회답을 받으면, 얻어진 회답을 HTML 형식으로 고쳐 파일로 하고, 조합 에이전트에 Fig. 15와 같이 메시지를 보낸다.(10)

```
(def-class shutter comp-object ;; 조명
  : slots(
    (nic-name "조명")
    (partof "store") )
  : mv(
    (composite '(lighting-source))
    (attribute '(lighting-function
      lighting-method .....)) ) )
(def-class glass-screen mono-object ;; 조명방법
  : slots( (nic-name "조명방법") )
  : mv(
    (group '(window-group))
    (similar-relation '(show-window))
    (document '(05-010)) ) )
```

Fig.12 The definition of concept abstracted from a special knowledge ontology

```
(def-class 05-010 documentation ;; 조명방법
  : slots( (file-name "05-010")
    (title "조명방법") )
  : mv( (image ('("05-010-1" "05-010-2"))
    ) )
```

Fig. 13 Data expression of the document.

```
(tell
:content
  (by (lighting-method store-kind)
    ((attribute
      (store "점포" nil) (store -kind "업종"))
      (composite
        (store "점포" nil) (lighting "조명" nil))
      (attribute (lighting "조명" nil)
        (lighting-method "조명방법"
          ((| 05-010 | (| 05-010-1 |)"lighting")))))
    (example
      (store -kind "업종" nil)
      (super-market "슈퍼마켓"))
    (document
      (| 05-010 | (| 05-010-1 | )
      ((super-market ighting lighting-method))))
```

Fig. 14 Answer message from the document agent to the background knowledge agent

```
(ask-about
  : content
    (compile by (lighting-method store-kind)
      information
      ?file-name)
  : reply-with Q1)
```

Fig. 15 Message requesting combination from the interface agent to the interface combination agent.

3) 조합 에이전트

조합 에이전트는 조합을 의뢰하는 메시지가 오면, 내용을 판단하여 최종적으로 사용자 인터페이스에 정리하여 표시되도록 HTML 형식에 따라 조합을 행한다. 그때, 전문문서편의 파일명을 알고 있는 부분에 대해서는 실제로 그 파일의 내용을 전개하여 조합 속에 포함시킨다. 실제 조합의 내용은 어느 파일에 보존된다.(11)

```
(tell
  : content
    (compile about (storefront)
      information
"/home/masano-n/www/study/spool/compile-spool")
  :in-reply-to Q1)
```

Fig. 16 Answer message from the combination agent to the interface agent.

인터페이스 에이전트의 응답은 Fig. 16과 같이 질문 ?file-name인 곳에 조합 파일명(여기는 compile-spool)을 써서 보낸다.(12)

4) 재 인터페이스 에이전트의 예

인터페이스 에이전트는 조합 에이전트가 만든 조합 파일 compile-spool을 spool-file2라는 파일명으로 서버측(CGI-bin 프로그램이 있는 호스트)에 복사한다.(13) 이것으로 인터페이스 에이전트는 자동적으로 종료한다.(14)

5) CGI-bin 프로그램

CGI-bin 프로그램은 인터페이스 에이전트가 종료하면 spool-file2의 내용을 표준출력으로 출력한다. 이와 같이하여 출력된 내용이 client측에 보내진다.(15) CGI-bin 프로그램은 spool-file, spool-file2 등을 소거한 후 종료한다.(16)

6) 출력

회답표시의 처음은 질문내용인 『업종별 조명』이 표시된다. 그 밑의 『문맥의 구조』에서는 얻어진 ontology

구조에 문서의 타이틀 명을 표시한다.

4. 결 론

본 논문에서는 현재 많이 이용되고 있는 하이퍼 시스템 대용으로 지적 정보제공 시스템인 지식미디어의 사고를 도입한 전문문서 지식미디어 시스템을 제안하였다.

본 연구의 지식미디어 사고에 바탕을 둔 ontology는 배경지식의 ontology와 전문지식의 ontology로서 모두 대상분야의 전문서적을 보고 작성한다. 배경지식 ontology는 전문서적에 나타나는 어휘를 체계화시키고, 전문지식 ontology는 전문서적 내의 목차를 중심으로 한 정보로부터 가능한 것을 나타내고 있다. 전문문서편은 전문지식 ontology 중의 개념과 대응시켜 관리하는 것을 나타내고 있다.

이에 의해 전문문서 지식미디어 시스템의 프로토 타입 시스템을 구축하고, 일부 기능 실현을 제안하였다. 이 시스템은 이용자의 시점에서 재구성된 전문문서의 제시가 가능함을 나타내었다. 이상의 연구 결과로 지적인 정보제공시스템으로 지식미디어의 개념을 채택하여 대상분야의 지식을 ontology로서 관리, 이용하는 방법이 효과적이다.

전문분야 지식미디어 시스템의 응용은 문서의 편집, 통합시스템으로 이용이 가능하고, 동일분야의 전문서적을 여러개 준비하여 그것들을 본 연구에서 제시한 방법으로 지식미디어화 하는 것에 의해 복수의 전문서적 지식을 하나의 시스템으로 이용할 수 가 있다. 충분한 규모의 배경지식에ontology를 작성하면 전문지식 ontology 및 전문문서를 문서 에이전트로서 추가하여 취급하는 지식의 양을 증가시키는 것이 가능하다.

참 고 문 헌

1. Nicola Guarino, Massimiliano Carrara, and Pierdaniele Giarretta, An ontology of meta-level Categories, 1994.
2. Jed Krohnfeldt and Graig Steury. Frolic: Logic programming with frobs. Utah pass project opnote 86-08. Computer Science Department, Univ. of Utah, 1993.

3. Eric Muhle, Frobs user guide, Utah pass project opnote 87-05, Computer Science Department, Univ. of Utah, 1993.
4. 高嶋昌康, 店舗施設の 綜合知識, 誠文堂 新光社, 1984
5. 飯野健二, "KCL-KAPI에 대해서", 日本 NIAST, 1993.
6. 錦正信, 裴相賢, 에이전트 개발용 Tool, proto. lisp version 1.0, 日本 NIAST, 1994.
7. Robert Neches, etc., "Enabling technology for knowledge sharing," AI Magazine, Vol.12, No.3, pp. 36-56, 1991.
8. Rebert E.Kahn, etc., "The digital library project, Vol.1 : The World of Knowbots(DRAFT)," Technical Report, Corporation for National Research Initiative, 1988.
9. 西田豊明 etc., 智識 Community, pp. 176-189, 共立出版.
10. 清口理一郎, 전문가 시스템 I ~ III, 조창AI 라이브러리, 1993.