

## 종이책 환경과 유사한 지능형 Annotation 인터페이스 설계

손원성<sup>o</sup>, 김재경, 최윤철, 임순범<sup>†</sup>

연세대학교 컴퓨터과학과

<sup>†</sup>숙명여자대학교 멀티미디어학과

### Design of an Intelligent Paper-like Annotation Interface

Won-Sung Sohn, Jae-Kyung Kim, Yoon-Chul Choy, Soon-Bum Lim<sup>†</sup>,

Dept. of Computer Science, Yonsei University

<sup>†</sup>Dept. of Multimedia Science, Sookmyung Women's University

E-mail : {sohnws, ki187cm, ycchoy}@rainbow.yonsei.ac.kr, sblim@sookmyung.ac.kr

#### 요 약

전자문서 환경에서의 annotation은 종이 문서 환경에 비하여 다양한 장점을 제공하기 때문에, 현재 웹 및 hand-held 환경을 기반으로 하는 여러 분야에서 응용 및 적용되고 있다. 한편 annotation 인터페이스 중 paper-like annotation interface (PLAi)는 직관적인 자유형 마킹 생성 및 최소의 작업 오버헤드 보장 등의 장점을 제공한다. 그러나 현재의 annotation 시스템에서는 대부분 PLAi를 제외한 서로 다른 인터페이스를 사용하고 있으며 이는 모든 환경에 PLAi를 적용하기에는 기술적 혹은 사용성 측면에서 다수의 부정적인 요인을 포함하기 때문이다. 따라서 본 연구에서는 데스크탑 및 hand-held 등과 같은 다양한 환경에서도 적용 가능한 PLAi를 제안하고자 하며, 특히 자유형 마킹에 대한 인식 및 보정 인터랙션이 지원되는 지능형 annotation 입력 인터페이스를 설계한다. 그 결과 제안된 인터페이스의 인터랙션 기법들은 다양한 환경에서도 PLAi의 장점을 최대한 지원할 수 있는 동시에, e-Learning, 전자교본, 온라인 교정, 전자도서관, 전자책 등의 분야에 적용 가능하다.

#### 1. 서론

일반적으로 문서환경에서 annotation이란 문서의 주제 및 내용에 관한 해설, 설명, 그리고 강조를 목적으로 추가되는 문장 또는 텍스트를 의미하며[2,3,4,5] 특히 밑줄, 심볼, 노트 등과 같은 스타일 타입으로 표현된다[2,3]. 따라서 현재 다수의 전자문서[8] 환경 및 웹[6], 그리고 hand-held[15] 컴퓨터 환경에서도 annotation 기능은 다양하게 활용되고 있다.

한편 현재 사용되고 있는 annotation 인터페이스는 annotation 생성 관점에 기반하여 다음과 같은 4가지 범주(chat-boarding interface, line-based interface, symbol-based interface, paper-like interface)로 구분할 수 있다. CBI(chat-boarding interface)[6]는 동일한 주제에 관한 다중사용자간의 의견을 쓰레드 기반의 메시지로 표현하기 때문에 그 구조가 현재의 BBS의 경우와 매우 흡사하다. LBAI(line-based annotation interface)[7]는 인라인 형태의 라인, 하이라이트, 노트 등의 annotation 스타일 타입을 지원하며, annotation 생성을 위해서는 WIMP(Window, Icon, Menu, Pointing device or Pull-down menu)에 기반한 메뉴 선택과정을 수행한다. SBAI(symbol-

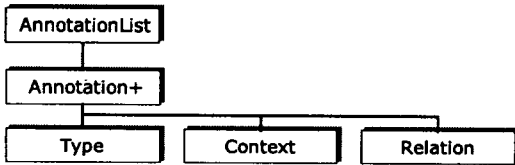
based annotation interface)[10]는 라인 및 노트 등과 같은 기본적인 annotation 타입 뿐만 아니라 타원, 박스 와 같은 심볼 및 마우스를 이용한 자유형 마킹(free-form marking) 기능을 제공할 수 있다. 한편 hand-held[15] 및 note-taking 시스템[8], 전자책 단말기[12] 등과 같은 환경에서는 대부분 펜 기반의 PLAi를 제공한다. PLAi는 종이문서 환경과 유사한 펜 기반의 자유형 마킹, 자유로운 공간 레이아웃(spatial layout), 그리고 기존 WIMP 보다 적은 작업 오버헤드(task overhead) 보장 등의 장점[11]들을 제공한다. 그러나 이러한 annotation 인터페이스들에서는 PLAi의 장점을 충분히 제공하지 못하고 있으며, 그 이유는 다양한 환경에 PLAi를 적용하기 위해서는 매우 정교한 기술적 또는 인터페이스 측면의 개선이 필요로 하기 때문이다[4].

따라서 본 논문에서는 데스크탑 및 hand-held 등의 다양한 환경에서 범용적으로 적용 가능한 PLAi를 설계 및 제안하며 특히 펜 및 마우스/키보드 환경에서도 사용하기 용이한, 영역 드래깅 annotation 입력 및 annotation에 대한 인식 및 보정 인터랙션을 지원하는 지능형 자유형 마킹 입력 인터페이스를 제안한다. 또한 본 인터페이스는 structural

context annotation markup language (SCAML)를 정의하여 annotation 정보를 XML 기반의 구조정보로 표현한다. 본 연구에서는 제안 인터페이스를 지원하기 위하여, 170여개의 규칙으로 구성된 규칙기반 시스템 및 annotation 인식 및 보정기법을 제안한다. 그 결과 본 논문에서 제안된 인터페이스는 데스크탑 및 hand-held 등을 포함한 다양한 환경에서도 PLAI의 장점을 제공할 수 있으며, 특히 웹 및 하이퍼텍스트에 기반한 온라인 교정 (proof-reading), e-Learnig, 전자교범 (Interactive Electronic Technical Manuals), 전자도서관, 전자책 등에 효과적으로 적용 가능하다.

## 2. Annotation 마크업 모델

제안된 인터페이스에서는 생성된 annotation 정보를 XML 기반의 구조 정보로 저장하며 이를 최종적으로 출력 및 저장한다. 이를 위하여 본 연구에서는 structural context annotation markup language (SCAML)를 정의하였다. SCAML은 다음 [그림 1]과 같이 크게 annotation type, context, 그리고 생성 annotation에 대한 relation 엘리먼트로 구성된다.

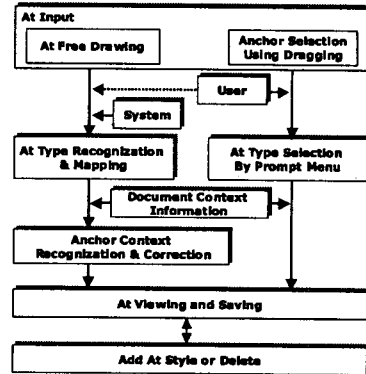


[그림 1] SCAML의 구조

Annotation type 엘리먼트는 lines, symbol, note, bracket 엘리먼트로 구성되며, 각 타입 속성에는 tightly 및 loosely type 정보를 포함한다. 이러한 속성 정보는 본 연구에서 제안한 규칙기반 annotation 인식 및 보정 시스템에서 각 타입별 규칙을 선택하는 기준이 된다. Context 엘리먼트는 annotation의 anchor 원문을 표현하기 위한 nonstructural 엘리먼트와 문서의 구조 및 읍셋정보를 포함한 structural 엘리먼트로 구분된다. Relation 엘리먼트는 placeholder, readingAssistant, selfThinking을 목적으로 하는 remind, mark, interpretive, idea, criticism 과 같은 의미적 (semantic) 속성을 제공하며, 동시에 사용자 고유의 의도를 반영하기 위한 userDefine 속성을 제공한다.

## 3. 지능형 annotation 인터페이스

본 장에서는 annotation 인식 및 보정 기능을 포함하는 지능형 annotation 인터페이스에 대하여 설명한다. 본 인터페이스는 XML기반의 전자책[1]을 대상으로 하며, 전체 구조는 다음 그림 2와 같이 annotation 입력, 인식 및 보정, 출력 그리고 저장 단계를 포함한다.



[그림 2] 제안 인터페이스의 전체 구조

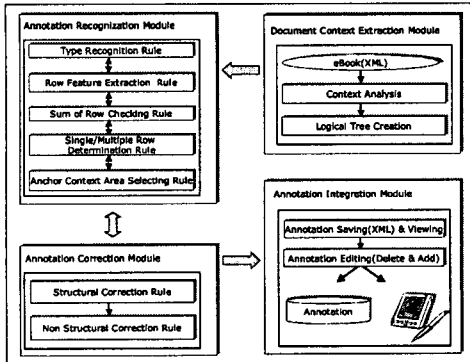
### 3.1 Annotation 입력 인터페이스

본 논문에서는 다양한 환경에서의 효과적인 annotation 생성을 위하여, 펜 및 마우스/키보드 인터페이스의 장점[9]을 최대한 지원하는 동시에, annotation 인식 및 보정 기능을 고려한 지능형 annotation 입력 인터페이스를 설계하였다. 이러한 입력 인터페이스는 자유형 마킹 방식 및 드래깅 방식을 지원하는 2가지 입력 모드를 제공한다. Annotation 타입 인식 및 영역 보정 인터랙션에 기반한 자유형 마킹 인터페이스는 스타일 팔레트 및 별도 윈도우 생성 등을 의한 메뉴 선택 과정을 생략할 수 있는 동시에[9], 마우스 환경에서도 마킹을 용이하게 입력할 수 있다. 또한 제안 마킹 인터페이스는 기존 PLAI 환경과는 달리 마킹에 대한 정확한 구조화가 가능하기 때문에 이를 지식정보로서 활용할 수 있다. 한편 펜 입력 방식은 마우스 방식에 비하여 정확성이 부족하기 때문에 초보자들에게는 비효율적인 결과를 제공할 수도 있다[9]. 따라서 본 인터페이스에서는 기존 마우스 방식의 장점[11]인 정확한 annotation 영역 지정[9] 기능을 제공하기 위하여 드래깅에 기반한 annotation 생성 방법을 사용한다. 이러한 인터페이스에서는 정확한 annotation 타입 및 영역 지정이 가능하며, 또한 발생할 수 있는 anchor 영역 인식 및 타입 매핑 오류를 방지할 수 있다. 이러한 입력 인터페이스의 자세한 내용은 4절에서 살펴본다.

### 3.2 Annotation 인식 인터페이스

Annotation 인식 및 보정, 그리고 출력, 저장을 위한 인터페이스는, 172개의 규칙 모델로 구성된 인식 및 보정 모듈, 그리고 최종적으로 보정된 SCAML 저장 및 출력을 위한 통합 모듈로 구성된다. Annotation 인식 및 보정은 1:1 매핑 처리가 불가능한 매우 다양한 경우의 수를 포함하기 때문에, 본 연구에서는 다음 [그림 3]과 같은 인지적 특성을 반영한 규칙 모델을 구축하고 이를 처리하기 위한 기법을 제안한다. 본 연구에서는 먼저 annotation 인식에 대한 내용을 살펴본

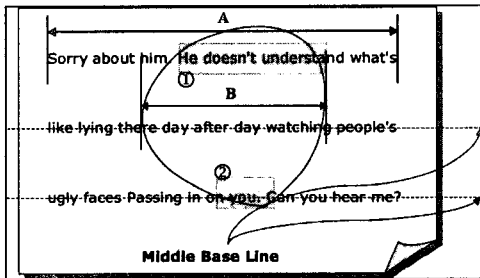
다.



[그림 3] 규칙기반 annotation 인식 보정 인터페이스

### 3.2.1 타입 인식 및 Row Feature 추출 규칙

입력 장치로부터 생성된 자유형 마킹은 생성과 동시에 그래픽정보를 이용하여 [13,14] 본 연구에서 정의한 annotation 스타일 타입과 매핑된다. 또한 매핑 결과에 따른 정보는 Row Feature Extraction 규칙 모델로 전달되어, 각 타입에 따른 적절한 정보를 추출하게 된다. 한편 Row Feature Extraction 규칙 모델에서는 생성된 annotation과 원본문서 문맥 (context)간의 접점 여부를 판별한다. 제안된 방법에서 Row Feature이란 생성된 annotation의 상하 영역과 원문 텍스트의 중간 지점 (MBL: Middle Base Line)과의 연관정보를 의미한다. 또한 이를 추출하기 위하여 본 기법에서는 상단 및 하단 Row와의 접점 여부 판별을 위한, 식 (1)와 같은 임계값 (Threshold of Upper or Lower Row Area) 및 IF 조건 AND ...AND 조건 THEN 적용될 방법) 형태로 표현되는 규칙 (11)을 적용한다. 본 논문에서는 이러한 규칙 모델을 설명하기 위하여 다음 [그림 4]와 같이 마킹된 타원을 예제로 사용한다.



[그림 4] 자유형 마킹에 의한 타원 입력의 예

[그림 4]의 예제는 제안된 임계값의 규칙에 따라, annotation 전체 영역(B)과 상하단의 Row가 MBL의 일정 영역 이상(1) 또는 이하(2)되는 경우만이 판별 대상으로 간주

되며, 규칙 적용에 의하여 최상단의 경우만을 접점으로 판별한다.

$$ThURA = \frac{\sum \text{최상단의 AnchorRow에서 MBL의 50\%이상되는 영역의 픽셀길이}}{\sum \text{생성된 Annotation의 Xmin, Xmax의 총길이}} \quad (1)$$

규칙 (11);

- IF: (1) Annotation 타입이 타원이다.  
 (2) Anchor Text 영역이 존재한다.  
 (3) Ax의 Ymin, Ymax 사이에 Row가 2개 이상 존재한다.  
 (4) 가장 상단의 Anchor Row가 존재한다.  
 (5) ThURA > 0.3 이다.
- THEN: (1) 상단의 Anchor Row를 Row로 간주한다.  
 (2) 하단 Row Feature 정보를 추출한다.

### 3.2.2 총 Row 개수 판별 규칙

본 규칙 모델에서는 Row Feature Extraction 규칙 모델에서 판별된 정보를 토대로 annotation 타입 내부에 포함된 총 열 (Row)의 개수를 추출하게 된다. 예를 들어 단일 라인에 생성된 annotation은 한 개의 열 정보를 추출하며, 복수 열에 걸쳐 생성된 경우, 각 상하 접점 여부에 따라 내부의 총 열 수가 변경될 수 있다.

### 3.2.3 단수 복수 판별 규칙

본 규칙 모델에서는 앞서 설명한 규칙 모델의 결과인 총 열 수를 이용하여 생성된 annotation에 대한 단/복수 여부를 판별하게 된다. 제안된 모델에서는 기본적으로 하나의 열을 단수, 2개 이상의 열을 복수의 경우라 지정한다. 그러나 앞서 설명한 모델에서 추출한 열 수가 2개 이상 되더라도 생성된 annotation의 좌우길이 (그림 4의 B)와 원본문서 컬럼의 총길이 (그림 4의 A)를 이용한 임계값 (Threshold of Single/Multiple Row)의 결과가 일정 이상이 되는 경우에만 복수로 판별한다. [그림 4]의 예제는 2개 이상의 열을 포함하고 있지만, 제안된 규칙에 의하여 단수로 처리된다.

### 3.2.4 앵커 영역 선택 규칙

Annotation 인식 모듈의 마지막 규칙 모델에서는 단복수 정보에 따른 annotation 영역의 최종 열 선택 및 시작과 끝 지점을 추출하게 된다. 예를 들어 [그림 4]의 예제는 해당 열 선택을 위한 임계값 (Select Upper or Lower Context) 및 다음 규칙 (82)를 적용하여 중간의 열을 앵커 열로 선택하고 동시에 시작 및 끝 지점을 추출하게 된다.

규칙 (82);

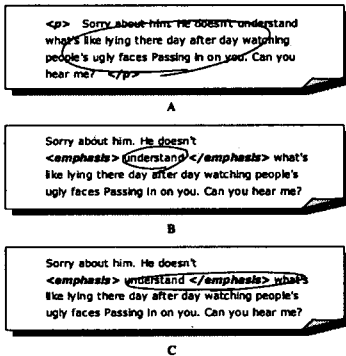
- IF: (1) Annotation 타입이 타원이다.  
 (2) 단일 Row 로 판별되었다.  
 (3) Row의 총 수가 1개이다.  
 (4) SUC < SLC 이다.
- THEN: (1) 생성된 Annotation의 Xmin, Xmax 지점과 만나는 하단의 Row 지점을 시작/끝 음셋으로 지정한다.  
 (2) 구조 보정 툴을 수행한다.

### 3.3 Annotation 보정 인터페이스

Annotation 인식모듈을 통하여 추출된 열의 개수 및 단/복수 여부, 앵커의 시작 및 끝 지점 등과 같은 모든 정보는 annotation 보정 모듈로 전달되어 구조 및 비구조 정보에 의해서 최종적으로 영역 보정된다. 이러한 과정은 다음의 내용과 같다.

#### 3.3.1 구조 보정 규칙

제안된 보정 규칙모델에서는 구조정보를 우선적으로 고려하며, 다음 [그림 5]와 같이 annotation 과 구조정보가 외포(Exclude; A), 내포(Include; B), 중복(Overlap; C) 된 경우를 보정 대상으로 간주하며 이를 구조적 annotation이라 명한다. 또한 각각의 경우에서 임계값의 일정 이상되는 경우에만 구조 정보를 우선적으로 보정 기준으로 판단한다. 한편 [그림 5]의 내포된 경우는 식 (2)와 같은 임계값 (Threshold of Include Structure Correction Area)과 다음 규칙 (110)을 적용하여 보정의 대상으로 간주된다.



[그림 5] 구조적 annotation의 예

$$ThISCA = \frac{\sum \text{엘리먼트 내부의 시작에서 끝까지의 음셋}}{\sum \text{생성된 Annotation의 전체 음셋}} \quad (2)$$

규칙 (110);

- IF: (1) Annotation 타입이 타원이다.  
 (2) Annotation의 내부에 완전한 엘리먼트가 존재한다.  
 (3) Annotation 내부의 Context 음셋이 10이상이다.  
 (4)  $ThISCA > 0.8$  이다.
- THEN: (1) 시작태그 지점을 시작 음셋으로, 끝 태그 지점을 끝 음셋으로 지정한다.  
 (2) 다음 규칙을 수행한다.

#### 3.3.2 비구조 보정 규칙

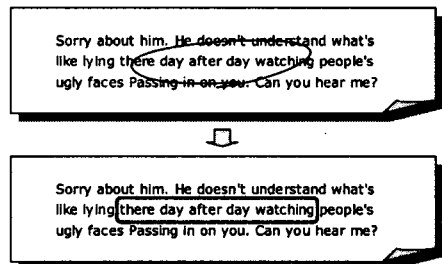
비구조 정보를 이용한 annotation 영역 보정은 앞서 설명한 구조 정보가 반영되지 않는 경우에만 적용된다. 특히 제안된 규칙모델에서는 annotation 영역이 context에 정확히 걸쳐 있지 않았을 경우 단어 및 문장을 기준으로 보정작업을 수행하며, 다음 식(3)과 같은 임계값 (Select Start or End Word

Area) 을 사용한다. 본 절에서는 [그림 4]를 비구조 보정하기 위하여 규칙 (139) 등을 적용하였고, 그 최종 보정된 결과는 다음 [그림 6]의 내용과 같다.

$$ThSSWA = \frac{\sum \text{현재 지점에서 X+방향으로 존재하는 공백이전까지의 음셋}}{\sum \text{현재 지점의 Context가 속한 단어의 전체 음셋}} \quad (3)$$

규칙 (139);

- IF: (1) Annotation 타입이 타원이다.  
 (2) Single Line의 속성을 포함한다.  
 (3) 구조보정 규칙을 수행하였다.  
 (4) Annotation의 Xmin 지점이 시작 지점이다.  
 (5) 현재 지점이 공백이 아니다.  
 (6)  $ThSSWA \leq 0.5$
- THEN: (1) X+ 방향으로 이동하여 공백 다음의 첫 문자를 시작 Anchor 지점으로 지정한다.  
 (2) Xmax End Rule 수행한다.



[그림 6] 그림 4 예제에 대한 비구조보정 결과

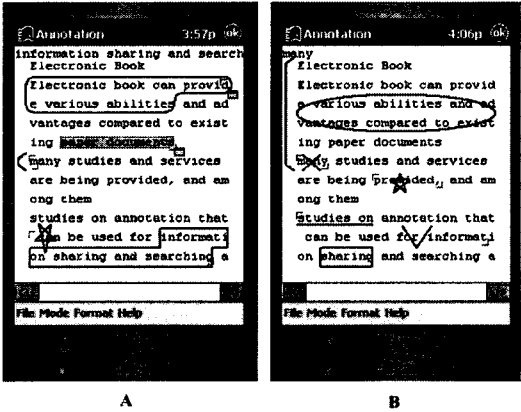
### 4. Annotation 시스템 구현

본 연구에서는 제안 인터페이스를 기반으로 하는 지능형 annotation 시스템을 구현하였으며, 본 시스템은 XML 기반의 전자책 문서 표준을 원본 문서로 사용하였다. 본 장에서는 이러한 구현 결과 중 PDA 환경에서 구동되는 annotation 브라우저와 이에 포함된 제안 PLAi의 적용 결과를 설명하도록 하며 시스템의 전체 화면은 다음 [그림 7]과 같다.

제안 시스템에서는 [그림 7]과 같이 2가지 입력 모드를 선택하기 위한 메뉴만을 필요로 하며, 부가적으로 annotation 컬러 등과 같은 편의기능을 지원한다. 한편 annotation 입력은 [그림 7. A]와 같은 드래깅 방식과 [그림 7. B]와 같은 자유형 마킹 인터페이스를 통하여 수행되며, 모든 결과는 제안된 annotation 모델 및 인터페이스에 기반한다. 특히 [그림 7. A]에서는 드래깅 기법에 의한 다양한 (idiosyncratic) 형태의 annotation 생성 결과를 나타내고 있으며, [그림 7. B]에서는 자유형 마킹 생성에 따른 annotation 인식 및 보정 결과를 보여주고 있다.

한편 [그림 7]에서 생성된 annotation은 본 논문에서 제안한 SCAML 형태의 구조정보로 저장되며, 특히 원문의 context 정보와 시스템 모듈에서 추출한 annotation 앵커 정보와 함께 원문과 별도의 형태로 저장된다. 이러한 저장 결과

는 다음 [그림 8]과 같으며 저장된 annotation 정보는 annotation 통합 모듈을 통하여 편집 및 출력 그리고 재저장 된다.



[그림 7] 시스템 구현 결과화면

```
<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE report SYSTEM "annotation(new).dtd">
<annotationList>
<annotation>
<type>
<character color="16744576" page="1" input type="dropping" x1="124" y1="287"
x2="187" y2="348" id="user" aid="533422"/>
</type>
<context>
<nonStructural>electronic book can provide various abilities</nonStructural>
<structural>
<startSegment offset="0" part="/body[2]/chapter[0]/p[4]/font[0]" totalOffset="137"/>
<endingSegment offset="16" part="/body[2]/chapter[0]/p[4]/font[0]" totalOffset="148"/>
</structural>
</context>
<relation type="idea"/>
</annotation>
<annotation>
<type>
<addNote color="16744576" page="1" id="user" aid="ref533422">
characteristics of ebook
</addNote>
</type>
</note>
</context>
.....
</annotation>
</annotationList>
```

[그림 8] 생성 annotation에 대한 저장 결과

### 5. 결론 및 향후연구

PLA는 종이책 메타포에 기반한 펜 기반 마킹, 최소의 작업 오버헤드 보장 등의 장점을 제공하며, 이러한 PLA를 다양한 환경의 컴퓨터환경에 적용하기 위해서는 인터페이스 측면의 개선을 필요로 한다.

이를 위하여 본 논문에서는 지능형 annotation 인터페이스를 설계하였고, 특히 마킹에 대한 인식 및 보정 인터랙션과 이에 대한 처리 기법, 그리고 드래깅의 의한 다양한 annotation 입력을 보장하는 인터페이스를 제안하였다. 또한 제안된 인터페이스에 기반한 규칙 기반 시스템을 제안 및 구현하였고 본 시스템에서는 정확성을 고려한 드래깅 입력과 자유형 마킹에 대한 인식 및 보정 인터랙션을 포함한다.

이러한 인터페이스 및 기법은 웹 및 hand-held 환경에서의 cyber-class, 그리고 IETM등의 도메인에 직접 적용 및 활용

가능한 동시에 다양한 장점을 제공할 수 있을 것으로 예상된다. 한편 제안된 시스템에서는 원본문서 구조 변경 및 다중사용자간의 annotation 교환은 고려하지 않았다. 따라서 향후 본 연구에서는 갱신된 구조문서에서의 다중사용자간 annotation 처리 기법을 제안하고, 그 결과를 다양한 도메인에 적용할 예정이다.

### [참고문헌]

- [1] Won-Sung Sohn, etl, al., Standardization of eBook documents in the Korean Industry, Computer Standards & Interfaces, Vol. 24, issue 1 (2001)
- [2] Bill N., Gene G., and Morgan N. (2000), Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations, Proc. ACM Int'l Conf. CHI (1998)
- [3] Catherine C. Marshall, Annotation: From Paper Books to Digital Library, Proc. of the 2nd ACM International Conference on Digital Libraries, ACM, Philadelphia (1997)
- [4] Iliia A. Ovsianikov, Annotation Technology, Int. J. Human-Computer Studies V.50. N.4. (1999) 329-362
- [5] Catherine C. Marshall, Toward an Ecology of Hypertext annotation, Proc. of HyperText, ACM, Pittsburgh (1998)
- [6] Laliberte, D., HyperNews, <http://www.hypernews.org/HyperNews/get/hypernews.html> (1997)
- [7] YEE, K.-P., The CritLink Mediator, <http://crit.org/critlink.html> (1997)
- [8] Price, M., Schilit, B., XLibris: the active reading machine, CHI98, ACM, LA (1998)
- [9] Takayuki, K., Potentials and Limitations of Pen-Based Computers, Proceedings of the 1993 ACM conference on Computer science, ACM, (1993)
- [10] iMarkup, <http://www.imarkup.com>, Last modified (2001)
- [11] André Meyer, Pen computing: a technology overview and a vision, ACM SIGCHI, Volume 27, Issue 3 (1995) 46-90
- [12] Microsoft, MS Reder, <http://www.microsoft.com/reader/>, Last modified (2001)
- [13] Krzyzak, Adam, W. Dai, Ching Y. Suen, On the Recognition of Handwritten Characters Using Neural Networks, Pattern Recognition—Architectures, Algorithms & Applications, World Scientific, Singapore (1991) 115-135
- [14] Suen, Ching Y., etl, al., Computer Recognition of Unconstrained Handwritten Numerals, Proc. of the IEEE, Vol. 80, Issue 7, (1992) 1162-1180
- [15] Baldonado, M., etl, al., Notable: At the intersection of annotations and handheld technology, Handheld and Ubiquitous Computing 2000. Springer Verlag. LNCS, 1927