

휴대용 단말기 환경을 위한 Annotation 모델링 및 시스템 구현

손원성

경인교육대학교 컴퓨터교육과

요약

어노테이션(annotation)은 문서에서 개인의 의견, 정리, 요약 등을 표현하기 위한 주석을 의미한다. 따라서 전자문서에서도 어노테이션은 중요하게 사용되며 특히 전자 잉크(digital inking)를 이용한 이동 단말기 환경에서 효과적으로 사용된다. 그러나 기존 연구에서는 휴대용 단말기 환경의 단점인 적은 디스플레이 공간을 전혀 고려하지 않기 때문에 어노테이션 작성 및 활용이 매우 불편하다. 따라서 본 논문에서는 전자펜과 이동식 단말기 환경을 고려한 어노테이션 모델 및 시스템을 제안한다. 제안 어노테이션 모델은 다양한 컨텍스트(context)를 고려하고 이에 기반한 어노테이션 마크업 언어를 정의한다. 본 모델은 다양한 어노테이션 타입 및 의미(semantic) 모델, 펜 기반 어노테이션의 자동 인식 및 영역 보정 기능 등을 고려하며, 이것을 기반으로 CAML(Context-based Annotation Markup Language)를 정의한다. 또한 본 모델을 이용하여 XML 기반의 전자책문서 및 단말기 환경을 고려한 어노테이션 시스템을 구현하고 그 활용 가능성에 대하여 살펴본다. 본 연구의 결과는 eLearning, Cyber-Class, IETM(Interactive Electronic Technical Manuals)에서 적절히 응용 가능하다.

Annotation Modeling and System Implementation for Hand-held Environment

Won-Sung Sohn

Gyeongin National University of Education, Dept. of Computer Education

ABSTRACT

For the accurate creation of annotation information in a free-form annotation environment, the ambiguity that arises in the analysis stage between the geometric information and annotations needs to be resolved. Therefore, this This paper identifies, analyzes, and proposes presents solutions methods for the ambiguity that can occur between free-form marking and various contexts in XML-based annotation environment. The proposed method is based on context which includes various textual and structure information between free-form marking and annotated part. The proposed method show that the annotated portions areas included in the free-form marking information are more accurate, achieving more accurate exchange results amongst multiple users in a heterogeneous document environment. This study can be effectively applied to eLearning, Cyber-Class, and IETM

Keywords: Annotation, XML, Annotation Modeling, Annotation Markup Language, Hand-Held GUI

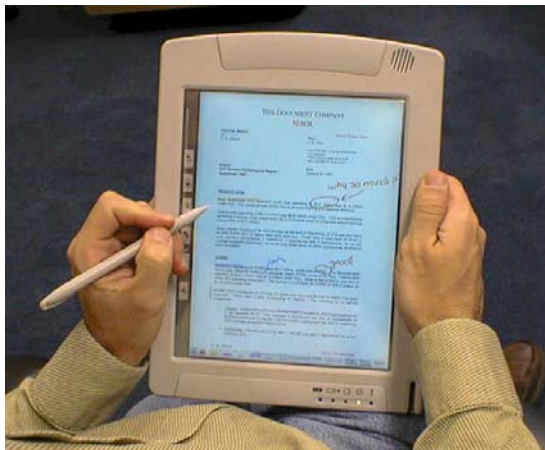
1. 서론

어노테이션이란 일반적으로 문서의 주제 및 내용에 관한 해설, 설명, 그리고 강조를 목적으로 추가

되는 문장 또는 텍스트를 의미한다[1, 2, 3]. 종이문서에서의 어노테이션은 문서 내용에 대한 정리, 요약, 이해, 기록 등을 위하여 사용되며, 특히 밑줄, 심볼, 노트 등으로 표현된다[1, 2]. 한편 전자문서

환경에서의 어노테이션 정보는 일회성 정보가 아닌, 재사용 및 공유될 수 있는 중요한 정보이기 때문에 다양한 분야에서 응용되고 있다[4, 5].

한편 전자펜(digital inking)을 이용하는 PDA, 태블릿 PC, 핸드폰 등과 같은 휴대용(hand-held) 단말기(그림 1)에서 어노테이션은 필수 기능으로 제공되며, 그 이유는 실세계에서와 유사한 어노테이션 입력 인터페이스를 사용하기 때문이다.



(그림 1) 휴대용 기기에서의 어노테이션 사용 예[6]

따라서 유비쿼터스 교육 환경에서도 어노테이션은 중요한 요소로 활용될 수 있다. 실제로 마이크로소프트사에서는 책가방 없는 교실을 목표로 자유로운 필기가 가능한 태블릿 PC를 무상으로 제공하는 동시에 다양한 하드웨어 및 소프트웨어를 개발하고 있다[7].

그러나 위와 같은 휴대용 기기들은 적은 디스플레이 화면과 전자펜을 이용하기 때문에 다음과 같은 단점[8]을 포함한다. 첫째 전자펜에 익숙하지 않은 사용자의 경우 정확한 어노테이션 작성에 상당한 시간과 노력이 요구된다. 따라서 정확한 어노테이션을 생성하기 위해서는 입력, 삭제, 재입력 등과 같은 일련의 과정을 반복하게 되며 이로 인한 피로도가 증가하게 된다. 둘째로는 기존 연구의 어노테이션은 이미지 형태로 저장되기 때문에 차후에 이것을 재사용하기 어려운 구조를 포함하고 있다. 그 결과 유비쿼터스 환경과 같이 지식에 대한 공유를 위해서는 컴퓨터와 사람이 정확히 인식할 수 있는 형태의 정보로 가공되어야 한다[9, 10].

따라서 본 연구에서는 휴대용 단말기에 익숙하지 않은 사용자들도 쉽게 어노테이션을 작성할 수 있는 지능형 인터페이스를 제안한다. 제안 인터페이스

는 사용자의 입력과 동시에 주변의 문자 및 그림, 여백, 구조 등과 같은 다양한 컨텍스트(context)를 이용하여 자동으로 영역을 보정한다. 또한 입력된 어노테이션을 본 연구에서 정의한 XML 기반의 어노테이션 모델로 저장되어 다중 사용자간에 정확히 공유될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 어노테이션과 관련된 선행 연구에 대하여 알아본다. 3장에서는 컨텍스트 기반의 어노테이션 모델을 제시하고 4장 및 5장에서는 제안 시스템과 구현 결과를 설명한다. 마지막으로 6장에서 결론 및 향후 연구 방향에 대하여 기술한다.

2 관련연구

2.1 어노테이션 시스템

어노테이션의 가장 기본적인 형태는 흔히 우리가 '리플'이라 칭하는 Chat-board 형태의 인터페이스(CAi: Chat-board Annotation interface)이며 Hyper News[2]에서 그 예를 살펴볼 수 있다. 이는 사용법이 간단하고 구현이 쉬우나, 원본문서와 사용자들의 어노테이션이 시각적으로 연결되지 않는 단점이 있다.

한편 ComMentor[11], CoNote[12]등의 시스템에서는 보다 시각적으로 직관적인 어노테이션 입력 인터페이스(LAi: Line-based Annotation interface)를 제공하며, 특히 인라인(문서 내부에 직접 입력되는 경우)형태의 밑줄, 형광펜, 메모 등의 어노테이션 타입 및 검색기능[9] 등을 제공한다.

이와는 달리 iMarkup[13], Equill[14]등에서는 심볼 형태의 어노테이션이 지원되는 그래픽 기반의 인터페이스(SAi: Symbol-based Annotation interface)를 제공하며 이는 사용자가 그래픽 툴을 이용하듯이 간단한 선이나 도형 등을 메뉴를 통하여 입력한다.

Xlibrary[6]와 전자책 단말기[15] 등에서는 단말기 환경에서 펜을 이용한 자유 입력(free drawing) 형태의 어노테이션 입력 인터페이스(PAi: Paper-based Annotation Interface)를 제공한다. 그러나 이 경우는 입력된 어노테이션들을 이미지로 삽입하여 교환이나 재사용시 응용성이 떨어질 수 있다.

2.2 기존 인터페이스에 대한 사용성 평가

본 논문에서는 세부적인 설계에 앞서 앞서 살펴본 4가지 인터페이스의 프로토타입을 구현하고 이

에 대한 사용성 평가(usability inspection)를 수행하였다. 이를 위하여 본 논문에서는 각 프로토타입에 대하여 휴리스틱 평가(heuristic evaluation)[16]를 시행하였고, 평가에 포함된 사용자들의 요구사항을 도출하여 그 결과를 본 논문의 어노테이션 모델 및 인터페이스 설계에 반영하였다. 평가를 위하여 우리는 대부분 어노테이션 시스템을 처음 사용해본 컴퓨터과학 전공의 대학원생들 및 개발자로 구성된 총 24명의 8개 그룹을 구성하였고, 또한 각 그룹 간의 평가자(evaluator)들에 대하여 서로 상이한 순서를 포함하는 평가 절차를 적용하였다. 또한 평가의 원칙은 Jakob Nielsen이 제안한 10가지 원칙을 기본으로 하며, 16개의 휴리스틱 및 1부터 6까지의 severity에 기반하며, 동시에 3이하의 심각도(severity) 항목에 대해서는 별도의 설문 조사를 이용한 경험적 평가(empirical study)를 수행하였다. 본 내용에서는 테스트 결과 중 중요한 항목만을 다음 표 2와 같이 정리하였고, 이를 기준으로 평가에 대한 내용을 설명하도록 한다.

〈표 1〉 4가지 어노테이션 인터페이스에 대한 휴리스틱 평가 결과¹⁾

Heuristics	CBi	LAi	SAi	PAi
인터페이스 구성 및 기능 만족도	12	58	54	54
Error preservation	95	67	33	17
Visual feedback	17	54	66	67
Readability	38	71	42	54
Flexibility and efficiency of annotation input	17	63	38	71
Satisfaction rate of annotation type	13	58	75	63

먼저 전체 인터페이스 구성 및 기능과 관련하여, CBi를 제외한 나머지 인터페이스들에 대한 만족도는 비교적 유사하였다. CBi의 경우 평가자들은 대부분 스타일타입이 지원되지 않는 점을 지적하였고, SAi의 경우 심볼 타입 기능에 대해서는 대부분 만족하지만 입력 인터페이스가 너무 복잡한 것이 오히려 LAi의 경우보다 만족도가 낮은 이유가 되었

1) 본 평가에서는 1에서 6까지의 심각도 중 4이상의 심각도를 평가 항목에 만족하는 것으로 간주하였다. 또한 다음의 수치들은 4이상의 심각도를 선택한 평가자들의 비율을 의미한다.

다. 그리고 PAi가 예상외로 낮은 만족도를 나타내었는데, 이는 단말기에서 심볼 타입을 지원하지 않는 것과 펜을 이용한 마킹(marking) 결과에 만족하지 못한 것이 가장 큰 요인으로 분석되었다.

어노테이션 입력시 발생할 수 있는 오류와 관련해서는 PAi와 SAi에서 자유형 마킹(free-form marking) 결과에 대한 불만이 만족도를 저해하는 가장 큰 요인이었으며, 전자펜에 기반한 어노테이션 인터페이스 설계시 에러 방지(error preservation)에 대한 중요성을 인식할 수 있었다.

어노테이션 생성 결과에 대한 시각적 피드백(visual feedback)과 관련하여 CBi를 제외한 나머지 경우가 비교적 비슷한 수치를 나타내었으나, PAi 평가자에 대한 설문조사에서는 상당수가 전자펜 기반의 어노테이션 입력시 정확한 영역을 선택하기 어려운 것을 개선사항으로 제시하였다.

원본문서와 어노테이션간의 가독성(readability)과 관련하여 흥미로운 점은 SAi와 PAi에서 생성된 불규칙한 자유형 마킹이 가독성의 저해요소가 될 수 있다는 것이며, 어노테이션 타입 및 생성 영역에 대한 제한이 오히려 가독성을 높일 수 있다는 가능성을 확인하였다.

어노테이션 입력 인터페이스와 관련하여 반수 이상의 참여자들은 현재의 인터페이스에 만족하고 있음을 알 수 있었으나, 나머지 평가자들에 대한 설문조사 결과 간결하면서도 정확한 입력이 가능한 인터페이스를 요구하였다. 그리고 평가자들은 각 인터페이스에서 지원되는 어노테이션 타입과 관련하여, 심볼 타입 및 자유형 마킹 타입을 모두 생성할 수 있는 SAi를 가장 선호하였다.

평가를 종합해볼 때 사용자들은 하이라이팅, 심볼, 그리고 자유형 타입을 선호하였으며, 마우스에서 사용되는 드래깅 인터페이스와 전자펜 입력 환경에서 높은 만족도를 나타내었다. 또한 인터페이스 설계와 관련하여 PAi의 장점인 간결하면서도 직관적인 어노테이션 입력환경이 사용성(usability)에 절대적인 영향을 미치는 것을 알 수 있었다.

본 연구에서는 평가에서 제시된 다양한 의견 및 개선사항을 바탕으로 데스크탑 및 펜 컴퓨팅 환경에서도 PAi의 장점을 최대한 지원할 수 있는 인터페이스를 제안하고자 한다. 이러한 본 연구의 세부 목표는 최종적으로 다음과 같은 내용을 포함한다.

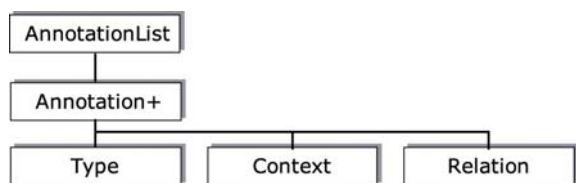
- 펜 및 마우스/키보드 환경에 적용 가능한 어노테이션 인터페이스 설계
- 정확하고 간결한 펜 입력 기능을 제공하기 위한

- 어노테이션 영역 인식 및 보정 인터랙션 제안
- 웹 표준문서인 XML을 이용한 어노테이션 정보 구조화 및 관련 기법 제안

3. 어노테이션 모델

어노테이션은 그 목적에 따라 서로 다른 형태로서의 정의 및 분류 될 수 있으며 또한 다양한 타입 및 의미(semantic), 구조 등을 포함할 수 있기 때문에 어노테이션 표준과 관련된 기존 연구가 부족하다 [17]. 따라서 XML과 같은 구조정보를 이용한 명확한 어노테이션 모델의 정의가 반드시 요구되며, 이러한 이유에서 컨텍스트에 기반한 어노테이션 모델 및 마크업 언어 CAML(Context-based Annotation Markup Language)를 제안한다. 본 모델에서는 다양한 어노테이션 타입, 의미정보, 인식 및 보정을 위한 자세한 정보를 포함하며 XML 문서를 적용 대상으로 한다. 구조기반의 어노테이션 정보는 차후 교환 및 재사용에 유리하다.

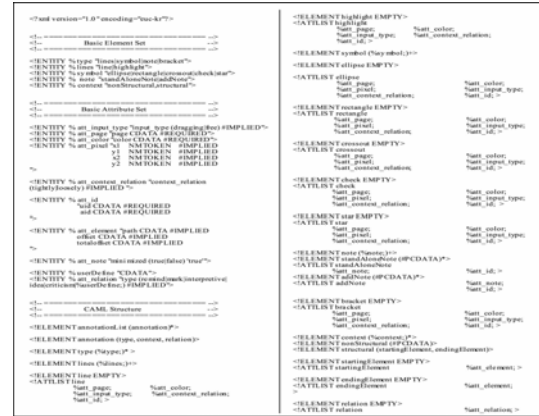
CAML은 원본문서를 대상으로 생성된 어노테이션에 대한 어노테이션 타입 및 Semantic 정보, 원본 문서에 대한 구조적, 물리적 속성 등을 표현하며, 어노테이션 출력시 CAML 문서와 원본문서를 매핑하여 이를 반영한다. CAML의 전체구조는 다음 그림 2와 같으며, 루트 엘리먼트인 어노테이션 List 하부에 복수의 어노테이션이 반복 가능하며 각 어노테이션에는 Type, Context, Relation들이 존재한다.



(그림 2) CAML의 전체구조

어노테이션 타입은 컨텍스트 정보에 따라 강 연결(tightly) 혹은 약 연결(loosely coupled) 타입으로, 그리고 형태에 따라 닫힘 형(closed) 혹은 열린 형(open set)으로 정의된다. 어노테이션 컨텍스트는 비구조(nonstructural), 구조(structural)에 따라 구분되며, 비구조는 어노테이션의 대상이 되는 문서(앵커: anchor)의 내용을, 구조는 문의 구조 및 속성 정보를 포함한다. 어노테이션 Relation은 placeholder, readingAssistant, selfThinking 카타고리를 중심으로 하부에 각각의 어노테이션 의미들을

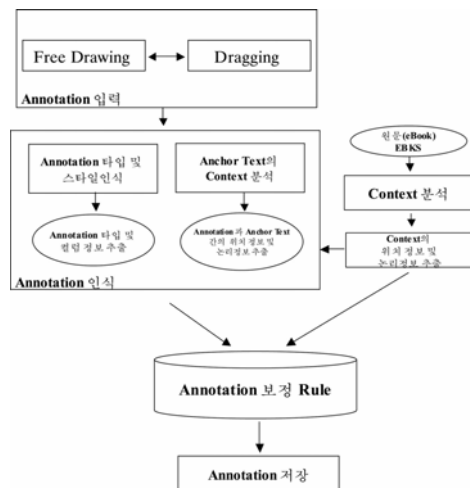
제공하고 있다. 이러한 모든 내용을 포함하는 CAML의 전체내용은 다음 그림3과 같다.



(그림 3) CAML 전체 내용

4. 시스템 설계

본 장에서는 지금까지 살펴본 어노테이션 모델을 이용하여 설계된 어노테이션 시스템에 대하여 살펴본다. 제안 시스템은 XML기반의 전자책[18]을 대상으로 하며, 다음 그림 4와 같이 어노테이션 입력, 출력, 저장, 그리고 인식 보정 과정을 수행한다.



(그림 4) 전체 시스템 및 인터페이스 구조

4.1 어노테이션 입력 인터페이스

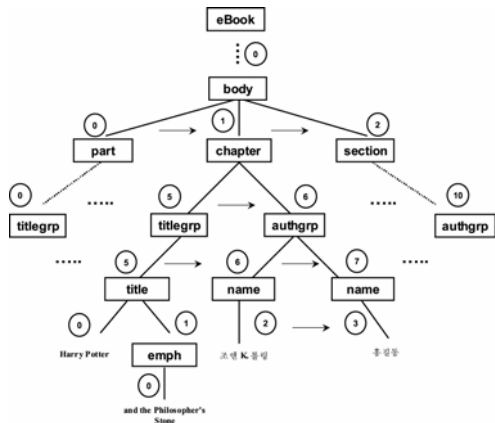
그림 4에서와 같이 어노테이션 생성은 펜 입력 환경에 기반하여 어노테이션을 스케치 할 수 있는 자유형 입력 방식과, 기존 마우스 기법을 이용한 드래깅 방식을 사용한다. 펜 기반의 자유형 [8, 12] 기법은 단말기와 같은 환경에 적절히 사용될 수 있

으며, 동시에 마우스 드래깅 기법을 통하여 정확한 어노테이션 영역 선택할 수 있다[8].

4.2 어노테이션 타입 및 영역 인식

어노테이션 인식 인터페이스는 생성된 어노테이션의 타입 인식과 원본 문서에 대한 논리정보 인식 단계로 구성된다.

어노테이션 타입 인식은 펜 디바이스로부터 드로잉(drawing)된 픽셀정보를 분석[19, 20]하여 본 연구에서 정의한 어노테이션스타일 타입으로 변환한다. 어노테이션 논리정보 인식과정에서는 CAML에서 정의된 어노테이션 앵커 컨텍스트의 위치, 읍셋, 구조 관계를 추출하여 보정 인터페이스로 전달한다. 이를 위하여 Xlink[21] 및 Xpointer[22]와 유사한 형태의 정보를 포함하는 DOM[23] 기반의 구조문서에 대한 논리구조트리(logical tree)를 그림 5와 같이 생성하고 논리트리에서의 단말노드(원본문서의 텍스트 및 기타 미디어)에 대한 경로와 읍셋 정보를 모두 추출하여 별도로 저장하고 이를 보정 모듈로 전달하게 된다. 단말노드에 대한 위치정보는 논리구조 트리에서의 각 깊이(depth) 별 순번(sequential number), 단말노드까지의 경로(path), 단말노드 텍스트의 읍셋(offset)으로 표현되며 그 예는 다음 그림 6과 같다.



(그림 5) 논리구조 트리 생성 과정

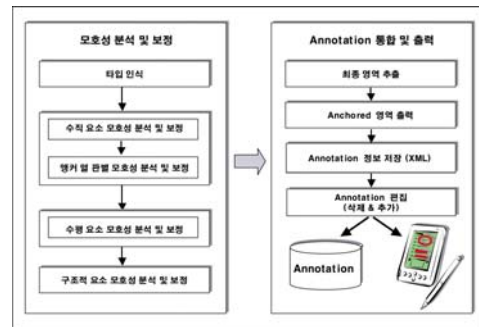
```
eBook/0 body/1 chapter/5 titlegrp/5 title/0 [text] harry porter/12/512
eBook/0 body/1 chapter/5 titlegrp /5 title/1 emph /0 [text] and the
Philosopher's Stone/28/530
eBook/0 body/1 chapter/6 authgrp/6 name/2 [text] 조앤 K. 롤링/8/538
eBook/0 body/1 chapter/6 authgrp/7 name/3 [text] 홍길동/3/541
```

(그림 6) 논리트리 생성에 따른 컨텍스트 정보 추출의 예

그림 6에서 왼쪽부터 오른쪽으로 진행되는 위치 정보는 각 노드별 레이블, 깊이에서의 순번, 그리고 “[]”로 표현되는 단말노드의 미디어 타입, 마지막으로 단말노드 내용 및 읍셋 정보를 나타내고 있다. 또한 마지막 숫자는 전체 텍스트의 첫 텍스트로부터 해당 단말노드 텍스트까지의 총 읍셋 정보를 뜻하며 이는 어노테이션 출력시 원활한 처리를 위하여 사용되었다.

4.3 어노테이션 영역 보정

어노테이션 보정은 인식과정에서 추출된 정보를 이용하는 다음 그림 7과 같은 규칙 모듈을 통하여 수행되며 그 내용은 본 연구에서 제안한 어노테이션 모델에 근거한다.



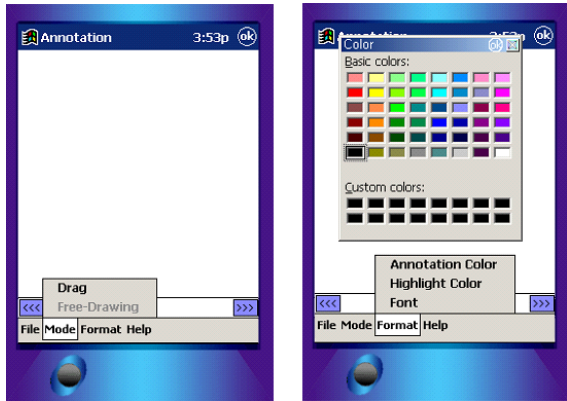
(그림 7) 어노테이션 보정 규칙

그림 7과 같이 보정 가장 먼저 ‘타입 인식 규칙’을 통하여 어노테이션의 타입 분석 결과에 따라 적용할 룰을 선택하게 되며, 각각 순서대로 어노테이션과 컨텍스트 행간의 점점여부 판별, 행의 개수 및 이에 근거한 단/복수 행의 결정, 앵커 영역 결정, 구조에 따른 영역 보정, 컨텍스트 정보에 따른 영역 재보정 과정 등을 수행하게 된다. 이러한 보정과정을 모두 수행한 후에는 최종적으로 어노테이션 출력 및 저장 과정을 수행하게 된다.

5. 구현 결과

본 장에서는 제안 시스템의 구현 결과를 설명한다. 제안 시스템은 Windows XP 및 Windows CE 기반의 브라우저에서 실행되며, XML 기반의 전자책 문서 표준을 원본 문서로 사용하였다. 본 절에서는 이중 PDA 환경에서 구동되는 어노테이션 시스템 및 이에 포함된 제안 인터페이스의 내용을 설명한다. 전체 인터페이스의 화면은 다음 그림 8과 같

다. 제안 시스템에서는 2가지 입력 모드를 선택하기 위한 기본 메뉴와, 부가적인 옵션을 선택하기 위한 보조 메뉴를 포함한다.

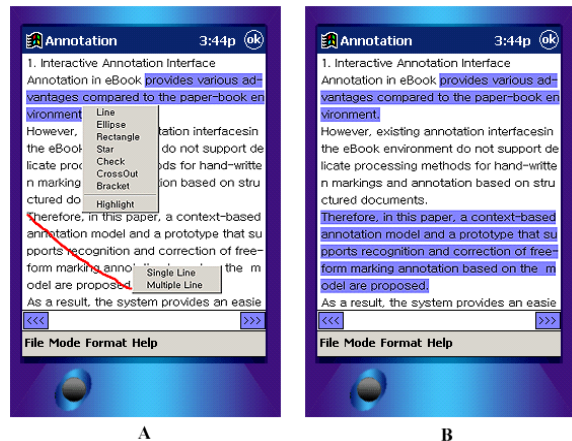


(그림 8) 제안 시스템의 입력 인터페이스

한편 제안된 인터페이스의 모든 어노테이션 타입들은 각각의 입력 방식에 따라 서로 다른 입력 및 처리 과정을 수행하며 이에 대한 자세한 설명은 다음과 같다.

5.1 라인

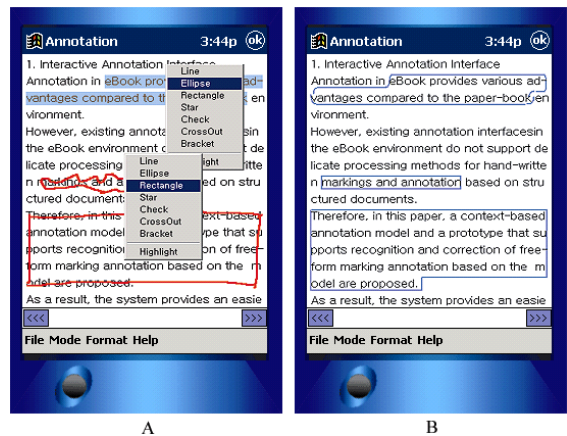
드래킹에 의한 라인 어노테이션 입력은 1차적으로 그림 9.A의 첫번째 예제와 같이 영역 선택 및 어노테이션 타입 선택과정으로 구분된다. 제안 인터페이스는 마우스 사용에 익숙한 사용자나 초보자에게 정확한 어노테이션 영역 및 타입 선택을 가능케 한다. 또한 내부적으로도 드래킹 기법은 앵커의 시작 및 끝 지점과 타입 등에 대한 정보를 정확히 추출할 수 있기 때문에 비교적 구현과정이 용이하다. 한편 라인이외의 다른 어노테이션 타입들도 대부분 위와 같은 형태의 드래킹 인터페이스를 포함한다. 그림 9.A 두번째 예제와 같이 자유형 마킹에 의하여 생성된 라인 어노테이션은 다른 어노테이션 타입에 비하여 비교적 명확하지 않은 위치정보를 포함한다. 따라서 라인을 입력할 경우에만 그림 9.A와 같이 영역의 단/복수 정보를 메뉴를 통하여 사용자가 입력하게 하여, 생성 라인의 열(row) 정보에 대한 인식 오류를 최소화하고자 하였다. 또한 마킹에 의한 복수라인의 생성은 드래킹에 비하여 어렵기 때문에 제안 인터페이스는 간략한 제스처(guesture)를 제공하며 그림 9.B는 이러한 드래킹 및 마킹에 의한 라인 생성 결과를 나타내고 있다.



(그림 9) 라인 타입의 입력 및 출력 예

5.2 타원 및 사각 형태의 어노테이션

타원(ellipse)과 직사각형(rectangle) 타입의 인터페이스는 라인과 마찬가지로 2가지 방식으로 입력되며, 그 예는 다음 그림 10.A와 같다. 또한 그림 10.A 및 그림 10.B의 첫번째 예제와 같이 제안 드래킹 인터페이스에서는 기존과는 달리 다양한 형태의 심볼 타입을 생성할 수 있으며, 그림 10.A의 마지막 예제와 같이 자유형 마킹 영역 및 타입을 자동으로 인식 및 보정하여 그림 10.B와 같이 출력하게 된다.



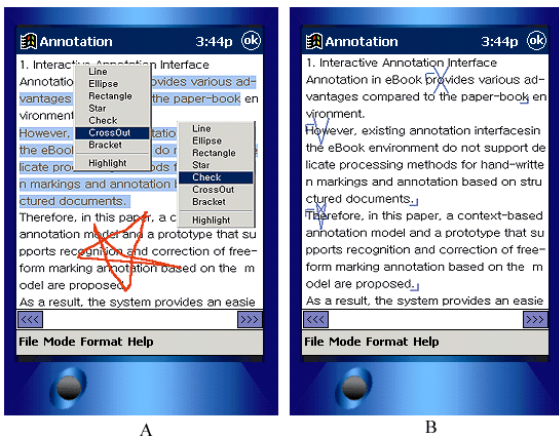
(그림 10) 타원 및 사각 타입의 입력 및 출력 예

한편 입력된 어노테이션 타입의 형태가 제안된 어노테이션 타입에 일치하지 않아 인식 모듈에서의 타입 매핑이 불가능할 경우, 제안 시스템에서는 사용자에게 자신이 의도한 어노테이션 타입을 메뉴를 통하여 직접 선택하도록 하며 그 예는 다음 그림 10.A 및 그림 10.B의 두 번째 예제와 같다. 그 이유는 부정확한 어노테이션 타입 매핑 결과는 사용자에게 오히려 어노테이션 삭제 및 재입력을 위한 테

스크롤 유발하기 때문에 이를 최소화하기 위함이다.

5.3 크로스아웃, 체크 및 스타

Star 및 check 등과 같은 심볼 어노테이션 타입은 일반적으로 문서의 상단 혹은 마진(margin)에 입력한다. 그러나 다수의 사용자들이 원문 내부에 다양한 형태의 심볼을 입력하거나 마킹하는 경우를 사용성 평가 과정에서 빈번하게 살펴볼 수 있었다. 한편 이러한 인라인 형태의 심볼 출력 영역은 라인, 타원 등과 같은 도형과는 달리 어노테이션 대상의 영역이 정확하게 일치하지 않기 때문에 그 영역을 별도의 격자로 표기하여 문서의 가독성을 저하시키지 않도록 하였다. 다음 그림 11.A에서는 이러한 crossout, check, star에 대한 드래깅 및 자유형 입력 방법을 나타내고 있으며, 그림 11.B에서는 입력에 대한 최종 출력 결과를 나타내고 있다.

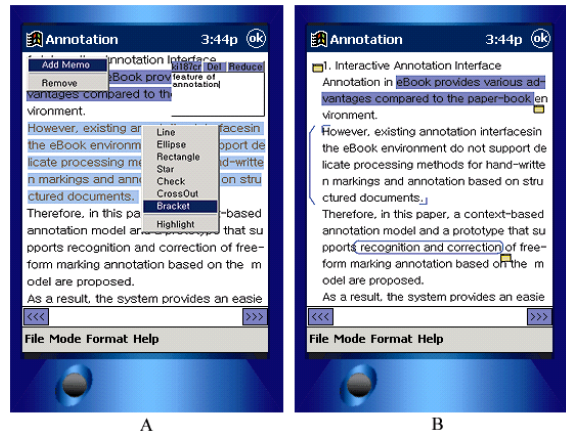


(그림 11) Cross, check, star의 입력 및 출력 예

5.4 노트 및 브라켓

제안 인터페이스의 노트(note) 타입은 이미 생성된 어노테이션에 추가적으로 입력하는 'add note'와 문서 상하좌우 마진에 단독으로 입력하는 'stand-alone note'로 구분된다. Add note는 다음 그림 12.A의 첫번째 예와 같이 2가지 입력 모드에 관계없이 노트이외의 어노테이션 타입을 생성한 후에 별도로 추가(add)되며, 이는 컨텍스트와 노트와의 연관성을 명확히 하기 위함이다. 또한 'stand-alone' 노트의 경우는 그림 12.A의 왼쪽 상단의 예와 같이 컨텍스트 영역을 제외한 상하좌우 마진 어느 곳이나 입력 가능하지만, 최종적인 출력을 위해서는 열(row)을 기준으로 위치를 보정한다. 또한 브라켓(bracket) (그림 12.A 및 그림 12.B의 두 번째 예)은 문서의 좌우 여백에만 입력 가능하며 노트와

는 달리 드래깅 또는 마킹 기법으로 입력 가능하다.



(그림 12) 노트 및 브라켓의 입력 및 출력 예

5.5 어노테이션 저장

제안 시스템에서는 생성된 어노테이션 정보를 XML에 기반한 CAML로 저장한다. 이를 위하여 제안시스템의 통합 모듈에서는 원문의 컨텍스트 정보와 규칙모듈에서 최종 보정된 어노테이션 컨텍스트 정보를 통합하고 그 결과를 원문과 별도의 형태의 CAML 정보로 저장한다. 이러한 어노테이션 정보의 구조적 표현의 예는 다음 그림 13과 같으며, 그림 12에서 생성된 드래깅 기반의 하이라이트 타입 및 노트 정보를 포함하고 있다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<!DOCTYPE report SYSTEM "scaml.dtd">
<annotationList>
<annotation>
<type>
<lines>
<highlight color="16744576" page="1" input_type="dragging" page="1" x1="10000" y1="10000" x2="0" y2="0" uid="k1187" aid="533422"/>
</lines>
</type>
<context>
<nonStructural>
eBook provide various advantages compared to the paper-book
</nonStructural>
<structural id="1">
<startingElement offset="15" path="/body[2]/chapter[0]/p[2]/#text[0]" totalOffset="52"/>
<endingElement offset="76" path="/body[2]/chapter[0]/p[2]/#text[0]" totalOffset="113"/>
</structural>
</context>
<relation type="idea"/>
</annotation>
<annotation>
<type>
note
<addNote color="16744576" page="1" id="user" aid="ref533422">
feature of annotation
</addNote>
</type>
</annotationList>
```

(그림 13) 어노테이션에 대한 CAML 저장 예

6. 결론

본 연구에서는 휴대용 단말기에서 정확한 어노테이션을 입력하고 생성하기 위한 기법을 제안하였다. 제안 기법에서는 컨텍스트를 고려하여 어노테이션 모델과 마크업 언어(CAML)를 정의하고, 펜 기반의 어노테이션 영역을 자동으로 보정하여 사용성을

극대화하였다. 또한 생성된 어노테이션 정보는 XML 정보로 표현되어 다중사용자간에 정확히 공유될 수 있다.

그 결과 전자펜에 익숙하지 않은 사용자들도 휴대용 기기에서 보다 적은 노력으로 정확한 어노테이션을 입력할 수 있다.

제안 기법은 휴대용 기기를 이용한 , 등에 적용 가능하다. 따라서 전자책 단말기, 이동 단말기, 태블릿 PC 등을 이용하는 유비쿼터스 환경의 eLearning, IETM(Interactive Electronic Technical Manuals), 노트 시스템(note taking system) 및 협업시스템에 효과적으로 적용 가능하다.

참고문헌

- [1] Catherine C. Marshall.(1997). Annotation: From Paper Books to Digital Library, Proc. of the 2nd ACM International Conference on Digital Libraries, ACM, Philadelphia.
- [2] Iliia A. Ovsiannikov.(1999). Annotation Technology, Int. J. Human-Computer Studies V.50, N.4. 329-362.
- [3] Catherine C. Marshall.(1998). Toward an Ecology of Hypertext Annotation, Proc. of HyperText, ACM, Pittsburgh.
- [4] Jacques A., Helene R.,(1999). Paper-less editing and proofreading of electronic documents," Proc. of EuroTex'99.
- [5] Baldonado, M., Cousins S., Gwizdka, J., Paepcke.(2000). Notable: At the intersection of annotations and handheld technology, Springer Verlag. Lecture Notes in Computer Science 1927. 100-113.
- [6] Price, M., Schilit, B.,(1998). XLibris: the active reading machine, CHI98, ACM, LA.
- [7] Tablet PC.(2006). <http://www.microsoft.com/tablepc>
- [8] Takayuki, K.,(1993). Potentials and Limitations of Pen-Based Computers, Proceedings of the 1993 ACM conference on Computer science, ACM.
- [9] Laurent D., Laurence V.,(2000). An annotation tool for web browsers and its applications to information retrieval, Proc. of RIAO2000, Paris April
- [10] Venu V., Mark P.,(1999). On Web annotations: promises and pitfalls of current Web infrastructure, Proc. of the Thirty-second Annual HICS, IEEE.
- [11] Stanford Digital Library Project.(1999)., <http://hci.stanford.edu/commentor>.
- [12] Davis J, Huttenlocher D.,(1994). <http://dri.cornell.edu/pub/davis/Annotation/annotation.html>, CoNote(Annotation)
- [13] iMarkup.(2001). <http://www.immarkup.com>.
- [14] EQuill Visual Teaming, EQuill Copr,(2001). <http://www.e-quill.com>.
- [15] MS Reder, Microsoft.(2001). <http://www.microsoft.com/reader>.
- [16] Heuristic Evaluation.(2006). <http://www.useit.com/papers/heuristic>.
- [17] Nichols D.M., Pemberton D., etl. al.,(2000). DEBORA: Developing an Interface to Support Collaboration in a Digital Library, Proc ECDL 2000, Lisbon Portugal, 239-248.
- [18] Won-Sung Sohn, etl. al.,(2002). Standardization of eBook documents in the Korean Industry, Computer Standards & Interfaces, Vol. 24, issue 1.
- [19] Krzyzak, Adam, W. Dai, Ching Y. Suen.(1991). On the Recognition of Handwritten Characters Using Neural Networks, Pattern Recognition Architectures, Algorithms & Applications, World Scientific, Singapore, 115-135.
- [20] Suen, Ching Y., etl. al.,(1992). Computer Recognition of Unconstrained Handwritten Numerals, Proc. of the IEEE, Vol. 80, Issue 7, 1162-1180.
- [21] XML Linking Language (XLink) Version 1.0, W3C Recommendation 27 June 2001, <http://www.w3.org/TR/2000/REC-xlink-20010627>
- [22] XML Pointer Language (XPointer) Version 1.0, W3C Candidate Recommendation 11 September 2001, <http://www.w3.org/TR/2001/CR-xptr-20010911/>
- [23] Document Object Model (DOM), <http://www.w3c.org/TR/1998/REC-DOM-19981001>, World Wide Web Consortium (1998)