

펜 입력 장치 환경을 고려한 컨텍스트 기반 Annotation

(Context-based Annotation for Pen Input Device Environment)

김재경 † 손원성 † 임순범 †† 최윤철 †††
 (Jae-Kyung Kim) (Won-Sung Sohn) (Soon-Bum Lim) (Yoon-Chul Choy)

요약 Annotation은 개인의 의견, 해설, 요약 등을 표기한 것으로서, 현재 전자문서 환경에서 annotation을 효과적으로 처리할 수 있는 많은 연구가 이루어지고 있다. 그러나 기존 연구에서는 annotation의 기능성에 치중하여 직관적인 펜기반 입력 인터페이스를 제공하지 못하거나, 이를 지원하는 일부 시스템에서도 원본 문서와 annotation의 관계가 명확하지 못하여 재활용성이 떨어지는 문제점이 있다. 이에 본 연구에서는 전자문서에서 펜 입력 장치 환경을 위한 컨텍스트 기반 annotation 모델링을 정의하고, 이를 활용한 인터페이스를 제안한다. 이를 위하여, annotation 유형, 문서의 컨텍스트 정보, 그리고 annotation과 원본 문서간의 관계를 정의하고, 모델링 정보에 기반한 펜기반 annotation 입력 인터페이스와 annotation DTD를 정의하여 다양한 활용이 가능한 시스템을 구현하였다. 그 결과, 본 연구에서는 기존 연구와는 달리 펜기반 입력 환경에서 컨텍스트 기반 annotation의 정확한 정의가 가능하며 이를 활용한 다양한 기능을 제공하는 동시에 앞으로의 응용 가능성을 제시하고 있다.

키워드 : 주석, 컨텍스트 기반, 모델링, 펜기반, 인식, 보정

Abstract Annotation is used for inscribing personal opinion, explanation, and summary. Various methods for processing annotation efficiently in digital document environments are being studied. However, previous studies placed much emphasis on function of annotation, so either they did not support intuitive paper-based input interface or the systems that support it still have low reusability problems, because relation between annotation and original document are not explicit. Thus, in our study, we define context-based annotation modeling for digital document environments, and suggest annotation interface based on the modeling. To design annotation model, we define annotation types, context information of document, and relationship between annotation and original document. Also, a system based on the modeling is implemented to support pen-based annotation and annotation DTD. As a result, unlike previous studies, it is possible to explicitly define context-based annotation in pen-based input environments. We present various functions using the modeling and various possibilities of application.

Key words : Annotation, Context-based, Modeling, Pen-based, Recognition, Correction

· 이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2002-013-E00062).

† 학생회원 : 연세대학교 컴퓨터학과
ki187cm@rainbow.yonsei.ac.kr

† 학생회원 : 연세대학교 컴퓨터학과
sohnws@rainbow.yonsei.ac.kr

†† 중신회원 : 숙명여자대학교 멀티미디어학과 교수
sblim@sookmyung.ac.kr

††† 중신회원 : 연세대학교 컴퓨터학과 교수
ycchoy@rainbow.yonsei.ac.kr

논문접수 : 2002년 4월 18일

심사완료 : 2003년 3월 11일

1. 서론

컴퓨터 기술의 발달로 인해 기존의 문서, 책, 잡지 등의 출판물들이 점차 전자 문서화되고 있으며 실제로도 이에 대한 다양한 연구 및 서비스가 진행되고 있다. 일반적으로 종이 문서는 전자 문서에 비해 소수의 사용자 간의 정보 교환용으로 사용되는 경향이 있으나, 전자 문서는 디지털 환경이라는 특성으로 다수의 사용자 간에 공유가 가능하다. 또한 사용자들은 전자 문서 자체의 정보를 습득하는데 그치지 않고 사용자들 간의 의견 및

추가적인 정보를 쉽게 공유 및 교환하는 것이 가능하다. 이러한 것을 가능케 하여주는 것이 annotation인데, 이것은 일반적으로 문서의 주제 및 내용에 관한 해설, 설명, 그리고 강조를 목적으로 추가되는 문장 또는 텍스트를 의미한다[1]. 또한 문서 내용에 대한 정리, 요약, 이해, 기록 등을 위하여 사용되며, 일반적으로 밑줄(Line), 도형(Symbol), 쪽지(Note) 등으로 표현된다[2][3]. 그 결과 annotation 정보는 일회성 정보가 아닌, 재사용 및 공유될 수 있는 중요한 정보라 할 수 있으며, 특히 전자 문서 환경(웹 환경)에서 더욱 그 필요성이 요구된다[2]. 그 이유는 전자문서에서의 annotation은 종이문서의 경우와는 달리 문서 내용에 대한 링크[4], 설명, 해석, 강조, 질문 기능은 물론이고, 입력된 annotation에 대한 재사용, 검색[5], 공유[6] 등의 기능을 제공할 수 있기 때문이다.

디지털 환경에서의 annotation은 전자 문서를 종이문서와 차별화 할 수 있는 가장 큰 특징으로 볼 수 있다. 그러나 기존의 전자 문서 및 관련 연구에서는 annotation을 단순한 일회성 정보로 저장하고 처리하고 있어서 생성된 annotation의 다양한 활용이 불가능하였다. 더욱이 가장 큰 단점은 annotation에 대한 정확한 정의 및 활용을 위한 모델제시가 결여된 단순한 시스템이 제공된다는 것이다.

기존의 annotation 시스템에서는 텍스트 및 스타일 기반의 annotation을 지원하는 것과 종이 문서에 펜으로 그림을 그리듯이 화면에 구애받지 않고 자유로운 입력을 지원하는 펜기반[7] annotation 입력 시스템으로 구별되고 있다. 현재 대부분의 시스템들이 텍스트와 스타일 기반의 인터페이스를 지원하고 있고, 최근 핸드 헬드 기기와 같은 모바일 환경이 발달하면서 펜기반 인터페이스를 지원하는 시스템들이 증가하는 추세이다[8][9].

그러나 텍스트 및 스타일 환경의 시스템은 사용자가 펜기반 환경에서 사용하는 다양한 annotation을 입력할 수 없는 단점이 있으며, 펜기반 환경의 시스템의 경우, 사용자가 입력한 그래픽을 원본문서의 텍스트를 고려하지 않고, 위치 정보만을 저장, 처리하고 있어서 생성된 그래픽 annotation의 다양한 활용이 불가능하였다. 또한 표준적인 형식을 지원하고 있지 않고 대부분 자체 형식만을 지원하고 있다.

이에 기존 시스템의 문제점을 해결하고자 annotation을 효과적으로 입력 및 활용할 수 있는 시스템을 위해 컨텍스트 기반의 annotation 모델링을 제시한다. 이를 위하여 본 연구에서는 펜기반 환경에서의 annotation 모델링을 제시하고 이에 대한 활용 및 가능성을 제시하

고자 한다. 또한 이를 응용한 펜기반 annotation 입력 인터페이스를 구현하고, 기존 연구의 단점을 극복하기 위하여 구조정보 및 콘텐츠와 annotation간의 정확한 관계를 포함한 정의를 통하여, 향후 이를 활용한 다양한 가능성을 제시한다.

본문의 구성은 다음과 같다. 2장에서는 전자 문서 환경에서 annotation과 관련된 선행 연구에 대하여 알아 본다. 3장에서는 컨텍스트 기반의 annotation 모델링을 제시하고 4장에서는 제안한 모델링을 응용한 펜기반 annotation 인터페이스 및 DTD를 설명한다. 마지막으로 5장에서 결론 및 향후 연구 방향에 대하여 기술한다.

2. 관련 연구

Annotation과 관련된 기존 연구는 annotation 표현 방식에 따라 텍스트 및 스타일, 그리고 펜기반 annotation 시스템으로 나눌 수 있는데 먼저 텍스트 및 스타일 기반의 시스템에 대해 살펴본다.

이것은 초기 annotation 환경에서 많이 사용한 기법 [10]이다. 원본 문서의 텍스트에 작은 도형 등을 삽입하고 이곳에 링크를 연결하여 사용자의 의견 등을 추가하는 방식으로 기능이 간단하고 구현이 쉽다. 대표적인 예로 CritSuite[11]가 있다. [그림 1]은 CritSuite의 초기화면이며 사용자가 annotation을 입력한 곳에 작은 도형이 삽입되어 있는 것을 볼 수 있다. 이 방식은 구현이 매우 쉽고 사용하기가 쉽다는 장점이 있지만 펜기반 환경에서 쓰이는 도형과 같은 다양한 종류의 annotation 표현이 불가능하다.

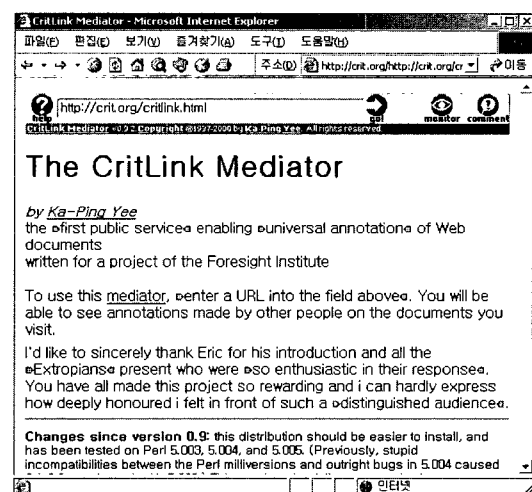


그림 1 텍스트 기반의 annotation 시스템인 CritSuite

다음으로 스타일 기반 annotation 기법[12]이 있는데 이것은 텍스트 기반의 시스템과 유사하나 메뉴를 이용한 밑줄, 형광펜(Highlight) 등의 기능이 지원된다. 텍스트 기반 기법에 비해 펜기반 환경과 유사한 인터페이스를 제공하며 원본 문서와 annotation간의 구조적인 연결도 고려하고 있다. 이러한 인터페이스를 가진 것으로 YAWAS(Yet Another Web Annotation System)[13]라는 시스템이 있다.

이 시스템은 [그림 2]와 같이 원본 문서에서 annotation 입력을 원하는 텍스트에서 메뉴를 생성하고 새로운 팝업창에서 annotation을 생성하는 인터페이스를 가지고 있다. 텍스트 기반 기법에 비하여 좀더 다양한 스타일의 annotation 표현이 가능하고 원하는 원본 문서의 텍스트 위치에서 바로 입력이 가능하기 때문에 직관적이라 할 수 있겠다. 또한 구조적 문서로 annotation을 저장하기 때문에 원본 문서와 annotation간에 구조적으로 명확한 관계가 성립되며 또한 구조 정보를 이용한 효율적인 검색 및 저장이 가능하다.

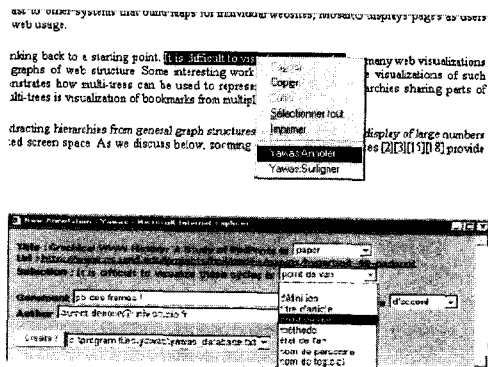


그림 2 YAWAS의 스타일 기반 입력 인터페이스의 예

그러나 이러한 스타일 기반의 시스템은 도형과 같이 그래픽에 기반한 인터페이스를 지원하지 않기 때문에 펜기반 환경에서 사용자가 생성하는 다양한 annotation 스타일을 지원하지는 데 한계가 있다.

마지막으로 펜기반 annotation 시스템에 대해서 알아보도록 한다. 이것은 사용자가 포토샵과 같은 그래픽 툴을 이용하듯이 간단한 선이나 도형등을 자유롭게 입력할 수 있는 인터페이스[14]를 가지고 있다. 펜기반 시스템은 스타일 기반의 그것과 비교하여 다양한 annotation 종류를 입력할 수 있는 장점이 있으며 대부분 텍스트 및 스타일 입력 인터페이스 뿐 아니라 그래픽 입력도 함께 가지고 있기 때문에 가장 사용자의 입력 편의성이

뛰어나다고 할 수 있다.

이러한 펜기반의 인터페이스를 가지고 있는 시스템에는 MS Reader, iMarkup, AudioGraph, Forward Anywhere, Classroom 2000등이 있다. [그림 3]은 펜기반 annotation입력을 지원하는 Classroom 2000[14]이다. 그림에서 보듯이 펜을 이용한 자유형식 그리기(Free-Form Drawing) 입력 방식으로 다양한 형태의 annotation 입력이 가능하다.

그러나 입력된 그래픽 annotation들은 단순히 화면상의 좌표 정보만을 가지고 처리되어 저장 및 출력을 하고 있다. 따라서 annotation과 원본 문서의 텍스트와의 연결이 이루어지지 않아 어느 텍스트에 annotation이 입력되었는지 파악하기 힘들다. 또한 사용자가 입력한 그래픽 annotation을 밑줄, 사각형, 및 그 외 도형 유형으로 분류하지 않고 단순히 동일한 종류의 그래픽 이미지로 인식한다. 따라서 사용자는 다양한 종류의 annotation을 입력하였으나 실제 시스템에서는 이것을 모두 동일한 형태로 처리하기 때문에 이중 시스템과의 annotation 교환이나 추가적인 의미정보를 이용한 정보 검색등과 같은 재활용성이 떨어지는 것이다. 더욱이 대부분의 시스템들이 annotation 저장에 있어서 표준구조 형식을 사용하지 않고 자체 형식을 사용하기 때문에 이러한 재활용성은 더욱 저조해지는 단점이 있다.

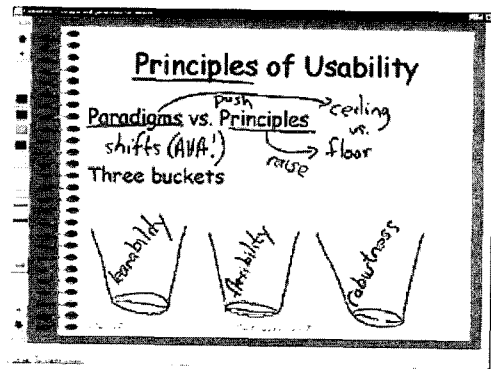


그림 3 Classroom 2000에서의 annotation 사용 예

따라서 펜기반 환경에서 쓰이는 그래픽 annotation을 지원하며, 동시에 입력시 이들의 종류를 인식하고 원본 문서와의 연결을 고려하는 컨텍스트 기반의 annotation 인터페이스가 요구되어진다. 뿐만 아니라 생성된 annotation들을 표준형식에 기반하여 저장함으로써 annotation 활용성을 높이도록 하는 것이 바람직하다. 여기서 컨텍스트라는 것은 annotation이 입력되어지는 원본 문서의 일부분

을 뜻하며 앵커 텍스트라고도 불리운다[15].

다음 표 1은 지금까지 설명한 기존 시스템들과 다음 장에서 설명할 컨텍스트 기반 시스템의 특성을 비교하여 표로 정리한 것이다. 컨텍스트 기반의 시스템은 직관성이 높은 기존 그래픽 기반의 인터페이스를 가져야 하기 때문에 텍스트나 스타일 기반의 시스템에 비하여 구현이 어려운 단점이 있다. 그러나 사용자에게 편의성을 제공하기 위하여 그래픽 기반의 시스템을 지원하는 것이 바람직하다. 또한 사용자가 입력하는 annotation을 인식하여 annotation을 유형별로 분류하고, 구조 정보를 이용한 원본 문서와 annotation간의 명확한 관계 정의 방법을 도입하여야 한다. 그리고 표준구조문서 형식을 사용하여 annotation의 표현하여 향후 저장, 검색 및 공유와 같은 활용성을 높이도록 해야 한다.

표 1 기존 및 컨텍스트 기반 annotation 시스템의 비교

	텍스트 기반	스타일 기반	그래픽 기반	컨텍스트 기반
구현의 용이성	아주 쉽다	쉽다	어렵다	어렵다
직관적인 입력 인터페이스	하	중	상	상
annotation 유형 인식	안함	부분적	안함	인식함
원본 문서와 annotation과의 관계	단순 텍스트 정보 이용	구조 정보 이용	화면 좌표 정보 이용	구조 정보 이용
표준구조문서 형식의 annotation 표현	미지원	지원	미지원	지원
annotation 활용성	낮다	보통	낮다	높다

다음 장에서는 이와 같은 기능들을 만족시킬 수 있도록 컨텍스트에 기반한 annotation 모델링을 정의하고 이에 기반한 시스템의 구현 및 활용에 대하여 알아보도록 한다.

3. 컨텍스트에 기반한 Annotation 모델링

앞서 설명하였듯이 컨텍스트라는 것은 annotation이 입력되어지는 원본 문서의 일부분을 뜻하며 컨텍스트 기반의 annotation 시스템에서는 이러한 컨텍스트의 크기, 위치, 그리고 구조정보를 이용한 annotation의 인식 및 활용이 필수적이라 하겠다. 따라서 본 연구에서는 이러한 기능을 모두 제공하기 위한 컨텍스트 기반 annotation 모델링을 제시하며 이에 대한 자세한 내용을 살펴보고

록 한다.

Annotation이 사용되는 환경에 따라서 그 정의는 달라질 수 있다. 텍스트 환경의 시스템에서는 텍스트 정보를 표현할 수 있는 annotation 정의가 필요할 것이며 스타일이나 그래픽 환경의 시스템에서는 텍스트 뿐 아니라 그래픽 정보도 표현할 수 있는 정의가 필요하다.

본 연구에서는 펜기반 환경에서 컨텍스트에 기반한 annotation 입력 및 인식, 그리고 스타일 보정을 위한 annotation을 정의하였다. 이를 위해 다음의 [정의 1]부터 [정의 4]를 통하여 펜기반 환경에서 쓰이는 그래픽 annotation 유형, 전자 문서에서 annotation의 대상이 되는 컨텍스트, annotation의 기본 요소 등을 정의하고 있다.

먼저 annotation 모델은 다음과 같이 2가지 기본 요소로 구성되며, 각 요소의 내용은 다음 [정의1]과 같다.

정의 1 : Annotation은 다음 식과 같이 기본 유형과 컨텍스트 정보로 정의된다.

$$A = \{T, C\}$$

여기서 T는 annotation을 구성하는 기본적인 유형을 의미하며 C는 annotation이 입력된 컨텍스트 정보, 즉 앵커 텍스트를 의미한다. 본 연구에서는 펜기반 환경에서의 annotation을 그 유형 및 컨텍스트 요소로 이루어진 집합으로 정의한다. 여기서 컨텍스트는 단어, 문장 및 절 등이 될 수 있다.

정의 2 : Annotation 유형은 스타일 유형에 따라 닫힌 유형(T_c), 열린 유형(T_o), 기호(Symbol), 쪽지(Note) 및 형광펜(Highlight) 유형으로 구성된다.

$$T = \{T_c, T_o, Symbol, Note, Highlight\}$$

$$T_c = \{Circle, Rectangle\}$$

$$T_o = \{Line, Bracket\}$$

$$Symbol = \{Check, Star, CrossOut\}$$

T_c , T_o 및 기호는 펜기반 환경에서 가장 많이 이용되는 유형들로써 사용자의 다양한 annotation 스타일 입력을 지원해주기 위해 반드시 정의되어야 한다. 닫힌 및 열린 유형은 주로 컨텍스트의 영역 표시를 목적으로 하는 annotation들이며, 닫힌 유형에는 타원(Circle)과 사각형(Rectangle)이 있으며 열린 유형에는 밑줄(Line)과 괄호(Bracket)가 있다. 기호는 중요 혹은 기억할 것 등과 같이 컨텍스트에 어떤 의미를 부여하는 annotation들이며 체크(Check), 별표(Star) 및 삭제(CrossOut)가 있다.

다음으로 쪽지 및 형광펜은 기존의 스타일이나 텍스트 기반의 시스템에서 주로 사용되는 것으로 부연 설명의 추가나 강조의 목적으로 사용되는 annotation들이다. 모든 annotation 시스템에서 필수적으로 필요한 기본적

인 annotation들이기 때문에 본 모델링 정의에 포함하였다.

정의 3 : Annotation 유형은 컨텍스트와의 관계에 따라 밀결합(tightly-coupled) 유형(T_t)과 약결합(loosely-coupled) 유형(T_l)으로 구분할 수 있다.

$$T = \{T_t, T_l\}$$

$$T_t \cap T_l = \emptyset$$

$$T_t = T_c \cup \{Line, Highlight\}$$

$$T_l = \{Bracket, Symbol, Note\}$$

밀결합 유형은 annotation 영역 및 스타일이 구조정보, 단어, 문장, 절 등에 따라 직접적으로 영향을 받는 경우이며, 약결합 유형은 그 영향의 정도가 덜한 경우를 의미한다. 이러한 밀결합 및 약결합 유형 구성요소와 [정의 2]와의 관계는 위 식과 같다.

정의 4 : 컨텍스트(C)는 annotation의 대상인 앵커 텍스트를 의미하며, 비구조적 컨텍스트(C_n)와 구조적 컨텍스트(C_s)로 구성된다. 비구조적 컨텍스트는 문서의 기본 구성요소인 단어, 구, 문장으로 구성되는데 이들은 알파벳들의 집합인 단어(W)의 반복된 집합으로 이루어진다. 구조적 컨텍스트는 비구조적 컨텍스트 요소와 시작(E_s) 및 끝(E_e) 엘리먼트를 기본으로 하는 구조정보 요소(S_i)로 이루어져 있다.

$$C = \{C_n, C_s\}$$

$$W = \{a, b, \dots, z, A, B, \dots, Z, 0, 1, 9\}$$

$$C_n = W^*$$

$$S_i = \{E_s, E_e\}$$

$$C_s = S_i \cup C_n$$

지금까지 펜기반 인터페이스 환경에서 컨텍스트에 기반한 annotation 모델링의 정의를 살펴보았다. 이와 같은 모델링을 통하여 annotation의 다양한 활용이 가능하며, 나아가 기존 그래픽 기반의 시스템이 가진 문제점들의 해결이 가능하다. 예를 들어 펜기반 환경에서의 일반적인 입력 방식인 자유형식 그리기로 입력된 annotation의 인식을 통한 유형 분류, annotation과 컨텍스트 사이의 구조적인 연결 정보를 이용한 annotation의 재사용성 증가, 컨텍스트 영역 정보를 이용한 그래픽 annotation 출력과 같은 것이 그것이다.

다음 장에서는 모델링을 적용한 실제 시스템의 구현을 통하여 활용 사례들을 알아보도록 한다.

4. 컨텍스트 기반의 Annotation 인터페이스

본 장에서는 컨텍스트에 기반한 annotation 인터페이스를 살펴보도록 한다. 4.1절에서는 입력 인터페이스에

대해 알아보고, 4.2절에서는 컨텍스트 정보를 이용한 입력된 annotation의 스타일 보정 및 출력, 4.3절에서는 annotation DTD 정의 및 저장에 대해서 논한다.

4.1 Annotation 입력 인터페이스

기존의 그래픽 annotation 시스템에서 자유형식 그리기 방식으로 입력된 annotation은 사용자의 입력의도를 파악할 수 없는 단점을 가지고 있다.

즉, 사용자가 중요한 컨텍스트에 별표를 그리거나 불필요한 곳에 가위표(Cross-Out)를 입력하였을 때, 이러한 annotation들은 단순한 이미지 형식으로 처리되어 사용자가 어떤 형태의 annotation을 입력하였는지 파악할 수 없다. 몇몇 시스템들에서 미리 정의된 도형이나 기호들을 메뉴방식으로 제공하고, 마우스 클릭과 드래깅으로 이러한 도형들을 끌어다 입력하는 방식으로 그래픽 기반의 annotation 유형을 구분하는 방법을 사용하기도 한다. 그러나 이것은 자유형식 그리기 방식이 아니기 때문에 직관적이 못한 입력이라는 단점을 가지고 있다.

따라서 이러한 단점을 극복하기 위하여 사용자가 펜기반 환경에서 주로 이용하는 annotation 유형[4]들을 자유형식 그리기로 입력하도록 하고 이들의 유형을 인식하는 입력 인터페이스를 전장의 annotation 모델링 정의에 따라 설계하였다.

전체적인 시스템의 인터페이스는 그림 4와 같다.

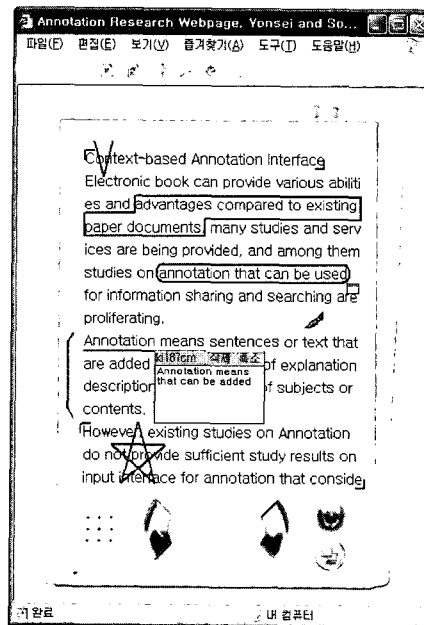


그림 4 컨텍스트 기반 Annotation 시스템의 인터페이스

향후 다중사용자 간의 협력 작업등을 고려하여 웹 환경을 기반으로 하고 있으며 일반 웹 브라우저에서 ActiveX 컨트롤로 포함되어 실행이 가능하다. 본 시스템은 annotation 입력을 위한 최소한의 기능을 제공하고 있으며, 하단의 버튼을 이용하여 페이지 넘기기, 색상 지정, 입력 모드 변환을 할 수 있다.

사용자는 이러한 최소한의 기능을 메뉴를 사용하여 실제 annotation의 입력 및 삭제는 오직 펜기반의 드로잉으로 수행한다. 이것은 펜입력의 특성을 최대한 활용하고 잦은 메뉴 선택으로 인한 입력시 불편함을 해소하기 위함이다[16].

사용자는 선, 도형, 기호등과 같은 다양한 annotation 유형을 자유형식 그리기 방식으로 마우스나 펜을 이용하여 메인화면의 컨텍스트에 입력한다. 입력된 annotation은 제스처 인식 알고리즘[17]에 의해 annotation의 종류를 인식하게 된다. 이들 종류는 [정의 2]에서 설명한 바와 같이 밑줄, 사각형, 타원, 별표, 가위표, 체크, 괄호, 쪽지, 형광펜의 총 9가지로 인식된다.

표 2 모델링에 따른 annotation의 명칭, 형태 및 기능










명칭	형태	기능	명칭	형태	기능
타원		영역선택	별표		매우중요
사각형		영역선택	가위표		강요요, 삭제
밑줄		중요부분 선택	쪽지		텍스트 추가
괄호		영역묶음	형광펜		강조
체크		상기, 기억			

표 2는 9가지 종류의 annotation을 표로 명칭, 형태 및 기능을 쉽게 알아 볼 수 있도록 표로 정리한 것이다. 타원, 사각형, 밑줄, 괄호, 체크, 별표 및 가위표는 자유형식 그리기로 입력 가능한 유형들이며, 이들에 대한 각각의 기능들은 일반적으로 표 2와 같으나 이것은 사용자에 따라서 달라질 수 있다. 왜냐하면 본 연구에서는 각 annotation의 기능이나 의미는 전적으로 각 사용자에 의해 개별적으로 결정되고 이용되고 있기 때문이다. 향후 모델링에 원본 문서와 annotation간의 관계를 추가하고 사용자 프로파일등을 통하여 annotation의 명시적인 기능을 정의하거나 분류할 수 있는데 이것은 현재 진행중인 연구이므로 여기서는 더 이상 언급하지 않기로 한다.

나머지 쪽지와 형광펜은 스타일 기반 시스템에서 지원되는 것으로 형광펜의 경우 제스처 인식 과정을 거치지

않고 메뉴 선택 후 마우스 드래깅을 통하여 입력되고, 쪽지의 경우 키보드에 의한 문자 입력시에 사용된다. 문자 입력시 펜기반 입력을 쓰지 않는 이유는 아직까지 펜보다는 키보드가 문자 입력에 월등히 빠르고 효율적이 입증되어 있기 때문이다[18].

다음으로 인식된 annotation 종류에 따라 [정의 3]을 이용하여 annotation을 밀결합 및 약결합 유형으로 구분한다. 유형별로 구분된 annotation은 다음 절에서 설명할 유형별 보정 기준에 따라 스타일 보정이 적용되어 그림 4와 같이 컨텍스트 영역에 출력되게 된다.

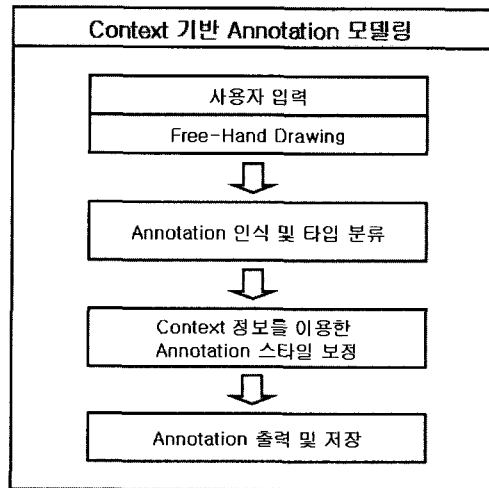


그림 5 Annotation 모델의 입력 및 출력 과정

Annotation 출력 후 마지막 과정으로 저장이 이루어지게 된다. Annotation은 [정의 1]에서 정의된 것처럼 annotation 유형과 컨텍스트간의 관계로 이루어져 있으며 annotation 저장 단계에서는 입력 단계에서 인식한 annotation 유형과 컨텍스트 정보간에 링크를 형성하여 저장한다. 본 연구에서는 annotation 및 annotation의 대상이 되는 원본문서에 XML을 사용하고 있으므로 효율적인 연결을 위해 이들 간에 구조적 정보를 이용한 연결을 생성한다. 그림 5는 이러한 입력 및 출력 과정을 그림으로 나타낸 것이다.

4.2 컨텍스트 정보를 이용한 Annotation 스타일 보정

기존의 그래픽 기반의 annotation 입력 인터페이스에서는 앞서 언급한 바와 같이 미리 정의되어 있는 이미지 형태의 도형이나 선들을 끌어다 놓고 사이즈 및 위치를 조절하는 방법을 이용하고 있다.

이와 같이 annotation을 단순한 이미지 형식으로 처

리하는 방법은 annotation과 원본 문서의 컨텍스트와 전혀 연결이 되지 않는 상황에서 이미지의 크기와 위치를 변경시키는 결과만을 가져온다.

본 연구에서는 이러한 방식을 벗어나 컨텍스트 정보에 따라 annotation의 위치, 크기 및 형태를 결정하는 기법을 제안한다. 이것은 화면상에서의 annotation 표시 영역과 실제 사용자가 원하는 컨텍스트 영역을 최대한 일치시킴으로써, annotation 컨텍스트 영역을 쉽게 파악할 수 있으며, 원본 문서의 수정이나 화면의 크기 변화에도 annotation이 원본 문서의 컨텍스트에 일치되도록 표시되므로 문서 변화에 크게 영향을 받지 않는 장점이 있다. 이러한 기법을 적용하기 위해 본연구에서는 컨텍스트에 기반한 보정 기법을 제안한다.

Annotation 보정(A_c)은 크게 영역 보정과 스타일 보정으로 구분된다. 영역 보정은 사용자가 입력을 원하는 목적 컨텍스트 영역을 컨텍스트 정보를 이용하여 보정하는 것이다.

종이 문서 환경에서의 펜을 이용한 입력 방식은 사용자에게 가장 친숙한 입력 방식이기 때문에 비교적 정확한 위치에 입력이 가능하다. 그러나 전자 문서 환경에서 펜 혹은 마우스를 이용한 입력 환경에서는 친숙하지 못한 입력기구나 방식으로 종이 문서에 비해 입력시 정확도가 떨어질 수 있다. 따라서 사용자가 입력을 원하는 목적 컨텍스트 영역을 효율적으로 파악하여 영역을 보정해주는 것이 필요하다. 이를 위해 본 연구에서는 비구조적 및 구조적 컨텍스트 정보를 이용한 보정 기법을 제안한다.

비구조적 컨텍스트 정보를 이용한 영역 보정의 경우, [정의 4]에서 정의된 워드를 기본 단위로 하여 영역 보정을 적용한다. 사용자가 자유형식 그리기 방식으로 입력한 annotation의 시작 및 끝부분이 워드에 걸쳐져 입력되었을 경우 해당 워드에 입력된 annotation 길이(AL_i , i 는 시작 및 끝부분)가 워드 길이(WL_i , i 는 시작 및 끝부분)의 절반이상을 차지한 경우 해당 워드를 유효한 목적 컨텍스트(VTC_n)라 판단한다. 이를 수식으로 표현하면 다음 식(1)과 같다.

$$VTC_n = \frac{AL_i}{WL_i} \geq 0.5 \quad (1)$$

본 연구에서 사용된 원본 문서는 XML에 기반하고 있으므로 다양한 구조정보들이 존재한다. 따라서 비구조적 정보 뿐 아니라 구조정보들을 이용하여 좀더 효율적인 보정을 적용할 수 있다. 일반적으로 구조적 문서들은 제목, 단락, 이메일 주소등과 같은 의미적인 내용들을 구조적으로 구분하고 있다. 그러므로 어떤 경우는 의미적으로 구분되는 컨텍스트 영역이 annotation 입력의

대상이 될 것이다. 사용자가 구조정보로 구분되는 컨텍스트에 annotation 입력을 시도한다고 판단되면 해당 구조 정보를 가지는 컨텍스트 영역에 보정을 시도한다.

구조적 컨텍스트 정보를 이용한 영역 보정을 위해, 먼저 [정의 4]에서 살펴본 구조적 컨텍스트 정의에 따라 원본 문서내의 시작 및 끝 엘리먼트로 구분되는 컨텍스트 영역을 파악한다. 사용자의 입력이 들어오면 입력된 annotation 길이(AL)와 주변의 구조정보 영역의 길이(SIL)의 차가 30%이내일 때 해당 구조정보 영역을 유효한 목적 컨텍스트 영역(VTC_s)으로 결정한다. 이는 다음 식(2)와 같이 표현되어진다.

$$VTC_s = |1 - \frac{AL}{SIL}| < 0.3 \quad (2)$$

그림 6은 실제 사용자의 입력을 받아 컨텍스트 정보를 이용한 영역 보정 적용의 과정 및 결과의 예를 보여준다. (a)에서 사용자가 'Annotation'이란 단어 근처에 밑줄을 자유형식 그리기 방식으로 입력한 경우에는 (c)에서 'Annotation'이 'keyword'라는 구조정보를 가지고 있으므로 식(2)의 보정 기준에 따라 (b)와 같이 'Annotation' 컨텍스트 영역에 영역 보정을 적용하였다. 또한 'the purpose of explanation' 영역에 입력된 annotation은 컨텍스트가 식(2)의 보정 기준에 속하지 않으므로 식(1)의 비구조적 영역 보정기준을 적용하여 비구조적 컨텍스트인 단어 단위로 (b)와 같은 영역 보정 결과를 얻었다.

proliferating.
Annotation means sentences or text that are added for the purpose of explanation description and emphasis of subjects or

(a) 사용자 annotation 입력

proliferating.
Annotation means sentences or text that are added for the purpose of explanation description and emphasis of subjects or

(b) 영역 보정 후 결과

and searching areproliferating. </p>
 <p><keyword>Annotation</keyword> means sentences or text that are added for the purpose of explanation description and emphasis of subjects or contents. </p>

(c) 실제 XML 원본 문서

그림 6 컨텍스트 정보를 이용한 영역 보정

다음으로 annotation 스타일 보정(A_c)은 annotation의 유형이 밀결합 혹은 약결합 유형인지에 따라 식(3)과 같이 밀결합 보정(TA_c) 혹은 약결합 보정(LA_c)으로 구분된다.

$$A_c = (TA_c, LA_c) \quad (3)$$

$$f : T_i(t, b, r, l) \rightarrow TA_c \quad (4)$$

$$\sigma : T_i(t, b) \rightarrow LA_c \quad (5)$$

밀결합 유형(T_i)에 속하는 annotation들의 모양은 컨텍스트 영역에 밀접하게 영향을 받는다. 즉, 식(4)와 같이 컨텍스트의 상하좌우 영역의 정보를 이용하여 밀결합 유형의 annotation의 크기와 위치가 일치하게 annotation을 출력하는 것이다.

다음으로 약결합 유형(T_i)은 밀결합 유형에 비해 비교적 컨텍스트 영역에 영향을 덜 받는 annotation들의 집합이다. 이 경우에는 식(5)와 같이 컨텍스트 영역의 상, 하 영역의 정보만을 이용하여 annotation 위치와 크기를 결정하게 되며, 따라서 전체 annotation 컨텍스트 영역과 동일한 위치와 크기로 annotation이 입력되지 않는다. 따라서 사용자가 이러한 약결합 유형에 속하는 annotation을 보고 정확한 컨텍스트 영역을 파악하기 힘들어질 수 있다. 이러한 점을 보완하기 위해 실제 annotation 컨텍스트에는 「, 」와 같은 보조 지시자를 삽입하여 쉽게 실제 영역을 알아 볼 수 있도록 하였다.

4.2.1 밀결합 유형

밀결합 유형에는 [정의 2]에서 분류한 T_c (타원, 사각형)와 밑줄 및 형광펜이 있는데, 이러한 유형의 annotation들은 컨텍스트 영역에 꼭 들어맞게 출력된다.

컨텍스트에 기반한 스타일 보정을 위해서는 사용자가 annotation 입력을 목적으로 하는 컨텍스트 영역을 파악한 뒤, 이 영역에 각 유형에 해당하는 스타일 보정을 적용한다. 즉, 컨텍스트가 화면상에서 차지하는 영역을 R_c 라고 하고 시간 t 에 따라 사용자가 입력한 annotation의 위치를 A_t 라고 할 때, A_t 를 포함하는 R_c 가 사용자가 annotation을 입력한 컨텍스트 영역이다. 여기에 컨텍스트 정보를 이용한 영역 보정을 적용하여 최종 목적 컨텍스트 영역을 얻게 된다.

영역 보정을 하는 절차는 먼저 구조적 영역 보정을 시도하고 구조적 영역 보정 기준에 해당하지 않는 경우 비구조적 보정을 적용한다. 구조적 보정을 우선시하는 이유는 XML의 태그로 구분되는 컨텍스트가 단어로 구분되는 비구조적 컨텍스트보다 의미적으로 명확히 구분되기 때문이다.

마지막으로 밀결합 보정은 컨텍스트의 상하좌우 영역 정보를 이용하여 밀결합 유형에 따른 annotation 스타

일인 TAS_{type} (type은 밑줄, 사각형, 타원 혹은 형광펜)을 적용한 최종 annotation을 출력하는 것으로 이루어진다. 이를 수식으로 표현하면 다음과 같다.

$$TA_c = TAS_{type} \cdot T_i(VTC(R_c \supset A_t)) \quad (6)$$

이러한 밀결합 보정에 의해 얻어진 annotation의 출력결과를 그림 7과 같다.

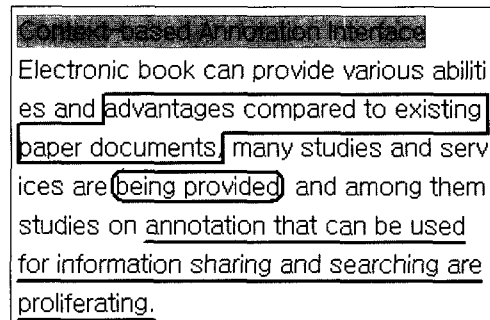


그림 7 밀결합 Annotation의 예

그림에서 보듯이 밀결합 유형의 annotation들은 식(2)에 따라 목적 컨텍스트 영역의 상하좌우 영역의 정보를 이용하여 컨텍스트 영역에 일치하는 스타일로 출력됨을 알 수 있다. 밀결합 유형의 annotation의 특징은 annotation이 출력되는 영역 자체가 곧 annotation 컨텍스트 영역이 되므로 사용자가 쉽게 annotation 컨텍스트 영역을 확인할 수 있다는 데 있다.

4.2.2 약결합 유형

약결합 유형에는 [정의 3]에서 정의한 바와 같이 T_c (밑줄, 괄호), 쪽지 및 기호의 합집합에서 밑줄을 제외한 것들이 속한다.

약결합 유형의 입력 및 영역 보정 과정은 밀결합 유형과 동일하다. 다만 스타일 보정 과정에서 컨텍스트의 상하 영역 정보만을 이용하는 것이 다르다. 약결합 유형에 속하는 기호나 괄호 등은 스타일 특성상 컨텍스트 영역과 정확히 일치되는 스타일로 표현되기 어렵기 때문이다.

따라서 약결합 보정은 컨텍스트의 상하 영역 정보를 이용하여 약결합 유형에 따른 annotation 스타일인 LAS_{type} (type은 괄호, 체크, 가위표, 별표, 혹은 메모)을 적용한 최종 annotation을 출력하는 것으로 이루어진다. 이를 식으로 표현하면 다음 식(7)과 같다.

$$TA_c = LAS_{type} \cdot T_i(VTC(R_c \supset A_t)) \quad (7)$$

약결합 유형의 경우 실제 annotation 컨텍스트를 직관적으로 확인하기가 불분명해질 수 있기 때문에 보조

영역 지시자를 사용한다. 그림 8에서 볼 수 있듯이 이들 유형은 보조 지시자로 표시된 실제 annotation 컨텍스트 영역의 상하 영역과 일치하게 annotation이 입력되었음을 알 수 있다.

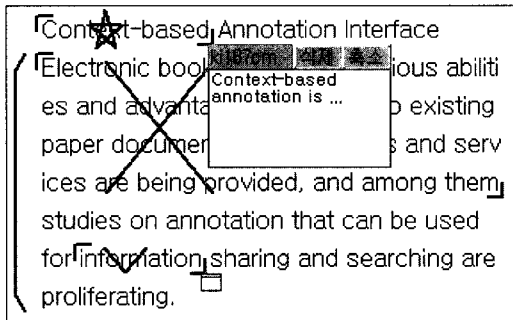


그림 8 약결합 Annotation의 예

약결합 유형을 가지는 annotation의 특징은 별표, 체크, 가위표, 괄호 등과 같이 다양한 스타일을 가지며, 따라서 강조, 상기, 삭제, 구분 등과 같이 여러 의미를 표현할 수 있다는 것이다. 또한 쪽지를 추가하여 다른 유형의 annotation에 부가정보를 추가하는 역할을 할 수 있다.

4.3 Annotation DTD

전자문서 환경의 annotation 시스템에서 사용자가 생성한 annotation은 원본 문서와 별도로 저장되는 것이 일반적이다. 원본 문서가 annotation에 의해 변경되어서는 안되며 annotation을 별개로 저장하여 관리하는 것이 효율적이기 때문이다. 따라서 각 시스템들은 대부분 자체 포맷을 이용하여 각 시스템에서 사용되는 annotation을 저장 및 관리하고 있다. 본 연구에서는 컨텍스트 기반의 annotation의 표현을 위해서 모델링에 기반한 annotation DTD를 제안한다. 이 annotation DTD는 XML에 기반하여 정의되었는데, 이는 국제 표준 형식인 XML을 사용하여 annotation의 확장성 및 호환성을 높이는 장점이 있다.

그림 9는 제안된 DTD의 구조를 나타내고 있다. 모델링의 [정의 1]에 따라 annotation은 크게 유형, 컨텍스트로 구분되며, 각 항목들은 다시 모델링 정의들에 따라 세분화 되어진다. 이 DTD에 정의된 유형들은 전절에서 언급한 바와 같이 7가지의 그래픽 annotation 외에 형광펜과 쪽지를 표현 할 수 있다. 또한 이들은 컨텍스트와의 관계에 따라 밀결합 혹은 약결합 형태로 분류되어 annotation 스타일 보정에 이용될 수 있다.

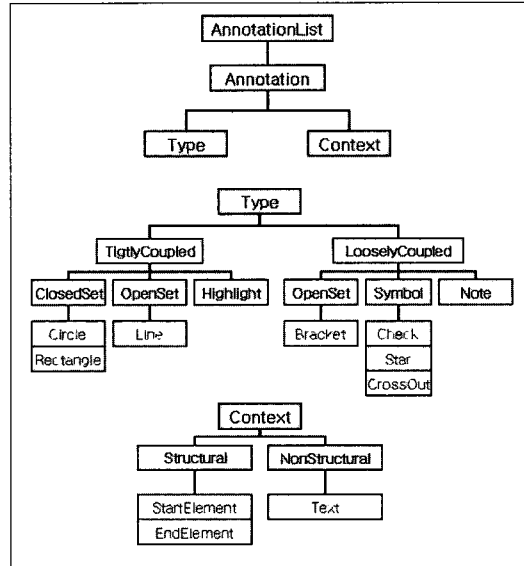


그림 9 Annotation DTD의 구조

DTD의 컨텍스트 항목은 구조적 정보와 비구조적 정보를 표현하고 있으며 이를 이용하여 annotation과 원본 문서의 구조적 연결 및 컨텍스트 정보 추출이 가능하여진다.

이와 같은 DTD에 따라 사용자가 입력한 annotation을 저장한 결과는 그림 10과 같다. 이 결과에는 annotation의 유형, annotated 컨텍스트, 원본 문서에서의 annotation 위치의 구조적 및 비구조적 정보를 포함하고 있다.

이러한 저장결과를 활용하면, 타 시스템으로의 annotation 이식이 용이해질 수 있다. 즉, annotation이 위치한 원본 문서의 비구조적 및 구조적 정보를 가지고 있으므로 XML을 지원하는 전자문서에서 원본 문서의 정확한 위치에 명시된 종류의 annotation의 출력이 가능하여진다. 반면 기존의 시스템에서는 annotation의 위치 정보를 페이지 및 화면상의 좌표값으로 처리하고 있으므로 타 시스템간의 annotation 교환이 어렵다. 또한 원본 문서의 변화시 기존 시스템에서는 annotation의 위치는 화면상에 고정되어 있어 annotation과 컨텍스트의 위치가 어긋나는 단점이 있으나 컨텍스트 기반의 시스템에서는 annotation의 위치가 항상 컨텍스트와 일치하기 때문에 일관성있는 annotation 출력이 가능하다. 그 외에 annotation 검색과 같은 재활용 분야에서도 구조적인 검색이 가능하기 때문에 효율성을 높일 수 있다.

```

<annotation uid="ki187cm" aid="ki187cm705546">
  <type areaSelect="true">
    <tightlyCoupled>
      <openSet>
        <line color="16711680" page="1"/>
      </openSet>
    </tightlyCoupled>
  </type>
  <context>
    <nonStructural>Electronic book can provide advantages </nonStructural>
    <structural>
      <startingElement offset="1" path="/body[2]/chapter[0]/p[2]/#text[0]"/>
      <endingElement offset="61" path="/body[2]/chapter[0]/p[2]/#text[0]"/>
    </structural>
  </context>
</annotation>

```

그림 10 XML로 저장된 Annotation의 예

5. 결론

본 연구에서는 컨텍스트에 기반한 Annotation 모델링의 정의를 내리고 이를 활용한 Annotation 인터페이스 구현 결과에 대하여 알아보았다.

Annotation 모델링에서는 기존의 펜기반 환경의 annotation 입력 시스템이 가지는 문제점을 해결하기 위한 컨텍스트 기반의 annotation 모델을 제시하였다. 기존 시스템에서는 사용자의 입력을 별다른 처리과정 없이 단순한 이미지 형태로 출력하였으나, 본 연구에서는 사용자 입력을 인식하여 이것을 모델링에 정의된 유형별로 구분하고, 컨텍스트 정보를 이용한 영역 및 스타일 보정 기법을 적용하였다. 또한 모델링 정의에 따라 컨텍스트 기반의 annotation을 표현할 수 있도록 XML을 형식의 annotation DTD를 정의하였다.

컨텍스트 기반의 annotation 모델링을 사용하였을 때의 장점은 그래픽 형태의 annotation과 컨텍스트와의 관계가 명확해지므로 다양한 활용이 가능하다는 것이다. 예를 들어 본 연구에서는 컨텍스트와 annotation간에 구조적인 연결이 생성이 되기 때문에 이를 활용한 펜기반 annotation 입력 시스템에서의 annotation 정보를 이용한 원본 문서의 검색이 효율적으로 이루어 질 수 있다. 기존에는 텍스트 혹은 스타일 기반의 시스템에서만 일부 이러한 검색 기능이 제공되어왔다. 또한 기존 펜기반 시스템에서는 annotation의 유형이 존재하지 않았으므로 모든 annotation에 대한 인텍싱만 가능하였으나 본 연구에서 제안한 모델링을 사용하면 유형별 인텍싱도 가능해진다. 또한 컨텍스트 기반의 시스템에서는 컨텍스트의 위치에 따라 annotation이 출력되므로 화면 크기의 변화에도 annotation이 올바른 컨텍스트의 위치에 출력이 되며 다른 시스템과의 교환시에도 컨텍스트 정보를 이용하여 쉽게 annotation을 변환할 수 있다.

향후에는 일반 웹 페이지 혹은 가상 환경과 같이 다양한 환경의 특성에 알맞는 annotation 모델의 정의와 이에 기반한 인터페이스 연구가 요구되며, 또한 이를 통한 전자 문서에서의 서비스 다양화와 정보 공유의 효율성이 기대된다. 즉 annotation 검색 및 공유, 컨텍스트 정보를 이용한 데스크탑과 핸드 헬드 기기와 같은 이기종간의 annotation 변환 및 교환과 같이 전자 문서의 활용성을 높일 수 있는 연구들이 진행되어야 한다.

참고 문헌

- [1] 정필모, 문헌정보학원론, 구미무역(출판사), pp. 254-257, 1996.
- [2] Catherine C. Marshall, Annotation: From Paper Books to Digital Library, Proceedings of the 2nd ACM International Conference on Digital Libraries, Philadelphia, pp. 40-49, 1997.
- [3] Ilia A. Ovsiannikov, Annotation Technology, International Journal of Human-Computer Studies Vol.50 no.4 pp. 329-362, 1999.
- [4] Catherine C. Marshall, Toward an Ecology of Hypertext Annotation, Proceedings of HyperText, ACM, Pittsburgh, pp. 1, 1998.
- [5] Joo-Hwee Lim, Learning Visual Keywords for Content-Based Retrieval, Conf., IEEE, pp. 3-4, 1999.
- [6] Martin Roscheisen, Christian Mogensen, Terry Winograd., Shared web annotations as a platform for third-party value-added information providers: architecture, protocols, and usage examples, Technical Report CSDTR/DLTR, 1994.
- [7] Andre Meyer, Pen Computing, ACM SIGGHI, July 1995.
- [8] Kenneth P. Fishkin, Anuj Gujar, Beverly L. Harrsion, Thomas P.Moran, Roy Want, "Embodied User Interfaces for Really Direct Manipulation",

- Communications for the ACM, Vol. 43 No.9, p 75-80, September 2000.
- [9] Brad A. Myers, "Using Handhelds and PCs Together", Communications of the ACM, Vol. 44, No. 11, pp.34-38, Nov. 2001
 - [10] Davis J, Huttenlocher D., <http://dri.cornell.ed-u/pub/davis/Annotation/annotation.html>, CoNote (Annotation), 1994.
 - [11] CritSuite, <http://crit.org/http://crit.org/index.html>
 - [12] Stanford Digital Library Project, <http://hci.stanford.edu/commentor>
 - [13] Laurent.Denoue <http://julie.univ-savoie.fr/labos/syscom/Laurent.Denoue>, YAWAS, Oct 2000.
 - [14] Classroom 2000 project, <http://www.cc.gatech.edu/fce/eclass/index.html>
 - [15] A.J. Bernheim Brush1, David Bargeron, Anoop Gupta, and JJ Cadiz, "Robust Annotation Positioning in Digital Documents", CHI, ACM SIGCHI, p. 285-292, April 2001.
 - [16] Wojahn, P., Neuwirth, C., and Bullock, B. Effects of interfaces for Annotation on Communication in a Collaborative Task, Proc. of CHI '98, Los Angeles, CA, pp. 456-457, 1998.
 - [17] Hanmandlu, M.; Murali Mohan, K.R.; Kumar, H. "Neural based handwritten character recognition", IEEE Document Analysis and Recognition, pp.241-244, 1999.
 - [18] Marcel Gotze, Magdeburg, Stefan Schlechtweg, "The intelligent pen: toward a uniform treatment of electronic documents", Proceedings of Smart graphics, ACM, pp. 129-135, 2002.



임 순 범

1982년 서울대학교 계산통계학 학사. 1983년 한국과학기술원 전산학 석사. 1992년 한국과학기술원 전산학 박사. 1989년~1992년 (주)휴먼컴퓨터 이사/연구소장. 1992년~1997년 (주)삼보컴퓨터 부장. 1997년~2001년 건국대학교 컴퓨터학과 조교수. 2001년~현재 숙명여자대학교 멀티미디어학과 조교수. 관심분야는 컴퓨터 그래픽스, 멀티미디어 응용, 전자출판 (폰트, 전자책, 사이버교재)



최 윤 철

1773년 서울대학교 전자공학과 졸업(공학사). 1975년 6월 Univ. of Pittsburgh(공학석사). 1979년 6월 Univ of California, Berkeley, Dept. of IE/OR(공학박사). 1979년 8월~1982년 7월 Lockheed 사 및 Rockwell International사 책임연구원. 1982년 9월~1984년 1월 Univ. of Washington 전산학과 박사과정. 1990년 9월~1992년 1월 Univ. of Massachusetts 연구교수. 1984년 3월~현재 연세대학교 컴퓨터학과 교수. 관심분야는 멀티미디어, 컴퓨터 그래픽스, 가상 현실, 지리정보시스템



김 재 경

2000년 단국대학교 전산통계 학사. 2002년 연세대학교 컴퓨터과학 석사. 2003년 현재 연세대학교 컴퓨터학과 박사과정. 관심분야는 Animated Character, 웹 Annotation 생성 및 응용



손 원 성

1998년 동국대학교 컴퓨터공학 학사. 2000년 동국대학교 컴퓨터공학 석사. 2003년 현재 연세대학교 컴퓨터학과 박사과정. 관심분야는 웹 Annotation 생성 및 응용