

전자책 환경에서 Context에 기반한 Annotation Interface

○
김재경, 손원성, 임순범*, 최윤철
연세대학교 컴퓨터학과
*숙명여자대학교 멀티미디어학과

Context-Based Annotation Interface in Electronic Book Environments

○
Jae-Kyung Kim, Won-Sung Sohn, Soon-Bum Lim*, Yoon-Chul Choy
Dept. of Computer Science, Yonsei University
*Dept. of Multimedia Science, Sookmyung Women's University
E-mail : {aqua,sohnws,ycchoy}@rainbow.yonsei.ac.kr, sbkim@sookmyung.ac.kr

요약

전자책은 기존 종이 문서에 비하여 다양한 기능 및 장점을 제공할 수 있기 때문에 현재 다양한 연구 및 서비스가 제공되고 있다. 또한 전자책 환경에서의 정보 공유 및 검색과 같은 다양한 활용을 위해서는 반드시 Annotation 지원이 가능하여야 하며 이에 대한 정확한 Annotation 정의가 요구된다. Annotation이란 일반적으로 문서의 주제 및 내용에 관한 해설, 설명, 그리고 강조를 목적으로 추가되는 문장 또는 텍스트를 의미한다. 그러나 기존 전자책 환경에서의 Annotation과 관련된 연구에서는 이에 대한 심도있는 연구 결과가 미비한 실정이다. 이에 본 연구에서는 전자책 환경을 위한 Context 기반 Annotation Modeling을 정의하고 이를 활용한 인터페이스를 제안한다. 현재 전자책 환경은 대부분 XML에 기반하고 있으며 이에 본 논문에서는 구조정보와 컨텐츠, 그리고 Annotation간의 관계 및 이를 활용하기 위한 모델링을 제시한다. 또한 모델링 정보를 이용한 다양한 장점 및 활용이 가능한 시스템을 구현하였다. 그 결과 본 연구에서는 기존 연구와는 달리 Context 기반 Annotation의 정확한 정의가 가능하며 이를 활용한 다양한 기능을 제공하는 동시에 앞으로의 응용 가능성을 제시하고 있다.

1. 서론

컴퓨터 기술의 발달로 인해 기존의 문서, 책, 잡지 등의 출판물들이 점차 전자 문서화 되고 있으며 실제로 이에 대한 다양한 연구 및 서비스가 진행되고 있다. 이러한 전자 문서의 대표적인 예로 최근 급속도로 활성화되고 있는 전자책을 들 수 있는데, 현재 진행되고 있는 전자책 서비스 및 연구에서는 전자책 내용에 대한 독자 들의 정보부가 기능 및 공유, 교환과 같은 기능이 매우 미비한 실정이다.

따라서 위와 같은 문제점을 해결하기 위해서는 종이책 환경에서 사용되고 있는 Annotation에 대한 활용이 반드시 요구된다.

Annotation이란 일반적으로 문서의 주제 및 내용에 관한 해설, 설명, 그리고 강조를 목적으로 추가되는 문장 또는 텍스트를 의미한다[1]. 종이문서에서의 Annotation은 문서 내용에 대한 정리, 요약, 이해, 기록

등을 위하여 사용되며, 특히 밑줄, 심볼, 노트 등으로 표현된다[2][3]. 그 결과 Annotation 정보는 일회성 정보가 아닌, 재사용 및 공유될 수 있는 중요한 정보라 할 수 있으며, 특히 전자책 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, Annotation에 대한 재편집, 재사용, 검색[5], 공유[6] 등의 기능을 제공할 수 있기 때문이다.

그러나 기존의 전자책 및 Annotation 관련 연구에서는 Annotation을 단순한 일회성 정보로 저장, 처리하고 있어서 생성된 Annotation의 다양한 활용이 불가능하였다. 더욱이 가장 큰 단점은 Annotation에 대한 정확한 정의 및 활용을 위한 모델체계가 결여된 단순한 Annotation 만이 제공된다는 것이다.

따라서 본 연구에서는 전자책에서의 정확한

Annotation 모델링을 제시하고 이에 대한 활용 및 가능성을 제시하고자 한다. 따라서 본 논문에서는 기존 연구의 단점을 극복하기 위하여 구조정보 및 콘텐츠와 Annotation간의 정확한 관계를 포함한 Annotation 정의 및 활용이 가능하며, 향후 이를 활용한 다양한 가능성을 제시하고 있다.

논문의 구성은 다음과 같다. 2장에서는 웹 및 전자책 환경에서 Annotation과 관련된 선행 연구에 대하여 알아본다. 3장에서는 Context 기반의 Annotation 모델링을 제시하고 4장에서는 제안한 모델링에 기반한 Annotation 인터페이스를 설명한다. 마지막으로 5장에서 결론 및 향후 연구 방향에 대하여 기술한다.

2. 관련 연구

Annotation과 관련된 기존 연구는 입력 환경에 따라 웹 환경 및 전자책 환경의 Annotation 입력 시스템으로 구분할 수 있다. 먼저 웹 환경은 Annotation 표현 방식에 따라 텍스트, 스타일, 및 그래픽 기반의 Annotation 시스템으로 나눌 수 있는데 먼저 텍스트 기반의 시스템에 대해 살펴본다.

초기 Annotation 환경에서 많이 사용한 기법 텍스트 기반의 Annotation 시스템[8]이 있다. 흔히 볼 수 있는 게시판 형식과 유사하며 기능이 간단하고 구현이 쉽다. 대표적인 예로 HyperText[4]가 있다. [그림 1]은 HyperText의 초기화면이며 사용자가 입력한 내용에 의견을 추가하는 형식을 취하고 있다. 이 방식은 구현이 매우 쉽고 사용하기가 쉽다는 장점이 있지만 원본 문서와 사용자들의 Annotation이 전혀 연결되지 않아 직관적이 못한 인터페이스라는 단점이 있다. 또한 이와 같이 원본 문서와 독립적인 Annotation은 일시적인 사용자간의 의견 교환에 그치는 일회성 정보로 사용될 수밖에 없어 낮은 재사용성을 가진다.

다음으로 스타일 기반 Annotation 기법[10][11][12][13]은 대부분의 Annotation 환경에 적용되고 있으며 메뉴를 이용한 밑줄, 형광펜, 메모 등의 기능이 있다. 텍스트 기반의 Annotation 기법에 비해 어느정도 직관적인 인터페이스를 제공하며 원본 문서와의 연결도 고려하고 있다. 이러한 인터페이스를 가진 예로서 YAWAS(Yet Another Web Annotation System)[9]라는 스타일 기반의 Annotation 시스템이 있다.

이 시스템은 원본 문서에서 Annotation을 원하는 텍스트에서 메뉴를 생성하고 새로운 팝업창에서 Annotation을 입력하는 인터페이스를 가지고 있다. 텍

스트 기반의 Annotation 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 검색 및 저장과 같이 재활용성을 높였다.

그러나 이러한 스타일 기반의 시스템은 도형과 같이 그래픽에 기반한 인터페이스를 지원하지 않기 때문에 실제 세계에서 사용자가 생성하는 Annotation 스타일을 지원하지는 데 한계가 있다.

마지막으로 그래픽 기반의 Annotation 시스템이 있다. 이것은 사용자가 포토샵과 같은 그래픽 툴을 이용하듯이 간단한 선이나 도형등을 자유롭게 입력할 수 있는 인터페이스[14]를 가지고 있다. 그래픽 기반의 시스템은 스타일 기반의 그것과 비교하여 다양한 Annotation을 입력할 수 있는 장점이 있으며 대부분 그래픽 입력 뿐 아니라 텍스트 및 스타일 입력 인터페이스도 함께 가지고 있기 때문에 가장 사용성이 뛰어나다고 할 수 있다.

그러나 입력된 그래픽 Annotation을 단순히 이미지로 처리하여 저장 및 출력을 하고 있으며 따라서 원본 문서와의 연결이 전혀 되지 않고 입력된 그래픽 Annotation의 종류를 인식하지 않아 재사용성이 크게 떨어지는 단점이 있다.

다음은 전자책 환경에서의 Annotation 시스템에 대해 살펴본다. 서론에서 전술한 바와 같이 전자책 환경에서의 Annotation 사용은 필수적이다. 그러나 현재 전자책 시스템에서는 Annotation에 대한 생성 및 뷰잉 등 단순한 기능만을 지원하고 있으며, 기타 Annotation과 관련된 기술 연구는 많이 부족하다. 전자책에서 Annotation을 활용하기 위한 핵심 기술은 다음과 같이 두 가지로 나눌 수 있다. 첫째, 사용하기 용이한 인지적인 Annotation 인터페이스, 둘째, Context에 기반한 Annotation 인식 기술이 그것이다.

Context라는 것은 Annotation이 입력되어지는 원본 문서의 일부분을 뜻하며 이러한 Context의 크기, 위치, 그리고 구조정보를 이용하여 Annotation의 인식 및 활용이 필수적이라 하겠다. 따라서 본 연구에서는 이러한 기능을 모두 제공하기 위한 Context 기반 Annotation 모델링을 제시하며 이에 대한 자세한 내용은 다음절에서 살펴보도록 한다.

다음 [표 1]에서는 지금까지 살펴본 기존 시스템의 기능들에 대하여 설명하고 있다.

	Annotation 인터페이스			Annotation 인식	
	graphic	style	text	Annotation Recognition	XML Approach
MicroSoft Reader	○	○	○	×	×
netLibrary	×	○	○	×	×
HieBook	×	○	○	×	○
WiseBook	×	○	○	×	○
Adobe eBook Reader	×	○	○	×	×
SoftBook	○	○	○	×	×

[표 1] 전자책 리더의 Annotation 인터페이스

3. Context에 기반한 Annotation 모델링

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

본 연구에서는 Context에 기반한 Annotation 입력 및 인식 보정을 위한 Annotation을 정의하였다. 이를 위하여 다음의 [정의1]부터 [정의8]을 제안하여 Context와의 관계에 따른 Annotation의 타입 인식 및 보정, 그리고 사용자의 의도를 반영하는 의미정보의 표현을 만족할 수 있도록 하였다.

[정의 1]

Annotation Model은 다음과 같이 3가지 기본 요소로 구성되며, 각 요소의 내용은 다음과 같다.

$$A = (T, C, R)$$

T : T는 Annotation을 구성하는 기본적인 타입을 의미한다.

C : C는 Annotated된 Context 정보, 즉 Anchor 텍스트를 의미한다.

R : R은 부가된 Annotation과 문서간의 Relation을 뜻하며, Annotation의 Semantic 정보를 포함하고 있다.

[정의 2]

Annotation 타입은 스타일 유형에 따라 Closed, Open, Symbol, Note 타입으로 구성되며 각 내용은 다음과 같

다.

$$A_t = (T_c, T_o, Symbol, Note)$$

$$T_c = (Circle, Box)$$

$$T_o = (Line, Bracket)$$

$$Symbol = (Check, Star, CrossOut)$$

[정의 3]

Annotation 타입은 Context와의 관계에 따라 Tightly Coupled 타입과 Loosely Coupled 타입으로 구분할 수 있다. Tightly Coupled 타입은 Annotation 영역이 구조 정보, 워드, 구, 구문에 따라 직접적으로 영향을 받는 경우이며, Loosely Coupled 타입은 영향의 정도가 덜한 경우를 의미한다. 이러한 Tightly, Loosely 타입 구성요소와 [정의 2]의 관계는 다음과 같다.

$$T = \{T_i, T_l\}, T_i \cap T_l = \phi$$

$$T_i = \bigcup_{i=1}^n (T_{ci}, \dots, T_{cn}) \cup (Line_1, \dots, Line_n)$$

$$T_l = \bigcup_{i=1}^j (T_{oi}, \dots, T_{oj}) \cup (Line_1, \dots, Line_j) \cup (Symbol_1, \dots, Symbol_j) \cup (Note_1, \dots, Note_j)$$

[정의 4]

Context는 Annotation의 대상인 Anchor Text를 의미하며, NonStructural Context와 Structural Context로 구성된다.

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

[정의 5]

Non_Structural Context는 문서의 기본 구성요소인 Word, Phrase, Sentence로 구성되며, 각 요소들은 일련의 연속된 형태라 가정한다.

$$C_N = (w, p, s)$$

이때에 일련의 알파벳(문자)에 의한 스트링 집합(T)을 워드(w)라 하며, 구(p)는 마침표를 제외한 문자심볼(CS)을 포함한 워드의 반복으로 이루어진다. 문장(s)는 워드와 구문을 포함하되, 마침표를 기준으로 종료되는 일련의 스트링 집합을 의미한다.

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

$$W = \bigcup T$$

$$P = \bigcup_{i=1}^j (T_i, \dots, T_j) \cup (C'S_1, \dots, C'S_j) - (.)$$

$$S = \bigcup_{i=1}^j (T_i, \dots, T_j) \cup (C'S_1, \dots, C'S_j)$$

$$C'_n = \bigcup_{i=1}^j (W_i, \dots, W_j) \cup (P_i, \dots, P_j) \cup (S_i, \dots, S_j)$$

[정의 6]

Structural Context는 NonStructural Context 요소 및 엘리먼트(시작, 끝 엘리먼트)를 기본으로 하는 구조정보 요소로 이루어져 있다.

$$C'_s = \bigcup_{i=1}^j (S_{i1}, \dots, S_{ij}) \cup (C'_{m1}, \dots, C'_{mj})$$

$$S_j = \bigcup (E_s, E_n)$$

[정의 7]

본 연구의 Annotation 보정은 스타일 보정과 영역 보정으로 구분된다. 이때에 각 보정은 Annotation의 타입이 Tightly 혹은 Loosely Coupled 타입인지에 따라 Tightly 혹은 Loosely Coupled 보정으로 구분된다. 또한 Tightly Coupled Type일 경우 Context 상하좌우 정보를, Loosely Coupled 타입은 Context 상하 정보를 우선시 하도록 한다.

$$A_s = (T.A_s, L.A_s)$$

$$f: T_i(t, b, r, l) \longrightarrow T.A_s$$

$$s: T_i(t, b) \longrightarrow L.A_s$$

[정의 8]

Relation(R)은 Annotation의 Semantic 기능으로서, Annotation 작성에 대한 사용자의 의도를 나타낸다. 본 연구에서는 이러한 Annotation의 기능을 다음과 같이 구분한다.

$$R = \{ Placeholder, readingAssistant, selfThinking \}$$

$$Placeholder = \{ remind, mark \}$$

$$ReadingAssistant = \{ interpretive, mark \}$$

$$SelfThinking = \{ idea, criticism \}$$

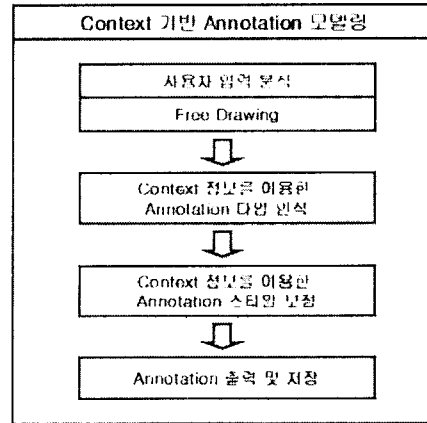
또한 이러한 Relation은 Annotation Type과 Context에 대하여 다음과 같은 관계를 갖는다.

$$REL: (T, C) \xrightarrow{R} A$$

4. Annotation 인터페이스 구현

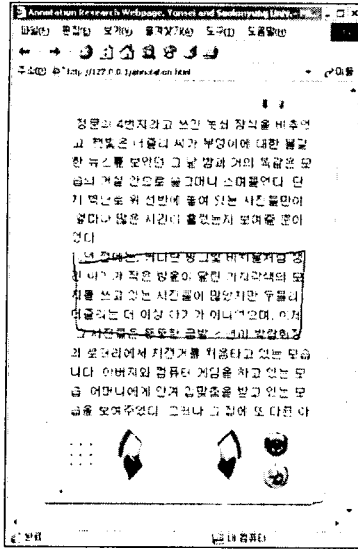
본 연구의 Annotation 인터페이스는 편의성을 고려하기 위하여 Free Drawing 방식으로 입력을 하도록 한다. Annotation 입력은 메뉴 선택 방식보다는 펜이나 마우스를 이용한 드로잉 인터페이스를 제공하여 종이 문서에서와 동일한 환경을 제공할 수 있도록 하기 위함이다.

본 연구에서 인식되는 Annotation 타입은 3장의 에서 정의한 바와 같이 밑줄 및 형광펜, 도형(타원, 사각형) 및 심볼(별표, 체크, 가위표, 괄호)이며 이러한 그래픽 입력 외에 텍스트를 입력하기 위한 메모 기능 또한 지원한다.

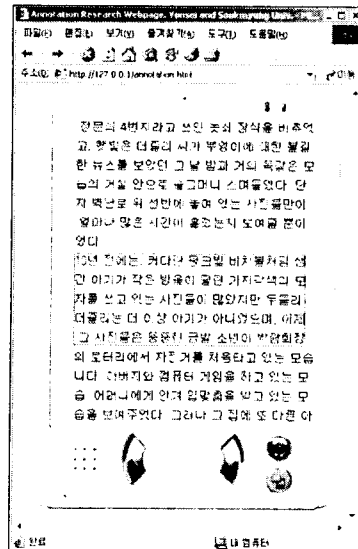


[그림 1] Annotation 입력 처리 과정

Annotation 입력 과정은 [그림 1]과 같다. 사용자의 Free Drawing 입력이 들어오면 이것을 분석하여 [정의 2]에 따라 기본적인 타입 분류를 수행한다. 그 다음 Context 정보를 이용하여 [정의 3]~[정의 7]에 따라 알맞은 Annotation의 스타일을 적용한다. 마지막으로 결과를 출력 및 저장하게 되는데 결과물은 Context와 연관되어 구조적 문서로 저장되므로 검색 및 공유와 같은 재활용성을 높일 수 있다.



[그림 2] 사용자 Annotation 입력 직후



[그림 3] 모델링 정의에 의해 인식 및 보정 후 결과

[그림 2]는 사용자가 사각형을 입력하였을 경우이며 [그림 3]은 인식과정을 거친 결과의 예이다. 모델링의 정의들에 따라서 타입분류를 수행하고 이에 따른 스타일을 적용하여 원본 문서의 Context에 정확하게 위치되어 출력됨을 볼 수 있다.

5. 결론

본 연구에서는 Context에 기반한 Annotation 모델링의 정의를 내리고 이를 활용한 Annotation 인터페이스에 대하여 알아보았다.

본 연구에서 정의한 Annotation 정의 및 인지적 인터페이스는 전자책 이용자에 대한 사용편의성 향상과 다양한 활용성을 제공할 수 있으며 그 실례를 살펴볼 수 있었다.

본 논문은 전자책 환경을 기반으로 Annotation 모델링을 정의하고 인터페이스를 구현하였으나 향후 사이버 클래스나 일반 웹 페이지에서도 각 환경이 가지고 있는 특성에 알맞는 Annotation 모델링의 정의와 이에 기반한 인터페이스연구가 요구되며 이를 통한 서비스의 다양화와 정보 공유의 효율성이 기대된다.

[참고문헌]

[1] 정필모, 문헌정보학원론, 구미무역(출판사), 1996.
 [2] Catherine C. Marshall, Annotation: From Paper Books to Digital Library, Proceedings of the 2nd ACM International Conference on Digital Libraries, Philadelphia, 1997.

[3] Iliia A. Ovsianikov, Annotation Technology, International Journal of Human-Computer Studies v.50 n.4 pp.329-362, 1999.
 [4] Catherine C. Marshall, Toward an Ecology of Hypertext Annotation, Proceedings of HyperText, ACM, Pittsburgh, 1998.
 [5] Joo-Hwee Lim, Learning Visual Keywords for Content-Based Retrieval, Conf., IEEE, 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] Laurent Demoue, Laurence Vignollet., An annotation tool for web browsers and its applications to information retrieval, RIAO2000, Paris, 2000.
 [8] Davis J, Huttenlocher D., <http://dri.cornell.edu/pub/davis/Annotation/annotation.html>, CoNote(Annotation), 1994
 [9] Laurent.Denoue <http://julie.univ-savoie.fr/labs/syscom/Laurent.Denoue>, YAWAS, Oct 2000
 [10] Venu Vasudevan, Mark Palmer., On Web annotations: promises and pitfalls of current Web infrastructure, Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences, IEEE, 1999.
 [11] Price, M., Schilit, B., XLibris: the active reading machine, CHI98, Los Angeles, 1998.
 [12] Annotation Engine : Harvard Law School, <http://cyber.law.harvard.edu/projects/annotate.html>
 [13] Stanford Digital Library Project, <http://hci.stanford.edu/commentor>
 [14] Microsoft, <http://www.microsoft.com/reader/>, 2001