

# XML 문서에서의 펜 기반 교정 인터페이스

## (A Pen-based Proofreading Interface in XML Documents)

손원성<sup>†</sup> 김재경<sup>\*\*</sup> 최윤철<sup>\*\*\*</sup> 임순범<sup>\*\*\*\*</sup> 김우성<sup>\*\*\*\*\*</sup>  
 (Won-Sung Sohn) (Jae-Kyung Kim) (Soon-Bum Lim) (Yoon-Chul Choy) (Woo-Sung Kim)

**요약** 펜 기반 교정 환경에서 정확한 마킹영역을 판별하기 위해서는 사용자가 드로잉한 교정마킹과 문서영역간의 관계를 판별하는 과정에서 발생하는 모호성(ambiguity)을 해결하여야 한다. 또한 웹 및XML과 같은 구조문서 환경에서는 변경된 문서구조가 반드시 정의된 DTD에 따라야 한다. 본 논문에서는 자유형 마킹에 의한 교정부호 입력이 가능하며, 교정 마킹과 원본문서의 영역을 판별하는 과정에서 발생하는 모호성 문제를 최소화하기 위한 컨텍스트 기반의 영역인식 및 보정 인터페이스(Context-based Proofreading Interface)를 제안한다. 제안 인터페이스에서는 교정부호의 입력에 따른 문서내용 및 구조의 변화가 발생할 경우 올바르게 문서 구조를 유지하기 위한 구조변경 방법을 제공한다. 그 결과 본 논문에서 생성된 교정마킹 정보는 기존 교정시스템에 비하여 보다 정확한 영역정보를 포함할 수 있으며, 교정부호 입력에 따른 구조문서 변경시에도 원본문서의 DTD에 따르는 문서구조를 유지할 수 있다.

**키워드** : 펜 기반 교정, XML, 어노테이션, 모호성, 컨텍스트

**Abstract** Accurate proofreading interface requires resolving the ambiguity that occurs when the system determines the relations between the free-form by the user and editing scopes of the document. Proofreading for structure documents such as XML/XHTML involves modification of document structures and modified document also should follow the pre-defined DTD. This paper present a CPI (Context-based Proofreading Interface) based on the XML. The CPI supports free-form drawing of correction marks and provides context-based scope recognition and revision methods. CPI provides both implicit and explicit modification methods for document structures. As a result, the correction mark information produced in this paper includes more accurate scope information than that in other systems.

**Key words** : Pen-based Proofreading, XML, Annotation, Ambiguity, Context

### 1. 서론

전통적인 문서교정(document proofreading) 환경에서는 종이문서를 대상으로 교정자가 교정부호를 드로잉하고, 그 결과에 따라 문서 작성자 혹은 편집자가 내용을 수정하는 일련의 단계를 포함한다[1-4].

컴퓨터기술의 발달에 따라 전자문서(웹) 환경에서 교

정부호를 이용한 교정 기법 및 시스템들이 제안되었으며 이는 기존 종이문서 환경에 비하여 다중사용자간의 자유로운 교정부호의 편집 및 재활용 등과 같은 다양한 장점을 제공할 수 있다[1,5]. 사용성(usability) 측면에서 효과적인 교정 입력인터페이스를 제공하기 위해서는 전통적인 종이 문서 환경에서 사용되었던 펜기반 교정 인터페이스를 지원하는 것이 바람직하다[6,7]. 그러나 전자 문서 환경에서 펜을 이용한 문서 교정기능을 제공하기 위해서는 기술적으로 해결되어야 할 문제점들을 포함하고 있다. 전자문서 환경에서 펜 입력기반의 교정 마킹 기능을 제공하는 시스템[2,5,6]에서는 사용자가 입력한 자유형(free-form) 마킹과 대상문서의 영역을 판별하는 과정에서 발생할 수 있는 모호성(ambiguity) [5,8] 문제가 필수적으로 발생하며, 이러한 모호성은 사용자의 의도와 전혀 다른 교정결과를 생성하게 된다. 그러나 펜 기반의 교정 인터페이스를 제공하는 기존 시스템들은

<sup>†</sup> 정 회 원 : 경인교육대학교 컴퓨터교육과 교수  
sohnws@ginue.ac.kr

<sup>\*\*</sup> 학생회원 : 연세대학교 컴퓨터과학과  
ki187cm@rainbow.yonsei.ac.kr

<sup>\*\*\*</sup> 중신회원 : 연세대학교 컴퓨터과학과 교수  
ycchoy@rainbow.yonsei.ac.kr

<sup>\*\*\*\*</sup> 중신회원 : 숙명여자대학교 멀티미디어과학과 교수  
sblim@sookmyung.ac.kr

<sup>\*\*\*\*\*</sup> 비 회 원 : 호서대학교 컴퓨터공학과 교수  
wskim@office.hoseo.ac.kr

논문접수 : 2003년 9월 9일

심사완료 : 2005년 12월 15일

앞서 설명한 모호성 문제를 고려하지 않는다.

최근의 교정 시스템은 일반 전자문서 뿐만 아니라 웹 환경에서 HTML 문서를 교정하기 위한 기능을 제공하고 있으며 동시에 명확한 문서 구조를 포함한 XML 문서를 교정의 대상으로 간주할 필요가 대두되고 있다[3]. 웹 환경에서의 교정 작업은 마크업문서 구조에 대한 변화를 반드시 수반하며, 특히 XML[10]과 같은 구조문서 환경에서는 교정에 따라 변경된 문서 구조 또한 정의된 DTD(document type definition)에 따라야 한다. 그러나 현재 대부분의 웹 기반 교정 시스템에서는 교정작업에 따른 문서 구조 변경을 고려하지 않거나 HTML 문서만을 고려한다[11,12].

본 논문에서는 XML 문서를 기반으로 하며 모호성 문제를 해결하기 위한 교정 인터페이스(CPI: Context-based Proofreading Interface)를 제안한다. CPI에서는 자유형 마킹에 의한 교정부호 입력이 가능하며, 기존 연구와는 달리 교정 마킹과 원본문서의 영역을 판별하는 과정에서 발생하는 모호성을 최소화하기 위한 컨텍스트 기반 영역 인식 및 보정 기법을 포함한다. 또한 본 논문에서는 입력된 마킹을 XML 형태의 구조정보로 표현하며 이를 위하여 Context-based Proofreading Markup Language(CPML)를 정의하였다. 마지막으로 CPI에서는 기존 HTML 기반의 시스템과는 달리 교정부호의 입력에 따른 문서 내용 및 구조의 변화가 발생할 경우 시스템 내부에서 자동으로 문서 구조를 유지하기 위한 묵시적인 구조변경 기법과 사용자와의 인터랙션을 통하여 문서 구조를 유지하는 명시적인 문서 구조변경 방법을 제공한다.

그 결과 본 논문에서 생성된 교정 마킹 정보는 기존 시스템에 비하여 보다 정확한 영역 정보를 포함할 수 있으며, 교정 부호 입력에 따른 구조문서 변경에도 원본 문서의 DTD에 위배되지 않는 문서 구조를 유지할 수 있다.

## 2. 관련연구

최근의 교정시스템은 종이 문서가 아닌 전자문서에 교정부호를 입력하고 그 결과를 즉시 확인할 수 있는 형태의 문서편집 과정을 수행한다. 이러한 교정시스템은 간결한 교정 절차, 자유로운 교정부호 입력 및 편집, 다중사용자간의 협업, 교정 결과에 대한 재활용 등과 같은 다양한 장점을 제공할 수 있다.

특히 전자문서 환경에서도 전통적인 종이환경에서 사용하는 마크업 모델[3]을 적용한 교정 인터페이스가 제안된 바 있으며, 쉽게 교정자의 마킹을 이해할 수 있다는 장점을 포함한다. 이러한 인터페이스는 교정부호의 입력방식에 따라 마우스를 이용하는 아이콘기반 교정

인터페이스 및 펜 입력장치를 사용하는 펜 기반 교정 인터페이스로 구분된다.

아이콘 기반의 교정 인터페이스에서는 종이문서 환경에서 사용하는 일반적인 교정부호를 아이콘화하고, 마우스를 통해 해당 아이콘을 선택하여 문서의 교정을 원하는 부분에 클릭함으로써 교정자의 교정 의도를 표현한다. 대표적인 아이콘 기반 인터페이스로는 CoCoA (Commutative Collection Assistant System)[12]를 살펴볼 수 있으며, 다음 그림 1과 같이 아이콘을 이용한 교정부호 입력화면 및 그 편집 결과를 확인할 수 있는 편집뷰어 화면을 포함하고 있다.

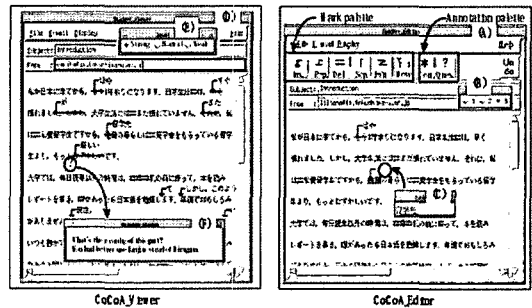


그림 1 아이콘 기반 온라인 교정 시스템(CoCoA)의 예

아이콘 기반 교정 시스템은 기존 교정 인터페이스에 비하여 직관적인 교정부호 입력 및 편집이 가능하며, 편집자와 문서작성자간의 의견을 보다 정확히 전달할 수 있는 장점을 제공한다. 그러나 교정부호 입력을 위한 아이콘 선택 및 영역 지정을 위해서는 펜 입력에 비하여 다소 높은 오버헤드를 필요로 하며, 마우스 드래깅을 이용하여 다중문단과 등과 같은 영역을 교정하기에는 다소 불편한 작업을 수행하여야 한다[2,4].

따라서 전자문서 환경에서도 전통적인 문서교정 환경에서 사용하던 표준적인 교정 부호를 펜 입력장치를 통하여 드로잉할 수 있는 인터페이스가 제안된 바 있다. 이러한 펜 기반의 교정 인터페이스는 저자와 교정자에게 실제계와 유사한 교정 환경을 제공하며 기존 인터페이스에 비하여 가장 직관적인 문서 교정이 가능하다[3,5,13]. 펜 기반 교정 인터페이스에서는 교정부호를 입력(mark)하기 위하여 digital inking을 제공하며, 입력된 교정부호는 별도의 인식처리를 통한 인식과정을 통하여 적절한 편집동작으로 변환된다.

이러한 펜 기반 교정 모델을 채택한 기존 시스템으로는 MATE[4], PenEdit[4], 그리고 Amaya+PEN[11] 등을 살펴볼 수 있다. MATE에서는 펜을 이용하여 체크처를 드로잉하고 입력된 체크처는 다음 그림 2와 같이 별도의 인식과정을 거쳐 적절한 편집동작으로 변환된다.

ANNOTATION VIEW	EDIT VIEW
<p>This is a sample document with several annotations mark<sup>ed</sup> on it. Note that some annotations correspond to <u>specific</u> editing commands.</p> <p>Whereas others are more general comments.</p>	<p>This is a sample document with several annotations marked on it. Note that some annotations correspond to editing commands.</p> <p>Whereas others are more general comments.</p>

그림 2 MATE 시스템의 화면 예

따라서 사용자는 편집자의 의도 및 편집결과를 직관적으로 파악할 수가 있다.

PenEdit는 다음 그림 3과 같이 스크린을 통한 펜기반 입력 및 제스처 인식 기능을 제공하며, 편집자는 자신의 의도를 교정부호와는 별도의 어노테이션 형태로 표현할 수 있는 인터페이스를 제공하여 저자와 편집자간의 정확한 커뮤니케이션을 가능케 한다. 그러나 디지털 잉킹(digital inking)이 가능한 기존 교정 시스템에서는 편집자가 입력한 자유형 마킹과 원본 문서의 텍스트간의 영역을 판별하는 과정에서 모호성 문제가 발생할 수 있다. 이러한 모호성은 교정 마킹의 정확한 영역 판별을 방해하여 편집자가 의도한 편집 영역과 상이한 결과를 초래하기 때문에 반드시 이러한 문제를 해결할 수 있어야 한다.

Much of the background knowledge & and guidelines for structure ~~look~~ plan design comes from the field of exhibit plan design (e.g., Klein, 1988; Konikow, Aur. Spall 1 1984; Miles & Alt, 1988), Aur. Date 1 Indeed, a major principle of exhibit plan design is to create displays that attract attention & and invite participation. However, the many technology components & and the nature of the interaction possible with interactive systems poses many new challenges even for experienced exhibit designers(FN 2).

End Page 1

그림 3 PenEdit 시스템의 예

한편 Aamaya+PEN[11]에서는 앞서 살펴본 교정 인터페이스와는 달리 HTML 문서를 교정 타겟으로 간주한다. Aamaya+PEN은 다음 그림 4와 같이 W3C에서 제공하는 HTML 편집기인 Amaya[11]를 기반으로 펜기반의 웹문서 편집이 가능하다. 펜기반의 제스처 인식에서는 교정 환경의 특성을 고려한 기하학적 알고리즘을 사용하였으며 텍스트 정보와 부호간의 관계를 고려한 교정 부호 인식 기법을 적용하였다. 그러나 Aamaya+PEN에서도 자유형 마킹에 대한 모호성 문제가 발생할 수 있으며, 동시에 HTML 문서 편집으로 인한 구조 정보의

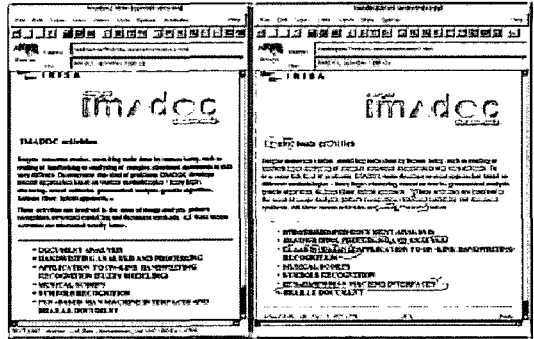


그림 4 펜기반 온라인 교정시스템(Amaya+Pen)

변경시 단순히 <P> 태그를 삽입하거나 삭제하는 수준의 단순한 기능만을 제공하고 있다.

본 논문에서는 XML문서에 대한 교정을 위해서 전통적인 교정환경에서 제공하던 펜 기반의 교정부호를 자유형 마킹 형태로 제공한다. 또한 입력된 교정부호의 정확한 영역을 판별하기 위하여 마킹과 원본문서간의 영역 모호성 문제를 해결할 수 있는 컨텍스트 기반 영역 인식 및 보정기법을 제안한다. 또한 본 논문에서는 교정부호 입력시 발생하는 구조변경의 문제를 해결하기 위하여 명시적인 구조 변경 기법 및 사용자와의 인터랙션을 통한 명시적인 구조 변경 기법을 제시한다.

### 3. 컨텍스트 기반 교정 마크업언어

XML 문서환경에서 디지털 잉크 형태의 교정마킹을 생성하기 위해서는 입력된 마킹의 그래픽 정보와 원본 문서의 컨텍스트 정보를 외부링킹[14] 형태로 표현할 수 있어야 한다. 외부링킹 기능을 통하여 원본 문서와 독립적인 형태로 자유로운 교정마킹을 입력할 수 있을 뿐만 아니라 다중 사용자 환경에서 교정마킹을 정확히 공유할 수 있다. 이러한 기능을 제공하기 위해서는 가장 먼저 교정마킹의 타입 및 원본 문서의 컨텍스트를 명확한 형태의 정보로 표현할 수 있어야 한다.

본 논문에서는 XML 형태로 교정부호를 정의하고자 하며 이를 위하여 XML 기반의 Context-based Profreading Markup Language(CPML)를 정의하였다. 제안 논문에서 정의한 CPML의 전체 구조는 다음 그림 5와 같다.

CPML은 그림 5와 같이 루트 엘리먼트인 *correctionList* 하부에 복수의 *meta* 엘리먼트 및 *correction* 엘리먼트를 포함한다. *Meta* 엘리먼트는 교정자의 이름 (*editor*), 교정 날짜(*date*), 교정버전을 기록하기 위한 *version* 엘리먼트를 포함한다. *Correction* 엘리먼트는 교정부호를 표현하기 위한 *type* 엘리먼트, 교정문서와

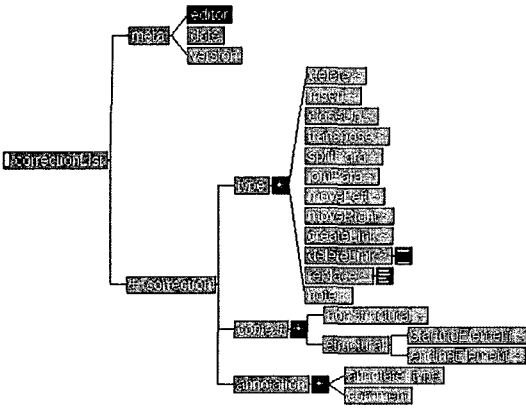


그림 5 CPML의 전체 구조

Correction	Notation	Correction	Notation
Delete		CloseUp	)
Insert	<	MoveLeft	↶
Transpose	↷	MoveRight	↷
SplitPara	↵	CreateLink	00
JoinPara	↵	DeleteLink	00
Replace	≡	Note	≈

그림 6 CPML의 교정부호(DeleteLink 부호의 경우 링크가 생성된 이후에만 적용된다.)

교정마크간의 교정영역 및 구조정보를 포함하는 context 엘리먼트, 그리고 입력된 교정부호에 대한 교정자의 의도 등을 나타내기 위한 annotation 엘리먼트를 포함한다.

CPML에서는 그림 6과 같이 총 12개의 교정부호 타입을 정의하였으며 이는 교정 부호 표준인 Chicago Style Proofreader's Marks[15]를 전자문서 환경에 맞

게 간소화한 것이다. CPML의 교정 타입은 delete, insert, transpose, closeUp, splitPara, joinPara, moveLeft, moveRight, createLink, deleteLink, replace, note 엘리먼트를 포함한다.

CPML의 context는 structure 엘리먼트와 nonstructure 엘리먼트를 포함한다. 구조 엘리먼트는 다음 그림 7(a)처럼 타겟 컨텍스트 중 구조문서의 엘리먼트 시작태그 및 끝태그 정보를 포함한다. 또한 구조 엘리먼트의 속성에는 원본문서와 교정기호간의 링크를 위하여 시작 및 끝 태그의 패스정보와, 오프셋 정보를 포함한다. 본 연구에서 교정 부호를 구조와 비구조 엘리먼트로 표현한 이유는 다음 그림 8과 같이 원본 문서에 생성된 교정부호는 외부링크로 표현되며, 또한 원본문서는 구조문서이기 때문이다.

비구조 엘리먼트는 그림 7(b)와 같이 교정의 대상인 문서의 텍스트 정보를 포함한다. 이러한 텍스트 정보는 단순히 하나의 교정 대상에 국한되지 않기 때문에 CPML에서는 교정 부호에 따라 상이한 타겟을 포함한다. 예를 들어 joinPara와 같은 교정 부호는 front와 end의 타겟 텍스트를 포함하여야 하며, delete와 같은 교정 부호는 단순히 하나의 타겟 텍스트만으로도 그 의미를 정확히 전달 할 수 있다.

#### 4. 펜 기반 교정 인터페이스

본 논문에서는 XML 문서를 기반으로 하는 교정 인터페이스를 제안한다. 제안 인터페이스에서 드로잉된 교정부호의 스타일 타입 및 영역 정보는 XML 문서 컨텍스트에 기반한 제안 기법을 통하여 인식 및 최종 보정된다. 또한 제안 시스템은 교정부호의 입력에 따른 문서 내용 및 구조의 변화가 발생할 경우 시스템 내부에서 문서 구조를 유지하기 위한 목시적인 구조변경 기법과 사용자와의 인터랙션을 통하여 문서 구조를 유지하는 명시적인 문서 구조변경 방법을 제공한다.

```

<!ENTITY % context "nonStructural,structural">
<!ENTITY % att_structural "startPath CDATA #IMPLIED
endPath CDATA #IMPLIED
offset CDATA #IMPLIED">
<!ELEMENT context (%context;)*>
<!ELEMENT structural (startingElement, endingElement)>
<!ELEMENT startingElement EMPTY>
<!ATTLIST startingElement %att_structural;>
<!ELEMENT endingElement EMPTY>
<!ATTLIST endingElement %att_structural;>
    
```

(a) CPML의 구조 엘리먼트

```

<!ENTITY % context "nonStructural,structural">
<!ELEMENT nonStructural EMPTY>
<!ATTLIST nonStructural %att_nonStructural;>
<!ENTITY % att_nonStructural
"targetContext CDATA #IMPLIED
frontContext CDATA #IMPLIED
endContext CDATA #IMPLIED">
    
```

(b) CPML의 비구조 엘리먼트

그림 7 CPML의 컨텍스트 구조

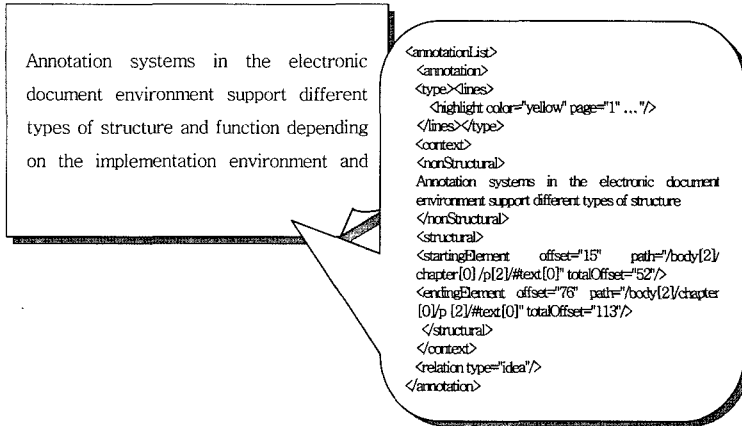


그림 8 교정 마킹에 대한 외부링크 표현 예

4.1 모호성 해결을 위한 컨텍스트 기반 교정영역 인식 및 보정기법

본 절에서는 온라인 교정 부호와 편집 대상 텍스트간 에 편집 영역 간의 관계를 판별하는 과정에서 발생하는 모호성을 최소화하기 위한 컨텍스트 기반 영역인식 및 보정 기법에 대하여 설명한다.

본 연구에서 제안하는 교정부호 인식은 두 가지 관점으로 분류를 할 수 있다. 첫째는 사용자가 입력한 교정부호의 타입을 인식[16]하는 것이고 둘째는 교정의 대상인 원본 문서의 편집 영역을 인식하는 것이다. 또한 인식된 편집 영역은 원본 문서와 교정 부호간의 컨텍스트 정보에 근거하여 최종 보정된다. 편집 영역을 인식하고 보정하기 위한 제안 기법은 IF-THEN 문법으로 표현되는 규칙 모델을 통하여 처리된다. 본 논문의 규칙 시스템은 총 12개의 교정 부호 타입과 편집 영역 인식을 위한 조건으로 이루어진 120여개의 규칙[17]으로 정의되어 있다.

원본 문서의 편집 영역을 인식 및 보정하기 위해서 본 논문에서는 교정마크와 텍스트 영역간의 관계를 다음 그림 9와 같은 컨텍스트 기반 섹터로 표현한다. 제안된 섹터들은 각각 문자, 단어, 문장, 단락 과 같은 컨텍스트 단위로 구성되며 각 단위는 박스로 구분한다. 또한 각각의 박스 내부영역은 컨텍스트 단위에 따라 문자인 경우 VCL(vertical center of letter) 및 섹터 단위에 대한 TL(top of letter) 과 BL(bottom of letter) 등으로 구분된다.

제안 인식 및 보정기법은 교정부호의 타입에 따라 각기 다른 섹터 단위를 포함한다. 예를 들어 delete 부호의 경우는 문자, 단어, 문장 단위로 인식 기법을 적용하며, joinPara와 같은 교정부호는 문단 단위로 편집 영역을 인식한다. 본 절에서는 delete 교정부호 입력에 대한

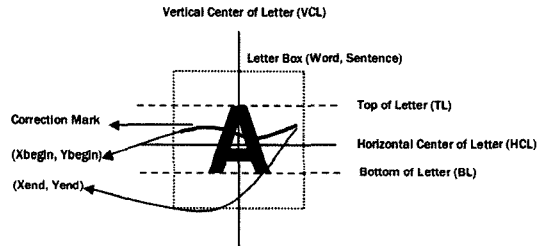


그림 9 편집 영역을 분석하기 위한 제안 논문의 컨텍스트 기반 섹터(문자단위의 경우)

예제를 중심으로 제안 기법에 대하여 설명한다.

4.1.1 문자단위의 교정영역인식 및 보정

다음 그림 10에서 입력된 교정부호는 펜기반 제스처는 인식 알고리즘[16]을 이용하여 제안 DTD의 교정타입과 매핑된다. 예를 들어 본 연구에서는 텍스트의 중간을 가로로 긋는 선을 delete라고 정의 하였다. 따라서 그림 10의 교정 부호의 타입은 delete가 된다.

교정타입에 대한 인식과정 후에는 교정 영역에 대한 인식과정을 수행한다. 이를 위해서는 먼저 입력된 교정부호와 원본문서의 컨텍스트간의 연관관계를 제안된 섹터를 이용하여 추출한다. 이때의 섹터는 입력된 교정부호의 타입에 따라 각기 다르게 선택되며, 본 절에서는 다음 그림 10과 같이 문자 단위의 'delete' 교정의 경우를 살펴본다. 그림 10에서 입력된 delete 부호는 왼쪽을 기준으로 6,7번째 'letter box'의 'w', 'o'와는 접하고 있지만, 5번째 'o'의 접점의 여부는 명확하지 않다. 이때에 제안 기법에서는 이러한 모호성을 해결하기 위하여 Xbegin과 Xend의 위치가 'letter box' VCL의 좌우측에 존재하는가의 여부에 따라 그 위치의 문자를 편집 대상 영역으로 넣을 것인지 결정한다.

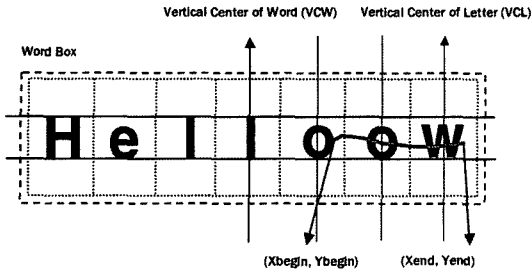


그림 10 'Delete' 연산에 대한 문자단위의 교정영역 분석과정

그림 10의 예제는 5번째 LETTER인 'o'는 교정 부호의  $X_{begin}$ 이 VCL 우측에 존재하므로 규칙(11)에 따라 편집 대상 텍스트 영역을 'o'가 포함된 다음 'letter box'로 이동한다.  $X_{end}$ 는 마지막 'letter box'의 VCL 우측에 존재하므로 텍스트 'w'를 편집의 대상으로 간주한다. 그 결과 그림 6의 예제는 6,7번째 'o, w'를 포함하는 편집 영역으로 보정되며 그 결과는 다음 그림 11과 같다.

규칙 (11) ;

- IF :
- (1) 교정타입이 삭제이다.
  - (2)  $X_{begin}$ 과  $X_{end}$  영역이 word box를 넘지 않는다.
  - (3)  $X_{begin}$ 과  $X_{end}$  사이에 복수의 letter box가 존재한다.
  - (4)  $Y_{begin}$ 과  $Y_{end}$  사이의 좌표점들은 TL와 BL 영역에 포함된다.
  - (5) 교정 부호의  $X_{begin}$  지점이 letter의 VCL 우측에 위치한다.

THEN :

- (1)  $X_{begin}$  지점을 다음 letter box로 이동한다.
- (2) 다음 해당 규칙을 수행한다.

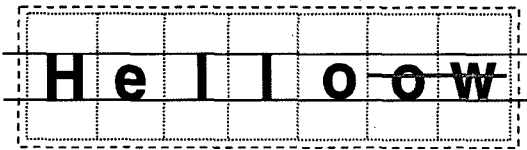


그림 11 그림 10에 대한 영역 보정 결과

한편 그림 11의  $X_{begin}$ 과  $X_{end}$  사이 'letter box' 영역의 길이는 'letter box'로 구성된 'word box' 영역의 근소한 부분을 포함하고 있다. 따라서 그림 11은 다음 규칙 (23) 및 식 (1)을 만족하지 않기 때문에 단어영역 내에 포함되는 컨텍스트의 편집 단위를 문자단위의 편집 연산으로 간주한다. 그러나 그림 11의 예제가 규칙(23) 및 식 (1)과 같은 임계값(Threshold of Word Area)을 만족한다면 제안 규칙에서는 문자단위가 아닌 단어단위의 연산으로 간주하여 새롭게 교정영역을 보정하게 된다. 본 논문에서의 임계값은 사용자 평가를 통하여 사용

자들로부터 최상의 만족도를 나타내는 수치를 선택한 것이다. 따라서 그 결과는 평가의 성별, 연령, 컨텍스트의 요소, 시간 등의 요소로부터 영향을 받을 수 있으며, 앞으로 이에대한 분석적인 연구가 필요할 것으로 판단된다.

규칙 (23) ;

- IF :
- (1) 교정 타입이 삭제이다.
  - (2) 인식된  $X_{begin}$ ,  $X_{end}$ 의 길이가 ThWA의 일정 기준을 만족한다.
- THEN :
- (1) 교정 영역을 Word box 단위로 간주한다.
  - (2) 다음 해당 규칙을 수행한다.

$$ThWA = \frac{\sum Total Lengthof Xmin And Xmax of The Correction Mark}{\sum Total Lengthof Word Box Area} \quad (1)$$

4.1.2 단어단위의 교정부호 인식

다음 그림 12는 단어단위에서의 delete 교정의 경우로 복수의 단어들을 편집의 대상으로 한다. 따라서 제안 기법에서는 교정부호에 대한 편집 영역을 추출하기 위하여 교정부호와 워드박스간의 연관관계를 분석한다.

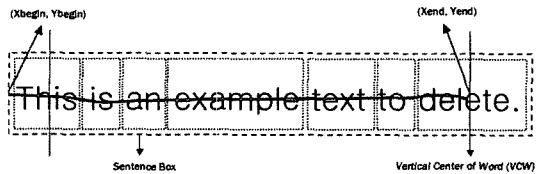


그림 12 Delete 연산에 대한 단어단위의 교정영역 분석과정

그림 12의 경우 'delete' 부호의  $X_{begin}$ 과  $X_{end}$  영역은 'sentence box'를 넘지 않으며 동시에 복수의 'word box'를 포함하고 있다. 따라서 제안 기법에서는 먼저 편집의 단위를 단어단위로 간주하여 시작 및 마지막 'word box'를 찾게 된다. 그림 12의 첫번째 워드박스에서는 VCW의 좌측에  $X_{begin}$ 이 존재하기 때문에 규칙 (42)에 근거하여 해당 워드박스를 편집의 대상으로 간주한다. 또한 마지막 워드박스에서는  $X_{end}$ 가 VCW의 왼쪽에 존재하므로 해당 워드박스의 왼쪽 워드박스를 편집의 대상으로 간주한다. 따라서 delete 교정부호의 편집 영역을 'This'로 부터 'to'까지의 워드박스로 교정한다.

규칙 (42) ;

- IF :
- (1) 교정 타입이 삭제이다.
  - (2)  $X_{begin}$ 과  $X_{end}$  영역이 sentence box를 넘지 않는다.
  - (3)  $X_{begin}$ 과  $X_{end}$  사이에 복수의 word box가 존재한다.

(4)  $Y_{begin}$ 과  $Y_{end}$  사이의 좌표점들은 TL와 BL 영역에 포함된다.

(5) 교정 부호의  $X_{begin}$  지점이 word box VCW 좌측에 위치한다.

THEN : (1) 해당 워드박스를 편집의 영역으로 간주한다.  
(2) 다음 해당 규칙을 수행한다.

한편 그림 12의 예제는 사람이 판단할 때 교정 영역이 단어단위가 아닌 전체 문장영역을 포함하고 있음을 직관적으로 알 수 있다. 따라서 그림 12의 편집 영역은 규칙에 의해 추출된 영역 'This'와 'to' 보다는 'This'와 'delete'까지로 영역을 변경할 필요가 있다. 이를 위하여 제안 기법에서는 다음 규칙(48) 및 식 (2)와 같은 임계값(Threshold of Sentence Area)을 만족하는 경우 다음 그림 13처럼 교정 영역을 문장 단위로 보정한다.

규칙 (48) ;

IF : (1) 교정타입이 삭제이다.  
(2) 인식된 교정 부호의  $X_{begin}$ ,  $X_{end}$  길이가 ThSA의 일정기준을 만족한다.

THEN : (1) 편집 영역을 sentence 박스 전체로 지정한다.  
(2) 다음 해당 규칙을 수행한다.

$$ThSA = \frac{\sum Total\ Length\ of\ X_{min}\ And\ X_{max}\ of\ The\ Correction\ Mark}{\sum Total\ Length\ of\ Sentence\ Box\ of\ Document} \quad (2)$$



그림 13 그림 12에 대한 영역보정 결과 화면

#### 4.1.3 문단단위의 교정부호인식

제안 기법에서는 동일 타입의 부호 외에도 각 교정 부호의 타입별 특성에 따라 서로 다른 인식 규칙을 적용한다. 예를 들어 Split/JoinPara와 같은 부호는 주로

문단 단위로 연산이 이루어지기 때문에 이에 맞는 문단 단위의 텍스트 영역 인식 규칙을 적용한다.

그림 14는 문단 단위 편집의 예로써, 입력된 교정 부호는 JoinPara이며 (A) 문단과 (B) 문단을 합치는 역할을 한다. 이 경우 교정 부호의 시작위치는 (A) 문단의 마지막 문장의 끝에 위치하고 부호의 종료 지점은 (B) 문단의 두 번째 문장의 시작 부분에 위치한다. 이 경우 입력의 모호성으로 (A) 문단을 (B) 문단의 어느 문장 앞에 병합하여야 하는지가 불명확하다.

그러나 일반적으로 문단의 병합시에는 인접한 상하 문단을 합치게 되며 문단 병합 교정 부호를 사용하여 한 문단을 다른 문단의 중간에 삽입하는 경우는 극히 드물다. 따라서 그림 14와 같은 예에서는 사용자가 (A) 문단과 (B) 문단을 병합시키는 목적으로 교정 부호를 입력하였다고 판단해야 한다.

이와 같은 사용자의 의도를 파악하기 위해 본 연구에서는 JoinPara와 같은 교정 부호의 경우 문단 단위 편집 명령에 대한 규칙을 적용한다. 그림 14의 예제는 다음 규칙(62)에 의하여 (A)와 (B) 문단을 병합하는 명령을 실행하여 그림 15와 같이 편집 영역을 보정하도록 한다.

규칙 (62) ;

IF : (1) 교정 타입이 JOIN\_PARAGRAPH이다.  
(2)  $X_{begin}$ ,  $Y_{begin}$  및  $X_{end}$ ,  $Y_{end}$ 은 서로 각기 다른 PARAGRAPH BOX 영역 내에 위치한다.  
(3)  $X_{begin}$ ,  $Y_{begin}$  및  $X_{end}$ ,  $Y_{end}$ 가 위치한 PARAGRAPH BOX들은 서로 인접한다.

THEN : (1) PARAGRAPH BOX단위의 JOIN\_PARAGRAPH 연산이다.  
(2) 유효한 PARAGRAPH BOX 영역들을 추출하여 문단을 병합한다.  
(3) 다음 해당 규칙을 수행한다.

#### 4.2 문서 변경에 대한 구조정보 변경규칙

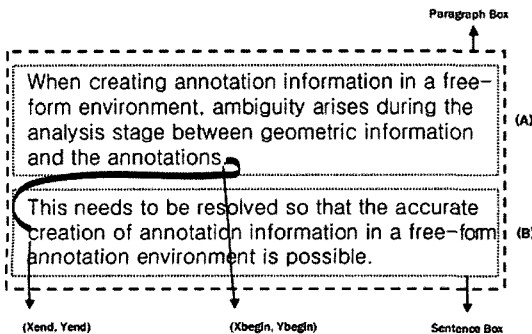


그림 14 문단단위의 영역분석 과정의 예

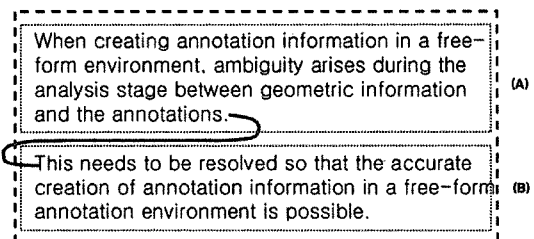


그림 15 그림 14 예제에 대한 영역보정 결과화면

XML과 같은 하이퍼텍스트 문서는 일반 텍스트 문서와 달리 구조를 포함한다. 따라서 이러한 문서를 편집할 때는 텍스트의 삭제 및 추가와 같은 연산으로 발생하는 문서구조 변경이 문서 DTD에 항상 따를 수 있어야 한다. 그러나 교정 시스템에서는 상세한 구조정보의 편집을 목적으로 하기보다는 이미 작성된 문서에 대하여 오류가 있는 곳을 교정 부호로 수정하는 것을 목적으로 하기 때문에 일반 XML 에디터와 유사한 구조 정보 편집기능을 지원하기에는 어려움이 따른다.

본 연구에서는 XML 기반의 교정 시스템에서 문서 편집 중 발생할 수 있는 구조정보의 변경에도 DTD에 올바른 구조를 유지하기 위한 인터페이스를 제안한다. 이를 위하여 제안 기법에서는 명시적인 구조 변경 및 명시적인 구조 변경 기법을 제안한다.

4.2.1 명시적 구조변경 기법

제안된 구조 정보 편집 규칙모델에서는 입력된 교정 부호와 구조정보와의 관계를 파악하여 예측 가능한 모든 구조정보 변경에 대한 경우의 수를 산출한다. 다음으로 추출한 경우의 수가 정의된 DTD에 올바른(validation) 경우만을 선택하여 시스템 내부적으로 적절한 구조 변경이 가능하다면 그 과정을 명시적으로 처리한다.

다음 그림 16에서 (a)는 삭제연산을 입력한 것이고 (b)는 연산 적용에 따른 문서 구조를 명시적으로 변경한 결과를 나타내고 있다. 이때에 그림 16의 문서 구조는 다음과 같이 (title, author, affiliate?)로 정의되었다면 'title'과 'author'는 반드시 발생하여야 하고, 'affiliate'는 발생하지 않아도 된다. 따라서 제안 기법에서는 DTD의 발생 지시자(occurrence indicator)에 따라 그림 16(a)의 (1)(2) 삭제는 텍스트를 삭제하되 그 구조는 그림 16(b)와 같이 남겨둔다. 그림 16(a)의 (3)의 경우는 삭제연산에 따라 노드자체를 삭제하여도 DTD에 위배되지 않으므로 그림 16(b)와 같이 내용 전체를 삭제하도록 한다.

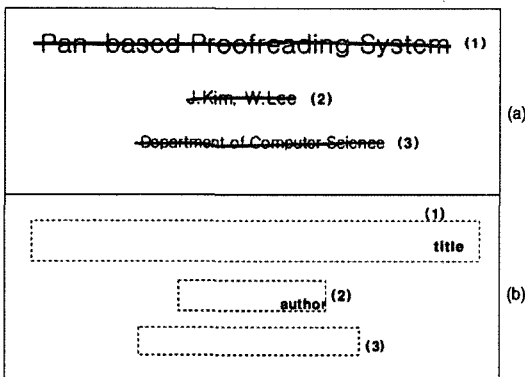


그림 16 명시적 구조 변경의 예

그림 16의 명시적 구조 변경절차는 다음 규칙(92)에 의해서 이루어진다.

규칙 (92) ;

- IF : (1) 교정 타입이 삭제이다.  
 (2) 영역 인식과정을 수행하였다.  
 (3) 영역 대상 내부에 1개이상의 구조정보가 존재한다.  
 (4) 포함된 구조정보는 DTD에서 필수 요소로 정의되어 있다.
- THEN : (1) 영역 내부의 노드를 잔류시킨다.  
 (2) 영역 내부의 텍스트를 삭제한다.  
 (3) 다음 해당 규칙을 수행한다.

4.2.2 명시적 구조변경 기법

교정 부호의 특성이나 구조정보의 위치에 따라 편집 명령으로 변경된 구조정보의 형태가 다수가 될 수 있다. 이러한 경우에 시스템에서 명시적으로 판단하여 가장 적절한 구종 요소를 교정자에게 출력하여 줄 수 있지만 교정자가 원하지 않는 결과가 나올 수 있다. 따라서 본 연구에서는 가능한 결과들을 교정자가 화면으로 확인하면서 원하는 결과를 선택할 수 있는 명시적인 구조 변경기법을 제안한다.

다음 그림 17(a)의 예제에서는 삽입 연산을 수행하는 과정을 설명하고 있다. 이때에 본 문서는 다음 그림 18과 같은 DTD 구조를 포함한다고 가정할 때에 시스템 내부에서는 삽입하려는 구조 'author'가 포함하는 총 8개의 하위 구조 중 하나를 자동으로 선택할 수 없다. 따라서 제안 기법에서는 이러한 경우 그림 17(b)와 같이

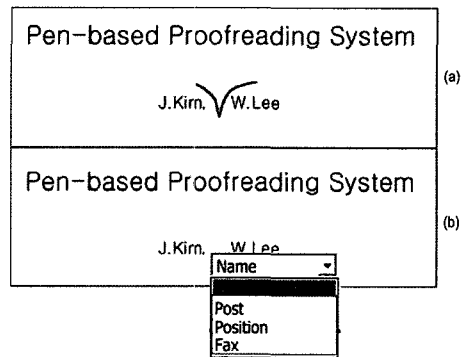


그림 17 명시적 구조 변경의 예

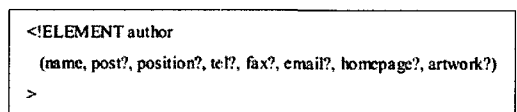


그림 18 그림 17 예제의 DTD 구조



해당 영역에서 발생가능한 하위 구조를 리스트박스 형태로 출력하여 사용자로부터 적절한 선택을 얻게 된다.

위 예제와 같이 명시적인 구조 변경을 위한 규칙모델은 다음 규칙(114)의 내용과 같다.

규칙 (114) :

- IF :
- (1) 교정 타입이 삽입이다.
  - (2) 영역 인식과정을 수행하였다.
  - (3) 영역 대상 내부에 1개이상의 구조정보가 존재한다.
  - (4) 해당 엘리먼트 내부에 복수의 하위 엘리먼트들이 존재한다.
- THEN :
- (1) 해당 엘리먼트의 하위 엘리먼트를 리스트 박스로 출력한다.
  - (2) 선택된 엘리먼트와 텍스트를 문서에 삽입한다.
  - (3) 다음 해당 규칙을 수행한다.

## 5. 교정시스템 구현

본 논문의 XML 기반 교정 시스템 (CPI: Context-based Proofreading Interface)은 Windows 및 Windows CE 플랫폼을 기반으로 한다. 또한 교정문서는 모두 XML 기반의 전자책 표준[18]을 사용하였다. 원본문서 및 교정정보 데이터를 처리하기 위한 파서로는 마이크로소프트사의 XML 파서를 사용하였다. 따라서 XML 파서를 내장하고 있는 Internet Explorer 5.0 이상의 버전이 설치되어 있는 시스템에서는 XML 데이터 처리를 위한 별도의 인스톨 과정은 필요하지 않다. 또한 본 시스템은 Internet Explorer 5.0 이상의 버전에서 ActiveX Control을 설치하여 웹 환경에서 동작하도록 하였다.

### 5.1 시스템 구조

본 연구에서 구현한 시스템의 구성도는 다음 그림 19와 같다.

본 시스템은 크게 4가지의 프로세스 단계를 포함한다. 먼저 교정의 대상이 되는 XML 문서를 서버로부터 호

출하고 이것을 문서처리 모듈을 통하여 필요한 컨텍스트를 추출한다. 컨텍스트에는 텍스트의 읍셋과 같은 비구조정보 및 엘리먼트와 같은 구조정보를 포함한다.

사용자로부터 입력된 교정부호는 시스템의 사용자 인터페이스를 통하여 픽셀정보로 처리되며 그 정보를 인식모듈로 전달한다. 인식모듈에서는 먼저 사용자 인터페이스로부터 전달 받은 물리적 교정부호 정보를 인식하여 적절한 교정부호 타입을 선택한다. 또한 문서처리 모듈에서 추출한 컨텍스트 정보를 이용하여 교정 영역에 대한 최종 영역을 판단한다. 인식모듈에서 처리한 최종 교정 타입 및 영역 정보는 XML 형태로 표현되며 이것을 통합 모듈을 통하여 저장 및 출력하도록 한다. 통합 모듈에서는 생성된 교정부호 정보를 제안된 CPML로 표현하고 이것을 원본 문서와 별도의 형태로 저장한다.

### 5.2 구현결과

본 연구의 교정 시스템은 다음 그림 20(a)와 같이 편의성을 고려한 자유형 드로잉방식과 정확성을 고려한 드래깅(dragging) 방식을 포함하는 2가지 교정부호 입력 인터페이스를 제공한다. 그림 20(a)와 같이 드로잉 방식으로 입력된 교정 마크의 타입은 제스처 인식 기법에 의하여 결정된다. 또한 입력된 교정마크의 최종 영역은 모호성 해결을 위한 컨텍스트 기반 인식 및 보정기법에 근거하여 자동으로 보정되어 화면에 출력된다. 출력된 교정 부호는 선택 옵션에 따라 곧바로 문서의 편집에 사용되거나 입력된 부호를 화면에 출력된 채로 남겨두었다가 메뉴에 의한 사용자의 실행 명령으로 일괄적 혹은 선택적으로 문서를 편집하게 된다. 그림 20(b)의 view 메뉴에서는 'original', 'edit', 그리고 교정연산이 적용된 결과를 확인하기 위한 'revise view'를 제공한다.

다음 그림 21에서는 실제 교정부호 입력 및 교정 결과를 나타내고 있다. 그림 21(a)에서는 총 4개의 교정부호를 입력한 예를 보여주고 있으며 그림 21(b)에서는 각각의 교정 연산이 적용된 결과를 보여주고 있다. 첫 번

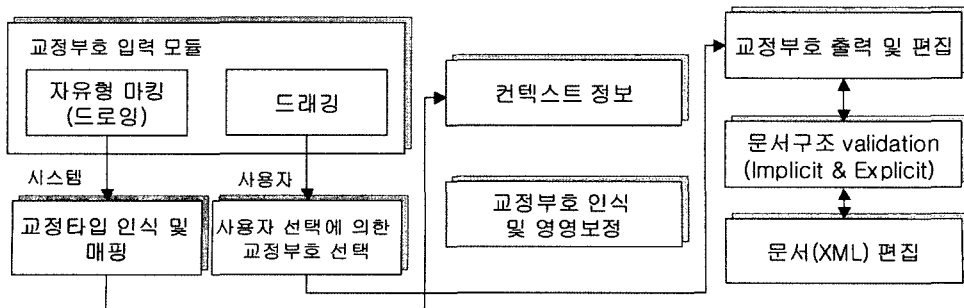
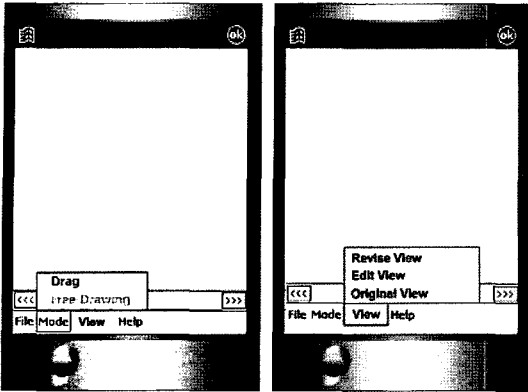
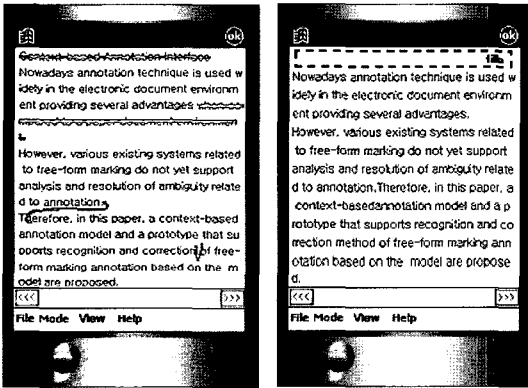


그림 19 교정 시스템 (CPI: Context-based Proofreading Interface)의 구성도



(a) (b)  
그림 20 제안 시스템의 메뉴 인터페이스



(a) (b)  
그림 21 제안시스템의 교정 부호 입력 및 편집결과 화면

제 삭제 교정부호는 DTD에서 반드시 발생하여야 하는 구조를 포함하기 때문에 삭제 후에도 그 영역을 별도로 출력되었다. 나머지 교정 부호들은 시스템 내부의 인식 모듈을 통하여 영역 보정되어 그림 21(b)와 같이 편집 된다.

### 6. 실험결과

본 논문에서는 제안 기법에 대한 성능을 평가하기 위하여 경험적(empirical) 사용자 평가를 수행하였다. 본 실험에서는 사용자가 생성한 교정부호 영역에 대한 인식 정확도를 측정하기 위하여, 제안 기법과 기존 Amaya+PEN[3]에서 사용하는 기법을 적용한 교정 프로토타입을 이용하여 사용자 평가를 수행하였다. 펜 기반의 교정부호 입력에 대한 정확도 측정은 정량적인 측정이 어렵기 때문에, 2가지 프로토타입에서 처리된 교정 결과에 대한 정확도를 사용자가 측정하도록 하였다.

### 6.1 실험 1

본 실험에서는 본 논문에서 제안한 기법과 기존 시스템에서의 교정 영역에 대한 정확도를 측정하기 위하여 자신이 의도한 교정 영역과, 시스템에서 처리한 영역과의 정확도를 개인으로부터 입력받도록 한다. 이를 위하여 본 실험에서는 먼저 30명의 사용자들에게 각각 삭제(10개), 대체(10개), 삽입(10개)을 위한 교정부호를 드로잉 하도록 한다. 프로토타입 내부에서는 입력된 교정 마킹에 대하여 제안 기법 및 기존기법[11]을 적용한 각각의 영역 판별 결과를 추출한다. 모든 실험이 끝난 후에는 시스템에서 추출한 판별 영역을 사용자에게 보여주고, 자신이 의도한 영역과 얼마나 차이가 나는지를 알아보기 위하여 1(lowest accuracy)부터 10(highest accuracy)까지의 스케일(scale)로 구성된 설문지를 이용한 설문 조사를 실시하였다. 본 실험에는 20명의 참여자가 참여하였다. 실험을 위해서는 반복있는 분산분석(ANOVA)를 사용하였다.

다음 그림 22는 입력된 교정영역에 대한 사용자들의 주관적 정확도를 나타내고 있다. 그림 22에 의하면 제안 시스템에서의 교정 영역결과에 대하여 대부분의 사용자들은 보다 정확하다고 평가하였으며, 또한 제안 기법과 기존 기법에 따른 영역정확도간에는 전체적인 유의도가 있음을 알 수 있었다.(F(1,114) = 36, P < 0.05). 또한 각 교정 부호에 대해서도 유의도가 있었음을 알 수 있었다. (삭제, F(1,38) = 12.1, P < 0.05, 대체, F(1,38) = 14.7, P < 0.05, 삽입, F(1,38) = 9.33, P < 0.05).

대부분의 사용자들은 제안 시스템에서 처리된 교정영역을 기본 기법보다 정확하다고 판단하였다. 특히 삭제와 대체 교정 연산의 경우 기존 시스템에서 처리된 영역보다 더 높은 정확도를 보였으며, 이는 복수의 단락에서 처리되는 경우가 많기 때문이다. 즉 제안 기법은 복수의 단락에서 수행되는 교정연산 기호에 보다 효과적인

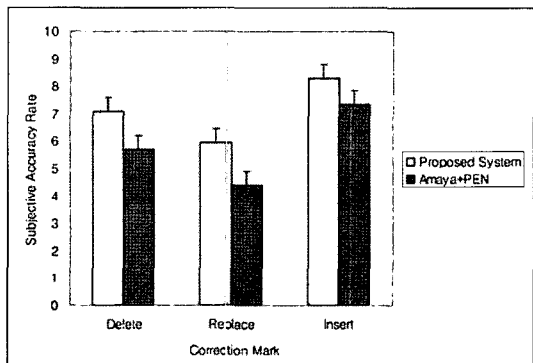


그림 22 실험1에서 교정 영역에 대한 주관적 정확도 (1=lowest accuracy, 10=highest accuracy)

으로 적용됨을 알 수 있었다.

## 6.2 실험 2

본 실험에서는 제안 시스템에서 제공하는 영역보정 기능에 대한 효과를 분석하고자 한다. 이를 위하여 입력된 총 30개의 교정부호에 대하여, 제안 기법을 적용한 것과 그렇지 않은 기존 시스템의 결과를 총 30명에게 설문하도록 한다. 이때에는 1차로 입력된 교정 영역과 각 시스템에서 처리된 결과를 서로 다른 30명의 사용자에게 그 정확도를 평가하도록 한다. 본 실험에서 사용된 설문방법은 앞절의 실험 1과 동일하다.

그림 23은 실험 2에 대한 분산분석 결과이다. 그림 23에 의하면 제안된 영역 보정기능이 주관적 만족도에 유의함을 알 수 있다( $F(1,114) = 23.6, P < 0.05$ ). 또한 삽입을 제외한 삭제, 대체 연산에 대해서도 기존 시스템보다 높은 만족도를 나타내었다(삭제,  $F(1,38) = 6.98, P < 0.05$ , 대체,  $F(1,38) = 14.78, P < 0.05$ , 삽입,  $F(1,38) = 3.22, P > 0.05$ ).

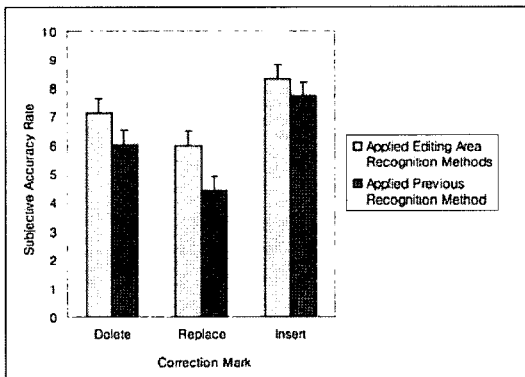


그림 23 실험 2에서 교정 영역에 대한 주관적 정확도 (1=lowest accuracy,10=highest accuracy)

실험 2의 결과 본 논문에서 제안한 교정환경에서의 영역 보정기능이 다중사용자들에게 보다 높은 이해도를 제공하는 것을 알 수 있었다. 특히 삭제와 대체 연산에서는 기존 시스템보다 대체로 높은 효과를 보였으며 이는 삭제 및 대체 연산시 발생하는 모호성을 감소시킬 수 있었던 것으로 분석된다. 그러나 삽입 연산의 경우 교정 부호 특성상 큰 모호성을 유발하지 않기 때문에 제안 기법에 대하여 큰 영향을 미치지 못한 것으로 판단된다.

## 7. 결론

본 연구에서는 XML 기반의 전자문서 환경에서 정확한 펜 기반의 교정 마킹 입력을 가능케하고 문서 교정에 따른 구조 변경시에도 DTD에 따르는 올바르게 유

지하기 위한 교정 인터페이스를 제안하였고 이를 포함하는 교정 시스템을 구현하였다.

제안 인터페이스에서는 입력된 마킹을 XML 형태의 구조정보로 표현하며 이를 위하여 Context-based Proofreading Markup Language(CPML)를 정의하였다. 또한 펜기반의 교정 부호와 편집문서 컨텍스트간의 속성을 파악하여 펜기반의 교정 부호와 편집 대상 텍스트간의 영역 매칭과정에서 발생하는 모호성을 해결하고자 하였다. 특히 텍스트의 영역을 다양한 기준선으로 세분화하고 입력된 교정부호의 특성 및 입력영역에 따라 서로 다른 규칙을 적용하여 텍스트를 글자, 단어, 문장, 절 등과 같은 컨텍스트 영역으로 인식하도록 하였다. 제안 시스템은 교정부호의 입력에 따른 문서 내용 및 구조의 변화가 발생할 경우 시스템 내부에서 자동으로 문서 구조를 유지하기 위한 묵시적인 구조변경 기법과 사용자와의 인터랙션을 통하여 문서 구조를 유지하는 명시적인 문서 구조변경 방법을 제공하였다.

그 결과 본 논문에서 생성된 교정마킹 입력결과는 기존 교정시스템에 비하여 보다 정확한 영역정보를 포함할 수 있으며, 교정 부호 입력에 따른 구조문서 변경에도 원본문서의 DTD에 위배되지 않는 문서 구조를 유지할 수 있다. 본 연구의 결과는 XML 문서환경의 annotation 시스템 및 레포팅도구(reporting tool), 그리고 저작도구(authoring tool) 등과 같은 도메인에 효과적으로 적용 가능하다.

향후 본 연구진은 문서 교정을 위한 다중 사용자 기반의 협업작업 및 문서 변경 탐지를 통한 문서 버전관리와 관련한 연구를 수행할 예정이다.

## 참고 문헌

- [1] Ackerman, S.S. & Turechek, W.W., "The risks and rewards of online editing," IEEE Transactions on Professional Communication, Vol. 31-3, pp. 122-123, 1988.
- [2] Andre, J. & Helene R., "Paper-less Editing and Proofreading of Electronic Documents," in Proceedings of EuroTex'99, 1999.
- [3] Farkas, D.K. & Poltrock, S.E., "Online editing, mark-up models, and the workplace lives of editors and writers," IEEE Transactions on Professional Communication, Vol. 38-2, pp. 110-117, 1995.
- [4] Richy, H. & Lorette, G., "On-line correction of Web pages," Document Analysis and Recognition. ICDAR '99. Proceedings of the Fifth International Conference, pp. 581-584, 1999.
- [5] Goldberg, D. & Goodisman, A., "Stylus user interfaces for manipulating text," in Proceedings of the fourth annual ACM symposium on User

interface software and technology, South Carolina, ACM Press, NY, pp. 127-135, 1995.

- [6] Alexander, G. A., "Applied Technology: A Closer Look at PenEdit," Seybold Report on Desktop Publishing, Vol. 8-3, <http://www.seyboldreports.com/SRDP/0dp8/D0803000.HTM>, 1996.
- [7] Hardock, G, Kurtenbach, G. & W. Buxton, "A Marking based Interface for Collaborative Writing," in Proceedings of the sixth annual ACM symposium on User interface software and technology, Atlanta, ACM Press, NY, pp. 259-266, 1993.
- [8] Christine, A. & Randall, D., "Resolving Ambiguities to Create a Natural Sketch Based Interface," in Proceedings of IJCAI-2001, 2001.
- [9] Gross, M. D. & Ellen, Y. D., "Ambiguous Intentions: A Paper-like Interface for Creative Design," in Proceedings of the 9th annual ACM symposium on User interface software and technology, Seattle, Washington, pp. 183-192, 1996.
- [10] Extensible Markup Language (XML), World Wide Web Consortium, <http://www.w3c.org/TR/1998/REC-xml-19980210>, 1988.
- [11] Amaya, <http://www.w3.org/Amaya>, 1997.
- [12] Ogata, H., Yano, Y. & Wakita, R., "CCML: Exchanging Marked-up Documents in a Networked Writing Classroom," Computer Assisted Language Learning, Vol. 11-2, pp. 201-214, 1998.
- [13] Meyer, A., "Pen Computing: A Technology Overview and a Vision," in Proceedings of ACM SIGCHI Bulletin, ACM Press, pp. 46-90, 1995.
- [14] XML Linking Language (XLink) Version 1.0, World Wide Web Consortium, <http://www.w3.org/TR/2000/REC-xlink-20010627>, 2001.
- [15] University of Chicago Press, The Chicago Manual of Style, 14th edition, Chicago: University of Chicago Press, 1993.
- [16] Rubine, D., "Specifying Gestures by Example," Computer Graphics, Vol. 25-4, pp. 329-337, 1991.
- [17] Won-Sung Sohn, Jae-Kyung Kim, Seung-Kyu Ko, Soon-Bum Lim, Yoon-Chul Choy, "Context-based Proofreading of Structured Documents," ICSSA 2004, Assisi, Italy, May 2004, LNCS 3046, pp. 743-753, Springer.
- [18] Sohn, W. S., et al., "Standardization of eBook documents in the Korean Industry," Computer Standards & Interfaces, Vol. 24-1, 45-60, 2002.
- [19] Miller, R. C. & Myers, B. A., "Multiple Selections in Smart Text Editing," in Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI 2002), San Francisco, pp. 103-110, 2002.



손원성

1998년 동국대학교 컴퓨터공학(학사)  
2000년 동국대학교 컴퓨터공학(석사)  
2004년 연세대학교 컴퓨터과학과(박사)  
2004년~현재 Carnegie Mellon University, Computational Design Lab, Associate Researcher. 관심분야는 웹 문서처리, Annotation, 스마트 홈, 유비쿼터스, XML



김재경

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



최윤철

1973년 서울대학교(학사). 1975년 Univ. of Pittsburgh(석사). 1976년 Univ. of California, Berkeley(석사). 1979년 Univ. of California, Berkeley(박사). 1979년~1982년 Lockheed 사 및 Rockwell International 사 연구원. 1990년~1991년 University of Massachusetts 교환교수. 2002년~2003년 일본 게이오대학 교환교수. 1984년~현재 연세대학교 컴퓨터과학과 교수. 관심분야는 멀티미디어와 웹, 멀티미디어 문서처리, 3D 사용자 인터페이스, 아바타 인터페이스, 컴퓨터 그래픽스, eLearning 및 Cyber Class



임순범

1982년 서울대학교 계산통계학과(학사)  
1983년 한국과학기술원 전산학과(석사)  
1992년 한국과학기술원 전산학과(박사)  
1989년~1992년 (주)휴먼컴퓨터 창업/연구소장. 1992년~1997년 (주)삼보컴퓨터 프린터개발부 부장. 1997년~2001년 건국대학교 컴퓨터과학과 교수. 2001년~현재 숙명여자대학교 멀티미디어과학과 교수. 관심분야는 컴퓨터 그래픽스, 모바일 멀티미디어 응용, 전자출판(폰트, XML, 전자책, e-Learning)



김우성

1993년 서강대학교(박사). 1984년~1987년 한국전자통신연구소 연구원. 1999년~2000년 미국 Univ. of Washington 방문교수. 1987년~현재 호서대학교 컴퓨터공학부 교수. 2002년~현재 호서대학교 첨단정보기술대학원장. 관심분야는 영상처리, 지식관리, 정보검색