

스타일을 설정할 수 있는 웹 기반 문서 편집 환경

Web Based Document Editing Environment with User-Defined Styles

황준형* · 윤정한* · 최석우** · 한태숙*

카이스트 전산학과* · KT 네트워크 연구소**

envia@envia.pe.kr, jeonghan.yun@gmail.com, choi.seokwoo@kt.com, han@cs.kaist.ac.kr

요 약

문서의 특정 구조를 표시할 때 쓰이는 표현 방법을 스타일이라고 한다. 웹 문서에서 스타일을 사용하면 문서에서 쓰이는 표현 방법들을 한 곳에 모아서 문서의 내용과 표현 방법을 분리하는 것이 가능하다. 이를 통해 문서에서 필요한 정보를 추출하거나 유지 보수하는 것을 편리하게 할 수 있다. 웹에 있는 다양하고 시간에 따라 변화하는 정보를 고정된 스타일만으로 나타내는 것에는 한계가 있으며, 사용자의 필요에 따라 스타일을 바꾸어줄 필요가 있다. 기존에는 원하는 형태의 스타일을 사용하기 위해서는 웹 사이트에 있는 스타일 관련 파일을 직접 편집해 주어야 했다. 따라서 파일을 수정할 권한이 없거나 웹 프로그래밍에 대한 지식이 부족한 경우 자신이 원하는 스타일을 지정해서 사용하는 것이 쉽지 않았다. 따라서 제공되는 스타일이 사용자가 원하는 것과 다를 경우 스타일을 사용했을 때의 장점을 얻는데 어려움이 있었다. 본 논문에서는 기본으로 주어지는 스타일들을 바탕으로 사용자들이 자신의 필요에 맞는 스타일을 설정하는 것을 지원하는 웹 기반 편집 환경을 제안한다. 제안한 편집 환경을 이용하면 웹 프로그래밍에 대한 특별한 지식이 없어도 스타일을 쉽게 작성하고 적용할 수 있다. 이를 이용하면 사용자가 스타일을 유연하게 변경할 수 있으므로 스타일을 사용했을 때의 장점을 충분히 얻을 수 있다. 스타일 정보가 웹에 저장되기 때문에 스타일 정보를 여러 곳에서 사용하거나 공유하는 것도 가능하다. 우리는 제안한 편집 환경을 위키 사이트에 적용해 보았다. 편집 환경을 구현한 방법과 스타일의 활용 사례도 함께 소개한다.

1. 서 론

인터넷이 보급되면서 점점 더 많은 정보들이 웹을 통해 제공되고 있으며, 웹에서 정보를 제공하는 사람들의 수도 늘어나고 있다. 따라서 웹 사이트를 유지 보수하는 일이 점점 더 중요해지고 있으며, 체계적인 방법으로 작성한 웹 문서의

가치가 주목을 받고 있다.

스타일은 문서의 특정 구조를 표시할 때 쓰이는 표현 방법을 말한다. 스타일을 사용하면 웹 사이트의 유지 보수를 편리하게 할 수 있다. 스타일을 이용해서 웹 문서에서 쓰이는 표현 방법들을 한 곳에 모을 수 있는데, 표현 방법들을 한 곳에 모으면 문서의 내용과 그 표현 방법을 분리

하는 효과를 얻을 수 있다. 문서의 내용과 표현 방법을 분리하면 한 쪽은 그대로 둔 채 다른 한 쪽만 독립적으로 바꾸는 것이 가능하며, 한 쪽의 정보만을 따로 이용하는 것도 가능하다. 따라서 사이트의 구조를 바꾸거나 필요한 정보를 추출하는 일이 간단해진다.

스타일을 이용해서 최대의 효과를 얻기 위해서는 실제 문서의 내용에 맞는 스타일을 제공해야 한다. 제공되는 스타일이 실제 문서의 내용과 맞지 않으면 스타일을 사용할 수 없거나 사용하더라도 혼란을 초래하게 된다. 웹 사이트의 내용은 다양하고 시간에 따라 변화하기 때문에 고정된 스타일만으로 나타내는 데에는 한계가 있다. 사이트에 맞는 스타일을 장기적으로 제공하기 위해서는 문서의 내용에 맞추어 스타일을 변경하는 것이 가능해야 한다.

기존에는 원하는 형태의 스타일을 사용하기 위해서 웹 사이트에 있는 스타일 관련 파일을 직접 편집해 주어야 했다. 파일을 수정할 권한이 있는 웹 사이트 운영자는 사이트에서 사용할 스타일을 쉽게 설정하여 사용할 수 있지만, 권한이 없고 웹 프로그래밍에 대한 지식도 부족한 사이트 사용자의 경우 자신이 원하는 스타일을 설정하여 사용하는 데 어려움을 겪게 된다. 사이트에서 제공하는 스타일이 사용자가 원하는 것과 다를 경우 이를 수정하는 것이 힘들기 때문에 스타일을 이용해서 얻을 수 있는 장점을 경험할 수 없게 된다.

본 논문에서는 사용자들이 자신의 필요에 맞는 스타일을 직접 설정하는 것을 지원하는 웹 기반 편집 환경을 제안한다. 제안한 편집 환경을 이용하면 웹 프로그래밍에 대해 특별한 지식이 없는 사용자도 스타일을 쉽게 설정하고 적용할 수 있으며, 이를 통해 스타일을 사용했을 때 얻을 수 있는 이점을 온전히 누릴 수 있다. 스타일 정보가 웹에 저장되기 때문에 스타일 정보를 여러 곳에서 사용하거나 공유하는 것도 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 소개하고 그들의 문제점과 한계를 지적한다. 3장에서는 사용자가 스타일을 설정할 수 있는 편집 환경과 그 구현 방법을 제안한다. 4장에서는 편집 환경을 위키 사이트에 적용한 사례를 소개하며, 5장에서는 스타일의 활용 사례를 소개한다. 6장에서는 결론을 맺고 앞으로의 연구 방향에 대해 생각해 본다.

2. 기존 방법과 문제점

웹 문서의 내용과 표현 방법을 스타일을 이용해서 분리하려고 할 경우 각각에 해당하는 부분을 독립적으로 작성하게 된다. 보통 내용에 해당하는 부분은 HTML이나 XML로 작성하고 표현 방법에 해당하는 부분은 CSS로 작성한다. 내용에 해당하는 부분을 XML로 작성하고 표현 방법에 해당하는 부분을 XSL로 작성한 뒤 이들을 HTML로 변환하는 경우도 있다. 본 논문에서는 첫 번째 방법을 사용한 경우에 대해서만 다루지만, 본 논문의 내용은 두 번째 방법을 사용한 경우에도 유사하게 적용할 수 있다.

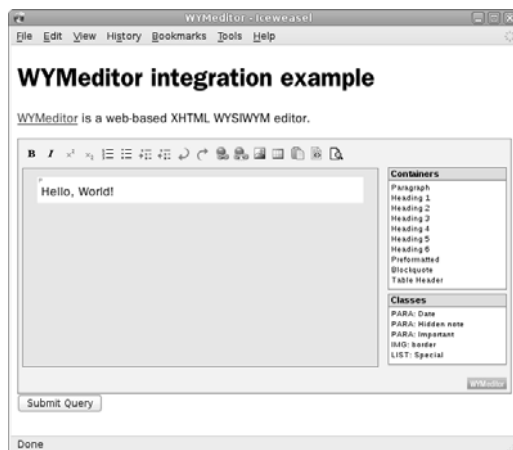
스타일을 이용해서 문서의 내용과 표현 방법을 분리하는 방법은 기존 문서 편집기에서 널리 쓰이고 있다. LaTeX 편집기인 LyX[1]에서는 이러한 개념을 WYSIWYM(What You See Is What You Mean)이라는 표현으로 나타내고 있다. 하지만 기존 웹 편집기들의 대부분은 문서 편집기와 같이 스타일을 설정하는 것을 지원하지 않는다.

웹 사이트에서 쓰이는 스타일을 바꾸는 가장 간단한 방법은 웹 사이트에 있는 CSS 파일을 직접 수정하는 방법이다. 하지만 CSS 파일을 직접 수정하기 위해서는 웹 사이트에 대한 높은 수준의 권한이 필요하거나 상당한 웹 프로그래밍 지식이 필요한 경우가 많아 일반 사용자가 이용하기에는 어려움이 있다.

일부 웹 편집기는 스타일을 편집할 수 있는 기

능을 제공한다. Amaya[2]는 사용자가 HTML 파일의 내용과 함께 CSS의 내용을 바꾸는 것을 허용함으로써 스타일 편집을 지원한다[3]. 하지만 Amaya는 웹 기반이 아니므로 Amaya를 이용해서 웹 문서를 수정하기 위해서는 Amaya를 먼저 설치해야 하는 번거로움이 있다. 한편 Amaya는 수정한 파일을 PUT 방식(method)으로 업로드하기 때문에 Amaya를 이용해서 웹 문서의 내용을 편집하려면 웹 서버에서 PUT 방식을 지원해야 한다. 현재 대부분의 웹 서버들은 보안상의 이유로 PUT 방식을 지원하지 않고 있으므로 이 방법을 이용해서 일반적인 웹 문서의 내용을 편집하는 데에는 한계가 있다.

스타일을 지원하는 웹 기반 편집기도 여럿 존재한다. 그림 1은 WYMeditor[4]의 화면이다. 사용자는 화면 오른쪽에 있는 목록에서 스타일을 선택할 수 있다. 화면 왼쪽의 편집 창에 있는 문서 내용은 해당하는 스타일을 나타내는 상자 안에 표시된다. 이와 같은 편집기는 이미 정해진 스타일을 사용하는 것은 지원하지만, 스타일을 새로 설정하거나 변경하는 것은 지원하지 않는다. 따라서 사이트에서 지원하는 스타일이 사용자가 원하는 것과 다를 경우 스타일을 충분히 활용하는 데 어려움이 있다.



[그림 1] WYMeditor의 화면

3. 스타일을 설정할 수 있는 편집 환경

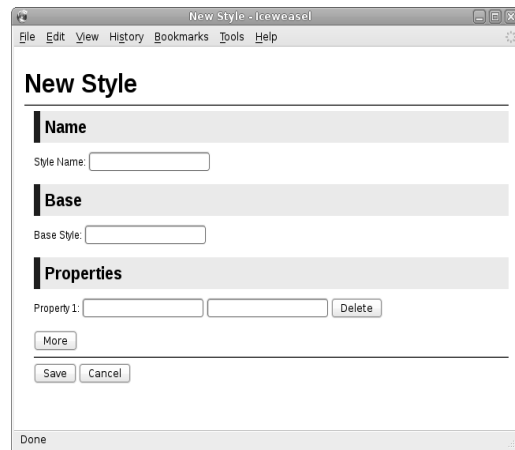
3.1. 구현 내용

우리는 웹 환경에서 사용자가 자신만의 스타일을 설정하여 사용할 수 있는 편집 환경을 개발하였다. 이를 위해 먼저 스타일을 관리하는 기능들을 개발하였다. 그 다음 기존의 편집기를 수정하여 스타일을 적용할 수 있도록 하였는데, 이것에 대해서는 4장에서 자세히 다루기로 한다.

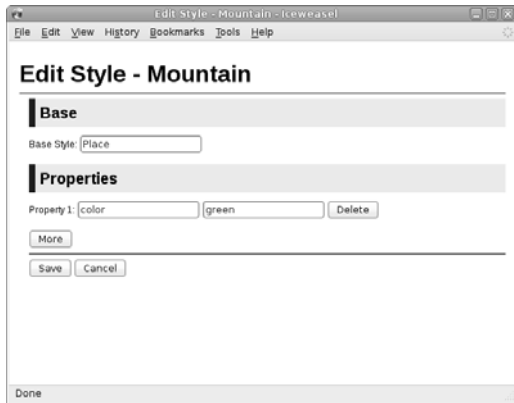
[그림 2]에서 [그림 5]까지는 스타일을 관리하는 기능에 해당하는 화면을 나타내고 있다. 구체적으로 다음 4가지 기능을 구현하였다.

- 스타일을 추가하는 기능 [그림 2]
- 스타일을 변경하는 기능 [그림 3]
- 스타일을 내보내는 기능 [그림 4]
- 스타일을 불러오는 기능 [그림 5]

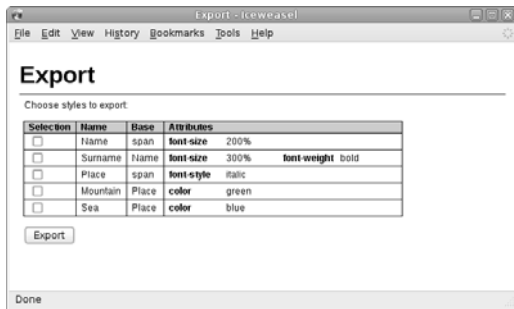
스타일을 추가하고 변경하는 화면에서는 스타일의 이름과 바탕 스타일, 스타일 속성을 지정할 수 있다. 스타일을 내보내는 화면에서는 내보낼 스타일을 선택할 수 있고, 스타일을 가져오는 화면에서는 스타일을 가져올 곳을 선택할 수 있다.



[그림 2] 스타일 추가 화면



[그림 3] 스타일 변경 화면



[그림 4] 스타일 내보내기 화면



[그림 5] 스타일 가져오기 화면

3.2. 스타일 구조

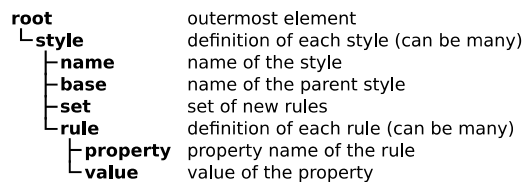
스타일을 설정할 때에는 스타일 이름과 바탕 스타일, 스타일 속성을 지정한다. 스타일 이름은 스타일을 구분하기 위해 사용하며, HTML과 CSS 파일을 만들 때 클래스(class) 이름으로도 쓰인다. 바탕 스타일은 기존 스타일 이름을 사용할 수도 있고, HTML 태그 이름을 사용할 수도 있다. 바탕 스타일에 HTML 태그 이름을 사용할

경우 실제 HTML 문서를 만들 때 해당 태그를 사용하게 된다. 바탕 스타일을 지정하지 않을 경우 기본 스타일을 사용하는데, 기본 스타일은 블록 요소의 경우 div이고 인라인 요소의 경우 span이다. 스타일 속성은 CSS 규칙에서 쓰이는 속성에 대응하며, 속성 이름과 값을 지정할 수 있다. 아무 속성도 지정하지 않을 경우 부모 스타일에 해당하는 표현 방법만을 사용하게 된다.

스타일을 적용할 때에는 모든 부모 스타일에 해당하는 표현 방법과 스타일 속성을 이용하여 지정한 표현 방법을 함께 적용한다. 부모 스타일과 자식 스타일의 표현 방법이 충돌할 경우에는 자식 스타일의 표현 방법을 적용한다.

3.3. 스타일 저장

스타일을 저장하는 것은 XML 형식을 사용하였다. 텍스트 파일 형식이기 때문에 직접 편집하기도 편하고, 많은 언어들 XML을 지원하기 때문이다. PukiWiki를 작성할 때 사용한 PHP의 경우 PHP 5부터 XML을 기본으로 지원한다.



[그림 6] 스타일 파일의 구조

```
<?xml version="1.0"?>
<root>
  <style>
    <name>Name</name>
    <base>span</base>
    <set>
      <rule>
        <property>font-weight</property>
        <value>bold</value>
      </rule>
    </set>
  </style>
</root>
```

[그림 7] 스타일 파일의 예

스타일 파일의 구조는 [그림 6]과 같다. 가장 바깥에 root 요소(element)가 있고, 그 안에 style 요소들이 있다. style 요소 안에는 name 요소 하나, base 요소 하나, set 요소 하나가 있다. set 요소 안에는 rule 요소가 여럿 있는데, 각각에는 property 요소 하나, value 요소 하나가 있다.

스타일 파일에서 root 요소는 가장 바깥에 있는 요소이며, style은 각각의 스타일에 해당하고, name은 스타일의 이름, base는 스타일의 부모 스타일, set은 새로 추가되는 규칙들에 해당한다. rule 요소는 각각의 규칙들에 해당하고, property는 규칙이 적용되는 속성의 이름에, value는 속성의 값에 해당한다.

[그림 7]은 표현 방법으로 굵은 글꼴을 사용하도록 하는 Name 스타일이 있는 스타일 파일의 내용을 나타낸 것이다.

4. 위키에서 스타일 사용하기

우리는 스타일을 편집하는 기능을 위키 소프트웨어에 적용하였다. 위키는 사용자가 위키 문법에 맞추어 입력한 내용을 HTML로 변환하여 표시한다. 위키는 구조가 단순하여 새로운 기능을 추가하기가 편리하다. 위키 중에서는 소스 코드가 비교적 간단한 PukiWiki[5]를 선택하였다. 이를 통해 제한한 편집 환경을 기존 웹 사이트에 적용하는 것이 가능하다는 것을 보였다.

4.1. 스타일 관리

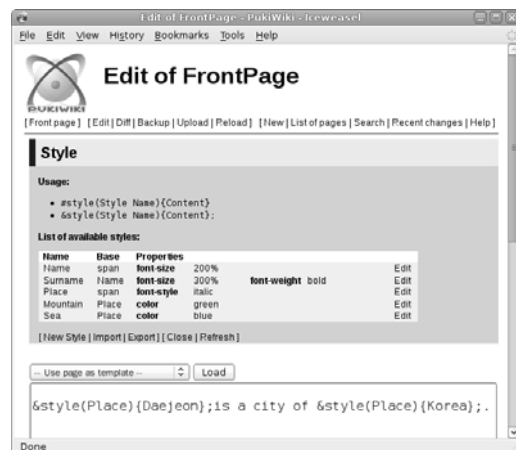
스타일을 추가하고, 변경하고, 내보내고, 불러오는 부분을 편집 창에 출력하기 위해 다음과 같은 방법들을 사용할 수 있다.

1. 전체 페이지를 새로 불러오는 방법
2. 팝업 윈도우를 사용하는 방법
3. AJAX를 사용하는 방법

4. IFRAME(내부 프레임)을 사용하는 방법

전체 페이지를 새로 불러오는 방법은 웹 표준에 가장 잘 맞는 방법이지만 반응 속도가 느리다는 단점이 있다. 팝업 윈도우를 사용하는 방법의 경우 팝업 윈도우에서 스타일을 관리한 다음 그 결과를 편집 창에 반영하는 방법인데 역시 느리고 사용하기 불편하다는 단점이 있다. AJAX를 이용할 경우 동적으로 자료를 받아와서 페이지를 고치게 되는데 반응성이 좋아서 널리 쓰이는 방법이지만 웹 표준이 아니라는 단점이 있다. IFRAME을 사용할 경우 편집기 화면의 일부를 IFRAME으로 지정해서 필요할 때마다 해당하는 페이지의 내용을 표시한다. IFRAME은 표준에 포함되어 있는 방법이고 반응 속도도 나쁘지는 않지만 한 페이지의 내용이 여러 파일로 나뉘게 되면서 사용성이 떨어진다는 단점이 있다.

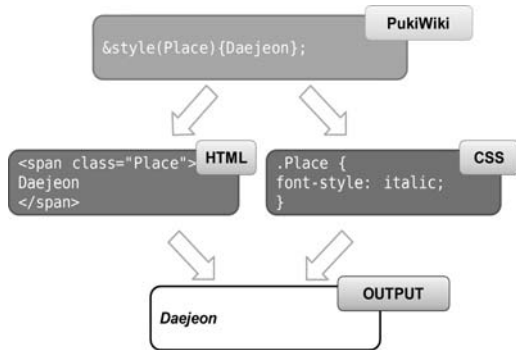
우리는 비교적 구현이 간단한 IFRAME을 사용하여 스타일을 관리하는 화면을 편집 창에 표시하였으며, 다른 방법을 사용하더라도 기능에 큰 차이는 없다. 다른 웹 사이트에서도 비슷한 방법으로 스타일 관리 화면을 추가하는 것이 가능하다. [그림 8]은 스타일 기능을 추가한 PukiWiki의 화면을 나타내고 있다.



[그림 8] 스타일 관리 화면이 추가된 편집창

4.2. 스타일 적용

PukiWiki에서 문서의 내용에 스타일을 적용할 수 있도록 하기 위해 style 플러그인을 만들어 PukiWiki에 포함시켰다. style 플러그인을 적용한 내용은 HTML을 생성할 때 스타일을 적용할 수 있는 형태로 출력되며, CSS를 생성할 때 해당하는 표현 방법이 포함된다. 이 과정을 그림으로 나타내면 [그림 9]와 같다.



[그림 9] 위키에서 HTML과 CSS의 생성 과정

HTML을 생성하는 부분은 PukiWiki의 플러그인 구조를 이용하여 구현하였다. 스타일을 적용할 내용을 스타일에 해당하는 HTML 태그로 둘러싼 다음 해당하는 클래스 속성을 추가하였다. 이 때 HTML 태그는 바탕 스타일을 이용하여 정하고, 클래스 속성은 스타일 이름을 이용하여 정한다. Daejeon라는 단어에 span 태그를 바탕으로 하는 Place 스타일을 적용할 경우 Daejeon과 같이 출력하게 된다.

CSS를 생성하는 부분은 PukiWiki의 플러그인 구조로는 변경할 수 없어서 CSS를 출력하는 코드를 직접 수정하였다. 수정한 코드는 스타일에 해당하는 규칙들을 CSS 파일에 추가한다. 이 때 규칙들은 스타일의 속성과 바탕 스타일의 속성들을 이용해서 정한다. 규칙들을 적용할 대상

찾을 때 쓰이는 선택자(selector)는 클래스 속성 이름을 이용하여 만든다. 예를 들어 위에서 언급한 Place 스타일에 해당하는 부분을 기울여서 표현해야 할 경우 .Place{font-style: italic;}을 추가하게 된다.

이와 같이 HTML과 CSS를 출력하는 방법으로 스타일이 의도한 것과 같이 적용되는 것을 확인할 수 있었다. PukiWiki에는 프린터에 맞도록 출력하는 기능이 있는데, 이 기능을 사용할 때에도 스타일은 의도한 바와 같이 적용되었다.

다른 웹 사이트에도 비슷한 방법으로 스타일을 적용하는 것이 가능하다. 웹 사이트에서 문서를 저장하는 방법을 스타일도 저장할 수 있도록 확장한 다음, HTML을 출력하는 부분과 CSS를 출력하는 부분을 각각 수정해 주면 된다.

5. 스타일의 활용 사례

스타일은 문서의 내용에 대한 표현 방법을 지정하기 위해 쓰이지만 웹의 특성을 활용하면 다른 용도로도 쓸 수 있다.

스타일은 일종의 메타데이터로 활용할 수 있다. 스타일의 이름을 이용해서 스타일을 적용한 내용이 어떠한 종류의 내용인지를 알아내는 것이 가능하다. 예를 들어 스타일의 이름이 Place라면 스타일을 적용한 표현이 특정 장소에 해당할 것이라는 추론을 할 수 있다. 사용자나 작성 시간에 따라 다른 스타일을 적용할 경우 문서의 해당 부분을 누가 언제 작성했는지를 쉽게 알 수 있도록 표시하는 것이 가능하다.

마이크로포맷(microformat)[6]은 웹 문서의 정보에 대한 메타정보들을 현재 존재하는 웹 표준들을 이용해서 제공하기 위한 형식이다. 마이크로포맷은 메타정보를 나타내기 위해 HTML 태그의 속성을 이용하고 있으며, 가장 많이 쓰이는 속성은 div 태그와 span 태그의 클래스 속성이다. 제안한 편집 환경에서도 스타일을 나타내기 위

해 클래스 속성을 사용하므로 스타일을 이용하여
면 마이크로포맷을 자연스럽게 나타낼 수 있다.

한 사람이 글을 쓰면서 사용하는 스타일은 일
정한 경우가 많으므로 하나의 스타일을 여러 사
이트에서 사용한다면 편리할 것이며, 개인의 개
성을 나타내는 데에도 도움이 될 것이다. 그리고
여러 사람이 하나의 스타일을 함께 사용한다면
통일된 효과를 주는 것이 가능할 것이며, 스타일
을 일관성 있게 변경하는 일도 쉬워질 것이다.

6. 결론과 향후 연구 과제

우리는 사용자가 스타일을 설정할 수 있는 웹
기반 편집 환경을 구현하였으며, 이를 위키 사이
트에 적용하였다. 사용자는 웹 프로그래밍에 대
한 특별한 지식이나 웹 사이트에 대한 특별한 권
한 없이도 필요에 따라 스타일을 설정할 수 있으
며, 스타일을 이용하여 웹 문서의 내용과 표현
방법을 분리하여 수정할 수 있다. 웹 사이트에서
사용하는 스타일을 유연하게 변경할 수 있으므로
정보의 활용과 유지 보수가 더욱 편리해진다.

본 연구에서는 기능의 구현에 중점을 두었기
때문에 전반적인 성능은 고려하지 않았다. 현재
제안한 편집 환경을 사용하기 위해서는 HTML
태그와 CSS 속성에 대한 기본적인 지식이 필요
하다. 제안한 편집 환경이 널리 쓰이기 위해서는
이러한 지식 없이 이용할 수 있도록 사용자 인터
페이스를 개선할 필요가 있다. 이 때 어떠한 편
집 환경이 사용하기 편리한지 정량적으로 평가
할 수 있는 방법을 고안하여 적용한다면 도움이
될 것이다. 스타일을 자동으로 생성하거나 스타
일의 일부 속성을 자동으로 생성하는 방법을 개

발하여 적용하면 사용하기가 더욱 편리해질 것
으로 기대한다.

참 고 문 헌

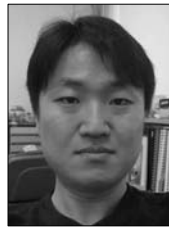
- [1] LyX Team. LyX. <http://www.lyx.org/>.
- [2] W3C and WAM Project. Amaya.
<http://www.w3.org/Amaya/>.
- [3] Vincent Quint and Irne Vatton. Editing
with Styles. In *Proceedings of the ACM Symposium
on Document Engineering*, pages 121-140, 2007.
- [4] WYMteam. WYMeditor.
<http://www.wymeditor.org/>.
- [5] PukiWiki Developers Team. PukiWiki.
<http://pukiwiki.sourceforge.jp/>.
- [6] Microformats Community. Microformats.
<http://microformats.org/>.

황 준 형



2002년~2009년 한국과학기술원
전산학과(학사)
2009년~현재 카이스트 전산학과
석사과정
관심분야: 프로그래밍 언어,
컴파일러.

윤 정 한



1997년~2001년 한국과학기술원
전산학과(학사)
2001년~2003년 한국과학기술원
전자전산학과 전산학전공
(석사)
2003년~현재 카이스트 전산학과

박사과정

관심분야: 프로그래밍 언어, 컴파일러



최 석 우

1994년~1998년 한국과학기술원

전산학과(학사)

1998년~2000년 한국과학기술원

전자전산학과 전산학전공

(석사)

2001년~2003년 카이스트 전산학과(박사)

2009년~현재 KT 네트워크 연구소 대리

관심분야: 네트워크 보안, 프로그램 분석



한 태 속

1972년~1976년 서울대학교 전자

공학과(학사)

1976년~1978년 한국과학기술원

전산학과(석사)

1990년~1995년 University of

North Carolina at Chapel

Hill(박사)

1996년~현재 카이스트 전산학과 부교수

관심분야: 프로그래밍 언어론, 함수형 언어
