

클라우드 컴퓨팅 환경에서의 문서 편집 인터페이스에 대한 제안

안성옥*, 김수균**

요약

클라우드 컴퓨팅 환경에서 여러 사용자들이 쉽게 사용할 수 있는 문서 편집기 인터페이스에 대해 제안한다. 현재 문서 편집기는 기능이 복잡해지고, 동작이 무거워지면서 사용자는 시스템 자원의 부족으로 인한 추가비용이 발생하게 되었다. 이러한 추가비용의 발생과 함께 사용하지 않은 기능으로 인해 시스템 전체가 느려지는 현상을 자주 겪게 된다. 본 논문에서는 문서 편집기의 장점은 부각 시키고 불필요한 추가기능은 배제하여 저 사양의 시스템 및 클라우드 컴퓨팅 환경에서도 무리 없이 구동이 가능한 문서 편집기를 객체지향언어와 API를 이용하여 구현하는 것을 목표로 한다.

Suggestion of Text Editor Interface in Cloud Computing Environment

Sung-Og An*, Soo-Kyun Kim**

ABSTRACT

This paper proposes a user friendly text editing interface that can be used in a cloud computing systems. The problem with the current text editor is that its functions are complicated and additional expenses are needed due its slow processing and lack of system resources. Often times, the additional expenses along with functions that are not used cause the entire system to become slower. The objective of this thesis is to cultivate the strengths and eliminate unnecessary features of the current text editor, thus embodying a text editor that can be used without difficulty in systems with low specifications and cloud computing systems by using object-oriented programming language and API.

Key Words : Hexa view, text editor, cloud computing, API, development environment

* 배재대학교 게임공학과(☐sungohk@pcu.ac.kr)

· 제1저자(First Author) : 안성옥 · 교신저자(Correspondent Author) : 김수균

· 접수일(2011년 2월 28일), 수정일(1차 : 2011년 3월 25일), 게재확정일(2011년 3월 31일)

I. 서론

클라우드 컴퓨팅 환경[1]은 인터넷상의 서버 이용하여 데이터 저장, 네트워크, 콘텐츠를 쉽게 사용할 수 있도록 해주는 소프트웨어 서비스이다. 이러한 환경 속에서 문서 편집기[2, 3, 4, 5] 또한 발전을 하고 기능이 많아지고 있지만, 이로 인해 편집기 프로그램은 사용자에게 있어서 사용하지 않는 불필요한 기능을 제공하게 되고, 또한 프로그램 자체가 무거워지고 있다. 현재 상용화 되어 있는 문서 편집기의 경우에는 성능이 우수하며 사용자에게 많은 기능을 제공하고 있다. 그러나 이러한 기능 중에는 사용 빈도가 높지 않은 부분이 있으며, 본 논문에서는 이러한 부분을 제거하고 필요한 기능만을 넣은 문서 편집기를 객체지향언어[6, 7]와 윈도우 API[8]를 이용하여 사용하기 편하고, 저 사양의 시스템 및 클라우드 컴퓨팅 환경에서 구동이 가능한 문서 편집기를 개발하는 것을 목표로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 설명하고, 3장에서는 본 논문에서 제안하는 프로그램의 전체 개요와 프로그래밍에 대하여 설명하였다. 4장에서는 본 논문에서 제안하는 문서 편집기의 코드를 설명하고 결과를 제시하여 효율성을 말하고 5장에서는 결론과 향후 연구 과제에 대해서 제안한다.

II. 관련연구

객체지향언어에 대해 설명하기 전에 사용자들이 현재도 많이 사용하고 있는 C언어는 전세계적으로 가장 많은 사람들로 부터 사랑을 받고 또한 널리 사용되는 프로그램 언어 중의 하나이다. 그러나 이 언어 역시 한계에 부딪히게 되었고, 이에 따라 고도로 복잡한 프로그램을 관리하기 위하여 객체지향언어인 C++가 탄생하게 된다. C++는 처음 소개된 이후로 1985년과 1989년, 그리고 C++에 ANSI표준을 적용하기 시작한 시기 등 3번에 걸쳐 개정되었다. 1994년에 최초로 표준안이

발표되었으며 ANSI C++위원회는 사실상 C개발자가 설정한 모든 사양을 그대로 수용하고 그 외의 약간의 사양을 덧붙였다. 그렇기 때문에 C++는 C프로그래머가 쉽게 C++를 사용할 수 있다는 관련성에서 큰 장점이 있다.

객체 지향 프로그래밍은 새로운 방식의 프로그래밍 접근 방법이다. 프로그래밍이 탄생한 이후로 그 방식은 많은 변화를 겪었는데, 프로그래밍이 현격하게 다른 변혁을 겪을 때마다 새로운 접근 방식이 나타나 프로그래머가 그 전 단계보다 훨씬 더 복잡해진 프로그램을 처리할 수 있도록 도왔다.

예전의 구조적인 프로그래밍은 복잡한 프로그램 작성 시 프로그램의 규모가 일정 한계를 넘으면 가끔씩 실패하는 경우를 제외하고 상당히 우수한 기능을 해왔지만 그 보다 좀 더 복잡한 프로그램을 쓸 수 있는 새로운 프로그래밍 방식이 필요했다. 그래서 탄생한 것이 객체 지향 프로그래밍이다. 이는 구조적인 프로그래밍에서 사용되는 개념들을 최대한 활용하여 종전과는 다른 방식으로 이러한 개념을 엮어내는 역할을 한다.

III. 문서편집기의 설계 및 구현

3.1 시스템 개요

본 논문에서는 객체지향언어를 이용하여 기존 문서 편집기의 장점은 남겨놓고 사용자들이 적게 사용하는 부분은 과감하게 배제하여 컴퓨터 자원을 최대한 줄여 사용하기 편리한 문서편집기를 제작한다.

본 절에서는 프로그램의 기본적인 설계 구조에 대해 설명한다. 본 문서 편집기는 C++를 기본으로 하여 기타 라이브러리를 이용하여 설계하였다.

시스템 개요도는 그림 1과 같은 구조를 가지며, 문서 편집기의 기능을 구현한 알고리즘을 이용하기 위해 각 기능의 CPP와 헤더파일에 정의 되어있는 알고

리즘을 호출하고 API를 이용하여 인터페이스를 만들고 HyperText.CPP 파일에서 호출하여 컴파일을 하게 되며, 최종적으로는 사용자가 사용하는 문서 편집기가 실행된다.

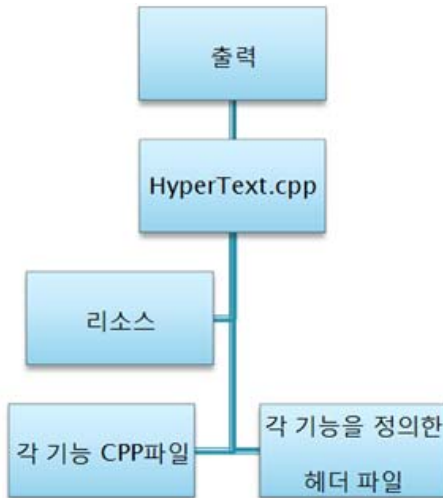


그림 1. 시스템 개요
Fig 1. System Overview

3.2 구문 분석기의 구현

본 절에서는 문서 편집기의 구현을 중심으로 다루며, 구현하기가 어려운 구문분석, 이진파일 뷰어의 가장 핵심적인 알고리즘의 설명을 위주로 설명하며, 또한 문서 편집기 프로그램이 컴퓨터에서 어떻게 구현되는지를 중심으로 다룰 것이다.

그림 2는 사용자가 사용하는 언어에 관하여 구문을 분석하여 적당한 공백과 색으로 표현을 해주는 작업 중에 구문을 분석하는 일련의 과정을 나타낸 흐름도이다. 그림 2처럼 처리 작업은 나누어져 있지만 사용자가 볼 수 없는 내부에서 구동되는 방식이고 실제 프로그램에서는 사용자의 입력과 구문 분석, 출력이 동시에 이루어져서 사용자는 구문분석시간이 따로 필요 없이 작업을 할 수 있다.

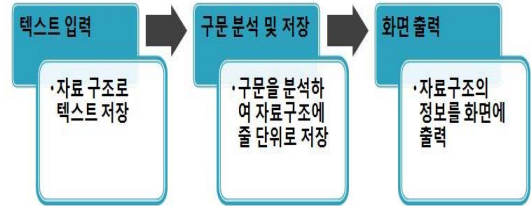


그림 2. 구문 분석 작업
Fig 2. Calculation extent of Object

그림 3은 C언어의 구문 분석기중 키워드를 선언하고 검색할 때 이용하는 함수이다. 위의 그림3과 같이 int, char와 같은 자료형 부터 #로 시작하는 전처리기 까지 C에서 사용하는 모든 키워드를 선언하여 입력 받은 구문을 분석할 때 검색하도록 한다.

```

BOOL CParseCcpp::IsKeyword(CApiEditText &e, int s, int e)
{
    static TCHAR *keyword= "auto break bool case char class const "
        "continue default delete do double else explicit enum "
        "extern float for friend goto if inline int long "
        "new namespace operator private protected public register "
        "return short signed sizeof static struct switch "
        "template this try typedef union unsigned using "
        "virtual void volatile while __asm __fastcall __based "
        "___cdecl ___pascal ___inline ";

    return IsStringExist(keyword, e, s, e, TRUE);
}

BOOL CParseCcpp::IsPreProcessor(CApiEditText &e, int s, int e)
{
    static TCHAR *preproc= "#define #elif #else #endif #error #if #ifdef "
        "#ifndef #include #line #pragma #undef ";

    return IsStringExist(preproc, e, s, e, TRUE);
}
    
```

그림 3. 키워드 선언
Fig 3. Keyword Declaration

그림 4는 실제 문서 편집기에 입력 받은 문자가 배열에 저장되는 모습을 나타낸 것이다. 위의 그림과 같이 실제 입력 받은 문자열은 배열에 저장하여 공백 칸으로 하나의 유닛을 구분하며, 각 유닛을 그림3의 함수를 이용하여 키워드를 검색하고, 그 외의 것은 정상 문자열로 아무런 처리도 하지 않게 된다. 각 유닛의 정보를 상대 값으로 받아와 적절한 공백 칸과 색으로 문자를 표현하여 여타의 개발도구처럼 구문을 분석하여 사용자에게 시각적으로 구분하기 편한 환경을 제공한다.

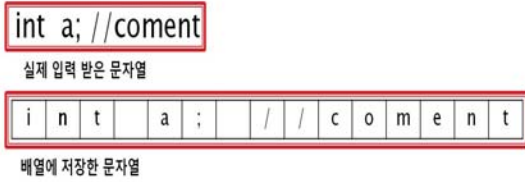


그림 4. 구문의 저장
Fig 4. Text store

3.3 hexa 뷰의 구현

그림 5는 이진파일이 출력이 되는 일련의 흐름을 나타낸다. hexa뷰 (HEXA View)가 필요한 이유는 바이너리 파일을 읽음으로써 폰트에 없는 그래픽 문자나 텍스트 파일에는 나올 수 없는 코드, 개발 중에 잘못 만들어진 파일, 하드 디스크의 물리적인 오류로 인해 손상된 파일 등의 경우 등 온전한 파일이 아닐 경우에도 읽어올 수 있다.

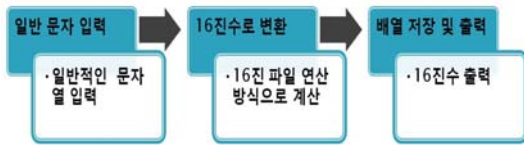


그림 5. 이진파일의 출력
Fig 5. Binary file

hexa 뷰의 구현으로 알고리즘을 이용하여 바이너리 뿐만 아니라 아스키 코드 및 UTF8 등의 다른 형식의 문서 인식기도 개발이 가능하다.

그림 6은 hexa 뷰를 실행했을 때의 결과를 보여준다. 변환 전에 입력했던 사항을 WinEdit 및 UltraEdit 처럼 쉽게 보여주는 것을 알 수 있다.



그림 6. hexa 뷰
Fig 6. Hexa view

IV. 개발환경 및 구현결과

4.1 개발환경

일반적인 사용자는 문서 편집기에서 사용하지 않는 기능 때문에 시스템의 리소스가 많이 낭비된다는 사실을 알지 못하고 있다. 특히 저사양 PC에서는 낭비되는 리소스로 인하여 사용자들은 컴퓨터 사용시 큰 불편을 느끼게 되기 때문에, 개발자들은 이러한 PC에 맞도록 가볍고 사용하기 편한 문서 편집기를 만들게 된다. 본 구현 프로그램은 사용자가 개발 및 작업을 할 때 용이하도록 프로그램 자체의 리소스를 조금 사용하는 문서편집기로 제작을 하였으며 사용자의 취향에 따라 사용할 수 있는 FTP접속 프로그램이나 문자뿐만 아니라 기타 진수로 보기 등 여러 편리한 기능을 제공하고 있다.

4.2 구현결과

본 프로그램에서의 목적은 개발자가 개발을 하기에 적절한 환경을 제공하여 개발자가 개발에 있어 부가 기능이 필요 없게 도와주고 사용하기 쉬운 프로그램을 개발하는데 있다. 이로 인해 개발자는 개발에 용이한 개발환경을 얻게 되어 작업 능력의 향상을 기대할 수 있다.

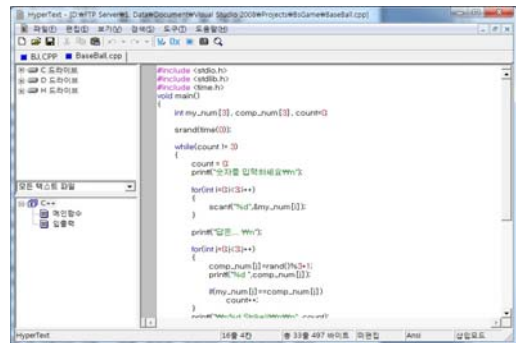


그림 7. 구현 결과 (편집기의 실행)
Fig 7. Result (Execution of text editor)

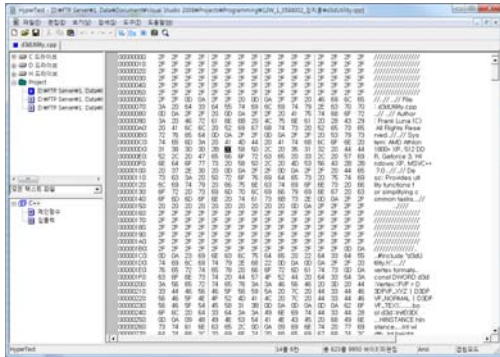


그림 8. 구현 결과 (Hexa view)

Fig 8. Result of Hexa view

V. 결론

본 논문은 클라우드 컴퓨팅 환경에서 여러 사용자들이 사용할 수 있는 문서 편집 인터페이스에 대해 제안 하였다. 논문에서는 문서 편집기의 장점은 부각 시키고 불필요한 추가기능은 배제하여 저 사양의 시스템 및 클라우드 컴퓨팅 환경에서도 무리 없이 구동이 가능한 문서 편집기를 객체지향언어와 API를 이용하여 구현하였다.

과거에는 프로그래밍이라는 작업이 특정 IT 개발자들에게만 한정된 작업 이었지만 현재는 다양한 기술이 개발되고 학문도 늘어남에 따라 배우려는 사람도 늘어나고 개발자도 늘어나 문서 편집기는 많은 사람이 사용하는 절반은 대중적인 프로그램이 되었다. 기존 문서 편집기와는 다른 사용하기 좋고 저 사양의 시스템에서도 무리 없이 구동이 가능한 프로그램을 제작하여 개발환경에 적합한 프로그램을 구동시키고자 하였다.

또한 필요 없는 기능은 빼고 기존 제품과 인터페이스 단축키 등은 최대한 비슷하게 하여 별도의 지침이 없어도 사용자의 혼란이 없이 손쉽게 사용할 수 있도록 디자인 하였다.

향후연구에서는 컴파일러와 같은 외부의 요소를 가져와서 소스코드를 편집하고, 컴파일까지 가능한 프로그램을 개발해 볼 것이다.

참고문헌

- [1] 방영환, "한국 과학 기술의 미래를 위한 클라우드 컴퓨팅 환경", *정보처리학회 논문지*, 제16권 2호, pp. 106-111, 2009
- [2] 김영철, 유두규, "추상구문트리를 이용한 구문지향 XML 문서 편집기", *한국인터넷정보학회*, 제6권 제2호, pp.117-126, 2005
- [3] 채원석, "SMIL 문서 편집기 개발을 위한 객체 모델링", *한국컴퓨터정보학회 논문집*, 제10권 제3호, pp.161-171, 2005
- [4] 김평규, 이종학, "문서병합 기능을 갖는 XML 문서 편집기의 설계 및 구현", *한국정보과학회 춘계학술발표회* 27권 1호, 2000
- [5] 임선미, "WYSIWYG방식의 웹 문서 편집기", *고려대학교 석사학위 논문*, 1997
- [6] 최용화, "플라톤의 이데아론과 객체지향 패러다임 비교 연구 :이데아론과 객체지향 프로그래밍의 구조를 중심으로", *연세대학교 석사학위 논문*, 2008
- [7] Andrew Koenig, Barbara E. Moo, *Accelerated C++*, Addison-WesleyProfessional, 2000
- [8] Andre Lamothe, *Windows Game Programming for Dummies*, IDG, 1998

저자소개



안성옥(Sung-Og An)

1983 고려대학교 수학교육과(이학사)
1985 고려대학교 컴퓨터학과(이학석사)
1989 고려대학교 컴퓨터학과(이학박사)

1991년~현재 배재대학교 게임공학과 교수

※ 관심분야: 가상현실, 데이터베이스



김수균(Sookyun Kim)

2006년 고려대학교 컴퓨터학과(이학박사)

2006.3~2008.2 삼성전자 통신연구소 책임 연구원

2008년~현재 배재대학교 게임공학과 조교수

※ 관심분야: 기하모델링, 게임그래픽, 실감미디어