

c-Edit: Input-Interface for e-Learning in Chemistry

Yoshimichi OIKAWA^{a*}, Tomizo OKUDA^b and Jiro TAKANO^c

^aLaboratory of General Education for Science and Technology, Tokai University
1117 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan

^bHigher Education Research Institute, Tokai University
1117 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan

^cDepartment of Chemistry, School of Science, Tokai University
1117 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan

*e-mail: oikawa@keyaki.cc.u-tokai.ac.jp

(Received: April 10, 2006; Accepted for publication: May 10, 2006; Published on Web: August 21, 2006)

An educational race is in the works to develop e-learning systems that can be considered ‘stand alone’, ‘off the shelf’, and easy to implement by both educators and students within the chemistry educational environment. Even the best systems available have several drawbacks as they have can be troublesome to utilize, limited in interactivity, and are time consuming, and in a few cases requiring as much time to get to learn how to operate as the educational operations they should be performing. Our team has been working on eradicating such drawbacks, having developed ‘c-Edit’ offering a multi-operational, simplistic input-interface program for educational chemical formula and chemical equation use. The ‘c-Edit’ e-Learning environment has taken the dynamic advantages of Macromedia Flash MX 2004 using action script. This web-based technology design, advances student users to input chemical formula answers (to instructor designed questions) and equations with a minimum of technical fuss. User input is systematically translated into TEX-like data, which is then sent and interpreted to interact with the learning management system or e-learning content provision side. Resulting, is a powerful, versatile student-interface, and as well for the instructor, a dynamic-learning environment, allowing easy in instructor modification to meet class environment needs.

Keywords: e-Learning input-interface program, Flash Animation, Chemical formula, Chemical equation

1 Introduction and Background

For those interested in a historical up-coming of assisted instructional systems, beginning with ancient mainframes up to the present, S.G. Smith offers the history buff a great time-line of events [1]. Here I merely want to state that even with the technological side of advancement, even with today’s web-based e-learning system, the main reason for the technology seems in many cases to have let the educational purpose lapse.

Most e-learning systems are still at the basics of ed-drilling with student results measured with less than intuitive ‘fill in the blanks’ and ‘multiple choice’ activities, using basic HTML text boxes, radio buttons, and check and list boxes [2–4]. Ways and means to fully involve the student with these systems in equation building and direct chemical formula input are feeble at best.

To be sure, a structural formula in certain circum-

stances can be inputted and targeted to remote computers, as in the case of the JME Molecular Editor [5]. But when it comes to chemical formulas or equations necessary in general chemistry, dynamic operational procedure is often wanting.

Some editing programs, for instance ChemDraw ActiveX/plugin Net [6] indeed have ability for formula/equation editing in a web-based setting, yet other program limitations do not allow full follow-through ‘send’ to remote computers desired. There is WebEQ [7], a good rounded math-expression editor, though for all its merits, it is not a worthy solution for chem.-based operational needs; becoming too cumbersome, requiring time lost in structure definition, expression creation, number values, etc. for the end user, who is simply in need of a systematic ‘idea to page’ unhindered working environment.

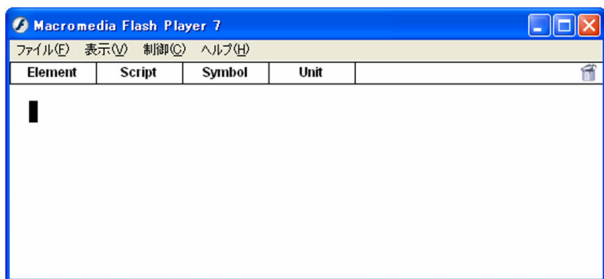


Figure 1. c-Edit Chemical formula/equation interface.

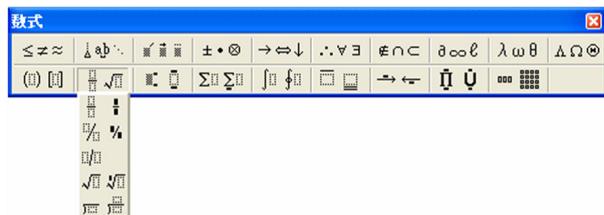


Figure 2. This is the panel of Microsoft Math Editor.

This is what 'c-Edit' was, by our team, created to change; to be an un-daunting tool for the chem. Student, who again, needs the grace to pursue his study, free of hindrances; able to put his or her mind to the creativity and experimentation of chemistry study. Our current and continued labors provide an input interface, unique for chemical formula and equation scripting student-exercises with the further power be both executable, intricate, web-savvy and available for monitoring via remote systems.

1.1 c-Edit Interface

Up front, the superiority of c-Edit lies within the versatile 'Macromedia Flash MX 2004 interactive interface. User events are done utilizing two simple areas: a selective menu bar, and an edit window, as shown here in Figure 1.

Components necessary for the creation of chemical formula/equation construction are conveniently selected from corresponding pull-down menus, and become visible in the edit window, where they can be arranged, rearranged and/or edited with simple 'drag and drop' maneuvers.

Equational structure definitions, i.e. fractional or super/subscript attributes are assignable at any time; superiority when compared to rigid restricted regimental input requirements necessary, for example, in commercial systems such as Microsoft Math Editor. MME's fractional creation alone requires complex 'select form> select numerator> select denominator' actions and requires a multi-step procedure in the creation of fractional components (Figure 2). c-Edit can handle such tasks in nearly single-steps, making c-Edit a tool whereby the user focuses on the educational, creative operant-end without so much concentration lost to daunting software tasks.

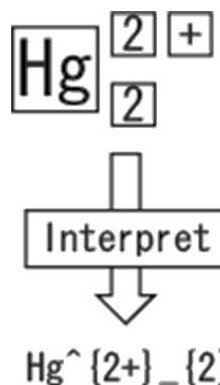


Figure 3. General case of interpretation.

Compared to c-Edit, with MME the user must devote extra time at the outset, defining the structure of expression via more complex panels, sorting out input components. Under this interface, structural equations must be defined by breaking them down based on relative positioning of each component. Thus the user's train of thought is stifled by a need to concentrate on components on a 'one by one' basis.

1.2 c-Edit Interpretation: From Entry to TeX

The next focus in the esthetic comfort of using the c-Edit system is in the simplistic transfer from entry tasks to TeX-like form, readying data for distribution.

With each component individually attributed its own data type, coordinates, and labels, c-Edit's automatic interpretation operant arranges components based on a coordinate-logic operation.

To explain a little more in-depth, components are script-ordered with focus on whether the component registers as possessing either an 'element type', 'script type', 'symbol', or 'unit type' element. For example, given a single or array of the above elements, attributed to each component, all components in an entered equation are interpreted and arranged accordingly with attention paid in the arrangement to the 'last non-script type' element. (Figure 3)

Given a 'special type' arrangement, i.e. a script type component appearing before a non-script type component (as in the case of 'atomic number or 'mass number' entry), the program logically registers the required sequence, automatically inserting a 'null-label' element type, rounding out the equation to grammatical perfection. (Figure 4)

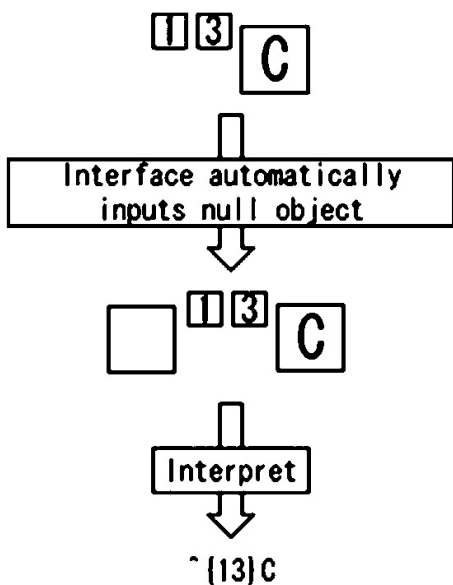


Figure 4. Interpretation case of mass number.

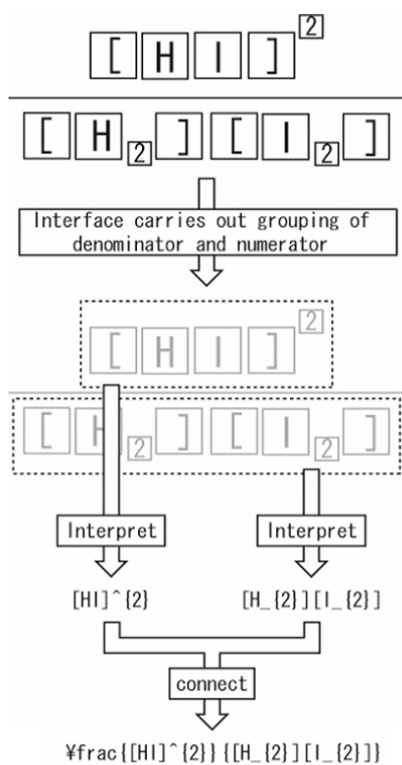


Figure 5. Interpretation case of fraction.

In dealing with fractional components, these are logically grouped by the program with due respect to their relative fraction bar positioning. (Figure 5)

The end product is the final interpreted data, readied and transferred from the 'interface' portion of the program, to the input area of the web page in the form of text string data.

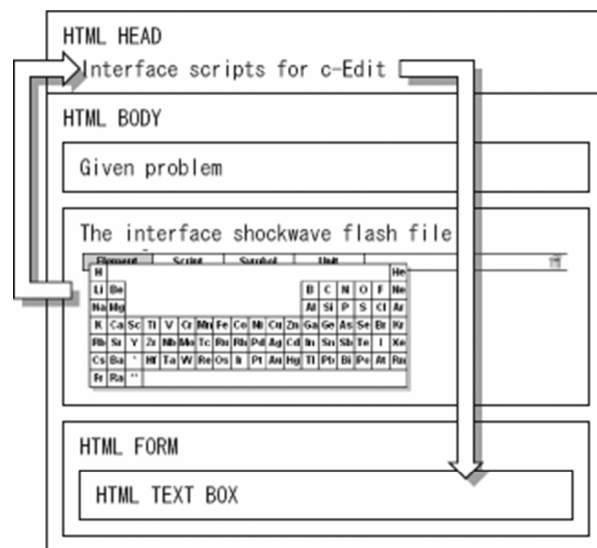


Figure 6. Web page structure with interface showing flow of interpreted text data.

1.3 Data flow

Figure 6 gives an example with arrows expressing the flow of interpreted text string and how it is sent from the initial embedded web page interface and subsequently stored as a subroutine defined in java-script. It is this method by which Text data becomes easier to evaluate by both web page and/or remote computer evaluation.

2 Preparation

2.1 Preparing for web page

c-Edit's programming ability is, again, created and published in 'Shockwave Flash' format. Once a 'Flash' file has been created, it is easy for the user to embed the created interface(s) into his or her web page simply by using 'tags'. Entering or retrieving data from the interface involves little more than the use of a couple of functions and an addition of 'form'. The form allows the versatile definition allowing the user ability to test text on either a client or remote computer system. Bold lettering allows the user unique names for script or object components embedded in the HTML, as shown below in Figure 7.

2.2 Operation

Returning to the interface menus and editor of c-Edit, all variations necessary to create formulas or equations can be selected from an upper panel menu bar with tabs which the user selects and calls on through simple drag-and-drop operations. Figure 8 demonstrates using a periodic table tab where the user easily chooses the symbol of the

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="javascript">
  function procFSCCommand(command, arg) {
    document.formName.textName.value = arg;
  }
</SCRIPT>
<script Language="VBScript">
  Sub objName_FSCCommand(ByVal command, ByVal arg)
    dummy = procFSCCommand(command, arg)
  End Sub
</script>
<SCRIPT LANGUAGE="JavaScript">
  function check() {
    // Answer check program
  }
</SCRIPT>
</HEAD>
<BODY>
<P>
  Question
</P>
<P>
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/sw
flash.cab#version=6,0,0,0"
  WIDTH="550" HEIGHT="200" id="ObjName" ALIGN="">
<PARAM NAME=movie VALUE="chef.swf">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
<EMBED src=chef.swf" quality=high bgcolor=#FFFFFF WIDTH="550"
HEIGHT="200" NAME="inpTest" ALIGN=""
TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
</EMBED>
</OBJECT>
</P>
<FORM NAME=" formName ">
<INPUT TYPE="TEXT" NAME=" textName " SIZE="80"
STYLE="display:0"/>
<INPUT TYPE="BUTTON" VALUE="Check!" onClick="check()">
</FORM>
<P>
</BODY>
</HTML>

```

Figure 7. Sample embedded HTML interface for web page.

desired element. Following, the chosen element can then be dragged to its proper positioning and be checked.

Adding super- or subscript, the user needs only to select the desired script from the upper script menu. Super-script will be interpreted when dragged above the center of an element, while the opposite subscript will be selected when dragged below the center (Figure 9).

Element	Script	Symbol	Unit	
H				He
Li	Be			B C N O F Ne
Na	Mg			Al Si P S Cl Ar
K	Ca Sc Ti V Cr Mn Fe Co Ni Cu Zn Ga Ge As Se Br Kr			
Rb	Sr Y Zr Nb Mo Tc Ru Rh Pd Ag Cd In Sn Sb Te I Xe			
Cs	Ba Hf Ta W Re Os Ir Pt Au Hg Tl Pb Bi Po At Rn			
Fr	Ra			

Figure 8. Elemental part selection.

Element	Script	Symbol	Unit	
Ca ²⁺	H ₂			

Figure 9. Sample screen shot of script operation.

Element	Script	Symbol	Unit	
H ₂ + I ₂ ⇌ 2 HI	K =	$\frac{[HI]^2}{[H_2][I_2]}$		

Figure 10. Sample screen shot of fractional operation.

Fractional creation is just as simple, operationally. Again, it is less cumbersome than other mathematical expression editors. An 'above' and 'below' area being automatically interpreted as numerator and denominator with the mere selection of the fraction-bar tool. The fraction bar can be positioned and resized (Figure 10). Note: undesirable or lost parts are disposable; thrown away into the garbage with no resultant trouble.

3 Testing – c-Edit In Use, and Comments

Our team has done and continues to do our best in creating a useful, interactive program; an offer to the field of chemistry education. Granted, c-Edit is a usable system but always a work in progress as any software designer can well appreciate.

To test the effectiveness of both the system and user adaptability in its current form, several test situations were created. Our test subjects were made up of 2 groups divided into approx 12 test subjects each. Test subjects were a mix of those who have had some chemistry ability, and others who have not.

The actual test (equal in questions regarding molecular formula, and element location on a periodic table) was then distributed. In preparation, two types of drill pages were designed to test interface comparative performance of the c-Edit program. The first test involved the c-Edit itself with Flash embedded interface; the second, a normal old style HTML format requiring direct input.

Q: Write a balanced equation showing the reaction between an aluminum ion and a hydroxide ion, producing aluminum hydroxide.

Element	Script	Symbol	Unit
Al	3+	+	3 OH ⁻

Al³⁺ + 3 OH⁻ →

Figure 11. Sample Question from Remote Computer.

c-Edit, as already described, frees the user to concentrate on the given exercise, again undaunted by formatting tasks. As a side note, let us describe a little on how this works (at the time of our student-subjects test): The remote computer (or teacher computer) is LAN connected to each test-subject student computer. The remote computer begins a 'drill,' singularly and sequentially sending a series of twenty 'questions' or 'problems' to each student (see Figure 11 to view one sample 'question' as it arrives on the readied-student interface).

Once received by the student, the student addresses each in the sequence of 'problems', and submits an answer, which then is redirected back to the remote (with the remote automatically sending the next question). Prior teacher or manual instruction in c-Edit is unnecessary, as, in the above process, students master the technical know-how of c-Edit simply by 'using' it (it normally has taken only 2-3 drill questions for the student to get the hang of the procedure). As students complete answers to the sequential drill and submit their answers back to the remote, the remote itself, is keeping track of each student's progress as well as registering for each, their answers, formatting the data for later teacher evaluation.

With the knowledge of the above in mind, let us return to the comparative testing being discussed.

The HTML entry page required an increasing amount of steps and procedures, occupying the test subject's attention. In this case the test subject (under normal HTML editor format) was given a text-box in which to work, and when faced with formulas requiring super-, subscript required operants, arrows, etc., the subject was not able to carry out the assignment without extra tasks. See Figure 12 as an example of a simple operation gone astray.

4 Results

Post the c-Edit program test and HTML test, all test subjects were subjected to a final written exam covering the same material. The purpose was to determine whether the student gained from the c-Edit system as compared to the older HTML testing method.

Superscript ... Ca²⁺

Subscript...H₂

Arrows... <-, >-, <>

Fraction... a / b

Figure 12. Example HTML input text box.

As for those students who have had some previous chemistry exposure, little or no difference was resultant on their paper test answers when it came to correctly writing element-form and simple equations. However, subjects with less chemistry knowledge, and who were subjected to the c-Edit system, scored remarkably high on the paper exam, compared to their same level peers, subjected to old style HTML testing.

However both low and high achievers, subjected to the c-Edit system, scored higher in the periodic table portion of the test as, again, compared to subjects of the old style HTML testing method.

This success demonstrated c-Edit's impressibility in the learning process, exceeding antiquated HTML format e-learning systems.

5 Conclusion

With modern programming technology, c-Edit as a modern e-learning system, houses the capability to drive a student's imagination in a genre (in this case chemistry), with little consideration for the technology itself. All elements, the historical background of e-learning, the design of differing programs to upgrade teaching methods, have been carefully considered in the design of c-Edit. The key, our team feels, is in the dynamics of the program to enhance and educate rather than limit a student's ability to interface with a program; and to enhance and assist the instructor's ability to see the mental process of the student in their learning and performing of tasks and exercises with the aid of such a system.

c-Edit is provided free from my web-site [8]. Please have a look, take it for a test drive, kick the tires. I am certain you will find a useful tool that may compliment your own needs.

References

- [1] S. G. Smith, *J. Chem. Educ.*, **75**, 1080-1087 (1998).
- [2] V. I. Bendall, *J. Chem. Educ.*, **73**, 323-324 (1996).

- [3] C W. Bown, *J. Chem. Educ.*, **75**, 1172-1175 (1998).
- [4] <http://www.jce.divched.org/JCEDLib/QBank/index.html>
- [5] <http://www.molinspiration.com/jme/>
- [6] <http://products.camsoft.com/>
- [7] <http://www.dessci.com/en/products/webeq/>
- [8] <http://chem.slge.u-tokai.ac.jp/c-edit/>