

The Virginia Tech Undergraduate Controls Lab: Past, Present, and Future

Peter R. Rony, Virginia Tech.

The Editor of CACHE News has invited me to describe how one of the CACHE software products – i.e., Doug Cooper’s PICLES and DIGEST – has impacted the teaching of a process controls laboratory, viz., ChE 3016, at Virginia Tech. I have decided to describe my use of DIGEST in the context of the evolution of the laboratory that I started to teach during the 1978 spring quarter. I believe that such evolution parallels that of many chemical engineering laboratories. In effect, I am providing a reminder of how far we all have come in laboratory automation since 1978.

I have taught the ChE 3016 lab almost continuously for 20 years, a period of time that has spanned the death throes of X-Y and Y-vs.-t pen recorders; the appearance of simple, laboratory microcomputers (e.g., the MMD1) equipped with breadboarded analog-to-digital and digital-to-analog converter chips; the emergence of the IBM PC and its subsequent iterations; the emergence of commercial data acquisition boards such as the Metrabyte and Data Translation boards; the emergence of inexpensive and powerful programmable-logic controllers (PLCs) and associated software such as the Texas Instruments TI545 and Application Productivity Tool, respectively; and the emergence of fast, Pentium-based PCs.

Past

Two decades ago (spring 1978 and 1979), the laboratory was primarily a measurements and instrumentation laboratory that primarily supported our summer unit operations laboratory. Students learned how to use orifice plates, rotameters, and thermocouples, and how to measure refractive index in order to determine component concentrations in liquid mixtures.

During spring 1980 and 1981, the lab course was transformed into a group of oscilloscope-based, recorder-based, and digital-multimeter-based experiments designed to teach students the basic principles of electronics and manual data acquisition. Though we had a variety of chart and X-Y recorders in the lab, their use proved cumbersome since the pens frequently became plugged. In addition, during a 3-hour afternoon lab very little data was produced, and portions of the recorder chart traces needed to be carefully converted to decimal data values.

By spring 1982, the first control experiment was added to the lab; it was based upon a M.S. thesis by Barry Archer. Archer’s “control of the temperature of flowing, heated air” experiment employed a simple laboratory microcomputer trainer — the Z80-based MMD-1 — that I used in several CACHE hands-on short courses early during the early 1980s. The controlled

system employed a hair-dryer heating coil and a glass vessel filled with glass beads.

As the 1980s progressed, IBM PCs [with and without hard disk] were added to the lab in order to automate at least three experiments. The chart and X-Y recorders were consigned to storage, where they remained until they were consigned to surplus in 1998. Early computer printers were simple, dot-matrix printers of the Epson type. During this period, we realized that the time constant of a controls experiment – most were first-order systems – was the key parameter that determined how much data a student could acquire during a 3-hour lab period. Experiments that had long time constants – e.g., the heating of a small, stirred flask of water (which had a time constant of several minutes) – were ultimately discarded in favor of experiments with time constants of less than 30 seconds.

Later in the 1980s, each IBM PC gave way to its replacement, a remarkably robust IBM PS/2 30/286 personal computer with 10 MB hard disk, a DOS operating system machine that remains in use even in 1998.

By 1990, Karl Rony (B.S. ChE Virginia Tech 1987), who worked in the APT software group at the Texas Instruments Programmable Logic Controller division at Johnson City, Tennessee, suggested to the author that the controls laboratory – with its use of Metrabyte data acquisition boards — was obsolete. He proposed, and together we implemented, the TI545 programmable logic controller and its powerful, high-level software, the Application Productivity Tool (APT). By spring 1991, five APT/TI-545-based experiments were in full operation in the lab. At this point, junior chemical engineering students started to use Doug Cooper’s DIGEST software to analyze laboratory data files that were very easy to acquire.

APT versions 1.3 and 1.4 had no capability to couple data acquisition with the production of DOS-based data files. Consequently, Karl and a colleague, Chris Manhard, at Texas Instruments developed a software driver that would interface Tutsim (an analog computer on a PC) to the extensive database residing in the PLC. A serial port was used to transmit, once a second, the decimal digits of a group of three monitored variables – process variable, manipulated variable, and setpoint – to the Tutsim program, which collected, as a function of time, the data in a data file.

Present

At this point, the controls laboratory has been substantially transformed from a 1980s-vintage laboratory — in which data was meager and tedious to analyze (e.g., one-point FO or two-point FOPDT curve fits) — to a 1990s-vintage laboratory in which data files are easy both to create (using real-time Tutsim) and to analyze (using DIGEST).

In spring 1998, the laboratory currently contains six, inexpensive, SISO experiments based upon the control of flowing water, a fourth-order operational amplifier system, the hot stream from a vortex tube, the pressure within a confined volume, the temperature of flowing, heated air, and, finally, temperature 1 mm above a light bulb within Dow's "pyroluminescent regulometer". The TI545 programmable logic controllers have proven to be robust, and not too difficult to program using APT, which allowed us to create three basic controller experiments: typical SISO control, statistical quality control (SQC), and Autotune Variation.

DIGEST was the critical software responsible for the transformation of data handling within our controls laboratory. Since the department has had no budget for software, the CACHE licensing cost of \$125 the first year and \$75 for subsequent years was distributed among the students, who paid no more than \$5 per year to a departmental kitty in return for the ability to use Digest/Picles in their undergraduate courses. With at least 40 junior ChE students per year, we were able to generate the income to license Digest/Picles software packages from CACHE every year for the past five years. Students used Digest both in the lab and in their rooms to analyze data files. At Virginia Tech, all engineering undergraduates have purchased their own personal computer since 1984.

In summary, in the 20 years since 1978, the ChE 3016 course has gone from an environment in which laboratory experiments were difficult to automate and lab data was difficult to obtain, scarce, and difficult to analyze to a current environment in which the situation is the complete reverse: laboratory experiments are easy to automate, and lab data is easy to obtain, plentiful, and easy to analyze. This changed situation places a premium on data acquisition and data analysis software in 1998.

Future

The Virginia Tech ChE controls laboratory is currently poised on the brink of its next transformation. Both the TI545 PLC hardware and the APT software are vintage-1989 products. During the next two years, the hardware/software combination will be replaced by Wonderware InControl and InTouch software and appropriate I/O data acquisition boards, perhaps some with Ethernet capabilities. InControl is state of the art

software developed in Johnson City by many of the same professionals who programmed the APT software at Texas Instruments during the mid- to late-1980s. Wonderware software requires Pentium-speed PCs and the Windows NT operating system, whereas APT was based upon the TI545 PLC and configuration on the DOS operating system.

The author views with great admiration the web-based laboratory experiments of Jim Henry at the University of Tennessee at Chattanooga and, following Jim's lead, plans to link all of our Wonderware-based experiments to the World Wide Web in order to facilitate remote data acquisition by our students. We agree with Jim that web-based data acquisition is a wonderful opportunity to promote interest in laboratory-based education.

Karl Rony, who now works at Wonderware in Johnson City, will once again be the patron saint of the Virginia Tech ChE 3016 controls lab course. He is also interested in working, through CACHE, to promote the use of Wonderware-based software for chemical engineering controls experiments. He can be reached by email at karl.rony@Wonderware.COM.

Industrial Associates (Industrial contributors to CACHE)

Aspen Technologies
Eastman Chemical
Merck & Company
Parke-Davis
Simulation Sciences
Union Carbide

Powerpoint Slides for Unit Operations Course

Richard H. Luecke, University of Missouri

Of the many changes of life styles brought by the “computer age”, none is potentially more dramatic than the effects on the delivery of education. Only a few decades ago, most of us taught by rubbing one rock on another. Then, in imitation of successful presentations at research meetings and industrial sales talks, many began to deliver lectures using prepared “overheads.”

The advent of presentation packages such as Microsoft’s Powerpoint has raised the stakes for this type of presentation. A quantum increase in visual quality as well as accuracy is possible but at the cost of much preparation time. It has seemed to me that the potential for mass sharing of this computer resource has not yet been exploited, at least not in Chemical Engineering education. A computer presentation should be passed around where it can be edited to fit various styles or syllabus content.

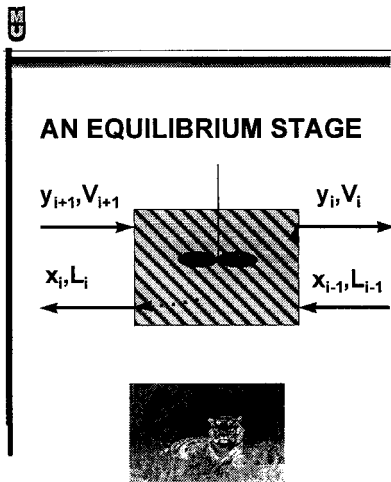
When I began to upgrade my overheads to Powerpoint about two years ago, I could not find anything on the Internet to save me the trouble. Now that I have a finished set, I will break the ice by offering, free of charge, these Powerpoint 95 classroom notes for a junior Unit Operations design semester to any Chemical Engineering department that sends me a self-addressed (100 meg) “ZIP” disk (with the return postage). The ZIP disk is required because of the large size of Powerpoint files.. . up to 10 meg for some of them.

1. These notes are keyed very closely to the second half of the text “Transport Processes and Unit Operations” by Christie Geankoplis, 3rd ed. (Prentice-Hall, 1993).
2. A list of topics and titles is given in Table I. They are listed in the order that I usually present them.
3. The presentations can be edited and then printed or copied as black-and-white or as color slides. During this (f97) semester, I have used a laptop plugged into a projector in our classroom. Direct projection of the Powerpoint slides allows me to move back and forth quickly, or to zoom to particular points on the slides. I can also bring up relevant spreadsheets.
4. A handout of most of the slides (6 to a side) allows students to concentrate on listening rather than writing. A collage of some of these notes in distillation (Figure 1) gives a sense of the content of the slides.

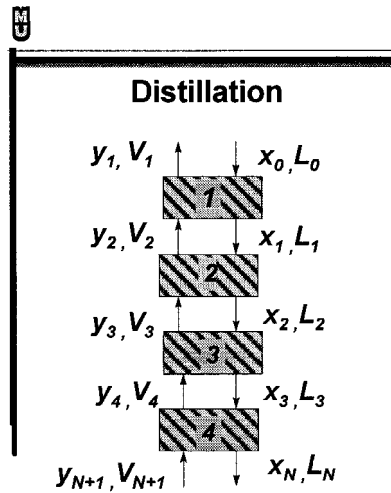
5. Students have uniformly told to me that they really appreciate the slides, particularly the handout “notes.” My teaching evaluation GPA went up nearly a point.
6. Part of my objective has been to move forward into teaching at a distance. However, it seems that face-to-face interaction is vital. Usually when I bring the laptop back to the office after class, it is covered with chalk dust.
7. One problem that other schools may encounter is the ubiquity of Mizzou Tigers throughout the presentations. Of course, the user can edit in pictures of aardvarks or whatever

Table I - Topics

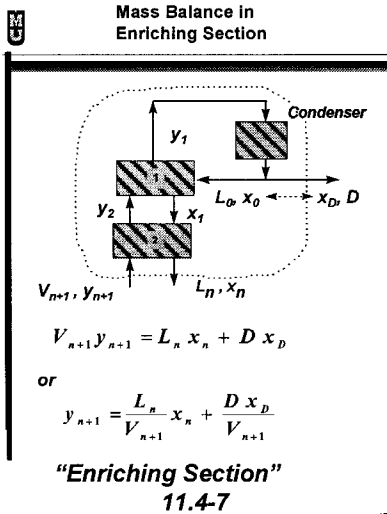
Topics	Slides
1. Introduction	11
2. Evaporation	10
3. Absorption	36
4. Packed Columns	30
5. Distillation	62
6. Extraction	25
7. Leaching	10
8. Adsorption	32
9. Drying	15
10. Membranes	11
11. Filtration	11



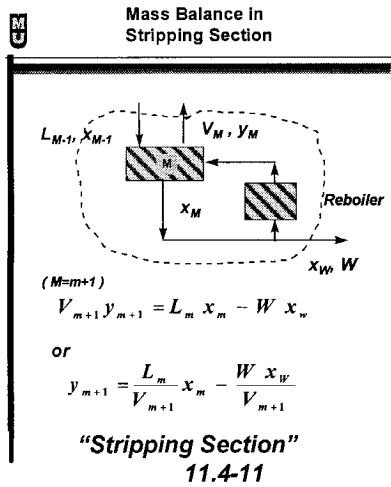
13



14



17



18

Constant Molal Overflow

For any section without an external flux, both L and V are constant....
 And so is L/V.

Enriching:

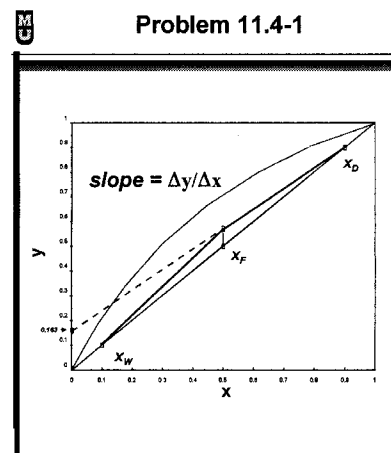
$$y_{n+1} = \frac{L_n}{V_n} x_n + \frac{D x_D}{V_n}$$

Stripping:

$$y_{m+1} = \frac{L_m}{V_m} x_m - \frac{W x_w}{V_m}$$

STRAIGHT LINES!

22



24

The Case for Excel and Visual Basic for Applications

Edward M. Rosen, EMR Technology Group

The CACHE Corporation's Recommendations of Desired Computer Skills for Chemical Engineering Graduates (1) includes familiarity with at least one scientific programming language. Many academic institutions have chosen to teach FORTRAN as the preferred scientific programming language. A possible alternative, however, is Visual Basic for Applications (2), the macro language used by Microsoft's Excel spreadsheet program.

The use of Visual Basic for Applications (VBA) has a number of attractions:

1. Excel is a widely used spreadsheet program in academia. It has a graphical and printing capability that is well suited to routine engineering and scientific usage. The VBA language is imbedded in this useful and familiar environment.
2. The language has many object oriented features.
3. It is both a superset and a subset of the Visual Basic language which is being broadly used in industry.
4. It has many of the features of the BASIC language with which many students are familiar. (Table 1).

VBA comes with a large number of scientific and engineering functions which complement Excel's built-in spreadsheet functions and its powerful equation solving and optimization program (Goal Seek and Solver). In addition, there is a mechanism for the integration of user libraries (written in VBA) into Excel.

VBA has been expanded in Microsoft's Office 97 to be the macro language for all of its applications, not only Excel. The comments here apply specifically to Excel 7.0 (Excel 95) which was released with Windows 95.

Subroutine Procedures, Functions and Function Procedures

The VBA language can be used to write *subroutine procedures*, *user defined functions* and *function procedures*.

Subroutine Procedures (or macros) are used to play out recorded keystrokes. More generally subroutine procedures may be used to control or alter the environment of the spreadsheet (i.e. addressing of cells, changing fonts etc.). They do not return a value.

Subroutine Procedures may be used to assign values to constants and may be called from functions.

Functions are used to carry out computations. *User Defined Functions* use input from the spreadsheet and output their results back to the spreadsheet. *Function Procedures* are invoked from VBA coding and return their results back to the coding. All available functions (both user defined and function procedures) are displayed in the *function wizard*.

Spreadsheet Functions

There are a number of functions available to the spreadsheet user.

Built-In Functions are supplied with Excel. They are invoked from the spreadsheet and may return a single value or an array of values to the spreadsheet. Functions that return an array are known as *array functions* (3).

User Defined Functions (UDFs) are written on a macro sheet in the VBA language and are used in the same way as built-in functions. They may be compiled (.xla file) and be included as an Add-In. They are invoked from the spreadsheet like built-in functions.

The Analysis ToolPak is a library of functions that can be added in and used like any other function. It is added in by using the Tools --- Add-Ins menu item.

VBA Functions

The user, coding in VBA, has available a number of functions.

VBA Function Procedures are supplied with the VBA language. They include such functions as Sqr (square root), Cos (cosine) and Exp (exponential). In general they are different from the spreadsheet built-in functions. They can be viewed from a macro sheet using View --- Object Browser --- VBA.

Spreadsheet built-in functions are available to VBA if invoked with Application preceding the name (e.g. Application.Max)

Function Procedures may be written on a macro sheet and be compiled and saved as a (.xla) file just as (and together with) user defined spreadsheet functions. The file may then be accessed (without being opened) by

pointing to it using Tool --- References from a macro sheet. Function procedures are called from VBA and return their results to VBA. The functions return a single value but may receive and pass arrays in their calling arguments.

Figure 1 is a listing of a the function procedure QuadFP for calculating the roots of a quadratic equation. Note that the results are returned to the calling program, the user defined array function QuadFPtest (Figure 2). The spreadsheet which invokes QuadFPtest is shown in Figure 3.

Analysis ToolPak-VBA is a library of engineering functions (ATPVBAEN.XLA) paralleling the Analysis ToolPak available to the spreadsheet. The *function wizard* indicates the VBA routines in lower case. Names in upper case refer to the spreadsheet functions in the Analysis ToolPak.

Excel and VBA as a Computational Language

FORTTRAN was developed in the late 1950's to give a computational capability to the engineer and scientist. For very large and complex programs it will probably continue to have a role to play though C or Visual Basic often appear to be the language of choice for new applications in industry.

There is of course the issue of speed of computation and the issue of recoding legacy programs written in FORTRAN. Both problems can be overcome by transferring out of Excel to carry out a very long/complex calculation (using the shell function in VBA) and then returning the results to VBA or the spreadsheet.

With the substantial computational capability available in Excel and VBA, it seems appropriate to consider it an alternate to teaching FORTRAN in academia. It can be the major computational tool for the student to carry out his routine calculations.

Distribution of Libraries

Function libraries containing user defined functions and function procedures can add a great deal of computational capability to Excel and VBA. CACHE will accept and distribute such libraries to other users.

Currently available is the library shown in Figure 4.

References

1. AIChE Education and Accreditation Committee, *Evaluating Programs in Chemical Engineering*, September 15, 1997.

2. Harris, Matthew, *Teach Yourself Visual Basic for Applications in 21 Days*, Sams Publishing, (Indianapolis) 1994.
3. Rosen, Edward, Visual Basic for Applications, Add-Ins and Excel 7.0., *CACHE News*, No. 45, Fall 1997

<u>Language</u>	<u>Comments</u>
BASIC	Developed in early 1960's
GWBasic	Supplied with MS-DOS versions prior to Version 5.0
QuickBasic	Included with DOS 6.0 and higher
Visual Basic for Windows	1992. Incorporated command to use Dynamic Data Exchange (DDE) and Object Linking and Embedding (OLE)
Visual Basic for Applications	Released in Excel 5.0. Essentially Visual Basic for Windows with slight differences.

```

Public Function QuadFP(Coef, Roots)

'Written by EMR Technology Group 10/3/97
'coef - The Vector of quadratic coefficients
'Roots - Array of roots of the quadratic

Dim Term1 As Double
Dim a As Double
Dim b As Double
Dim c As Double
Dim Arr(1 To 2, 1 To 2)

'Value of b2-4ac
a = Coef(1)
b = Coef(2)
c = Coef(3)

Term1 = b ^ 2 - 4 * a * c
If Term1 >=0 Then

    Arr(1, 1) = (-b + Sqr(Term1)) / (2 * a)
    Arr(1, 2) = 0
    Arr(2, 1) = (-b - Sqr(Term1)) / (2 * a)
    Arr(2, 2) = 0

Else

    Term2 = Sqr(Abs(Term1))

    if b = 0 Then
        Arr(1, 1) = 0
        Arr(2, 1) = 0
    Else
        Arr(1, 1) = -b / (2 * a)
        Arr(2, 1) = -b / (2 * a)
    End If

    Arr(1, 2) = Term2 / (2 * a)
    Arr(2, 2) = -Term2 / (2 * a)

End If

Output To Roots

For I = 1 To 2
    For J = 1 To 2
        Roots(I, J) = Arr(I, J)
    NextJ
Next I

QuadFP = 0

End Function

```

Figure 1. Function Procedure QuadFP

```

Public Function QuadFPtest(Coef)

    Coef are the Quadratic Coefficients a, b, c

    Dim Roots(1 To 2, 1 To 2)

    IR = QuadFP(Coef, Roots)

    QuadFPtest = Roots

End Function

```

Figure 2. User Defined Array Function QuadFPtest

Roots of a Quadratic Equation		
Coefficients	Output Array	
1	Real	Imaginary
9	-2.438447187	0
16	-6.561552813	0

Figure 3. Spreadsheet Invoking User Defined Array Function QuadFPtest

EMRLIB_Functions.xla		<i>Function Procedures</i>	
<i>User Defined Functions</i>		1. CubFP	Finds all the roots of a cubic equation
1. Bintx	First and second order interpolation in a two dimensional table.	2. DecompFP	Function used by LudFP
2. Cubic	Finds all the roots (real and complex) of cubic polynomial	3. LudFP	Finds the roots of a set of nonsingular set of linear equations
3. Interp	First and second order interpolation in a single dimensioned table.	4. LudIFP	Finds the inverse of a non-singular matrix
4. Lineq	Solves sets of (non-singular) linear equations	5. OrderFP	Function used by LudFP
5. Quad	Finds all the roots of a quadratic equation	6. PivotFP	Function used by LudFP
6. Romberg	Integrates area under curve using Romberg Algorithm	7. QuadFP	Finds the roots of a quadratic equation
7. Trapz	Integrates area under curve using the trapezoidal rule	8. SolveFP	Function used by LudFP
		EMR_Functions.xls	
		<i>User Defined Functions</i>	
		1. Rk4x	Integrates sets of differential equations using fourth order Runge Kutta
		2. Rhs	Right hand side for Rk4x

Figure 4. EMR Technology Group Library

CACHE Catalog of Products

To order CACHE Products, complete the Standard Order Form and send with payment to:

CACHE Corporation
 P.O. Box 7939
 Austin, TX 78713-7939

FAX: (512) 295-4498 or
 E-mail: cache@uts.cc.utexas.edu

We accept credit card (VISA/Mastercard), Purchase orders, and checks.
 Overseas orders are sent surface at no charge, airmail is extra

Web-based conference organization - a summary of computer experiences from PSE'97/ESCAPE-7

S. Hauan*, T. Krokstad, S. Skogestad and K.M. Lien
Norwegian Institute of Science and Technology (NTNU)

Background

This article is written to summarize our experiences with electronic document collection/(re)distribution in relation to PSE'97/ESCAPE 7¹ held in Trondheim, Norway May 25 - 29, 1997. More than 600 abstracts were received and, through review, reduced to 201 accepted papers published in a special issue of *Computers and Chemical Engineering*. Approximately 400 participants from 32 different countries were present at the symposium, with industry counting for approximately one third.

Conference organization obviously includes a significant amount of paperwork, both with respect to research material and practical matters. This article describes our approach to handling a major fraction of the documents involved in a suitable electronic form. Our main focus was to collect and (re)distribute the scientific articles through a web interface. The conference web site also included printable forms for accommodation and registration as well as an extended version of our official announcement.

Further information and source code for programs described herein are available at

<http://www.kkt.chembio.ntnu.no/research/PSE-ESCAPE/>

Motivation

By using Internet as information carrier, there is a large upside potential for fast and structured distribution. Most universities and major companies are now online, and more are expected to join shortly. Electronic collection and distribution of abstracts/articles offer several advantages to conventional, paper based systems through regular mail. We identified 6 major issues, roughly ordered by impact and importance:

1. Improved speed of distribution

Electronic transmissions are independent of slow and - from some countries - unstable postal communication. The ability to send a file, not paper, saves people outside western Europe up to 4 weeks compared to regular mail. Although courier is an alternative, it is usually quite expensive.

2. Easy updates

The authors are themselves responsible for paper layout, and, due to the possibility of speedy retransmissions, one may be (a lot) less restrictive to request changes. Usually, such requests are impractical due to time constraints and in practice hardly ever done "just for conference proceedings" unless the layout is completely wrong.

3. Reduced workload

Automatic collection and storage of cover-letters, abstracts, papers and eventual updates helps keep down the need for manual intervention. Some verifications/corrections should be expected, but they are significantly less than taking care of everything in person. Proper logging routines assures a complete history of communication with the author(s) with a minimum work of copying/archiving/indexing information that would almost certainly not be accessed except in the event of errors.

4. Higher level of consistency

By refusing to accept articles with incomplete cover-letter information (minimum: corresponding author with address, fax and phone number), we avoid "searching for people in charge" of a specific paper.

5. Systematic feedback; automatic and manual

Any submission may be immediately verified by mail or fax. This **both** checks our primary means of communication with the corresponding author and provides a receipt of submission.

6. More time for reviewer

The ability to redistribute documents to reviewers, in particular those living very far away, saves a lot of time compared to regular mail. Hence one may either extend the deadline(s) for when abstracts/papers are due, give reviewers more time/flexibility for their job or combinations thereof.

Specifically, we aimed at solving all or parts of the following tasks through a web interface:

- **collection** of abstracts/papers/updates
- structured **feedback** to authors

* Corresponding author. E-mail: steinhau@chembio.ntnu.no

¹ joint event of by the Executive Committee of the Process Systems Engineering Symposium Series and the EFCE Working Party on Computer Aided Process Engineering

² through our mail gateway

- **redistribution** of material to reviewers
- passive **information** to authors and participants

The information part was trivial from an implementation point of view. Menus on the conference home page were linked directly to documents generated in HTML and/or Postscript. Standard reminders or generic information was sent to authors using either email (preferred) or fax - in addition to the usual paper leaflets.

The remainder of this article will describe structural and implementational issues of the first three aspects. We will also present statistical results and briefly discuss possible improvements to a similar arrangement in the future.

Review and Publication Process

From the users' point of view, a submission consists of 3 basic steps

1. prepare a Postscript version of his/her abstract or full paper
2. transfer the Postscript file to our anonymous ftp server
3. complete the electronic cover letter in his/her web-browser, labelled FORM #3 in figure 1.

The final step iterates with the user to ensure a file has been sent and the cover-letter contains all required information. Step 3 is the "heart" of the process, iterating with the user to correctly glue together the file submitted in step 2 with information about the author(s). Details about the underlying programs are found in appendix A and B.

After successfully completing FORM #3, an *automatic receipt* is sent to the email/fax address given for corresponding author. The ftp file is also *locked* to avoid multiple cover- letters pointing to the same file. For security reasons, outside users are not allowed to browse any files outside the ftp area, neither is any freely accessible web-script allowed to write, move or change any files in the database or the ftp directory. Instead, a completely independent process - known as a *collecting daemon* - is run every 15 minutes with the single task of "safely" moving any successfully³ submitted document from the ftp area to our processing queue (spooling directory).

Once in the spooling directory, all documents are processed by the conference secretary through a protected web page. The main tasks are to verify printability of the submitted file, check for possible "spoofing"⁴ and get a copy for our paper based backup system.

There are three possible actions when fake submissions have been discarded:

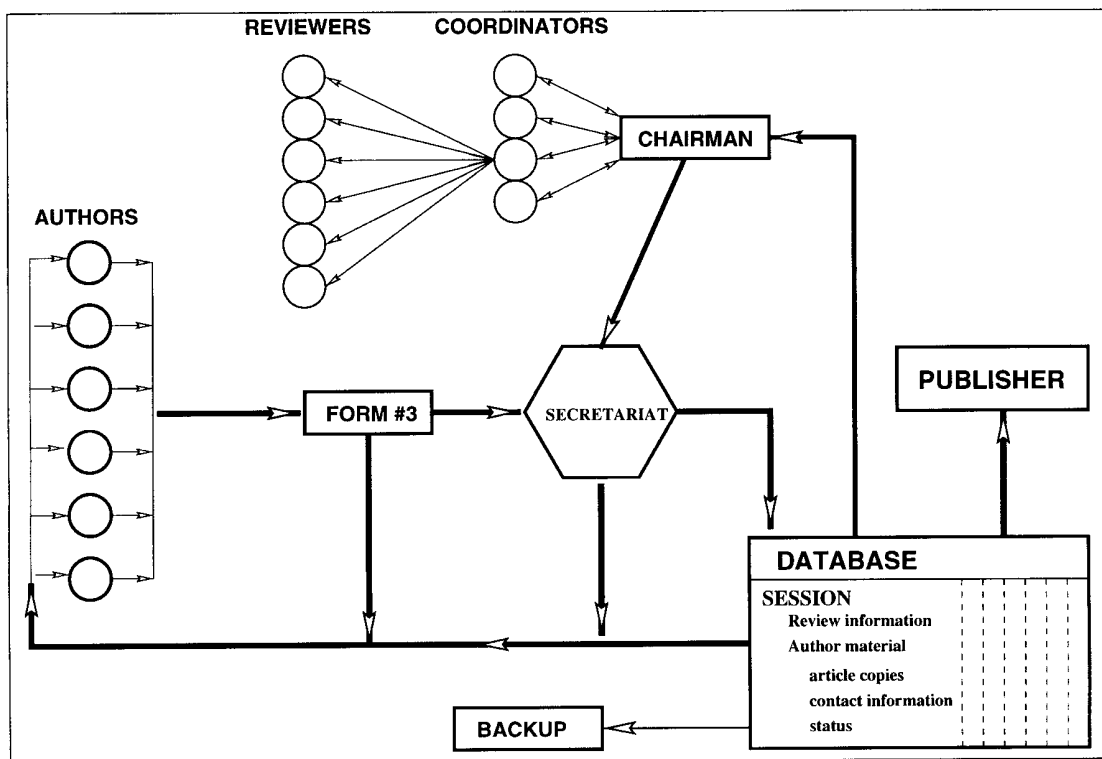


Figure 1. Electronic document collection and (re) distribution

³ meaning a completed electronic cover-letter attached to a file on the ftp server

⁴ An irrelevant submission of garbage by someone who just accidentally found our submission interface and attempted to make our life tougher.

- **ACCEPT:** sent back to the author.

Everything is ok. We mail/fax a *manual submission verification* to the corresponding author specifying the proper reference number and store the file in our database. Dependent of submission type (abstract, full paper or final paper) we add various messages like deadlines, registration information and reminders for copyright transfer forms.

- **TECHNICAL:** internal message to the administrator.

Something is wrong, our secretary is unable to print whole or parts of the document. The case - file and a short description of the error encountered - is forwarded to the "technical administrator" who looks at the file and attempts to figure out what is wrong and if anything might be done to make the file printable without involving the author. Usual problems and a few solutions are discussed later. After analysis, the file is moved back into the spooling queue for re processing; either with a fix enabling a successful printout or with a message intended for the corresponding author if no solution was found.

- **REJECT:** sent back to the author.

The document is incomplete or unprintable and we warn the author(s) *that we for technical reasons will be unable to use the file in it's current form*. Attached is also the above mentioned message from the administrator with more details about the cause of the error and (hopefully) the steps required to create a file in the proper format. The original submission is removed from the spooling queue and temporarily stored.

Results and Statistics

Reception: articles and registration

Overall, 733 of 968 (74%) successful submissions were made through our web-interface. Table 1 breaks down submissions by type. In particular, we were happy to see that almost every article update was sent electronically.

type	total	electronic copy	paper only
abstract	575	395 (69%)	180 (31%)
full paper	241	186 (77%)	56 (23%)
paper update	154	152 (99%)	2 (1%)

Table 1: Articles by type

Even though it is reasonable to assume almost everyone had been sent a copy of our official conference announcement, quite a large fraction found it more convenient to print and fax⁵ the registration form made available from our web site. Table 2 shows how people chose to register.

total received	from paper leaflet	from web page	other
387	354	117	16

Table 2: Registration by type

Error statistics: what was difficult?

Overall, there were 97 rejected submissions out of 733 attempts (13%), slightly higher than the 10% estimated in advance. Most errors were quickly corrected in one iteration with the corresponding author, and in only 8 cases (1%) we were unable to solve the computer issues involved and had to request a paper copy.

The errors may be broken down by type as follows:

	abstract	full paper	paper updated
electronic copies	395	186	152
#completed forms	474	214	195
#submissions in order	421	174	153
# technical rejects	37	24	36
other responses	16	16	6
# different people	26	20	27
wrong format	22	15	26
incomplete file	12	3	1
incorrect layout	n/a	2	9
other errors	3	4	0
never succeeded	5	2	1

Table 3: Electronic submission error statistics

Remarks:

- The errors are listed in priority order, meaning a submission containing an error with respect **both** to file format and to layout, only will be listed as a format error. In practice, the notification saying technical reject had a comment on both issues.

⁵ needed for credit card info and a valid signature

- Typical format errors, listed with decreasing frequency, were caused by modern, level 2, Postscript, raw word processor files, PCL files and non-conforming Windows or Macintosh printer drivers.
- The row for "other response" in table consists of submissions after the deadline and those who submitted the same file more than once or sent an updated copy before the previous version had been processed. This row also includes a few cases of manual intervention in cases where authors had two or more previous rejections for technical reasons.

Roughly 10 authors had severe trouble producing conforming Postscript and their initial submissions are found in the error columns for both abstracts and full paper. Their persistent attempts to use our submission interface were greatly appreciated, as it aided us in identifying flaws in our design and help system.

Redistribution

Redistribution by email got a mixed reception by the approximately 50 people involved in the review process. About a week before the deadline, an early warning was sent as a reminder of what would soon arrive. Electronic versions, where available, were sent by email as MIME encoded archive files in two stages. The first batch, including a review form, was sent immediately after the deadline to maximise the time-window available for the reviewers. The second (and final) batch with late papers was sent more than a week later to those who confirmed electronic distribution as a working alternative. A full set of paper copies was sent by regular mail as backup, but no late papers were included. Where airmail was expected to be slow or the reviewer was particularly busy, electronic redistribution was verified in advance. All review sheets were mailed or faxed to the session coordinators who summarised and forwarded individually to the chairman of the international program committee (IPC).

A very small group of reviewers specifically said they only wanted paper copies and did not care about electronic versions. On the other hand, only about 25 percent actually confirmed their use of the electronic batches sent to them. The latter group significantly increase the flexibility and speed in the review process and also had the opportunity to distribute their own workload more evenly in time. These reviewers also allowed acceptance of submissions made up to ten days after the actual deadline.

Discussion and Extensions

In general, most people had no trouble using the electronic form from their web-browser. The number of errors in transfer-

ring files by ftp was also surprisingly low. On the other hand, the total number of problems encountered were slightly above expected (13 versus 10 percent). Although most computer systems have the ability to generate proper Postscript files, this step is frequently completely transparent to the users, hence preventing actual insight in the underlying mechanisms. A step-by-step instruction for the most commonly used word processors in a web page would probably help keeping this number down.

The use of Postscript as an "exchange format" worked out well, mainly for 3 reasons:

- independence of machine- and word processor type and version
- the authors themselves are responsible for layout
- no binary file transmissions are needed

In order to minimise the number of file format errors, one should consider to allow download of proper printer drivers⁶ for the most commonly used PC and Mac platforms. From our experiences, no special actions are needed for UNIX users as they rarely had trouble with non-conforming Postscript.

There are several possible extensions to the scheme described in this paper. The most obvious one would be to require all authors to submit their work in electronic form. Considering the large fraction of authors who already did so, this is hardly expected to cause major concerns. With email as a backup solution, the need for shipment of paper based copies should be minimal. Similarly, one may opt to perform distribution and collection of reviews in electronic form only. Such a scheme is particularly attractive to minimise the time period between deadline and publication.

It is also possible to avoid the use of ftp for file transfers. In recent versions of HTML, one may create electronic forms capable of reading files directly from the users' home machines. A further refinement is found in Java where an ftp transfer process could be run in the background with the Postscript file selected from a dynamic menu. This latter option was considered and actually tested in the early phases of implementing our transmission system, but discarded to avoid turning anyone away purely for technical reasons due to the limited use and distribution of Java at the time.

A more futuristic approach would be to both collect, review and **publish** articles without ever shipping a single page of physical paper. Such an electronic journal opens for a large variety of interesting possibilities. There would be no page limits, no firm deadlines, complete cross references and easy mechanisms for searching the complete journal. Each article

⁶ beware of copyrighted drivers

could also contain links to demonstration packages on the author's home machine to allow readers to repeat simulations and study the details behind what is found in the publication itself without spending large amounts of time on their own implementation. A willing author could also allow reader-defined perturbations in parameters, physical states or specifications to study specific behaviour of the cases in question. Although an interactive, electronic journal should not be expected to replace existing ones, it would most certainly be a valuable supplement.

It should finally be noted that the sole author of the underlying programs was physically present in the U.S., several thousand miles away, while the whole thing was running. Any difficulty encountered processing incoming files was resolved through email and the occasional phone call.

Conclusions

We have successfully implemented and used a web-based system for collecting abstract and articles for a conference of reasonable size. A major fraction of the authors chose to submit their work in electronic form, saving both themselves and especially the conference secretariat time, money and effort.

Appendix A: Implementation details

The underlying programs were written in perl, a scripting language optimised for text-processing, and run through the CGI-mechanism⁷ in the web server. Our system included the following elements:

- main submission page (HTML) with links to help pages on how to generate Postscript files and how to use ftp
- main submission script, recursive on 3 levels:
 - 1) pick your filename from a dynamic pull-down menu, press continue on the bottom of the page when done. Except when submitting abstracts, the user is prompted for a reference number and a password⁸. Further details about specific checks and internal information flow is given in appendix B.
 - 2) iterate with the user for list of authors, paper title and contact information for corresponding author. Some fields are mandatory, others are not.
 - 3) create a temporary file for cover-letter information, lock the selected file to disable re submissions and send an automatic receipt to corresponding author. Print a message on the screen saying where the

automatic receipt was sent and that a manual verification should be expected, usually within a day or two.

- internal verification script, used to check the incoming abstracts/articles before they were moved to our database.
- collecting script, independent and inaccessible from the outside, in charge of safely moving submissions from the world-writable ftp area into the protected spooling queue.
- various utility script for consistency checks, reception status and statistics, review collection and message generation/distribution.

Appendix B: Internal checks

Several consistency checks should be performed as the users complete the electronic cover-letter.

- a) complain loudly and return if form is incomplete, the given "password" is incorrect or the reference number is invalid or previously rejected by review.
- b) complain loudly, print an error message and return if the user selection is obviously incorrect, for instance picking a directory or the README file instead of a Postscript file from the ftp directory.
- c) complain loudly and ask the user to contact the administrator by mail if the submission is identified as an "update", meaning it will replace any existing file in the database if processed as requested. Such files are NOT copied into the spooling queue to avoid possible mistakes by whoever is doing the manual inspection.
- d) complain softly, print a warning message, but continue if the selected file fails a few simple tests. Typically, we check for pure text, binary files or anything with zero size indicating error(s) during ftp transfer. The file is however still accepted and later processed; This way, the casual user will manage to submit an incorrectly generated file and then, in turn, get specific feedback from us. On the other hand, some experienced users might catch and correct their error(s) directly without our intervention.

⁷ Common Gateway Interface

⁸ as documents are further processed by our secretary and no direct access is permitted, this only serves as a rough filter against spoofing and typographic errors

SGI Thermodynamics Tutorial Available

Kenneth R. Jolls, Iowa State University

We have used the "Showcase" feature on Silicon Graphics IRIS workstations to produce a phase-equilibrium slide show. The tutorial contains computer-generated phase diagrams for binary, ternary, and quaternary fluid-phase systems and includes simple and azeotropic VLE and one example of ternary LLE. Data were generated using an equation-of-state option set with the various flash routines of ASPEN PLUS. Open Inventor graphics software was used for the three-dimensional visualizations (binary PT_x-y, ternary composition prisms, and quaternary tetrahedrons). Color and transparency are used to distinguish the components of the diagrams.

Approximately fifty text slides accompany the drawings to explain the thermodynamics and also the visual presentation. Text and graphics appear in a pre-programmed order, and the viewer can either follow that order or jump to particular sections.

There are both static images (such as the ternary LLE example shown here) and also movable images. The latter can be zoomed and rotated (and in some cases sectioned) to allow

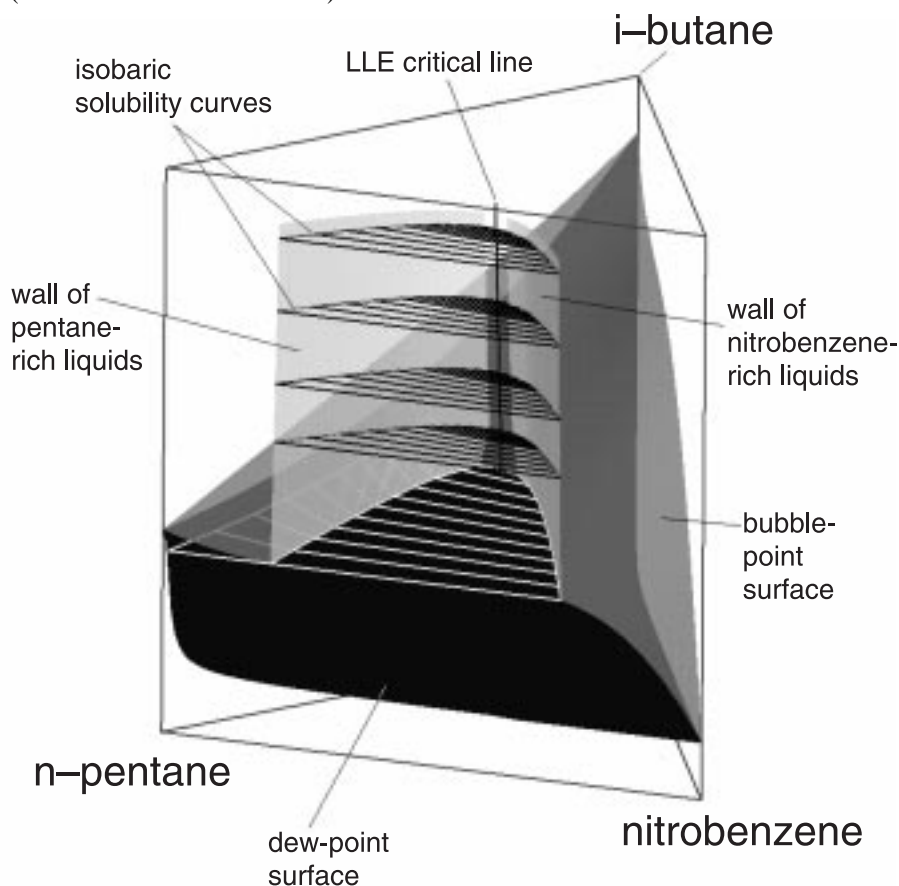
students to experiment with the drawings and reinforce their perception of the subject.

We are willing to offer this tutorial at no cost to teachers who can use it in their courses and who have the Silicon Graphics equipment needed to run it. Making it available to students is an effective way to alert them to the visualizability of thermodynamic information and to its basic hyperdimensionality.

Please contact Professor Jolls if you are interested in receiving the tutorial. Files can be retrieved via ftp and will contain instructions for use.

Kenneth R. Jolls, Chemical Engineering Department
Iowa State University, Ames, Iowa 50011-2230
(515) 294-5222; Fax: (515) 294-2689
e-mail: jolls@iastate.edu

or contact: Kong S. Tian, HTRI, Inc. College Station, Texas,
e-mail: kst@htri-net.com



Isothermal Pressure-Composition Prism