Developing Software to Broaden Usage and Lower Barriers

William F. Polik
Hope College
MU3C Online Conference, 2018
Outline

- Six Software Development Stories
- Lessons Learned
- Research Opportunities
- Discussion
Programming Projects and Hope Student Collaborators

- PCscan (1988-1992) / PyScan (2017-pres)
  - Brian Vroon ‘90, Rychard Bouwens ‘94, Peter Timperman ‘18, Julian Payne ‘19
- Discus (1997-2002)
  - Kevin Paulisse ‘99
- WebMO (2001-pres)
  - JR Schmidt ‘01, Jeff Largent ’11, Nathan Vance ‘17
- Build Your Own Cluster - BYOC (2004-2016)
  - Mike Poublon ’06, Nathan Vance ‘17
  - Nathan Vance ‘17

Turbo Graphic Tools is a Turbo Pascal graphics library which overcomes the hardware differences of the Hercules, IBM jr, IBM CGA, and IBM EGA graphics adapters. It is intended for programmers who wish to write transportable Turbo Pascal graphics programs capable of running under both the IBM and Hercules graphics standards.

New area filling algorithm \((n^2 \rightarrow \sim n^1)\)

The simplest seed filling algorithm

```
procedure Fill(x,y,NewColor: Integer);
var
    EraseColor: Integer;

    procedure RecursiveFill(x,y: Integer);
    begin [of procedure RecursiveFill]
        if PD(x,y) = EraseColor [if point needs filling...]
        then
            begin
                DP(x,y,NewColor); [fill it]
                RecursiveFill(x-1,y); [call fill algorithm with neighbors]
                RecursiveFill(x+1,y);
                RecursiveFill(x,y-1);
                RecursiveFill(x,y+1);
            end; [of procedure RecursiveFill]
        end; [of procedure Fill]
```

Run and its shadows

```
Figure 1.
```

New area filling algorithm ($n^2 \rightarrow \sim n^1$)

Licensed to and sold by UC Berkeley
PCscan (1988-1992) / PyScan (2017-pres)

Brian Vroon ‘90, Rychard Bouwens ‘94
PCscan (1988-1992) / PyScan (2017-pres)

ScanningInstrument Control and Data Acquisition
PCscan (1988-1992) / PyScan (2017-pres)

Peter Timperman ‘18, Julian Payne ‘19
Discus (1997-2002)

Kevin Paulisse ‘99
Discussion boards were a successor to listservs; Discus architecture included html, scripts, and data, with a web interface.
Discus (1997-2002)

Discus was the top-rated and 4th most popular of all CGI scripts in 1999.
Discus (1997-2002)

The internet + “viral marketing” + free software was an extraordinarily successful marketing plan
WebMO (2001-pres)

Text input files were extraordinarily difficult to create; needed to know unix to run jobs
WebMO (2001-pres)

JR Schmidt '01
WebMO Additional Features and Platforms

Jeff Largent '11

Nathan Vance ‘17
WebMO Software Usage

12.5 Gaussian 03 and WebMO Are the Quantum Chemistry Programs Most Commonly Used by Undergraduate Chemistry Students

Donald A. McQuarrie, Quantum Chemistry, 2nd Ed., 2008
Build Your Own Cluster - BYOC (2004-2016)

MU3C Cluster

Mike Poublon ’06, Nathan Vance ‘17

- **Build Your Own Cluster (BYOC)**
  - Cost: $680,000 for MU3C
  - Barriers: upfront cost, maintenance, expertise

- **Server On A Stick (SOAC)**
  - Cost: $4
  - Barriers: distribution, hardware speed, device needed

- **Server In The Cloud (SITC)**
  - Cost: $25 /mo
  - Barrier: limited to single computer servers

- **Cluster in the Cloud (CLIC)**
  - Cost: 3.4¢ / hr
  - Barrier: ???

Compute nodes are dynamically created and deleted; architecture remains versatile

**Job Submission Rate and Queue Length**
- Jobs Queued
- Job Submission

- Wait times: 0-12 minutes

**Jobs and Nodes Running**
- Jobs Running
- Nodes Up

\[
\int \text{Jobs Running} = 41 \text{ hrs} \quad \int \text{Nodes Up} = 54 \text{ hrs} = \$1.83 \quad \text{Efficiency} = 76\%
\]

CONTENTS
NOVEMBER 2017
ISSUE 283

FEATURES
78 Rapid, Secure Patching: Tools and Methods
   Control a heterogeneous server farm with the SSH agent.
   Charles Fisher

100 CLIC: Cluster In the Cloud
   Automatically create and delete high-performance computing resources in the cloud.
   Nathan R. Vance and William F. Polik

Nathan Vance ‘17
Computing Publications with Hope Student Co-Authors


Lessons Learned

• Address a need (not just a cool idea)
• Strive for platform independence and portability (hardware, OS, languages)
• Don’t re-invent the wheel (Be a hacker!)
• Know OS’s, scripting, application languages, algorithms, communication, database, web design, ... (and never stop learning)
• Work in your area of knowledge (or develop it)
• Follow new technology, but port only if it offers a distinct advantage (www, cloud)
Lessons Learned

- Low-cost is better (and free is best)
- Focus on ease of installation (none is best) and distribution (on demand)
- Many minor releases is better than few major releases (Discus stalled; WebMO is going strong)
- Develop a distribution / publicity / marketing strategy (hire out)
- Consider the novice user in your design (all users start as novices)
- Use your own software (big fixes, improvements)
Lessons Learned

• Have a good information retention and backup system (for software re-use and disaster prevention)
• Seek general solutions and look for synergy among projects (TGT -> PCscan; Discus -> WebMO)
• Be persistent (projects can take 5 to 10 years to develop)
Current Research Opportunities

- Instrumentation interfacing
- Computational chemistry
- System administration
- Cloud computing
- Web design and marketing
Acknowledgements

Brian Vroon '90
Rychard Bouwens '94
Kevin Paulisse '99
Ruud Van Ommen (exchange)
JR Schmidt '01
Ben Ellingson (NSF-REU)
Eli Scheele '06
Mike Poublon '06
Jeff Largent '11
Joshua Kammeraad '14
Nathan Vance '17
Peter Timperman '18
Julian Payne '19
...and many more Hope students!!!