

Just for the Kids School Reports: A data-oriented web site powered by SAS

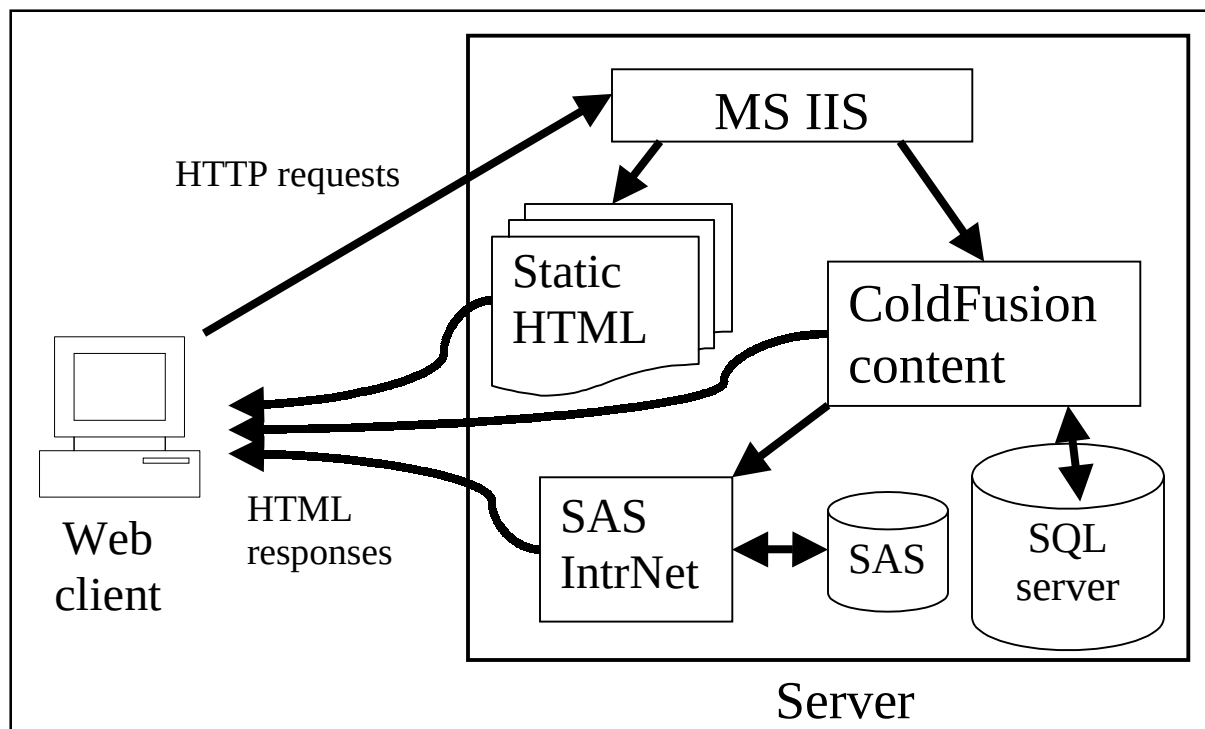
Bob Zambarano

National Center for Educational Accountability, Austin

Introduction

In 2001, the Education Commission of the States, The University of Texas at Austin, and Just for the Kids joined together to create the National Center for Educational Accountability (the Center). Among other things, the Center provides on the Internet a set of reports designed to compare schools on their students' standard test scores, (<http://www.just4kids.org>). The school reports were published on the web starting in the late 1990s by Just for the Kids (JFTK), which was founded in 1995 as a nonprofit Texas organization. Initially, the reports were just for Texas, but we now provide reports for any state that will provide us with the necessary data.

These reports enable parents and educators and other interested individuals to search for and select any public school from a participating state and view a set of reports detailing the performance of that school on the state's standard tests. Each school's performance is compared to the best performers among a set of comparable schools. Comparability depends on the characteristics of the student population; a school is only compared to schools serving similar or more disadvantaged students. Consistently high performing schools are also highlighted, with an additional set of reports providing comparative indicators for the selected school. The results from over 10,000,000 students attending over 60,000 schools are available at this site, which is still growing; we expect to have well over 40 states' data before summer's end, 2004.



We will not be reviewing the content or technical details of this site, but rather the project-level and operational processes and procedures, which are built primarily around SAS.

The site is served by Microsoft IIS, which calls ColdFusion services for any dynamic content. ColdFusion in turn calls SAS IntrNet services for generation of all school-specific charts and tables. The SAS IntrNet service calls a variety of SAS reporting and charting PROCs, which use ODS to dynamically generate web-ready output. The data is accessed from an indexed SAS data set, stored separately for each state. A relatively small amount of data is stored in MS SQL Server and accessed directly by ColdFusion to generate dynamic menus and pick-lists. Data is acquired from the states in a wide variety of storage formats; data processing is performed using SAS Base PROCs and macros. Where possible, we have embedded standard SAS code in SAS AF applications. All processing arrives at the standard SAS data storage structures.

We'll start by discussing the data we work with. From there we'll look at our data processing methods. Next we'll review the web service design and development process. Finally we'll close with a discussion of our staffing model for this system and our attention to quality assurance.

The Data

Our site presents information about the percent of students passing standard tests. For each state we break this out by:

- The school
- The year of the test

- The grade in school
- The subject tested
- The passing level achieved

School	Year	Grade	Subject tested	Passing level	Percent Passing
Lee Elem	2002	3 rd	Math	Basic	23%
Lee Elem	2002	3 rd	Math	Proficient	12%
Lee Elem	2002	3 rd	Reading	Basic	36%
Lee Elem	2002	3 rd	Reading	Proficient	21%
Lee Elem	2002	4 th	Math	Basic	...

Most states provide us the data already aggregated from individual students to the level shown above, but some give us student level data that we aggregate on our own. We actually prefer this as we can do much more with student level data. For these states, we are able to break the data out further by:

- The number of years that the students were enrolled at their current school
- The achievement level of the students before they enrolled at their current school

We process and present much more information, but this represents the core. We present each school's test results in comparison to the best performing comparable schools for the current year, and in comparison to consistently high performing schools.

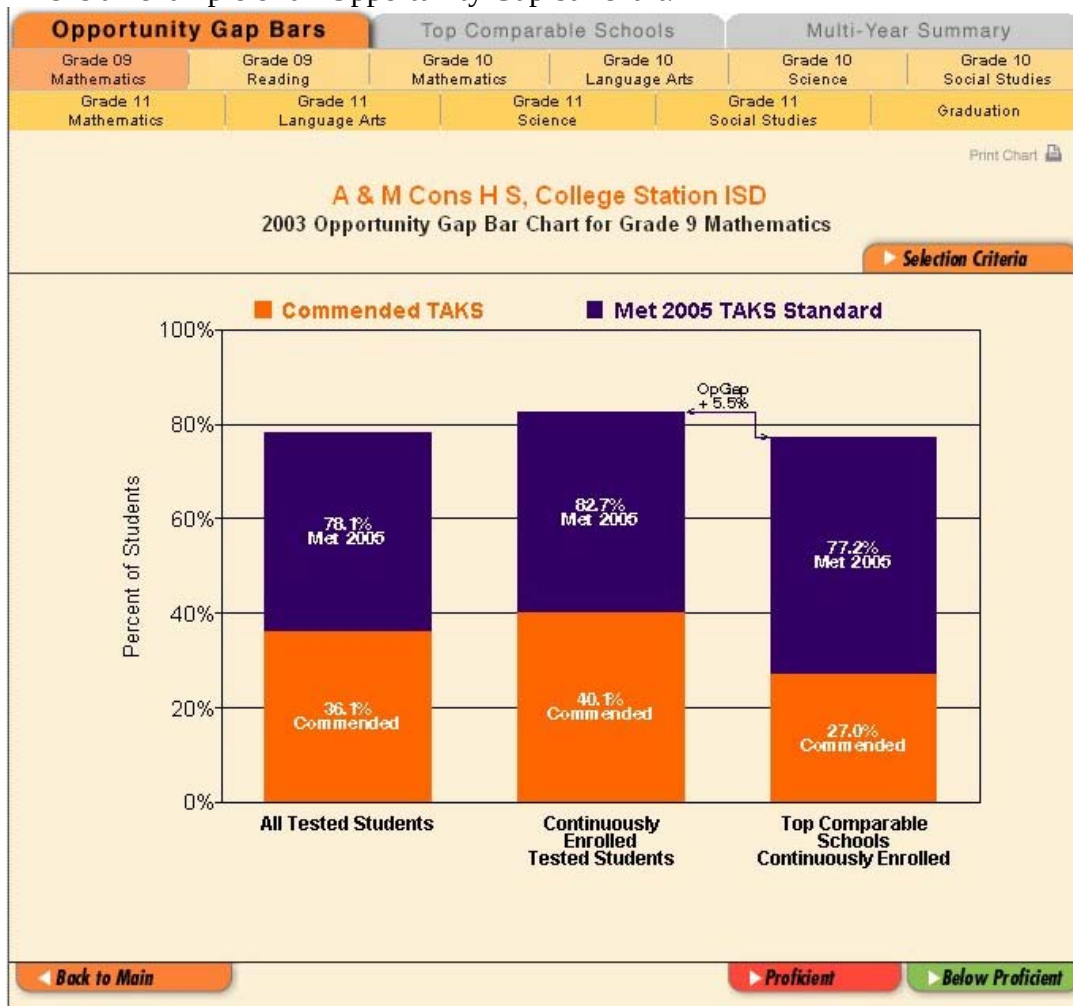
For current year comparisons, comparability is based on:

- The percent of economically disadvantaged students at the school
- The number of students tested in the grade
- The percent of English language learners at the school (if available)
- The percent of students tested in the grade (if available)
- Other comparability indicators as available and as requested by the state

For each school and grade, a pool of comparable schools is selected based on the above bulleted criteria. For each subject and for one or more selected achievement levels, we select the set of top performing schools and generate:

- The weighted average percent of students who passed
- The difference between the percent passing at the selected school and the weighted average of the top comparable schools. This is referred to as the Opportunity Gap.

This is an example of an Opportunity Gap bar chart.



Here is the table of Top Comparable schools' data.

Opportunity Gap Bars		Top Comparable Schools				Multi-Year Summary			
Grade 09 Mathematics	Grade 09 Reading	Grade 10 Mathematics	Grade 10 Language Arts	Grade 10 Science	Grade 10 Social Studies				
Grade 11 Mathematics	Grade 11 Language Arts	Grade 11 Science	Grade 11 Social Studies	Graduation					
Print Chart 									
Main Charts - Standard Picture A & M Cons H S, College Station ISD 2003 Top Comparable Schools for Grade 9 Mathematics									
▶ More Detail		▶ More Detail II				▶ Selection Criteria			
School Name (Grade Span) District Name		Continuously Enrolled (Tested on TAKS)					Schoolwide	Attrition	
		% Meeting 2005 Standard or Above	%Meeting Commended Standard	% Meeting 2003 Standard or Above	Average 8th Grade Test Score	Number of Students	%Low Income	% Dropout or Missing	% Retained
A & M Cons H S (09-12) College Station ISD		82.7	40.1	91.3	88	496	20.2	1.0	0.0
A & M CONS H S (09-12) COLLEGE STATION ISD		82.7	40.1	91.3	88	496	20.2	1.0	0.0
LIBERTY J H (07-09) RICHARDSON ISD		81.3	24.1	91.6	85	203	38.8	1.0	0.0
CYPRESS FALLS H S (09-12) CYPRESS-FAIRBANKS ISD		76.5	24.2	91.3	86	620	23.7	1.1	0.0
GEORGETOWN 9TH GRADE (09-09) GEORGETOWN ISD		75.3	26.0	86.5	87	542	29.7	0.9	0.0
DEER PARK H S - NORTH (09-12) DEER PARK ISD		74.6	22.3	89.7	88	759	22.8	1.1	0.0
Average of Top Comparable Schools		77.2	27.0	89.9					
Opportunity Gap		5.5							
Number in Pool		127							
◀ Back to Main		▶ Proficient			▶ Below Proficient				

In contrast, consistently high performing schools are picked using a WLS regression methodology that predicts percent passing rates from comparability indicators including those bulleted above. The regression model is used to generate standardized residuals, which are ranked within groups defined by demographic indicators. Consistently high performing schools are those with consistently high residual rankings for all subjects for three years. For the selected school, a person can see the residual values and ranks generated for each grade and subject and determine where their school's performance does not measure up to consistently high performing schools.

Here is an example of a high performing school table showing the criteria to be considered a high performing school, and how the selected school compares against those criteria.

A & M Cons H S (09 - 12)
College Station ISD
 2001 - 2003 High-Performing Criteria Table

Identification Criteria

The required criterion is in black, and your school's score is in either red or green (green indicates that this criterion was met, red indicates that it was not).

High-Performing Criteria			2001			2002			2003			Your School		
Subject	High-Performing Criterion	High-Performing Criterion Requirement	9th	10th	11th	9th	10th	11th	9th	10th	11th	Overall Performance Rank (66.6+)	Did they make AYP for subject?	Did they meet all the criteria for this subject?
Reading	Individual Rank	40+	n/a	37	n/a	n/a	20	n/a	50	n/a	n/a	n/a	Yes	No
	Proficiency Rate		n/a	89.9	n/a	n/a	90.9	n/a	87.6	n/a	n/a			
	Participation Rates	85%+	n/a	91.1	n/a	n/a	89.0	n/a	92.3	n/a	n/a			
	Size of Groups	10+	n/a	385	n/a	n/a	382	n/a	493	n/a	n/a			
Mathematics	Individual Rank	40+	n/a	82	n/a	n/a	84	n/a	95	98	97	93.3	Yes	No
	Proficiency Rate		n/a	84.6	n/a	n/a	87.2	n/a	82.7	86.0	78.9			
	Participation Rates	85%+	n/a	91.1	n/a	n/a	89.0	n/a	93.8	90.4	87.6			
	Size of Groups	10+	n/a	383	n/a	n/a	383	n/a	496	387	342			
Language Arts	Individual Rank	40+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	10	94	50.4	n/a	No
	Proficiency Rate		n/a	n/a	n/a	n/a	n/a	n/a	n/a	80.7	85.5			
	Participation Rates	85%+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	90.8	87.1			
	Size of Groups	10+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	389	346			
Writing	Individual Rank	40+	n/a	98	n/a	n/a	50	n/a	n/a	n/a	n/a	n/a	n/a	No
	Proficiency Rate		n/a	61.6	n/a	n/a	60.4	n/a	n/a	n/a	n/a			
	Participation Rates	85%+	n/a	91.1	n/a	n/a	89.0	n/a	n/a	n/a	n/a			
	Size of Groups	10+	n/a	380	n/a	n/a	379	n/a	n/a	n/a	n/a			
Number of subjects considered high performing:														
Was this school considered high-performing overall?														

Data Processing

Hardware, Software and file system configuration

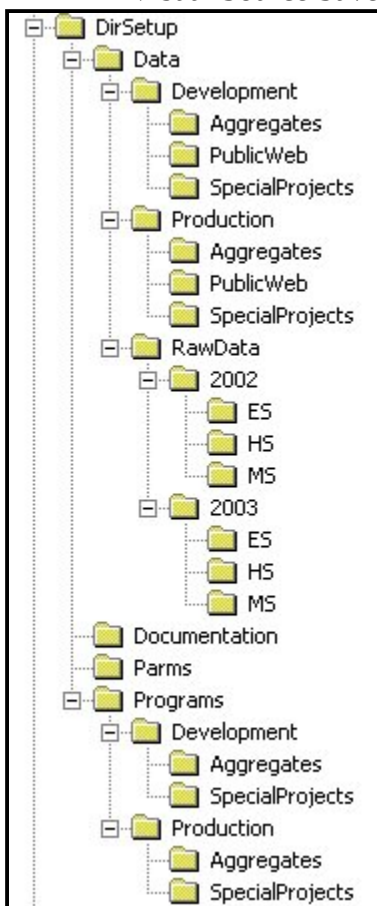
For performance and security reasons, data processing occurs on a physically separate system than web services. Programmers use Citrix clients to connect via their desktop systems, (or from home using VPN), to either of two MetaFrame servers where SAS data processing occurs; version 8.2 is installed on both. Both processing machines access the same shared file system managed by a third machine. All systems run Microsoft operating systems (XP or Server 2000).

All data processing steps are performed using SAS. SAS programs and data are created in a state-specific project directory, with identical subdirectory systems for development and testing, and for production. We use Microsoft Visual Source Safe (VSS) to maintain version control of SAS programs and project documents and to keep version histories in cases where we need to revert to prior code.

Data processing steps

The first processing steps involve obtaining data and setting up the project file storage systems, converting source raw data to raw SAS datasets if necessary, and checking data for completeness and reliability. A state programming project starts with data acquisition. We receive data from state education agencies. We accept data in whatever format the state provides. The most common formats are XLS, tab or comma delimited text, column-formatted text, MS Access database, SPSS SAV or POR files, as well as SAS datasets or transport files. SAS's ability to read multiple data file formats is a strength we rely on. Here is an outline of the specific steps:

- Data is received and data acquisition staff logs receipt.
- A programmer is assigned.
- A QA review programmer is assigned.
- Visual Source Safe (VSS) administrator is notified.



- VSS administrator creates a new project directory with matching development and production subdirectory systems. (See image on left.)
- VSS administrator creates and configures new VSS project folder.
- Programmer copies source data to project raw data directory.
- Programmer attempts initial programming step of checking, cleaning and converting received data from original file format to SAS. Data already in SAS format is programmatically reviewed.
 - SAS programs and data are created and tested in development directories.
 - Data problems are reported. Data retransmission is requested if necessary. Process steps are repeated as necessary.
 - When programmer is confident that raw data has been checked and is acceptable, the data cleaning and review program is checked into VSS and the QA reviewer may be notified.
 - Programmer proceeds to next phase.
- Review programmer reviews code against specifications, checks data, performs independent programming checks as deemed necessary. If problems are found, project programmer checks-out

program from VSS and programmer and reviewer repeats phase steps as necessary.

The next processing steps involve generating project programming specifications and then converting the raw data to conform to these specifications and our standard pre-processing structures. As mentioned in the prior section on data, data is either provided at the student level, or aggregated for each grade and subject tested at each school for a given year. We have standard data structures for both student-level and aggregated data.

Often the most complex processing task is transforming the raw data structure to our standard data structures. Here is an outline of the specific steps:

- Programmer develops a draft specifications document based on instructions from state, current standard practices and characteristics of the data.
- Specifications document is checked into VSS.
- Specifications are reviewed by Director of Programming. Programmer is given feedback and specifications document is updated as needed.
- Using the programming specifications document and standard data structures, programs are created to transform the data.
- As each major transformation program is completed to the programmer's satisfaction, programmer checks program in to VSS and the QA reviewer is notified. The programmer proceeds to next program.
- Review programmer reviews code against specifications, checks data, performs independent programming checks as necessary. If problems are found, project programmer checks-out program and programmer and reviewer repeat the steps of this phase steps as necessary.

The next processing steps are generally simpler – we have standard program templates we use for even the most complex state data processing requirements, and most states' data can be run through a set of highly parameterized macros that have been embedded in a SAS AF application. This application's GUI provides a tabbed dialog-box enabling configuration of macro variable values as appropriate for any given state. We call this application the Go-Go Gadget; it outputs a standard campus summary dataset for a given year. The campus summary contains all the information needed for specific grade, year and subject comparisons, and contains all that is needed to run the regression model for selecting consistently high performing schools. Each available year of data is finally concatenated into a single dataset. At this time, a final pass is made to omit data that would be inappropriate for web publication; data values are set to missing for data rows that do not meet a state's minimum group size requirements for protecting individual student's privacy, as specified in the Family Educational Rights and Privacy Act (FERPA).

Here are the three configurable tabs of the Go-Go gadget. Configuration options are to some extent dynamically based on other options selected – some options are mutually incompatible.

GO-GO GADGET AUTOMATER

General	Achievement Levels	Filters
State: TX	School Type: Elementary School	Year: 2003
Data Type: ☒ Longitudinal ☐ Snapshot ☐ Combination ☐ Include grade 6		Measure: TAKS
Select students to be included in Bar 1 (Tested) Group: ☒ Non-Special Ed Tested ☐ Absent Untested ☐ LEP Untested ☒ Special Ed Tested ☐ Spec Ed Untested ☐ Other Untested Minimum Students: 10		Mode: ☒ Development ☐ Production
Define Bar 2 (CE Tested) Group Parameters: Continuous Enrollment: ☒ Standard ☐ Custom Max Years: 3 Minimum Students: 5		Help
Run Previous Options Close Help		

GO-GO GADGET AUTOMATER

General	Achievement Levels	Filters
Current Achievement Levels: ☒ Level 3 clev3 ☒ Level 2 clev2 ☒ Level 1 clev1 ☒ Level 0 clev0 Current levels:		Sort Order: Sort By: level23 level3 Sort Variables: level1 level2 level1 2 level1 23
Help		
Run Previous Options Close Help		

GO-GO GADGET AUTOMATER

General | Achievement Levels | Filters

Comparable School Selection Filters

Comparable Schools: = 10 ☒ Include Magnet Schools ☒ %Tested >= 85

☒ % School Size >= 40 ☒ % Group* Size >= 40 ☐ % CE Tested >=

☒ % Grade Size >= 40 ☒ Min Group* Size >= 10 ☐ % Untested <=

* Group: Snapshot data - number tested (NTEST) Longitudinal data - number CE tested (NCETEST)

% Economically Disadvantaged

☒ Schoolwide Ceiling 90 ☐ Backoff ☐ Phase-in

☐ Student Group Ceiling ☐ Backoff ☐ Phase-in

% Limited English Proficiency

☒ Schoolwide Ceiling 75 ☐ Backoff ☐ Phase-in

☐ Student Group Ceiling ☐ Backoff ☐ Phase-in

Help

Run Previous Options Close Help

- For standard school level data or standard student level data, the Go-Go Gadget data transformation system is used to create a standard campus summary dataset.
- Other projects require custom programming as per specification. SAS programs and data are created in development area.
- Review programmer reviews code or Go-Go project settings against specifications, checks data, performs independent programming checks as deemed necessary. If problems are found, project programmer checks-out program and programmer and reviewer repeats phase steps as necessary.
- A program is created to concatenate current year campus summary data with prior years data. Data codes for generating web chart messages for problematic data are assigned. (Example: selected school's students are so disadvantaged, no comparable schools exist.)
- A program is created to generate a campus and district names dataset, containing all necessary information for web.
- A program is created to run the data codes macro, which assigns project specific messages for use with data codes assignments.
- A program is created to run the data stripper macro, in order to mask data values where number of students is below state threshold for ensuring student privacy. This program generates the final version of the campus summary dataset, along with composite index, typically based on school, grade, subject and year.

- As each program is completed to the programmer's satisfaction, programmer checks program in to VSS and the QA reviewer is notified and programmer proceeds to next program.
- Review programmer reviews SAS code against specifications, checks data, performs independent programming checks as deemed necessary. If problems are found, project programmer checks-out program and programmer and reviewer repeats phase steps as necessary.

Finally, all datasets are reviewed using a set of standard checks. Checking is performed using a system of standard macros and a database of logical vetting checks. This database grows each time a new checkable data error is detected by visual review. These processes are embedded in a SAS AF application that enables project specific configuration and generation of PDF reports. We call this application the Snazzy Checker.

Results are available for review by programmer, QA review programmer, and Director of programming. Results include:

- Model diagnostics: several pages of standard distribution statistics
- Volatility diagnostics: distribution of changes in summary data from year to year
- Gap diagnostics: summary tables of Opportunity Gaps across the years.
- Row level checks: a set of standard comparisons of values within each row of the summary dataset. Example: sum of the number of students passing the

test at each achievement level should equal the total number of students tested.

- Aggregate checks: comparisons across rows of standard data structures or against column aggregate summaries. Example: the percent of low-income students for a school should not differ among records for the same school.
- When all checks are run, the review programmer is satisfied with all programs and their output, the system of programs is run and data generated in the production area.

Web system design and configuration

The final result of data processing is a set of four data files:

- A dataset with a summary record for each school, year, grade and subject,
- A dataset with a set of state-specific messages to apply to data codes,
- A dataset containing general school information such as the school name, the name of the district, the grade span served, and so on,
- A dataset containing achievement results as compared to consistently high performance criteria.

As mentioned, data processing occurs on one cluster of machines while web services are provided from separate systems. Files must be manually transferred from one system to the other. Once transferred, the file containing general school information is loaded in SQL Server – this information is used by ColdFusion to generate dynamic content for navigation menus.

There are actually two web sites: a password protected site for internal development, testing and review, and a public site. Again, for security, stability and performance reasons, these systems are built on separate machines. The sites are essentially mirror images except content appears first on the development site and is moved to the public site after testing and review.

The basic configuration of the web servers is as follows: The operating system is Windows Server 2000. Microsoft IIS is used as the primary web service: it serves all static HTML pages directly. For ColdFusion content, the IIS calls the ColdFusion service, which dynamically generates pages for display. For this dynamic content, ColdFusion contacts SQL Server for data to populate dynamic menus and pick-lists. Selecting one of these dynamic menu choices passes a call to the SAS IntrNet Broker, providing information such as the selected school ID, grade, subject and chart-type. The Broker is configured to work with the IntrNet Load Manager to manage a minimum of two concurrent SAS Application Server sessions, with additional sessions created or closed as appropriate. The SAS session processes the request and generates HTML or .GIF charts.

Web development process

When data processing is complete, the chart programmer moves the data to the development server and loads the general school information data to SQL Server. A set of standard SAS programs is configured to generate all the necessary SAS charts and tables. (See the example charts and tables shown earlier.) These programs rely heavily on SAS Macro language and so configuration primarily involves setting values for Macro

parameters, but some state specific configuration is often required as well. ODS is used to generate the output, (and the same core programs used to create the HTML results are also used in an AF application that generates PDF for training sessions). For state projects that follow standard presentation formats, very little actual programming is required, but in many cases there are non-standard presentation requests which requires creation of new or modified charting programs. This is done on an as-needed basis and the development process depends on the requirements.

While the SAS charts and tables are programmed and configured, the ColdFusion pages that call them are created along with any state-specific static HTML pages. The ColdFusion pages are built from standard template files.

Once the site is up and running on staging a final page check review is performed. Final page checks involve the following:

- A random set of schools is selected for each school level (elementary, middle and high), for each subject tested, and each data code assigned. For these schools, a complete review of the charts and tables is performed. An extensive set of check lists is used.
- All static content is reviewed
- All dynamic menus are checked
- All links are followed

When this review is completed, the code and data is copied from the development server to the production server and the production site is launched.

Closing

Staffing details

This operation is currently well staffed. We have:

- One full time staff responsible for data requests, acquisition, problem tracking, updates, and so on.
- Three full time staff responsible for data processing
- One full time staff responsible for developing high performing school data
- One full time staff responsible for Chart development and generation
- Two full time staff responsible for web development and system administration
- One full time staff responsible for quality review
- One half time staff responsible for process development and maturation.
- One full time project and operations Director

During peak load periods (late fall, early winter), we have brought on up to one FTP contract programmer for data processing.

Quality management

We have two specific quality-checking phases, (data checks and page checks), all the process phases have quality assurance built in. For all phases of work we have

version control, a separate project development and production environment, standardized programs and processes and either peer review or formal product checks.

Capability maturity

There is always room for improvement. As I write, we are planning a redesign of the site from the ground up. Along with the obvious focus on improving usability, we will be incorporating new quality assurance procedures into the design. But notice that we are very comfortable with our current software and hardware technology choices; these are not being revisited. SAS does the job for us now and for as far into the future as we can imagine.