Student Block Assignments: a data driven algorithm

Saving Time and Resources in the Creation of Block Assignments for Dental Students

Author: Kathy Martin, Senior Systems Analyst
Purpose:

-an efficient and flexible approach for the assignment of students to block assignments (rotations) using a data driven algorithm.
Abstract:

Each Dental School has an approach for creating a schedule for students to rotate through the myriad of specialty clinics and experiences (Block Assignments), i.e., Oral Surgery, Pediatric, CPR class, etc. Your school has probably tried them all:

- butcher paper
- fancy excel manipulations
- 'teams'

The process is usually done by a team of people who have been doing it ‘forever’ or that requires hours of the school’s IT department, costing valuable, scarce resources.

By developing a data driven approach with a simple scheduling algorithm, the student and calendar information as well as basic parameters of each specialty are entered into data tables. This straightforward data entry can be done by any data entry staff person. The main algorithm is then easily manipulated by one non-IT staff person with minimal training.

By creating an algorithm that can be run repeatedly, the assignments are created to fill each specialty for the duration of the assignment. The algorithm is run for the longer assignment periods first, then each successive block is added to the schedules. 'Minimal clinic' sessions are also created between assignments. Holidays and Non-Clinic sessions are easily set in the data tables and skipped by the algorithm. Special situations or priority assignments can be hand entered first, which the algorithm then knows to skip.
Data Tables are the Central Communication Mechanism

Data Tables contain incoming:
- parameters
- block and student identification data sets
- calendar data sets
  which are acted upon by the
- algorithm
  and are used for outgoing information for
- exported block ‘appointments’
- quality assurance
- reports

Data Types Selected to ‘fit’ the Simple Algorithm

Data Types
- dates
- alpha-numeric codes
- bit maps and pointers for calendars
Select Simple but Complete Data Tables and Elements

- Students: class, clinic
- Calendars:
  - Academic Calendar: Is session open?  
    - days sessions on/off, weekends, summer, holidays  
  - Assignment Rotation Calendar: Needs for each block session  
    - Assignment Code and Description  
    - Start and Stop dates;  
    - # student/class; need full week or accept partial?  
    - # sessions per each rotation  
    - # rotations (CPR: once; Emergency: cover all open sessions)  
    - total number of sessions required  
    - days/sessions students are needed  
  - Student Calendar: Is student available?  
    - each student has own calendar  
  - Other: ‘guaranteed’ sessions between assignments

Put everything that might change in a data table.
Special ‘Bitmap’ Data Type for Calendars:

- String of 732 characters, where each character represents a date/session
- Day/session pointers for each calendar, stay in sync (same date/session)
  - Meaning of values of bit map characters depends on the calendar
    - **Academic** - Is the session open?
    - **Block** – Is the block open that session? If so, how many students are needed?
    - **Student** – Is the student available for that block?

Fits simple ‘next session, next day’ algorithm
**Academic Calendar: Is the Session Open?**

Day/Session legend:
- **ON** = A/B for Summer session selection;
- **Y** regular clinic year
- **OFF** = N for weekends, holidays, other clinic closures
Block Calendars: How Many Students Needed?

Is the block open that session? If so, how many students are needed?

Algorithm decrements assignment calendar as students are assigned.

Day/Session legend:
- N = OFF
- # represents students still needed to assign to the block
- 0 indicates session is FULL

Setup algorithm sets each unique Block Assignment calendar per class.

Block Calendar created from:
- Start and stop dates
- # students needed
- Full or partial week?
- # sessions per rotation
- # rotations required
- Total sessions required
- Days/sessions of need

ORAL SURGERY

<table>
<thead>
<tr>
<th>START: 8/17/2009</th>
<th>STOP: 4/30/2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon Tue Wed Thu Fri</td>
<td></td>
</tr>
<tr>
<td>Y Y Y Y Y</td>
<td></td>
</tr>
<tr>
<td>Y Y Y Y Y</td>
<td></td>
</tr>
</tbody>
</table>

SESSIONS BETWEEN BLOCKS: 6

COVERAGE COUNT? N

# SESSIONS: 10

# ROTATIONS: 2

SAME WEEK? Y

TOTAL SESSIONS:

SCHEDULE FOR THE SCHOOL YEAR: # Students Y=Open N=Closed

Block Assignment Code: G

Student ID: 0

Day/Session legend:
- N = OFF
- # represents students still needed to assign to the block
- 0 indicates session is FULL

Wednesday, April 7, 2010
Student Calendar: Is Student Available?

Each student has their own calendar. Initial student calendars are created with ‘.’ for each potential clinic session. As blocks are booked, calendar is filled with block or clinic-time codes.

Block Codes: ‘Letter code’ represents Block Assignment. (G=Oral Surgery)
Clinic-time Codes: ‘,’ /Comma represents ‘guaranteed’ clinic session between assignments.
‘.’ /Period represents regular clinic session that could be a block assignment.
Pointers keep Block and Student Calendars in Sync

### Academic Calendar

Pointers to each calendar point to the same day/session.
Many simple data manipulation routines

Block list:
- add/change/delete block; update block parameters

Student list:
- add/change/delete student
- get student from ‘next’ clinic
  - get next student

Academic calendar:
- Check if clinic is open

Block calendars:
- fill block
- are all blocks covered?

Student calendars:
- Check student availability
  - set student to block
  - are all students booked?

- summarize number of students per clinic per block (QA)
- summarize number of blocks per student (QA)

All calendars:
- convert 732 character sequence to monthly display
  - display 1 calendar (any type)
  - create initial calendars from parameters
  - support individual calendar adjustment
Repetitive Block Scheduling Process

After academic, block and student calendars are entered and verified:
- scheduler program is run discretely for each rotation of each block.
  - this allows access to the schedule developed thus far
  - to look for scheduling problems at their earliest conflict.

Problems found can be resolved by:
- modifying a few students schedules or
- clearing and restarting the block assignment

Based on policy, the algorithm can select the next student:
- from the same clinic or
- another clinic
Continuous use of the same data driven block algorithm through major operational changes:

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>Original version meets the needs of: 8 clinics, 1 floor, &gt;10 blocks per class, ’evenly distributed student assign by clinic’</td>
</tr>
<tr>
<td>2000</td>
<td>Y2K upgrade</td>
</tr>
<tr>
<td>2002</td>
<td>Compact clinics from 8 to 4 clinics</td>
</tr>
<tr>
<td>2003</td>
<td>Move to new Clinic Information System; Added export, import process</td>
</tr>
<tr>
<td>2004</td>
<td>Increase class size by 20 students</td>
</tr>
<tr>
<td>2006</td>
<td>Move to new Dental School: 4 clinics, 2 floors, added evening clinic</td>
</tr>
<tr>
<td>2006</td>
<td>Change to ‘student pull from 1 clinic’ approach</td>
</tr>
<tr>
<td>2007</td>
<td>Return to ‘evenly distributed assign by clinic’</td>
</tr>
</tbody>
</table>
Conclusion:

Use of a data driven approach has continued to be successful after 15 years, multiple clinic reorganizations, changing student rotation schemes, expansion of our student body as well as a move to our new Dental School.
Questions & Comments?

Kathy Martin
kmartin@umaryland.edu

University of Maryland