

THE BEAMLINER EXPERIMENTS SCHEDULING SOFTWARE*

Yuhong Yan[#], Ludeng Zhao, Zhiguo Wang, Yongxin Zhu, and Chun Wang,
 ENCS, Concordia University, Montreal, QC H3G1M8, Canada

Abstract

Scheduling the experiments to the beamlines of the synchrotron at the Canadian Light Source (CLS) is a manual procedure so far. Once every six months, the beamline scientists discuss before a whiteboard to schedule as many approved experiments as possible. There are so many constraints on resource capabilities, availabilities, user preferences, and experiment priorities to consider that none has ever been able to check if the manual scheduling results are optimal or not. In the Canarie funded project Science Studio, we are building an automatic scheduling module as a part of the User Office. After the synchrotron users submit their proposals via the User Office, the automatic scheduling module can find an optimal scheduling solution that satisfies all the constraints modelled, if such a solution exists, and display the results on a Web calendar. In this paper, we present our contributions on design and implementation of the scheduling module and our study on automatic scheduling of synchrotron experiments.

THE BACKGROUND

The automation of the scheduling activities at the CLS is part of the Canarie funded project Science Studio. The Science Studio project develops a complete experiment management system [1] that allows the researchers to control the experiment devices, observe the experiment processes, and collect data from their own home bases, instead of travelling to the CLS site.

There are about 30 plus the CLS like facilities [2] around the world. All the facilities have similar proposal approval procedures, regardless the different frequencies of calls-for-proposals and the length of minimal time slot. Scheduling the approved proposals is done manually. In the CLS, the beamline scientists who are in charge of scheduling experiments on the beamlines use e-mail and documents like spreadsheet and pdf as their primary tools to communicate with the users and manually scratch the schedules on a calendar. In order to make their lives easier, the beamline scientists tend to limit the possible combinations they should consider. Furthermore scheduling under conflicting constraints can easily become intractable as the number of users and proposals increase.

In this paper, we present our solution to automate the scheduling function. The User Office in the Science Studio platform has a proposal management module to manage the proposal submission and review procedure. In the CLS, the call-for-proposals occurs every six months

(aka. a cycle). The approved proposals can be scheduled into the next four cycles. Our schedule module reads in the proposal information, invokes the scheduling tool, and displays the scheduling results a Web calendar. The beamline scientists can review the schedule and manually fine-tune the schedule over the Web calendar. The design and the model of experiment scheduling can be reused for different facilities as well. The parameters of the scheduling model and the Web UI of the implementation are tuned for the CLS.

THE SYSTEM ARCHITECTURE

The Science Studio platform is a large J2EE enabled Web application. Figure 1 shows its system architecture. The core of the system architecture is the application tier composed by the User Interface (UI) services, the User Office, and the beamline services.

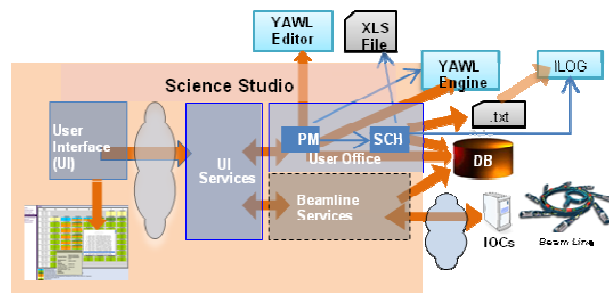


Figure 1: System architecture for Science Studio.

The major functions of the User Office are proposal management (PM in Fig. 1) and scheduling (SCH in Fig. 1). The proposal management module accepts user inputs and manages the proposal review process. The proposal review process can be executed by a workflow engine YAWL. YAWL Editor is used offline to design the proposal review process. When the approved proposals are decided, the scheduling function is invoked.

The scheduling module gets the approved proposals from the database or from the excel sheets currently used, and converts the information about the defined experiments into a text format that can be accepted by the ILOG - the automatic scheduler. A .txt file containing the converted data is the media of transferring the data between Science Studio and the ILOG. The ILOG is invoked by the scheduling module. The ILOG writes the scheduling results into a text file. The scheduling module reads the results and shows them on the Web calendar, meanwhile stores them into the database.

*Work supported by the Science Studio project which is funded by Canarie Network Enabled Platform Program contract number NEP-01
[#]yuhong@encs.concordia.ca

The UI service generates the Web UI. The users interact with the system through a common Web browser.

The beamline services offer the functions to observe the experiments and operate the physical devices from a Web browser. The underlying beamline control system is implemented using EPICS based Input/Out Controllers (IOCs) and with report access capability. This module is not the focus of this paper.

MODELING THE SCHEDULING PROBLEM

The experiment scheduling problem is modelled as an integer programming model.

Suppose the CLS has m beamlines $I = \{1, \dots, m\}$ and n approved proposals $J = \{1, \dots, n\}$. Currently a proposal defines one experiment. So we use experiment and proposal interchangeably in the following text. Experiments and beamlines are characterized by the following parameters which represent the clients' preferences and the scheduling constrains:

The cycle start time St defines when the synchrotron scheduling cycle starts. A cycle is six months, e.g. from 2010/1/1, 0:00AM to 2010/6/30, 12:00AM. Therefore, St can be the time point of 2010/1/1, 0:00AM.

The cycle end time Se defines when the synchrotron scheduling cycle ends.

The experiment unacceptable start time $Us[j, o]$, where $j \in J$, and $o \in \{1, \dots, Max1\}$, indicates when the o -th unacceptable period starts for the experiment j . $Max1$ is a constant that an experiment can define up to $Max1$ unacceptable periods. In the CLS, $Max1 = 6$.

The experiment unacceptable end time $Ue[j, o]$, where $j \in J$, $o \in \{1, \dots, Max1\}$ indicates when the o -th unacceptable period ends for the experiment j .

The experiment release time $R[j]$, where $j \in J$, is the earliest possible start time of the experiment j . Before that time, the experiment j cannot be scheduled.

The preferred end time $D[j]$ where $j \in J$ is the latest preferred finishing time for the experiment j . The experiment j should be scheduled before this time.

The processing time $P[j]$, where $j \in J$, is the time duration to complete the experiment j .

The weight $W[j]$, where $j \in J$, represents the priority given to the experiment j . Many factors can determine the priority for an experiment. For example, the proposals with biological samples have higher priority, and the commercial proposals have higher priority than the normal academic proposals.

The eligibility $E[i, j] \in \{0,1\}$, where $i \in I, j \in J$, is a Boolean value. When $E[i, j] = 1$, the experiment j can be conducted on the beamline i .

The beamline has some down time that is unusable for experiments. The beamline unusable start time $Ub[i, u]$, where $i \in I$, $u \in \{1, \dots, Max2\}$, indicates when the u -th unusable period starts. $Max2$ is a constant for the maximal number of unusable periods. The beamline unusable end time $Ua[i, u]$, where $i \in I$, $u \in \{1, \dots, Max2\}$, indicates when the unusable period ends.

The above variables contain the known facts of our model. The following variables are going to be assigned their values by the scheduling algorithm.

The experiment start time $S[j]$, where $j \in J$, is the scheduled time for starting the experiment j .

The assignment $X[i, j] \in \{0,1\}$, where $i \in I, j \in J$, is a Boolean value. $X[i, j]=1$ means the experiment j is assigned to the beamline i .

The scheduling has to adhere to the following rules:

Only the eligible beamlines can be selected:

$$\forall i, j, X[i, j] \leq E[i, j] \quad (1)$$

One beamline per experiment:

$$\forall j, \sum_{i=1}^m X[i, j] = 1 \text{ s.t. } X[i, j] \in \{0,1\} \quad (2)$$

The experiment start time should be greater or equal to the release time:

$$\forall j, S[j] \geq R[j] \quad (3)$$

The experiment start time should be greater or equal to the cycle start time:

$$\forall j, S[j] \geq St \quad (4)$$

The experiment end time should be less or equal to the cycle end time:

$$\forall j, S[j] + P[j] \leq Se \quad (5)$$

On a beamline, the experiments can't overlap:

$$\forall i, j, k, \text{ s.t. } j \neq k, k \in J$$

$$S[i, j] \geq S[i, k] + P[k] \vee S[i, k] \geq S[i, j] + P[j] \quad (6)$$

An experiment should be out of the unacceptable time window:

$$\forall i, j, u, S[i, j] + P[j] \leq Ub[i, u] \vee S[i, j] \geq Ua[i, u] \quad (7)$$

An experiment can only be arranged in the beamline available period:

$$\forall j, o, S[j] + P[j] \leq Us[j, o] \vee S[j] \geq Ue[j, o] \quad (8)$$

The objective of the problem is to minimize the total weighted lateness which is defined as the sum of time differences between the preferred end time of an experiment and its actual finish time. It is a criterion representing how much the schedule satisfies users' expectations in terms of users' preferred finish times.

$$\sum_{j=1}^J |W[j] * (S[j] + P[j] - D[j])| \quad (9)$$

SOFTWARE IMPLEMENTATION

The Web UI for the Calendar

The Science Studio platform has a common Web UI for all its modules (Figure 2). The left vertical bar shows the menu items, and the current item is the "automated schedule". A calendar in the content pane shows the schedule for a beamline.

The color encoding represents the different operation modes of the beamlines. For example, the normal mode is in green. Our scheduling application is implemented for two purposes: first, it supports manual scheduling by providing the beamline scientists with a calendar liked interface, on which the beamline scientists can define their facility operation modes and manually schedule eligible experiments onto the shifts; second, the scheduling application is able to invoke the automated algorithm resides the in ILOG and retrieve the results back for displaying them on its calendar UI. Figure 3 shows some of the manual operations. With a right click on a time slot,

a drop-down list pops up, containing all the schedulable experiments. The user can check the detail information of an experiment by right-clicking the experiment in the list.

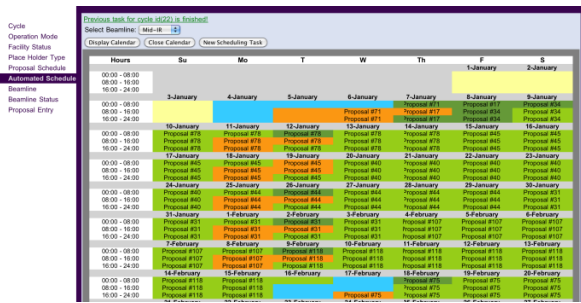


Figure 2: Example screen shot showing a calendar with schedule results.

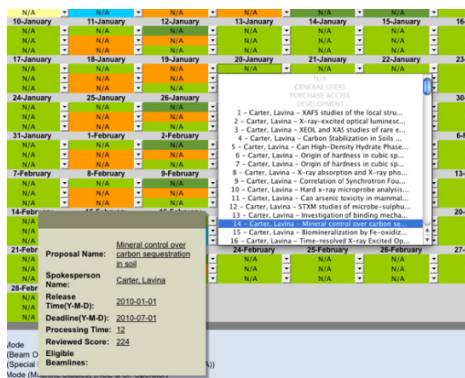


Figure 3: Example screen shot (enlarged) for illustrating some manual scheduling operations.

Testing with the real world data

We use the CLS proposals data for the first cycle of 2010 for testing the scheduling functions. A summary of these proposals is in Table 1. Totally 141 proposals by about 100 spokespersons from about 50 institutes are approved, most of which are from Canadian institutes.

Table 1: General user proposal summary

| Beamline | Total Requests | Total Shift Request |
|------------------|----------------|---------------------|
| 01B1-1 (Mid IR) | 5 | 77 |
| 02B1-1 (Far IR) | 15 | 258 |
| 06B1-1 (SXRMB) | 9 | 72 |
| 06ID-1 (HXMA) | 25 | 211 |
| 07B2-1 (VESPERs) | 5 | 60 |
| 10ID-1 (SM) | 36 | 343 |
| 11ID-1 (SGM) | 32 | 228 |
| 11ID-2 (PGM) | 14 | 120 |
| Total | 141 | 1369 |

These proposals are manually scheduled to 8 beamlines. Each row in Table 1 shows the number of the proposals scheduled on one beamline and the number of shifts used. We show we can do the same with the automatic schedule function in our module.

Figure 4 shows the input data file for the ILOG. We convert the shifts into nature numbers. For example, the first shift is from 2010/1/1, 0:00am to 2010/1/1, 8:00am,

and it is converted to 1. In Figure 4, $M=8$ is the number of the beamlines. $N=141$ is the number of experiments to be scheduled. R is the array of release time for all the experiments. As all the experiments have the release time from the first shift in the cycle, all the items in R are 1. W is the array of the weights of the proposals. Each of numbers in W is a proposal review score multiplied by 100. P is the array of processing time. E is an 8×141 -array of the eligibilities for the 141 proposals on 8 beamlines. We can see the first 5 proposals are eligible for beamline 1. Variable Ub and Ua are 8×30 -array of unusable start time and end time. 30 is the value for $Max2$ obtained from the real data. Variable Ts and Te represent the cycle start and end time. The available shifts are between Ts and Te except pairs of Ub and Ua . As the first numbers of Ub and Ua are 1 and 14, shifts 1 to 13 are unusable.

Figure 5 shows the schedule results in a text file outputted by the ILOG. Each item is in the format of *[the index of beamline, the index of proposal, the allocated start time, the allocated shifts]*. For example, the first item [7, 122, 14, 1] means on beamline 7 (11ID-1), the 122-th experiment is scheduled from the 14-th shift for 1 shift.

```
M=8;
N=141;
R=[1,1,1,1,1,...];
W=[171,200,229,291,300,...];
P=[24,8,9,21,15,...];
E=[[ 1,1,1,1,1,...],[ 0,0,0,0,1,...], ...];
Ub=[[ 1,16,140,242,340,343,431,542,...],...];
Ua=[[14,17,144,338,341,344,435,544,...],...];
Ts=1;
Te= 544;
Us=[[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],...];
Ue=[[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],...];
```

Figure 4: Example of the input data for the ILOG.

| | | | |
|------------|------------|--------------|--------------|
| 7.122.14.1 | 7.110.42.6 | 7.106.98.9 | 7.123.189.6 |
| 7.118.17.3 | 7.111.48.6 | 7.102.107.9 | 7.113.203.9 |
| 7.100.20.3 | 7.112.54.6 | 7.96.116.10 | 7.108.212.9 |
| 7.103.23.3 | 7.116.60.6 | 7.119.126.12 | 7.107.221.15 |
| 7.109.26.3 | 7.105.66.8 | 7.125.138.2 | 7.121.236.6 |
| 7.124.29.3 | 7.126.74.8 | 7.127.144.15 | 7.101.521.6 |
| 7.97.32.4 | 7.114.82.8 | 7.98.159.18 | 7.104.527.6 |
| 7.170.36.6 | 7.99.90.8 | 7.115.177.6 | 7.117.533.9 |

Figure 5: Part of the scheduling results from the ILOG for beamline SGM.

REFERENCES

[1] Canadian Light Source, Remote Instrument Control, http://www.lightsource.ca/operations/remote_access_project/

[2] Wikipedia Article, Synchrotron, Wikipedia, the free encyclopedia. en.wikipedia.org/wiki/Synchrotron. March, 2009.

[3] Zahid Anwar, Zhiguo Wang, Chun Wang, Dan Ni, Yaofeng Xu, Yuhong Yan, "A Integer Programming Model and Heuristic Algorithm for Automatic Scheduling in Synchrotron Facilities", 2009 IEEE Inter. Conf. on Systems, Man, and Cybernetics (SMC'09), Oct. 11-14, 2009, San Antonio, Texas, USA.