Fermilab

FERMILAB-Conf-02/168-A October 2002

Sloan Digital Sky Survey observing time tracking and efficiency measurement

Eric H. Neilsen, Jr.[a], Richard G. Kron[a,b], William N. Boroski[a]

[a]Fermi National Accelerator Laboratory, Batavia, IL
[b]University of Chicago, Chicago, IL

ABSTRACT

Accurate and consistent time tracking is essential for evaluating the efficiency of survey observing operations and identifying areas that need improvement. Off the shelf time tracking software, which requires users to enter activities by hand, proved tedious to use and insufficiently flexible. In this paper, we present an alternate time tracking system developed specifically for Sloan Digital Sky Survey observing. This system uses an existing logging system, `murmur`, to log the beginning and ending times of tracked circumstances, including activities, weather, and problems which effect observing. Operations software automatically generates most entries for routine observing activities; in a night of routine observing, time tracking requires little or no attention from the observing staff. A graphical user interface allows observers to make entries marking time lost to weather and equipment, and to correct inaccurate entries made by the observing software. The last is necessary when the change in activity is not marked by a change in the state of the software or instruments, or when the time is used for engineering or other observing not part of routine survey data collection.

A second utility generates reports of time usage from these logs. These reports include totals for the time spent for each observing task, time lost to weather and problems, efficiency statistics for comparison with the survey baseline, and a detailed listing of what activities and problems were present in any covered time period.

Keywords: Astronomical observatories, telescope operation, efficiency, time tracking

1. INTRODUCTION

The goal of the Sloan Digital Sky Survey[1] (SDSS) is to image $10,000\square''$ of the sky through 5 filters under photometric conditions, and collect spectra of $1,000,000$ objects identified in the imaging data. Observing consists of two stages. First, a 54 CCD imaging camera scans an area of the sky in time delay and integrate (TDI) mode. After the reduction of imaging data for a region of the sky, a multifiber spectrograph takes spectra of objects identified in this data.

The rate at which we collect survey data depends on many factors, including the weather at the observatory, time lost to maintenance and engineering activities, time spent taking calibration data, the exposure time needed to obtain the desired signal to noise in spectra, CCD readout time, time spent changing instruments, and other observing overhead. Some of these factors, such as the weather and the CCD readout time, are either beyond control or are determined entirely by design decisions which cannot now reasonably be changed. Other factors may be improved given sufficient effort.

Accurate measurement of how much time is lost to each factor is essential for determining which systems require the most attention, and what problems are causing us to loose the most data. Furthermore, they help determine the causes of differences between the true and expected data collection rate for the instruments.

Further author information: neilsen@fnal.gov; Telephone: 1 630 840 6720; Address: Mail Stop 120, Fermi National Accelerator Laboratory, P. O. Box 500, Batavia, IL 60510-5011

Table 1. Baseline plan for the northern galactic cap

	dark	im.	sp.			im.	sp.	im.	sp.	im.	sp.	sq.	No.
	hrs.	frac	frac.	weather	uptime	eff.	eff.	ops.	ops.	hrs.	hrs.	deq.	plts.
2000 Q2	277	0.3	0	0.6	0.7	0.86	0.65	0.25		7.5	0	140.7	0
Q3	40	0.3	0	0.5	0.8	0.86	0.65	0.5		2.1	0	38.7	0
Q4	196	0.3	0	0.6	0.7	0.86	0.65	0.75		20.5	0	140.7	0
2001 Q1	464	0.3	0.7	0.6	0.9	0.86	0.65	0.9	0.75	58.2	85.5	1090.9	114
Q2	277	0.3	0.7	0.6	0.9	0.86	0.65	0.9	0.9	34.7	61.3	651.2	82
Q3	40	0.3	0.7	0.5	0.9	0.86	0.65	0.9	0.9	4.2	7.4	78.4	10
Q4	196	0.3	0.7	0.6	0.9	0.86	0.65	0.9	0.9	24.6	43.3	460.8	58
2002 Q1	464	0.3	0.7	0.6	0.9	0.86	0.65	0.9	0.9	58.2	102.6	1090.9	137
Q2	277	0.3	0.7	0.6	0.9	0.86	0.65	0.9	0.9	34.7	61.3	651.2	82
Q3	40	0.3	0.7	0.5	0.9	0.86	0.65	0.9	0.9	4.2	7.4	78.4	10
Q4	196	0.2	0.8	0.6	0.9	0.86	0.65	0.9	0.9	16.4	49.5	307.2	66
2003 Q1	464	0.2	0.8	0.6	0.9	0.86	0.65	0.9	0.9	38.8	117.3	727.3	156
Q2	277	0.2	0.8	0.6	0.9	0.86	0.65	0.9	0.9	23.2	70.0	434.2	93
Q3	40	0.2	0.8	0.5	0.9	0.86	0.65	0.9	0.9	2.8	8.4	52.2	11
Q4	196	0.2	0.8	0.6	0.9	0.86	0.65	0.9	0.9	16.4	49.5	307.2	66
2004 Q1	464	0.2	0.8	0.6	0.9	0.86	0.65	0.9	0.9	38.8	117.3	727.3	156
Q2	277	0.2	0.8	0.6	0.9	0.86	0.65	0.9	0.9	23.2	70.0	434.2	93
Q3	40	0.2	0.8	0.5	0.9	0.86	0.65	0.9	0.9	2.8	8.4	52.2	11
Q4	196	0	1	0.6	0.9	0.86	0.65	0.9	0.9	0.0	61.9	0.0	83
2005 Q1	464	0	1	0.6	0.9	0.86	0.65	0.9	0.9	0.0	146.6	0.0	195
Q2	277	0	1	0.6	0.9	0.86	0.65	0.9	0.9	0.0	87.5	0.0	117
TOTAL	5162									411.0	1155	7706	1540

2. SURVEY REQUIREMENTS & THE BASELINE PLAN

A schedule for data collection, called the "baseline plan,"[2] was derived using early estimates for each factor. Some could be estimated reasonably well. Because the survey telescope was situated at an existing observatory, the expected fraction of time the telescope would be closed due to weather could be estimated using past data. Other factors were more difficult to estimate; before the completion of the instruments, the estimate of the time needed for calibration and instrument change could be educated speculation at best.

Table 1 shows the baseline plan for one part of the survey, the northern galactic cap. Similar tables describe the baseline for the southern survey and southern outriggers. Column 2, "dark hours," gives an estimate of dark time available for observing. Columns 3 through 10 list the factors and efficiencies that lead to the time spent collecting imaging and spectroscopic data. For example, in Q1 of 2002,

$$\begin{aligned}
\text{(im. hrs)} &= \text{(dark hrs.)} * \text{(im. frac.)} * \text{(weather)} * \text{(uptime)} * \text{(im. eff.)} * \text{(im. ops.)} \\
&= 464 \text{ hours} * 0.3 * 0.6 * 0.9 * 0.86 * 0.9 \\
&= 58.2 \text{ hours}
\end{aligned}$$

and

$$\begin{aligned}
\text{(sp. hrs)} &= \text{(dark hrs.)} * \text{(sp. frac.)} * \text{(weather)} * \text{(uptime)} * \text{(sp. eff.)} * \text{(sp. ops.)} \\
&= 464 \text{ hours} * 0.7 * 0.6 * 0.9 * 0.65 * 0.9 \\
&= 102.6 \text{ hours}
\end{aligned}$$

Note that these calculations assume that the various factors and efficiencies are uncorrelated.

The properties that determine these factors are often complex; weather considerations not included in the "weather" factor contribute to the imaging faction (and the corresponding spectroscopic fraction), for example.

The weather factor listed in the table is the fraction of time the weather can support useful observations of any sort. Only some fraction of this time has seeing and photometric conditions needed for imaging, and the imaging fraction obtained can be no higher than this fraction.

The imaging and spectroscopic efficiencies are yet more complex. An instrument's efficiency is the total science exposure time on that instrument expressed as a fraction of the time spent using that instrument. These efficiencies are therefore a measure of the overhead in observing with each instrument.

Without experience with the finished instrument, few of these factors can be well estimated. Even where historical data can be found (the weather data, for example), it is often a poor indicator of future performance.

3. MANUAL TIME TRACKING

Initial attempts at measuring observing efficiency used `titrax`,[3] a general purpose, "off the shelf" time tracking package. `titrax` provides a menu of activities. It was the observers responsibility to make sure that the current activity was checked at any given time. At the end of the night, one could then generate a report of the total time each activity was checked during that night.

This strategy had several major flaws, the most important of which is the level of attention it required of the observers, particularly at those times where their attention was most needed elsewhere. When problems occurred, the time tracker was often overlooked while the problems were being addressed. It was therefore difficult to reconstruct exactly how much time was lost to the problem. Even under routine, problem free observing, keeping `titrax` up to date was a nuisance; it would need to be updated between each stage in setting up for imaging a new area of sky, or changing spectroscopic fields. Observers would often forget to change the state, and had to estimate corrections after the fact. Therefore, in addition to being a nuisance, manual time tracking was also unreliable.

4. LOGGING AND AUTOMATED TIME TRACKING

Many components of the SDSS observing software use `murmur`[4] for logging. `murmur` is a software package for generating, reporting, and storing log messages over distributed systems. A variety of SDSS software products send `murmur` a significant quantity of data, which `murmur` saves in a log file together with its origin, type, and time. As used by SDSS observing software, `murmur` logs are stored as a text file suitably formated for processing by scripting languages and standard UNIX text manipulation utilities.

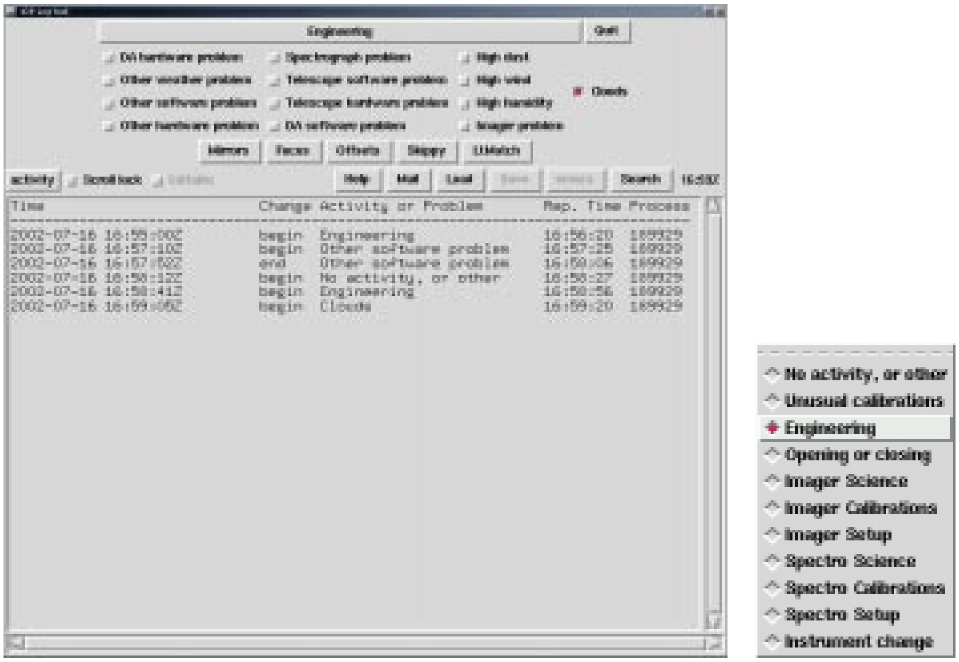
The `murmur` log is therefore a useful archive of information for determining when the observing software performed various actions. At first, however, none of the information it recorded was designed specifically for determining when various observing activities began and ended. It recorded only information intended for debugging, such as communication between operations programs and the instruments. Much of this could be used to determine what was happening when; if, for example, we can see when the spectrograph mechanicals where commanded to open the shutters, we can determine when an exposure started. If we look for commands to the flat field and arc lamps, we may be able to determine whether that was a calibration exposure or not. As useful as it might be, this sort of information was inconvenient to use, and often inadequate.

We needed three things to make the `murmur` log a convenient archive for time tracking data:

1. observing software that sends messages to the `murmur` log when commands corresponding to specific observing tasks are issued and completed;

2. tools the observers could use to supplement the log with time tracking data that cannot be recorded automatically, to verify that the messages being sent automatically are correct, and to accurately correct entries that were made incorrectly (whether automatically or manually); and

3. a tool to extract useful tables of results from archived `murmur` logs.



☐ DA hardware problem ☐ Spectrograph problem ☐ High dust
☐ Other weather problem ☐ Telescope software problem ☐ High wind ☒ Clouds
☐ Other software problem ☐ Telescope hardware problem ☐ High humidity
☐ Other hardware problem ☐ DA software problem ☐ Imager problem

☐ Scroll lock

```
Time                           Change Activity or Problem     Rep. Time Process
------------------------------------------------------------------------------
2002-07-16 16:55:00Z           begin  Engineering             16:56:20  189929
2002-07-16 16:57:10Z           begin  Other software problem  16:57:25  189929
2002-07-16 16:57:52Z           end    Other software problem  16:58:06  189929
2002-07-16 16:58:12Z           begin  No activity, or other   16:58:27  189929
2002-07-16 16:58:41Z           begin  Engineering             16:58:56  189929
2002-07-16 16:59:05Z           begin  Clouds                  16:59:20  189929
```

Figure 1. The logTool user interface, and menu of activity choices

In many cases, supplementing the murmur log with automatic time tracking information was a simple task. For example, the observers use the doNextScience command to begin a science exposure on the spectrographs. When doNextScience begins, it sends a message to murmur noting the beginning of spectrographic science observations. This works only because science exposures are almost always started with doNextScience, and doNextScience is rarely used for anything other than science exposures. Similarly, the instChange, gotoField, preCalib, smear, and postCalib commands correspond directly to different stages in spectrographic observing. Therefore, when the beginning and ending of any observing activity corresponds to the start and completion of one of these commands, the observers need not record the change in activity; the observing software does it for them.

Although these commands are almost always used in connection with their corresponding observing tasks, there are exceptions. For example, doNextScience may be executed as part of an engineering test, perhaps to test a code change in doNextScience itself. Furthermore, not all changes in observing activity correspond directly to observing software commands. Any activity may end prematurely due to an equipment problem. Even in problem free observing, there are times when activity changes cannot be automatic. For example, imaging setup ends and imaging science begins when the observers judge no further instrument adjustments are necessary.

Observers monitor the activity being recorded and supplement the murmur log with their own additions and corrections using logTool. Figure 1 shows the user interface to logTool. The bar at the top of the window shows the activity currently being reported. Clicking on the bar brings up a menu of activities, shown on the right in figure 1, from which the observers can choose. The checkboxes beneath the current activity bar should reflect the problems currently being experienced. This arrangement of problems and activities reflects the restriction that only one activity may be attempted at any given time, but many problems may occur simultaneously. When the users change an activity or toggle a problem state, they are prompted for the time the change occurred. logTool records both the time of the change in state and the time at which the change in state was reported in the murmur log.

logTool may also be used to help write the manual night log. The two rows of buttons immediately beneath

```
creationTime 2002-06-25 21:45:46Z

typedef struct {
int mjd;
char time[21];
char duration[10];
char activity[17];
char problems[128];
} TIMEPERIOD;

TIMEPERIOD 52441 {2002-06-16 06:35:56Z} 00:31:16 instrumentChange  {otherHardware}
TIMEPERIOD 52441 {2002-06-16 07:07:12Z} 00:03:12 instrumentChange  {}
TIMEPERIOD 52441 {2002-06-16 07:10:24Z} 00:09:37 spectroSetup      {}
TIMEPERIOD 52441 {2002-06-16 07:20:01Z} 00:04:56 spectroCalib      {}
TIMEPERIOD 52441 {2002-06-16 07:24:57Z} 00:06:10 spectroSetup      {}
TIMEPERIOD 52441 {2002-06-16 07:31:07Z} 00:21:38 spectroScience    {}
TIMEPERIOD 52441 {2002-06-16 07:52:45Z} 00:01:00 none              {}
TIMEPERIOD 52441 {2002-06-16 07:53:45Z} 00:21:35 spectroScience    {}
TIMEPERIOD 52441 {2002-06-16 08:15:20Z} 00:00:58 none              {}
TIMEPERIOD 52441 {2002-06-16 08:16:18Z} 00:18:29 spectroScience    {}
TIMEPERIOD 52441 {2002-06-16 08:34:47Z} 00:00:43 none              {}
TIMEPERIOD 52441 {2002-06-16 08:35:30Z} 00:11:13 spectroCalib      {}
TIMEPERIOD 52441 {2002-06-16 08:46:43Z} 00:00:35 none              {}
TIMEPERIOD 52441 {2002-06-16 08:47:18Z} 00:04:44 instrumentChange  {}
```

Figure 2. The header and a representative sample of entries in the time period description file generated from `murmur` logs.

the problem checkboxes are associated with this feature. A description of this use of `logTool` is beyond the scope of this paper.

The table at the bottom of the `logTool` shows the times of those changes of state, what those changes were, and the time those changes were recorded. It is possible for multiple, contradictory changes of state to be recorded for the same time. In these cases, changes recorded latest take precedence. This table includes both changes initiated by the user and those that occur automatically.

5. RESULTS

Once time tracking information is recorded in the `murmur` log as accurately as can be reasonably expected, the data must be extracted and presented in a cenvenient form. A set of `tcl` scripts gather `murmur` log entries from the archive of `murmur` logs, build a file describing the activities and problems of each time period during observing, and derive a set of tables summarizing the desired statistics.

Although `logTool` simplifies record keeping by the observers, time tracking data recorded in `murmur` logs is still occasionally incomplete or inaccurate. The most common problems involve reporting the begin and end times of problems, which still cannot be recorded automatically, yet occur at times when the observers attention is occupied elsewhere. The is one important failure of the manual time tracking efforts that could not be eliminated by the automation effort. However, automated time tracking entries in the `murmur` log do help considerably. By providing a detailed chronology of those portions of the night for which there were no problems, they make it possible to estimate the beginning and ending times for problems described in the text of manually written night logs. The collection of `murmur` log data into a self contained, readable file simplifies the manual editing of the records this process requires..

```
Baseline comparison
                    dark  dark + grey baseline
 total time         92:34   121:17    (92:34)
 imaging fraction   0.15     0.12       0.27
 spectro fraction   0.76     0.81       0.63
 weather            0.52     0.57       0.6
 uptime             0.98     0.97       0.9
 imaging efficiency 0.84     0.84       0.86
 spectro efficiency 0.59     0.58       0.65
 operations         0.91     0.92       0.9
```

Figure 3. The efficiency measured from total times.

Fortunately, the `murmur` log entries generated automatically by the observing software help place into context the detailed prosaic problem descriptions found in the manually written night logs, and reasonable estimates can be made for the times of the missed entries. The collection of `murmur` log data into a self contained, readable file simplifies the manual editing of the records this process requires.

Once an accurate time period description file is created, summary tables can be generated. Figures 3 and 4 show sample summary tables for both efficiency and total time.

6. INTERPRETATION

Figure 3 shows efficiencies measured useing actual time tracking data. Although they are measured according to the deffinitions used in the baseline, one cannot make a direct comparison of the results of the time tracking and the baseline described in section 2. The simplifying assumptions made in defining the elements of the baseline sometimes result in parameters that cannot always be mapped perfectly to actual observing events. For example, the baseline collects time lost to weather into a single factor, estimating that 40% to 50% of potential observing time will be lost. Estimates for other factors assume that time lost to weather is accounted for in the "weather" fraction, and are made based on good weather estimates. In practice, however, measuring the time weather conditions completely prevent observing does not account for time lost to weather. For example, passing clouds may prematurely stop an imaging scan, resulting in a shorter scan than planned. Because the observing overhead per scan is independent of the length of the scan, the measured imaging efficiency (the ratio of time spent taking science data to the total time spent imaging) will drop because of this. A lower measured imaging efficiency, therefore, cannot be blindly attributed to observing performance falling below expectations. The underlieing cause must be explored with more detailed statistics.

Figure 4 lists the time dedicated to different observing activities during the June 2002 dark run. These data relate directly both to the quantities measured by the time tracking and the quantity of data gathered. The "imaging" section of this plot clarifies the origin of the low imaging efficiency: the mean science time per run was only 2:00, while the baseline is 2:30. The setup time per science run roughly matched expectations. At this point, one can check the hand written night logs to discover that the reason for the short runs was interruption by clouds.

There are additional problems in comparing the time tracking output to the baseline. The "imaging operations" and "spectrographic operations" baseline values are supposed to reflect miscellaneous time losses, such as time lost at the end of the night when there is too little time left to start anything new, but after a set of observations is complete. While the baseline divides this time between imaging and spectroscopy, in practice only a single quantity is measured: time for which the time tracking does not record any activity.

Although baseline plan factors such as these cannot be derived directly from the time tracking data, several of the factors (such as uptime) can, and others may be approximated well. The overall operations efficiency can be measured, and compared with the baseline values of both the imaging and spectrographic efficiencies stated

```
Total Time Summary
                                    time  baseline
             Spectroscopy, Dark + Grey
 Inst. and cart. change time       4:17   4:20
 Spectro setup time                10:36  4:20
 Spectro calibration time          7:38   2:10
 Spectro science time              31:07  19:30
 Total spectro time                53:38  31:30
 Total plates                      <37    26
                Spectroscopy, Dark
 Dark inst. and cart. change time  2:47   4:20
 Spectro dark setup time           5:52   4:20
 Spectro dark calibration time     5:41   2:10
 Spectro dark science time         20:48  19:30
 Total dark spectro time           35:08  31:30
                Spectroscopy, Grey
 Grey inst. and cart. change time  2:47   0:00
 Spectro grey setup time           4:44   0:00
 Spectro grey calibration time     1:57   0:00
 Spectro grey science time         10:20  0:00
 Total grey spectro time           18:30  0:00
                      Imaging
 Imaging dark setup time           1:02   1:37
 Imaging dark calibration time     0:07   0:00
 Imaging dark science time         6:00   9:59
 Total dark imaging time           7:08   13:30
 Imaging setup time                1:02   1:37
 Imaging calibration time          0:07   0:00
 Imaging science time              6:48   9:59
 Total imaging time                7:56   13:30
 Number of Science Runs            3      4
 Mean science time per science run 2:00   2:30
 Mean setup time per science run   0:21   0:24
                  Dark time usage
 Total dark (moonless) time        92:34  ----
 Dark time lost to problems        1:54   9:15
 Dark time lost to weather         44:21  33:20
 Other calib dark time             0:00   0:00
 Engineering dark time             0:00   0:00
 Operations dark time              4:04   5:00
                 Total time usage
 Total night time                  121:17
 Night time lost to problems       4:11
 Night time lost to weather        51:08
 Other calib time                  0:00
 Engineering time                  0:00
 Operations time                   5:13
 Total science time                61:34
```

Figure 4. The summary of time usage.

in the baseline. The overhead per run may be calculated from the time tracking data, allowing us to determine whether we are loosing more time than expected to setup.

While a comparison between the baseline and observing in practice can be interesting, it is not the primary goal of time tracking, because differences may reflect inaccurate estimates in the baseline more than anything about observing. In some cases, the baseline is overly optimistic; the calibration time per spectrograph plate will never match the baseline estimate. In other cases, it is pessimistic; we routinely match or exceed the baseline for uptime and operations efficiency.

A more important use of the time tracking software is to identify observing time lost to tasks in principle within our control, and to track the effects of changes in procedure, software, and equipment on our actual data collection rate. For example, shortly after we began studying time tracking results, we discovered that setting up for imaging was taking far more time than initially estimated; exploration of why this was happening led to significant improvements to the imaging setup procedure. We could then track the improvement in the setup time as the imaging procedures changed.

7. DISCUSSION

The current time tracking software has been useful for observing optimization, and continues to be an important too for tracking and prioritizing changes. Experience in the implementation and use of this software has provided several valuable lessons.

- General purpose time tracking software requires significant attention from the observers to be useful, and may not be reliable enough even when it is attended religiously.
- Logging software of the sort commonly used for monitoring and debugging software can also form a useful repository for time tracking information.
- A close relationship between software commands and observing activities greatly simplifies automated observing, because the same command that is used to start an activity can also log the time tracking information. The ease with which time tracking was integrated into the observing software was a consequence of its initial design, although not an entirely coincidental one. Different commands were written to initiate different observing activities, because different activities required bookkeeping records that would not be supplied by lower level, more general commands. The time tracking records became part of the bookkeeping performed by these commands.
- Provided a significant fraction of time tracking information is recorded automatically, some reliance on manual adjustment by the observers is not only tolerable but necessary. The combination of detailed, manually written night logs and time stamped automatic time tracking information can provide a reasonably good reconstruction of the timing of events not recorded automatically. Prose descriptions of the events of a night are essential for interpreting the time tracking data.
- Measuring performance against a baseline not designed with time tracking in mind can be challenging. Designing the time tracking methodology in concert the baseline would have been helpful, but without actual experience taking data with the instruments, creating a baseline plan where every factor can be easily measured and intuitively understood will always be a challenge.

The usefulness and practicality of time tracking can benefit greatly from advanced planning, both in the design of the observing software and in the definition of the parameters used to measure performance. The SDSS has been fortunate that both its observing software and baseline plan could accomodate useful and relatively covenient time tracking with only minor modifications. Greater coordination between time tracking, software design, and baseline specification at an early stage would still have been helpful.

ACKNOWLEDGMENTS

Funding for the creation and distribution of the SDSS Archive has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Aeronautics and Space Administration, the National Science Foundation, the U.S. Department of Energy, the Japanese Monbukagakusho, and the Max Planck Society. The SDSS Web site is http://www.sdss.org/.

The SDSS is managed by the Astrophysical Research Consortium (ARC) for the Participating Institutions. The Participating Institutions are The University of Chicago, Fermilab, the Institute for Advanced Study, the Japan Participation Group, The Johns Hopkins University, Los Alamos National Laboratory, the Max-Planck-Institute for Astronomy (MPIA), the Max-Planck-Institute for Astrophysics (MPA), New Mexico State University, Princeton University, the United States Naval Observatory, and the University of Washington.

REFERENCES

1. D. G. York, J. Adelman, J. E. Anderson, S. F. Anderson, J. Annis, N. A. Bahcall, J. A. Bakken, R. Barkhouser, S. Bastian, E. Berman, W. N. Boroski, S. Bracker, C. Briegel, J. W. Briggs, J. Brinkmann, R. Brunner, S. Burles, L. Carey, M. A. Carr, F. J. Castander, B. Chen, P. L. Colestock, A. J. Connolly, J. H. Crocker, I. . Csabai, P. C. Czarapata, J. E. Davis, M. Doi, T. Dombeck, D. Eisenstein, N. Ellman, B. R. Elms, M. L. Evans, X. Fan, G. R. Federwitz, L. Fiscelli, S. Friedman, J. A. Frieman, M. Fukugita, B. Gillespie, J. E. Gunn, V. K. Gurbani, E. de Haas, M. Haldeman, F. H. Harris, J. Hayes, T. M. Heckman, G. S. Hennessy, R. B. Hindsley, S. Holm, D. J. Holmgren, C. Huang, C. Hull, D. Husby, S. Ichikawa, T. Ichikawa, Ž. Ivezić, S. Kent, R. S. J. Kim, E. Kinney, M. Klaene, A. N. Kleinman, S. Kleinman, G. R. Knapp, J. Korienek, R. G. Kron, P. Z. Kunszt, D. Q. Lamb, B. Lee, R. F. Leger, S. Limmongkol, C. Lindenmeyer, D. C. Long, C. Loomis, J. Loveday, R. Lucinio, R. H. Lupton, B. MacKinnon, E. J. Mannery, P. M. Mantsch, B. Margon, P. McGehee, T. A. McKay, A. Meiksin, A. Merelli, D. G. Monet, J. A. Munn, V. K. Narayanan, T. Nash, E. Neilsen, R. Neswold, H. J. Newberg, R. C. Nichol, T. Nicinski, M. Nonino, N. Okada, S. Okamura, J. P. Ostriker, R. Owen, A. G. Pauls, J. Peoples, R. L. Peterson, D. Petravick, J. R. Pier, A. Pope, R. Pordes, A. Prosapio, R. Rechenmacher, T. R. Quinn, G. T. Richards, M. W. Richmond, C. H. Rivetta, C. M. Rockosi, K. Ruthmansdorfer, D. Sandford, D. J. Schlegel, D. P. Schneider, M. Sekiguchi, G. Sergey, K. Shimasaku, W. A. Siegmund, S. Smee, J. A. Smith, S. Snedden, R. Stone, C. Stoughton, M. A. Strauss, C. Stubbs, M. SubbaRao, A. S. Szalay, I. Szapudi, G. P. Szokoly, A. R. Thakar, C. Tremonti, D. L. Tucker, A. Uomoto, D. Vanden Berk, M. S. Vogeley, P. Waddell, S. Wang, M. Watanabe, D. H. Weinberg, B. Yanny, and N. Yasuda, "The Sloan Digital Sky Survey: Technical Summary," *A. J.* **120**, pp. 1579–1587, Sept. 2000.
2. http://www.sdss.org/documents/5yrbaselineschedule.pdf.
3. http://www.alvestrand.no/h̃ta/titrax/.
4. G. Oleynik, L. Appleton, B. MacKinnon, C. Moore, G. Sergey, and L. Udumula, "Murmur: A message generator and reporter for unix, vms, and vxworks," Tech. Rep. FERMILAB-CONF-92-268, Fermi National Accelerator Laboratory, Oct. 1992.