

## Operation event logging system of the Swiss Light Source

Andreas Lüdeke\*

*SLS, Paul Scherrer Institute, Villigen, Switzerland*  
(Received 6 October 2008; published 17 February 2009)

Modern 3rd generation synchrotron light sources aim for 100% availability. No single beam interruption is acceptable and every beam disturbance should be investigated: what caused the interruption? Can it be avoided in the future? If it cannot be avoided, how can the recovery be accelerated? An automated event recording system has been implemented at the Swiss Light Source (SLS) in order to simplify beam distortion investigations with respect to a well-defined metrics. The system identifies beam disturbances and records automatically the type and duration of the event. Relevant information of the event, like control system archive data or shift summaries, is linked to the event and presented in Web pages. Tools for the automated evaluation of alarm logs are provided that generate summaries of a beam distortion. On the basis of this information each event will be assigned to a failure cause. The means to filter the events are provided. We will describe the concept and the implementation of the system at the SLS and our experiences with it. Finally, the SLS operation event logging system will be compared with failure analysis at other light sources.

DOI: [10.1103/PhysRevSTAB.12.024701](https://doi.org/10.1103/PhysRevSTAB.12.024701)

PACS numbers: 29.27.Fh, 29.20.dk, 29.85.-c

### I. MOTIVATION

A large fraction of the work of an operations manager is dedicated to the analysis of operations data. Each distortion of the beam quality needs to be detected, the reason for the distortion should be identified, and a repetition of the distortion should be prevented. The procedure is always in three steps: (a) identifying an undesired operation state (called in the following an “event”), (b) assigning this event to a cause, and (c) defining the actions to take. While the first part can be easily automated, the latter two depend on human interpretation of the facts. But the decision making process can be supported by applications: each type of event requires a certain set of information to decide on the cause and reliable statistical information on the different “events” are a prerequisite for an efficient planning of resources.

An example will illustrate the requirements. For example, a light source wants to know all about loss of the stored beam and about problems with a short beam lifetime. First, we define the two event types: a “downtime” event starts, when the beam current is below 50 mA and stops when 400 mA is stored. A “lifetime” event starts when the beam lifetime is shorter than 4 hours and stops when it is greater than 4 hours. For each event we want to collect some information automatically: a link to the shift summary in the electronic logbook, a link to an alarm log, and a link to archived data. An alarm log shows entries for each single warning or alarm of all devices required for beam operation. For example, a temperature that is too high, a bad vacuum pressure, or a failure of a magnet power

supply are all in the alarm log. Archived data are collected as a history of control system data, like time series of temperatures, vacuum pressures, or power supply current readouts. We want to see the archived data of the beam current for downtime events and of the beam lifetime for lifetime events.

The event logger records basic information: type of the event together with the start and end date. The logger will store the event data as soon as an event is finished. Additionally, the links to the relevant informations for this event are created and stored, too.

A beam loss is handled by the operator in the control room. He normally does not need to fully understand what caused the beam loss, he just refills the storage ring. Someone will later analyze the event data to determine what had really happened. Based on the analysis, he will assign a cause to the event. For example, a beam loss where an alarm shows that a magnet power supply in the storage ring failed will be assigned to the system “magnet power supplies.” The magnet power supply group will use the collected data to find out which power supplies caused beam losses in the past, how often, and how long. The operations manager will use the data to keep himself up to date with the machine operation, to find out when problems reoccur, for the planning of resources, for upgrade plans, and so on.

### II. BASIC SPECIFICATION

The operation event logging system does an automatic recording of defined events during operation of the light source. The events are detected from control system variables by start rules and stop rules, both depending on the operation mode. A set of references to related information

\*Andreas.Luedeke@psi.ch  
<http://people.web.psi.ch/luedeke>

is defined for each event type and the information references are added by the logger for each event.

In order to reference all related data of an event in a useful way, all those data sources need to support references. A simple and efficient way for supporting references is by Web interfaces that allow one to reference data by unified resource locators (URLs) [1]. The main sources of information for operation at the Swiss Light Source (SLS) allow Web access to the data of a time interval by an URL.

One or more causes can be assigned to each event. Each cause is defined by two categories: the affected area, e.g., “Linac” or “storage ring,” and the category of the failing system, like “RF” or “Magnets.” A comment describes the actual failure. The total event duration is split up into fractions for each cause if more than one cause is assigned to an event.

A Web-browser interface to the event database allows one to query it and to filter for specific event types, durations, and causes.

A statistical analysis of the events is automated, to provide for each cause category for a given time period the number of failures and their total duration. The generation of weekly operation reports is integrated with the event database.

### III. GENERIC IMPLEMENTATION

We will describe the operation event logging system in two parts: in this section we will explain the generic part of the application. The customization of the application for the usage at the SLS will be described in the next section.

#### A. System overview

The actual event logging is done by a server application called “event logger.” It connects to a given set of control system process variables (“channels”) and evaluates from the retrieved values the beginning or end of a defined event using “start-rule” and “stop-rule” formulas. After the event has stopped the logger generates references to relevant data for the time of the event and writes these references together with the event period and type to the database. This data is read by the “event browser” and the data references are used by the operation manager to assign a “cause,” i.e., a fault category, to the event. The data references in use at the SLS are: the operation alarm log from the *alarmhandler* [2], the *channel archiver* [3], and the operation shift logbook [4].

The *alarmhandler* and the *channel archiver* are applications provided within the “Experimental Physics and Industrial Control System” (EPICS) collaboration [5]. The EPICS *alarmhandler* is a tool to signal fault conditions in the control room. Additionally, it is used to write the occurrence of those fault conditions into a relational database. It is configured to signal warnings and alarms of all relevant process variables, like vacuum pressures, temperatures, or beam orbit deviations. The EPICS *channel ar-*

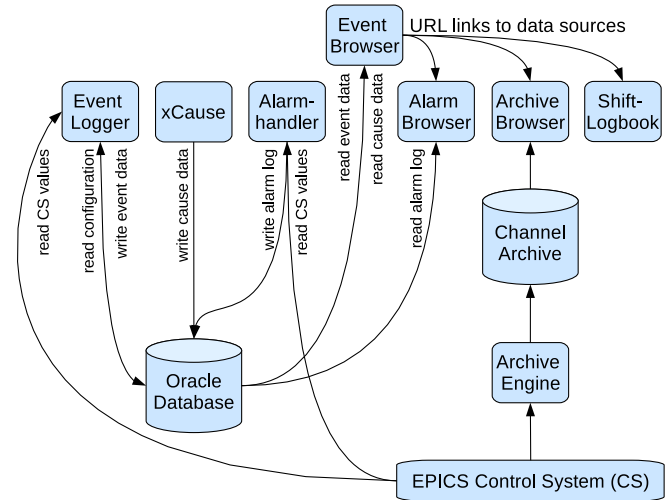


FIG. 1. (Color) System overview of the operation event logging.

*chiver* engine is a server application that monitors the values of a given subset of control system process variables. It is capable of monitoring an arbitrary number of channels and of sampling more than 80 000 values per second [6]. The amount of data and the archive frequency is mainly limited by the available storage capacity. A Web interface allows one to access and display this data.

The logbook application is part of the PSI Digital User Office (DUO). DUO is the central tool to organize the user operation at PSI. This Web application is used to submit proposals, experimental reports, publications, to request for PSI badges or dosimeters, or register for conferences. Many DUO logbooks are in use: in addition to the logbook for SLS shift summaries there are logbooks for the vacuum group, for the controls group, and for many beam lines.

Figure 1 shows a system overview. All database tables and their relations are shown in the database model view in Appendix C, Fig. 10.

#### B. Automatic event logging

An event is defined by the *event\_rules* database table. It has a *start\_rule*, a *start\_delay*, a *stop\_rule*, and a *stop\_delay*. An event of a certain type can have multiple definitions depending on the operation mode of the light source. The possible operation modes are defined in the *operation\_category* table.

If the start rule of an event is evaluated to be true for the current operation mode, a counter is started. If the counter exceeds the start delay, the event start is recorded. If it evaluated to false before the start delay has been exceeded, the counter is reset to zero. After the event start, the event-stop rule is evaluated until it is true for at least stop-delay seconds. The event stops as well in case of a change of the operation mode. Then the event is recorded to the database. Event-start and event-stop dates are defined as the beginning of the delay interval. For example, if a stop delay is

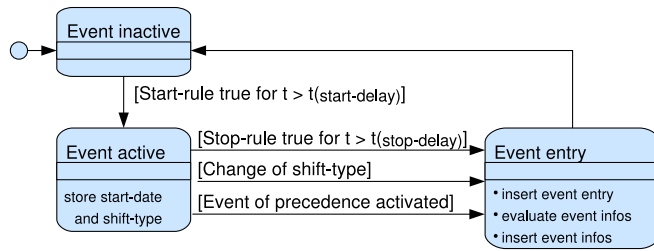


FIG. 2. (Color) Generic state diagram for each automatic event type defined by an event rule set.

one hour, the event entry is written to the database one hour after the event-stop date.

The start of one event can stop events of other types. Those event relations are defined in the *event\_precedence* table. Figure 2 shows a state diagram of the event logger for one event type. Each event has its own state set, but by the event precedence each event activation can deactivate other events. Once an event has been activated, the event data will be written to the database regardless of the way the event is deactivated: by the stop rule, a change of the operation mode or by giving precedence to the activation of another event. An example of the usage of the event counters and the event precedence is given in Fig. 3.

Actual events are recorded in the *event\_entry* table. The *event\_start* date is the primary key of this table, together with the *event\_stop* it defines when the event occurred. The combination of *event\_type* and *op\_type* defines the actual type of the event. The *op\_type* is in reference to an *operation\_category*. Each event entry can have many *event\_info* entries. Those contain comments for an event defined by its *event\_start*. Automatic entries in the *event\_info* table are made according to definitions in the *event\_links* table. The table defines for each *event\_type* a list of

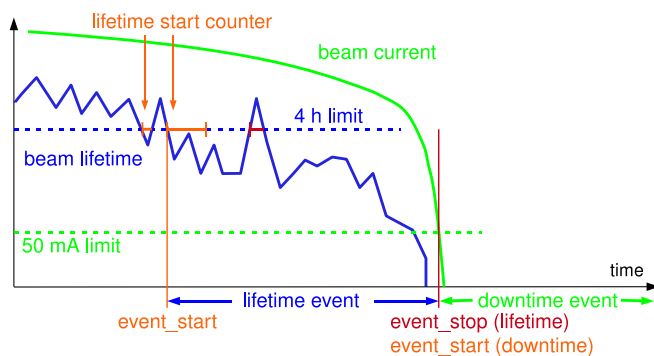


FIG. 3. (Color) The example shows a lifetime event as defined in Sec. I. The event-start counter had been reset the first time, since the beam lifetime came back above 4 hours before it finished. The second time the counter finished and the event started. The stop counter started once but had been reset when the lifetime dropped again below 4 hours. The event stopped when the beam current dropped below 50 mA: that started a downtime event which has precedence.

functions to be executed. For each function the timestamps for the start and the end of the event are passed as arguments. The output text is then written to the *event\_info* table. It is possible to manually add more information later on to a specific event. The table *event\_info\_hist* keeps a history of delete and update transactions for the *event\_info* table.

Another command line tool allows one to add or delete events manually. This is useful in case of control system failures, which can cause wrong events or prohibit the automatic generation at the time of the event. We also defined a nonautomatic event “miscellaneous” that can be used to manually document exceptional events.

### C. Event cause assignment

The expert manually assigns one or more causes to the event after its automatic generation.

Which person will assign a cause to an event may depend on the type of the event. In most cases a good knowledge of accelerator physics is useful, in order to avoid wrong cause assignments due to misinterpretation of the event data. The system experts of the beam feedbacks are best qualified to analyze those events and assign a cause accordingly.

The command line tool “*cause*” or the graphical user interface “*xcause*” allow privileged users to assign a cause to an event entry. A cause is characterized by the *event\_start*, an *area\_ID*, a *sys\_ID*, a *fraction*, and an *om\_description* (a description of the problem). All event causes for one event should sum up to 100%. The possible cause areas are defined in the *event\_area* table by an *area\_ID* and a description. The system categories are defined in the table *event\_category* by an *op\_type* and a description.

Both applications, *cause* and *xcause*, allow one to display those events that do not yet have a cause assigned or where the fractions of all assigned causes do not add up to 100%.

Delete or update transactions to the table *event\_cause* are logged in the table *event\_cause\_hist*, together with the modifying user and the modification date.

### D. Event alarm evaluator

Alarms of the *alarmhandler* are grouped and the groups belong to a hierarchy. This hierarchy is visible to the operator when he acknowledges the alarm in the interactive user interface, but it is not visible in the flat alarm log. Therefore the analysis of the list of alarms can be difficult, in particular, if the operator is not very experienced. To simplify the interpretation of an alarm list we have defined classes of important alarms and provided a command line tool to evaluate the alarms of a specific event. The alarm hint groups are defined in the table *alh\_hint\_groups*. This table is in relation to the alarm log. It is not part of the event database and is therefore not included in the database

model overview in Fig. 10. Only the time intervals for the alarms to evaluate are selected from the event entries.

Each alarm hint group has a number *hint\_count* defined to limit the output lines of one hint group for one event. For example, if a defective magnet power supply may cause dozens of alarms, the count could limit the output to the evaluation of the first few. The *alh\_hint* table is the basis of the evaluation. Each alarm in this table has its *group\_name*, which is the name of the original *alarmhandler* group it belonged to, a *hint\_group*, an *html\_comment*, and a failure system and area code. The *html\_comment* may contain tokens as a reference to (i) *%c* channel name: for the name of the channel that caused the alarm; (ii) *%d* device name: like *%c* but only until the first colon “:” in the string; (iii) *%v* value: the value of the channel that caused the alarm; (iv) *%s* severity: the alarm severity, e.g. “minor” for a warning, “major” for an alarm.

The *alh\_hint* table itself is filled by an application that parses the *alarmhandler* configuration files. This script is SLS specific; it is described below in Sec. IV E.

### E. Event browser

Web pages, applications, and command line tools have been provided to browse and modify the event database. Figure 4 shows a screen shot of the event browser Web page. The browser allows one to select events within a given time range. It provides filters for the operation mode, the event type, cause area, cause system, and event duration. All information to an event is shown in a single row, including the list of causes and their descriptions. The last column “Info” contains the automatic and manual information for the event. The three entries visible for each event in Fig. 4 are the automatically added links to the Web retrieval tools used for the SLS (see Sec. IV D).

### F. Automated event statistics

An application facilitates the compilation of overview statistics from the operation event logging database for a given time period. For each operation mode and event type, it calculates the number of events and their total duration. Then those numbers are categorized into the different

No.	Start	Stop	Event	Cause	Description	Info
1	Wed, W27 04-JUL-07 18:21:47	Wed 04-JUL-07 18:25:07	3min ofb-fail UO	BLID [100%]	Local bumps for X03MA	<a href="#">Alarms</a> <a href="#">Archiver OFB</a> <a href="#">Shift protocol</a>
2	Thu, W27 05-JUL-07 01:30:10	Thu 05-JUL-07 02:11:42	42min beamdrop UO	LIRF [100%]	Failure of Linac modulator 1 (MOD:PIHI, KLY:EOLCHI)	<a href="#">Alarms</a> <a href="#">Archiver OP</a> <a href="#">Shift protocol</a>
3	Sat, W27 07-JUL-07 17:54:15	Sat 07-JUL-07 18:13:08	19min downtime UO	RIRF [100%]	Failure of ARIRF-A1 (KLY:Arc)	<a href="#">Alarms</a> <a href="#">Archiver OP</a> <a href="#">Shift protocol</a>

FIG. 4. (Color) Example screen shot of the event browser Web page.

systems that caused those events. Some examples will be shown in the Sec. V, Figs. 8 and 9.

### G. Automated operation reports

An automated report shows plots of a set of important process variables, like beam current and lifetime, beam sizes, etc., for a given time period. In those plots the events are marked and the event summary is listed. Those reports are used to provide a quick overview, e.g., for periodic operation meetings.

The application to generate those plots can be configured freely. One can define the number of plots and their process variables, the axes, labels and colors of the plots, and even the event types to be displayed. Additional information can be added after the plot, like beam statistics, by providing external commands as arguments to the command line execution.

The current implementation expects one to retrieve the plot data from the *channel archiver*. But the data retrieval is encapsulated in an external wrapper script and can therefore be easily adapted to other data sources.

Figure 7 in Sec. V shows an example report.

### H. Technology of the implementation

All applications are written in Tcl/Tk [7] for the operating system Scientific Linux [8]. We use an Oracle database [9]. The database access uses the “Oratcl” package for tcl [10]. The control system of the SLS is EPICS [5]. All control system access is encapsulated in a single function within the event logger application and based on the EPICS Tcl/Tk Interface “ET” [11].

The event browser is written in PHP4 [12], running on an Apache Web server [13]. The database access uses the OCI8 library of PHP [14].

The Oracle database is a proprietary, commercial software. It is in use at the SLS since the commissioning in 1999 and utilized for many applications. The rest of the above technology has been chosen to be freely available for many different operating systems.

## IV. SLS SITE SPECIFICS

### A. Definitions of categories

The *operation\_category* table keeps the list of modes of operation of the SLS. The definitions are shown in Table I.

TABLE I. SLS operation categories.

<i>op_mode</i>	Description
UO	User operation
BL	Beam line tests
MD	Machine tests
SD	Shutdown

TABLE II. SLS failure areas.

<i>area_ID</i>	Description
LI	Linac
LB	Linac to booster transferline
BO	Booster
BR	Booster to ring transferline
RI	Storage ring
BL	Beam lines
NO	Not specific to area
OT	Other areas

TABLE III. SLS failure groups.

<i>sys_ID</i>	Description
BD	Beam dynamics application
BL	Beam lines
CO	Controls
DI	Diagnostics
DY	Delayed user operation start
EV	Electricity supply, mains
FE	Frontends
ID	Insertion devices
MA	Magnets
OP	Operator fault
OT	Other reason
PS	Magnet power supply
RF	Radio frequency
SA	Safety systems: PSA, VCS, etc.
S+	Scheduled interruption
SU	Radiation protection
TS	Thunderstorm/lightning
UK	Unknown reason
VA	Vacuum
WC	Water cooling

We keep a table to define user-run periods. It consists of the *op\_type* for the operation mode, a *run\_type* and the *start\_date* and *end\_date* of the run. Those run periods are used for the automated generation of operation reports.

Table II shows the failure areas of the cause assignment as defined for the SLS.

Table III shows the failure categories of the cause assignment as defined for the SLS.

## B. Event rules and precedence

The generic application has to be configured by rule definitions of the desired events. The current event rules for the SLS are shown in Table IV. Rules are defined for the operation modes “user operation” (UO) and “beam line tests” (BL). Event precedence is currently only defined for downtime events: it stops all other events.

## C. Event rule variables

All control system variables that are used in the event rules are defined in the *event\_pv* table. The table just keeps a short variable name to be used in the rule formulas and the control system process variable name to connect to.

One specific process variable is the “shift-type.” It reflects the current shift operation mode. A mapping table translates this into the operation mode used for the event logger. This allows one to map several different shift mode strings to the same *op\_mode* of the event database. We have, for example, the shift types “User Operation” and “User Reserve Time” both mapped to the operation category “User Operation.”

## D. Event information data references

The main data sources for the failure analysis at the SLS are the EPICS *channel archiver*, the alarm list of the EPICS *alarmhandler*, and the DUO logbook for SLS shift summaries. The logbook was Web based by design. We created

TABLE IV. Event rules defined for the SLS.

Event	Start Stop	Rule	Op.- Mode	Delay [s]
Downtime	Start	Beam current $I_{\text{beam}} < 50$ mA	UO, BL	1
	Stop	Top-up current $I_{\text{top-up}}$ has been reached	UO, BL	3600
Beamdrop	Start	$I_{\text{beam}} < I_{\text{top-up}}$	UO, BL	150, 300
	Stop	$I_{\text{beam}} > I_{\text{top-up}}$	UO, BL	3
Lifetime	Start	Beam lifetime $\tau_{\text{beam}} < 4.5$ h	UO, BL	60
	Stop	$\tau_{\text{beam}} > 4.5$ h	UO, BL	600
Blowup	Start	Beam height $\sigma_y > 15$ $\mu\text{m}$ or beam width $\sigma_x > 80$ $\mu\text{m}$	UO	3
	Stop	$\sigma_y < 15$ $\mu\text{m}$ and $\sigma_x < 80$ $\mu\text{m}$	UO	60
ofb-fail	Start	Orbit feedback stopped	UO, BL	10, 900
	Stop	Orbit feedback running	UO, BL	20, 900
fpf-fail	Start	Filling pattern feedback stopped	UO, BL	10, 900
	Stop	Filling pattern feedback running	UO, BL	20, 900

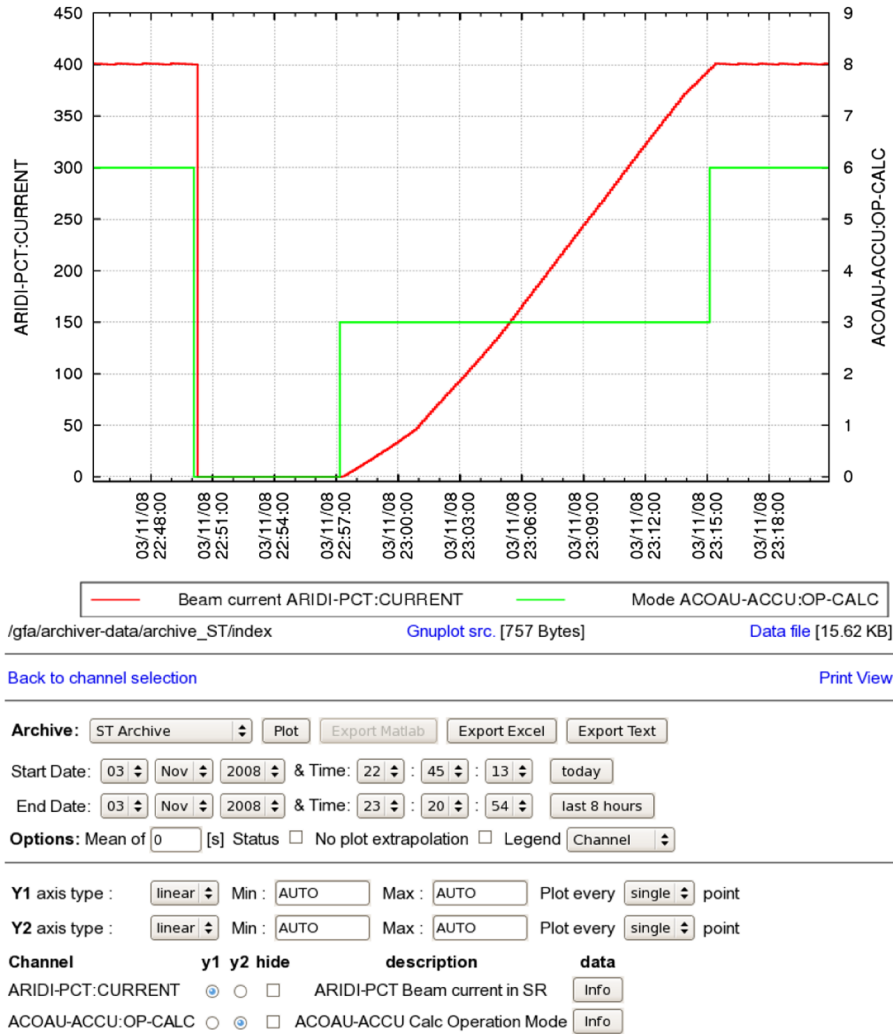


FIG. 5. (Color) A sample screen shot of the *channel archiver* Web interface.

custom Web interfaces for the other two in order to simplify the communication within the laboratory: Web links can be easily exchanged by Emails. All three interfaces are implemented in standard PHP. We will not describe the full functionality of those interfaces here, just how we use URLs to reference their data.

### 1. EPICS channel archiver Web interface

Figure 5 shows a screen shot of the channel archive retriever Web interface that allows one to access and display the archived data. Up to 100 process variables can be

displayed on two *Y* axes at a time. The data is selected by the time range and the names of the process variables. The selection of the data and the configuration of the display can be configured by parameters of the URL string. An example for the relevant part of the syntax is shown in the Table V. The tokens *startDate*, *startTime*, *endDate*, *endTime* facilitate the specification of the time range of the event. The *NAMES* token provides the means of specifying process variables which are relevant to the specific type of event.

TABLE V. Example URL to the channel archive retriever.

Host	http://archif.psi.ch/archiver/php/output.php?
Data	DIRECTORY = /import_arch/archiver_ST/index&
PVs	NAMES = ARIDI-PCT:TAU-HOUR%0D%0AARIDI-PCT:CURRENT&
Start	startDate = 09/25/2008&startTime = 00:00:00&
Stop	endDate = 09/25/2008&endTime = 13:51:50&
Plot	gpYMin = 0&gpYMax = 410&gpY2Min = 0& gpY2Max = 10&ARIDI-PCT:TAU-HOUR = y2

No.	Date	Channel	Status	Severity	Value
1	07-JUL-07 17:54:15	ARIRF-A1-FC.STATE-GET	STATE	MINOR	Standby
2	07-JUL-07 17:54:15	ARIRF-A3-Ph-COMP	LOLO	MAJOR	-56
3	07-JUL-07 17:54:15	ARIRF-A3-Ph-COMP	NO_ALARM	NO_ALARM	0
4	07-JUL-07 17:54:16	ARIRF-CAV1:Vcav	LOW	MINOR	20.28
5	07-JUL-07 17:54:18	ALIRF-GUNIDLE-TRIG	STATE	MAJOR	disable
6	07-JUL-07 17:54:18	ARIRF-A1:FirstError	LINK	MAJOR	KLY:Arc
7	07-JUL-07 17:55:28	ARIRF-A1:FirstError	NO_ALARM	NO_ALARM	none
8	07-JUL-07 17:55:38	ARIRF-A1-FC.STATE-GET	NO_ALARM	NO_ALARM	RF ON
9	07-JUL-07 17:56:41	ALIRF-GUNIDLE-TRIG	NO_ALARM	NO_ALARM	enable
10	07-JUL-07 17:59:42	ARIRF-CAV1:Vcav	NO_ALARM	NO_ALARM	400.53

FIG. 6. (Color) Example screen shot of the Web interface of the *alarmhandler*.

The function `ARlink` generates the text “Archiver <Link-Text>” with a URL reference to a plot of the archived data. This string is written to the *event\_info* entry. The syntax of this function is the following: “`ARlink <Link-Text> <PV1> <PV2> <event_start> <event_stop>`.” Up to two process variables <PV1> and <PV2> can be specified, each will be shown on a separate vertical axis. The time range <event\_start> to <event\_stop> is extended by 5 minutes in both directions for a better overview.

The channels to be displayed are configured in the *event\_links* table. For downtime and “beamdrop” events the link to the archiver shows the beam current and a process variable for the accumulation state. The latter changes state when no beam is in the machine, when top-up is stopped, when top-up current has been reached, etc. For lifetime events the beam lifetime is shown together with the beam current. In case of “blowup” events the horizontal and vertical beam sizes are shown.

The status information of the corresponding feedback is shown in case of “ofb-fail” for orbit feedback [15] failures and “fpf-fail” for filling pattern feedback [16] failure events.

## 2. EPICS alarmhandler Web interface

The SLS Web interface for the fault condition retrieval is a PHP Web page to display alarms of a given time range (see Fig. 6). An example URL to this page is shown in Table VI. An *event\_info* entry is generated by the function `ALHlink`. It has the syntax “`ALHlink <event_start> <event_stop>`” and displays the text “Alarms” with

TABLE VI. Example URL to the alarm log Web interface.

Host	<code>http://abk.web.psi.ch/operation/alh/?</code>
Start	<code>start = 22-Sep-08.11:56:34&amp;</code>
Stop	<code>stop = 22-Sep-08.12:01:57</code>

TABLE VII. Example URL to DUO logbook Web interface.

Host	<code>https://duo.psi.ch/duo/logbook_search.php?</code>
Log-book	<code>LBID = 7&amp;</code>
Start	<code>STARTDATE = 20080923&amp; STARTTIME = 1400&amp;</code>
Stop	<code>ENDDATE = 20080923&amp; ENDTIME = 1500&amp;</code>
Flag	<code>REDIRECT = YES</code>

a reference to a plot of the alarm list for the given time range. The shown time range starts 5 minutes before the <event\_start> since some alarms do not take effect on the beam immediately.

## 3. DUO logbook for SLS shift summaries

In order to provide links to a specific logbook entry, one had to specify the logbook ID of the entry. A reference by an ID does not work for the operation event logging system, since an event could be posted even before the shift summary had been written. Therefore we added the feature of referencing a logbook entry by date. An example URL is shown in Table VII. The *event\_info* entry is generated by the function `LOGblink`. It has the syntax “`LOGblink <event_start> <event_stop>`” and displays the text “Shift protocol” with a reference to logbook entries for the given time range. The used time range starts with the beginning of the shift during the <event\_start> and ends after the beginning of the shift during the <event\_stop>. Therefore it contains all shift summaries for the time range of the event. In the case that the event occurred within one shift, the flag “REDIRECT = YES” causes the Web server to directly display the logbook entry. A list of entry links is shown to cater for situations where more than one logbook entry exists for the specified time range.

## E. Event alarm evaluator configuration

A script has been written to populate the *alh\_hint* table. It allows one to select process variables of a given pattern from the *alarmhandler* configuration files and assigns the *hint\_group*, *html\_comment*, and failure system and area to each alarm when adding the alarm to the *alh\_hint* table. Table VIII shows the different hint groups currently de-

TABLE VIII. The alarm groups at the SLS.

Description	<i>hint_group</i>	<i>hint_count</i>	No. PVs
Coupled bunch modes	CBM	1	1437
Insertion device interlock	ID	3	288
Beam line access control	LAC	2	18
Machine interlock	MIS	2	214
Magnet protection interlock	MPS	2	682
Magnet power supply fault	PS	3	1128
Personal safety system	PSA	2	4
Radio frequency	RF	0	4
Vacuum interlock	VCS	2	338

fined at the SLS and the *hint\_count* to limit the output for each group in case of multiple alarms from one group.

## V. EVENT DATABASE EXPERIENCE

The event logger has been in operation since August 2006. Downtime and beamdrop events have been

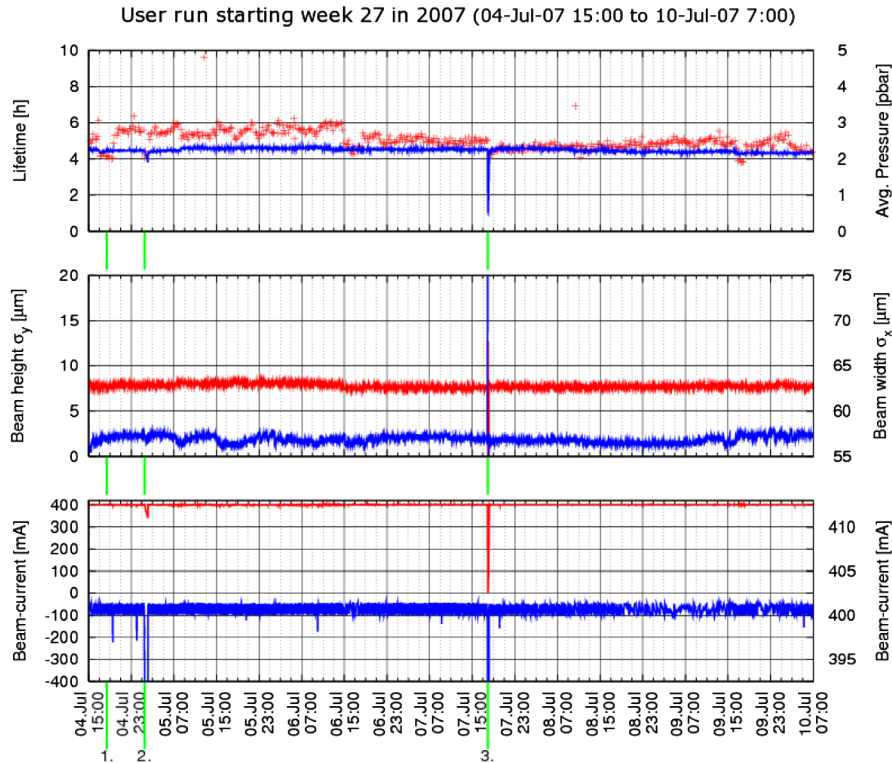
TABLE IX. Statistics of the recorded events at the SLS.

Event type	Rule added at	No. at UO	No. at BL
Downtime	January 2006	277	153
Beamdrop	January 2006	167	50
Orbit feedback failures	August 2006	163	14
Miscellaneous		8	0
Blowup	June 2008	5	0
Lifetime	August 2008	1	0
Filling pattern FB failure	September 2008	0	0

imported from a previous file based system for 2006 [17]. In total the number of 838 events have been recorded until the end of September 2008. Table IX shows the number of events of the different types from the first implementation of a rule until the end of September 2008. No rule exists for the event type “miscellaneous” since those events are added manually only. The average time between the recording of an event and the manual assignment of a cause has been two days. While the causes for most events are added by the operation manager, the “ofb-fail” event causes are added by the feedback experts.

About a third of the events during beam line tests were scheduled for a purpose. This is simply reflected by assigning the cause “scheduled” to those events.

The system has been extended continually since 2006. The event logger is now robust against all kind of failures, like disconnecting control system process variables or transient oracle database outages. If single events had been lost, e.g., due to an outage of the oracle database,



### Events:

- 1.) 04-JUL-07 18:21 ofb-fail BLID [100%]: Local orbit bumps for X03MA-Adress (3min)
- 2.) 05-JUL-07 01:30 beamdrop LIRF [100%]: Failure of Linac mod. 1 (KLY:EOLCHI) (42min)
- 3.) 07-JUL-07 17:54 downtime RIRF [100%]: Failure of ARIRF-A1 (KLY:Arc) (19min)

Operation from 4.7 15:00 to 10.7 07:00, Total hours: 136.0h, MTBF: 68.0h \*2  
 Top-up Ready : 0.04h = 0.03%  
 Machine Down : 0.16h = 0.12%  
 Accumulating : 0.84h = 0.62%  
 Light Available : 134.94h = 99.23%  
 Net User Time : 135.67h = 99.77%  
 Total Downtime : 0.31h = 0.23%

FIG. 7. (Color) User-run overview with events marked.



the error log file of the event logger contains the commands to add them manually later. No single event had to be added manually for the past six months. Some events had to be deleted manually, when the quality of the measurement did not match the formulation of the event rule. For example, a calibration failure of the diagnostics caused the measured horizontal beam size to drift over weeks to larger values. This caused about a dozen *blowup* events although the real beam size did not change. Those events were deleted manually and the rule was adapted to a larger threshold for the horizontal beam size until the measurement had been recalibrated.

Several SLS applications are using the event information. The events of a shift are automatically added to the shift summary. A weekly user-run overview is generated as a Web page and shows the event and cause information marked in plots of the main control system variables, like beam current, beam sizes, and beam lifetime (see Fig. 7). Those overviews are used to discuss the operations performance in the biweekly operation meetings and if necessary to decide on measures to improve the performance.

The application for statistical analysis of the downtime is used for yearly reports. Figures 8 and 9 show some statistics from the event database. Those reports are used for making strategic decisions about upgrade and maintenance plans.

The event browser is used for the daily analysis of faults. It forces the operation manager to document each outage in a timely manner and allows everyone to get a quick update on recent operation distortions. It helps all system experts to answer questions like: “Has a similar problem occurred earlier?” “Is a particular klystron arcing more than others?” “Which events caused beam interruptions longer than three hours last year?”

It is not easily possible to quantify the reduction of beam distortions due to the usage of the operation event logging system. But the author is convinced that the system does help to improve the quality of the beam delivery. In particular, the investigation of infrequent failures and the

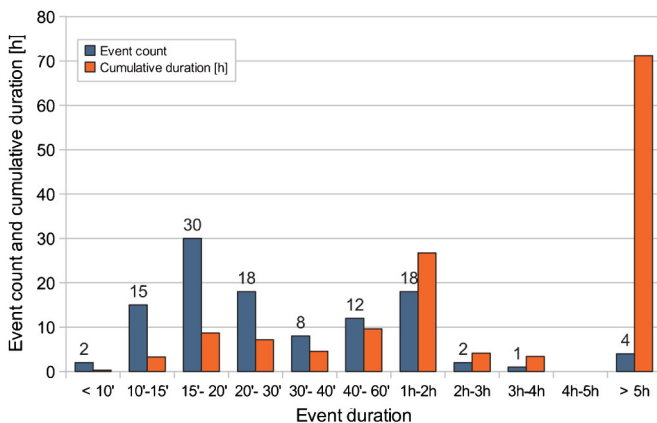


FIG. 8. (Color) Durations of downtimes in 2007, calculated from the data from the operation event database.

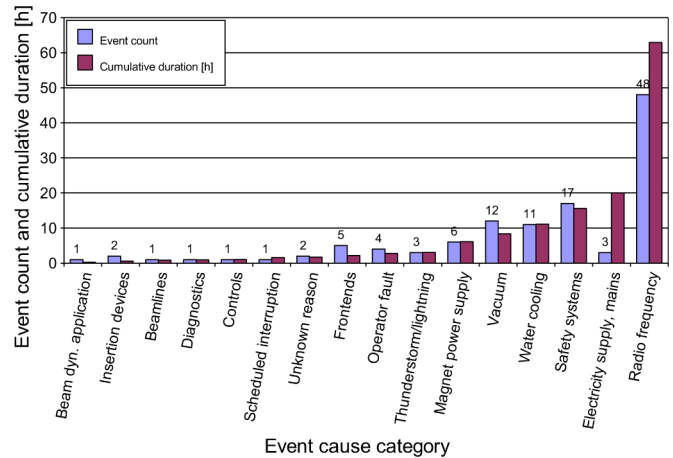


FIG. 9. (Color) A chart of the downtime event numbers and total duration per system, based on the data from the operation event database.

serious quantification of the beam effects of repeating failures is hardly feasible without such a database. It helps to provide a thorough overview of all problems and to put each one into perspective.

The system helps to accelerate many tasks of the operations manager. The amount of time to compile the operation statistics for yearly reports shortened from several days of work to a few minutes. At the same time the quality of the data improved significantly. The weekly operation report is automatically updated every hour and therefore available to all participants of the operations meeting beforehand.

## VI. SURVEY ON FAILURE ANALYSIS AT LIGHT SOURCES

While the operation event logging system proved to be very useful for the SLS, we wanted to know if it could be of any use to other light sources. Therefore we have asked nine major light sources about their experience with failure analysis. The questionnaire was split in two sections. The first part dealt with the used operation metrics, like definitions of beam downtime and beam availability. The second part queried their methods of failure documentation.

### A. Operation metrics

Four out of nine light sources apply common sense rules for the calculation of downtime. A downtime in those cases subsumes all kinds of problems that prohibit the majority of the users from making use of their beam time. Only APS and SPring-8 track the injector failures that cause decaying beam, since those facilities run normally in top-up mode. For most other light sources an injector failure is counted as downtime if it causes the beam current to drop below a certain threshold.

The calculation of the beam availability varies considerably among the light sources. Most light sources have a

“short uptime rule”: a beam delivery between two beam outages is considered downtime if it is shorter than a given time interval. But this time interval still varies between 15 and 60 minutes.

While seven out of nine light sources do provide compensation time to users, the accounting of this compensation time in the beam availability is different in most cases. Some light sources subtract the delivered beam time during compensation from the downtime before calculating the availability. Others just do not count downtime during compensation but otherwise count it as scheduled beam time. In one case the compensation time is not accounted for at all in the calculation of the beam availability.

All light sources do track other beam distortion types like increased beam sizes or orbit problems. But none of those light sources compile periodic statistics regarding such failures.

### B. Failure documentation

At all light sources, beam outages and beam distortions are analyzed with the help of process variable archives and alarm logs. A lot of effort has been spent on the development of powerful tools to help with the analysis of the archived data. In most cases additional data sources like electronic logbooks or special beam interlock data is utilized, too.

While eight of nine light sources maintain a failure database, the concepts of these databases differ. In most cases they are designed for equipment failures. Whatever failure was relevant to the operation of the machine is kept in those databases. The caused downtime is in most cases part of the failure report. Failure reports are mostly manually entered by the operator. Some light sources maintain a database dedicated to beam outages. SPring-8 keeps track of injector failures causing decaying beam during top-up in the same database. Only APS autogenerates the downtime entries in the so-called *fill history*. The operator adds a short description of the failure and assigns a responsible group. In all other cases the full report is manually entered by an operator. In some cases, like at APS, the data is postprocessed and extended by the operation manager for user-run summaries.

### C. Survey conclusions

The rules to calculate the beam statistics at light sources are far from standardized. The term “beam availability” is often only defined in common sense terms. This is convenient, since it allows a judging of every single event for its implication on the usability of the beam. In practice, a common sense rule for downtime combines several different events into one. Each individual event can of course be clearly defined: no beam, not enough beam, unstable beam, etc. The operation event logging system can then be used to

log all types of events while still leaving the final decision to the judgment of the statistician: which events should be accounted for as downtime, but based then on the recorded events.

The failure documentation has been considered important at all facilities. Many different types of applications have been developed to keep track of beam outages, beam distortions, equipment failures, and similar events. Every laboratory makes use of archived control system data and failure logs in order to analyze the reason for a failure. Nearly all light sources use a database to document those failures. In most cases those failure databases are not directly coupled to the operation metrics of the light source. The databases are instead used to manually document each type of failure and often they are intended to record equipment failures.

We consider the operation event logging system not to be a replacement for an equipment failure database but as a complement. Equipment failure reports have a specific life cycle: the report is submitted, assigned to a person, actions are taken, the problem is solved, and the report is closed. Those reports should contain a link to the documentation of the resulting beam distortion. Whenever a beam distortion has been in relation to an equipment failure, a link to the failure report should be added as an *event\_info* to that event.

## VII. SUMMARY

The operation event database proved to be an extremely valuable tool for operation management of the Swiss Light Source. It simplifies the tasks of the operation manager and helps to prioritize maintenance and performance upgrade plans.

A survey of nine major light sources showed that only very few facilities have an automated beam outage database. But several facilities do keep a manual database of beam outages and distortions. We think that an automation of such a database has some major advantages. The automated recording has defined rules to precisely account for each defined beam distortion, while the manual recording depends on the interpretation of the diagnostics by the individual operator. In addition, the automatic collection of the relevant event information reduces the effort to maintain such a database and therefore allows one to record a large variety of different types of beam distortions. The generic implementation of the operation event logging system would allow for its usage at other facilities. The chosen technology allows the implementation on many different platforms. Adaptation to other control systems would be simple in most cases. In order to use the application efficiently, one needs to provide URL controlled Web access to the major data sources of the facility like control system archives, alarm logs, and electronic logbooks.

## VIII. OUTLOOK

As a next step we plan to integrate our equipment fault tracking system with the event database. A commercial Web based fault tracking system has been recently introduced at the institute. We will develop a dedicated application to submit fault reports from the control room. This application will provide the means of relating the failure report to a beam distortion. In this case a link to the related beam distortions will be added to the failure report, and a link to the failure report will be added to the event information.

Currently the event causes are primarily added by the operations manager. With the help of the alarm evaluation tool this task could be delegated to the operators. A dedicated application should simplify the cause assignment with automated alarm evaluation for the operators.

The operation event logging system is not limited to the operation of light sources. At PSI we operate several large accelerator facilities. We plan to evaluate the applicability of the event database for the operation of those accelerators.

## ACKNOWLEDGMENTS

The author wishes to acknowledge the contributions of the operation managers who participated in our survey on failure analysis. The information on the different tools and procedures was inspiring and induced new ideas for the enhancement of our toolkit for operation management.

## APPENDIX A: QUESTIONNAIRE ON FAILURE ANALYSIS

Nine major light sources have been asked about their operation metrics and their failure documentation. The questionnaire is shown in Appendices A 1 and A 2. The following persons answered for their light sources: Randy Flood, APS Chief of Operations; Jean-Luc Revol, ESRF Operations Manager; Masaru Takao, SPring-8 Operation

Team Leader; Vince Kempson, Diamond Operations Manager; Jean-Marc Filhol, SOLEIL Division Head Sources and Accelerators; Peter Kuske, BESSY II Operations Manager; Emanuel Karantzoulis, ELETTRA Operations Manager; Anke-Susanne Müller, ANKA Operations Manager.

### 1. Operation metrics

How do you exactly define “beam downtime”? Is it “time where beam current is below  $x$  mA” or a more complex rule?

How do you exactly define “injector downtime”? (This is more important for top-up operation, but you may account for delayed refill?)

How do you exactly define “beam availability”? (i) Do you count each uptime, e.g., very short ones between two beam outages? (ii) Do you provide compensation time to users when a long downtime happens? (iii) If yes: how do you take this compensation into account in the availability?

Are other types of failures regularly analyzed and/or tracked? For example, (i) increased beam sizes or (ii) orbit feedback failures.

### 2. Failure documentation

What data sources are used to analyze beam outages or other beam distortions: (i) control system archive data; (ii) control system alarm logging data; (iii) shift protocols/ logbook entries; (iv) do you use other data?

Do you keep a failure database? If yes: (i) what type of failures are kept there (beam outages, injector failures, others); (ii) what type of data is filled into the database; (iii) who updates the database; (iv) what is the typical update period; (v) who has access to the database?

## APPENDIX B: EVENT DATABASE APPLICATIONS

In addition to the main applications shown in the overview in Fig. 1 there are several other applications and

TABLE X. List of applications that access the event database.

Name	Function	Usage
Event logger	Detect events according to rules and store them together with related data references to a database table.	Server application
<i>addinfo</i>	Add descriptive information and related data references to an event.	Command line
<i>event</i>	Manually add or delete an event in the database.	Command line
Event statistics	Compile the event statistics for a given time intervall.	Command line
Operation reports	Compile and publish an overview page for the current operation run.	Command line, hourly cron <sup>a</sup> job execution
Event browser	Displays filtered lists of events together with event cause and event info data.	Interactive PHP Web page
<i>cause</i>	List and manipulate event cause assignments.	Command line
<i>xcause</i>	List and manipulate event cause assignments.	Interactive, graphical Tcl/Tk application

<sup>a</sup>UNIX daemon to execute scheduled commands.

TABLE XI. List of auxiliary scripts with no direct access to the event database.

Name	Function	Usage
<i>ARlink</i>	Create data reference to the channel archive retriever.	Library function called by the event logger and <i>addinfo</i> .
<i>LOGBlink</i>	Create data reference to the SLS operation shift logbook.	Library function called by the event logger and <i>addinfo</i> .
<i>ALHlink</i>	Create data reference to the SLS alarm browser.	Library function called by the event logger and <i>addinfo</i> .
Alarm hints	Evaluate alarms of a given time range.	Command called from <i>cause</i> .

scripts interfacing with the database that were mentioned throughout the paper. An overview of those applications is listed in the Table X. Table XI lists the scripts used within the context of the event logging system that do not directly access the event database.

APPENDIX C: DATABASE MODEL OVERVIEW

Figure 10 shows a database model diagram of the operation event logging system. The tables are explained in Sec. III.

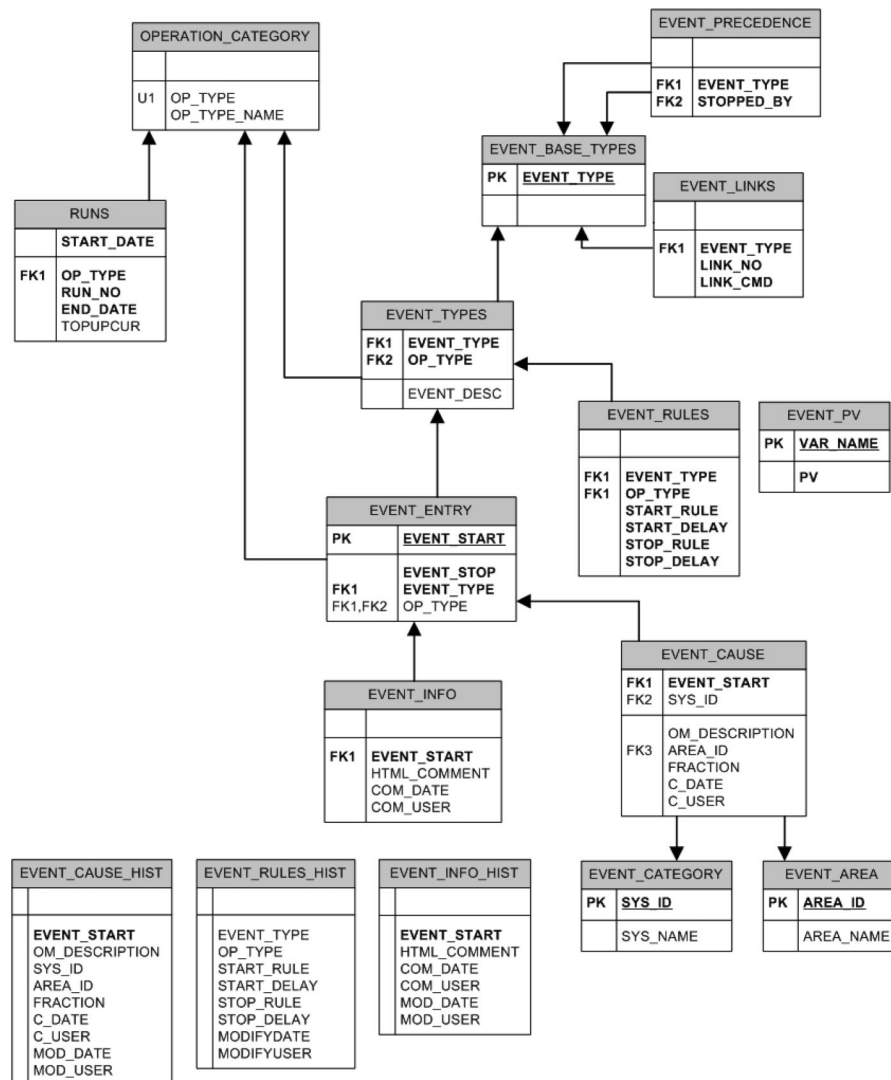


FIG. 10. (Color) Database model overview.

- [1] T. Berners-Lee, R. Fielding, and L. Masinter, Uniform resource identifier (uri): Generic syntax, <http://www.w3.org/Addressing/#rfc3986> (2005).
- [2] M. Kraimer, B. Cha, and M. Anderson, in *Proceedings of the Particle Accelerator Conference*, 1991, pp. 1314–1316.
- [3] K. Kasemir and L. Dalesio, in *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, 1999, pp. 463–465.
- [4] A. Bertrand and H. Weyer, *Nucl. Instrum. Methods Phys. Res., Sect. A* **546**, 591 (2005).
- [5] Experimental Physics and Industrial Control System, <http://www.aps.anl.gov/epics/>.
- [6] K. Kasemir, Channel Archiver Manual, <http://ics-web.sns.ornl.gov/kasemir/archiver/manual.pdf>.
- [7] Tcl developer site, <http://www.tcl.tk/>.
- [8] Scientific linux, <https://www.scientificlinux.org/>.
- [9] Oracle database, <http://www.oracle.com/database/>.
- [10] Oratcl: an extension that provides an oracle call interface layer to the tcl language, <http://sourceforge.net/projects/oratcl/>.
- [11] Et: Epics tcl/tk, <http://www.aps.anl.gov/epics/extensions/tcl>.
- [12] The php group, <http://www.php.net/>.
- [13] Apache Software Foundation, <http://www.apache.org/>.
- [14] Php oci8 Manual, <http://www.php.net/manual/en/book.oci8.php>.
- [15] M. Böge, B. Keil, A. Lüdeke, and T. Schilcher, in *Proceedings of the 21st Particle Accelerator Conference, Knoxville, 2005* (IEEE, Piscataway, NJ, 2005), pp. 1538–1540.
- [16] B. Kalantari, V. Schlott, and T. Korhonen, in *Proceedings of the 9th European Particle Accelerator Conference, Lucerne, 2004* (EPS-AG, Lucerne, 2004), pp. 2885–2887.
- [17] A. Lüdeke, in *Proceedings of the 10th European Particle Accelerator Conference, Edinburgh, Scotland, 2006* (EPS-AG, Edinburgh, Scotland, 2006), pp. 3318–3320.