USE OF E-LOGBOOK IN VEPP-5 CONTROL SYSTEM

D.Yu.Bolkhovityanov*, R.E.Kuskov
The Budker Institute of Nuclear Physics, Novosibirsk, Russia

Abstract

An electronic logbook (e-logbook) becomes a must for large experimental facilities not only during operation, but also at building and commissionning stages (where VEPP-5 is now). Unfortunately, the "market" of such products is almost nonexistent. So, the choice is narrow: either use some other lab's software (adapting it for local needs) or create your own one from scratch. We have chosen the former way and picked DOOCS e-logbook from DESY. Main changes concerned localization (since Russian uses cyrillic letters, not latin) and data feeding mechanism (due to different model of logging from applications). Integration with GIS and alarm system is being examined.

VEPP-5 INJECTION COMPLEX

VEPP-5 injection complex [1] is destined to provide electrons and positrons to all BINP electron-positron colliders — existing VEPP-3/VEPP-4M complex and VEPP-2000, which is currently being put into operation. VEPP-5 is able to produce more than 10^{10} positrons per second, which exceeds requirements of all users.

VEPP-5 consists of 300 MeV electron linac, conversion system, 510 MeV positron linac, and damping ring with beam injection and extraction channels.

REQUIREMENTS

From the very beginning of VEPP-5 construction, records are kept in many different ways. Applications perform local logging to text files. Operators write down "interesting data" to a register — just a thick note-book, lying on a control stand. Engineers, dealing with various VEPP-5 subsystems, make their own logs.

In such a way, valuable information is spread in many places, and even if someone knows that some records do exist, they are often hard to find even in a few weeks time, not speaking about years.

Thus, VEPP-5 required a unified electronic system, able to store various kinds of information — text, images and other types of binary data.

Beyond automation and unification of data storage, the paramount requirement is an ability to *search* among the stored data.

CHOICE

A similar requirement had risen in other high-energy physics centers some years ago, and it got an answer — *E-Logbook*[2, 3], and even more advanced uses of this technology were invented[4].

Since e-logbooks for control systems have narrow use, there's no "market" of such software. While many labs create their own e-logbook software, BINP in general and VEPP-5 in particular don't have enough manpower for this. So, we chose to adapt some other lab's e-logbook software for our needs.

We have made a brief analysis of available e-logbooks, where comparison table [5] was extremely useful (while it is dated, no newer analogues exist). Finaly, we have chosen DOOCS E-Logbook[2], due to following reasons:

- It is a complete product.
- It is easily available, and has a web-site where one can test it interactively.
- It requires very little specific software, just Java+Apache+Tomcat combo, which is readily available in most Linux distros. And all these are free software (as opposed to e.g. JLab's E-Log, which requires Oracle).
- It's code has moderate size, which allows to grok it and to tailor to our needs.
- It is still maintained by original author.

GENERAL APPROACH

DOOCS E-Logbook is browsed via web-interface, with Java-applet used for vivid time overview. For logging, E-Logbook provides 2 ways:

1. Via web-form — used for manual input; this allows to input textual data with optional binary attachments.
2. Via so-called "virtual printer" — this way is used by applications, which just print graphics to PostScript printer, redirected to E-Logbook.

Unfortunately, the 2nd way, which is the key to automatic logging, didn't fit into VEPP-5 reality:

- E-Logbook's "virtual printer" supposes use of LPD, not CUPS (which manages printing at VEPP-5).
- Printing Russian to PostScript is far more complicated than that for latin, and has many pitfalls. Translation of cyrillic PostScript back into text is also an intricate task.
- Most applications already have their own logging mechanism, which is *text*-oriented.

To solve this problem, we made some modifications to web input channel, so that now applications can log via HTTP, simulating the browser.

* bolkhov@inp.nsk.su

Thus, VEPP-5 variant of E-Logbook has following choices for logging:

1. Manual input is still performed via web-form.
2. Applications perform automatic logging via HTTP, using a newly-developed `DoELog()` C API.
3. For use in scripts a standalone utility was made — `elog`, which is just a wrapper around `DoELog()`.
4. Each GUI application has a “Make e-logbook record” button, giving user a short-cut to make subsystem-specific notes (see Fig.1).

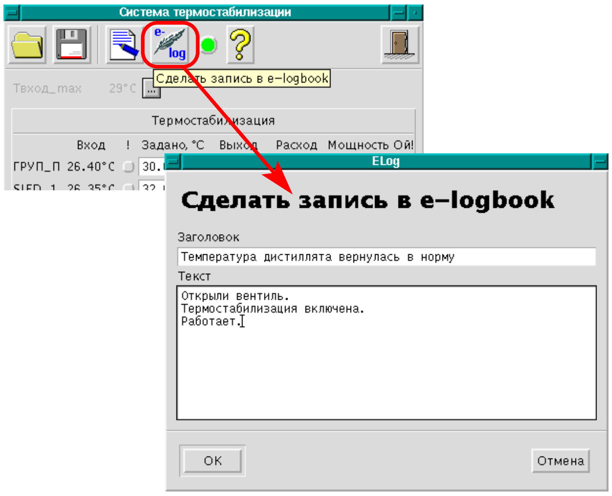

Figure 1: “Make e-logbook record” window.

“Virtual printer” isn’t used at all.

LOCALIZATION

Background

Internationalization (usually abbreviated as “i18n”) and localization (“L10n”), in their core aspect — characters’ encoding — rarely concern English-speakers, and, to a great extent, west-european-languages-speakers too. That’s determined by the fact that all english letters are contained in ASCII, and most west-european ones are included in ASCII superset — ISO8859-1 (and its variant Windows-1252). So, these languages can do with 1-byte encodings. And even in 2-byte Unicode (UCS-2) character codes are the same.

For cyrillic-based languages, situation is just the opposite: there are several different encodings (KOI8-R and Windows-1251 are most used), all of them are incompatible with ISO8859-1.

DOOCS e-logbook approach

As DOOCS e-log book was created in DESY, its target languages are English and German. So, despite being written in Java (which has i18n features), it stores all data directly, byte-by-byte, and assumes text to be ISO8859-1.

Consequently, cyrillic letters result in hardly-readable combinations like `роза`, where the numbers represent Unicode positions of respective non-Latin1 letters. The same would happen with CJK languages too. Sure, searching through such content also didn’t work.

Of course, that’s absolutely inappropriate for VEPP-5, as our main language is Russian.

Solution...

The simplest way was the most traditional one: to rewire E-Logbook to use KOI8-R instead of 8859-1 (KOI8 is also a single-byte encoding, and differs only in the 2nd half). That’s how most applications were localized up to the end of 1990s.

But that’s a dead-end track — while use of KOI8 solves a local problem, it immediately kills portability and interoperability with other systems.

In modern systems UNICODE and its text-safe presentation form — UTF-8 — became de-facto standards. UNICODE allows to represent *all* characters of all today’s living languages simultaneously.

That was an obvious way to go — to switch E-Logbook to UTF-8 completely.

...and its implementation

As Java is extremely widespread, similar problems had appeared numerous times. There’s a whole class of interconnected localization-related tasks, and their solutions are well-known[6].

The core problem (in case of E-Logbook) was mixing of `char` and `byte` types. For 8859-1 they can be used interchangeably, but any other encoding wouldn’t work in such a way. 8859-1 was in some places hardwired in E-Logbook sources, or was “supposed” as default in others.

Thus, the task was to introduce correct charset conversion at servlets’ boundaries, as shown on Fig.2.

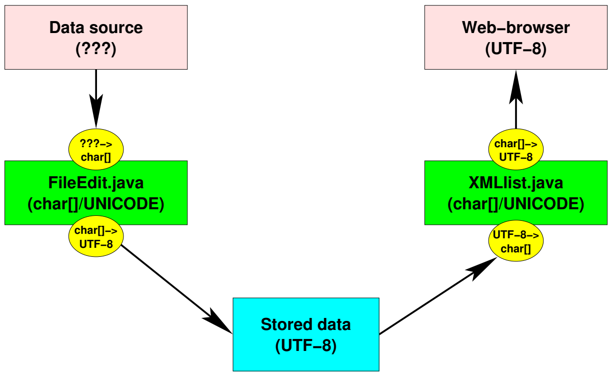

Figure 2: Encoding conversion at E-Logbook servlets’ boundaries.

As all modern web-browsers support UTF-8, it is used for output unconditionally.

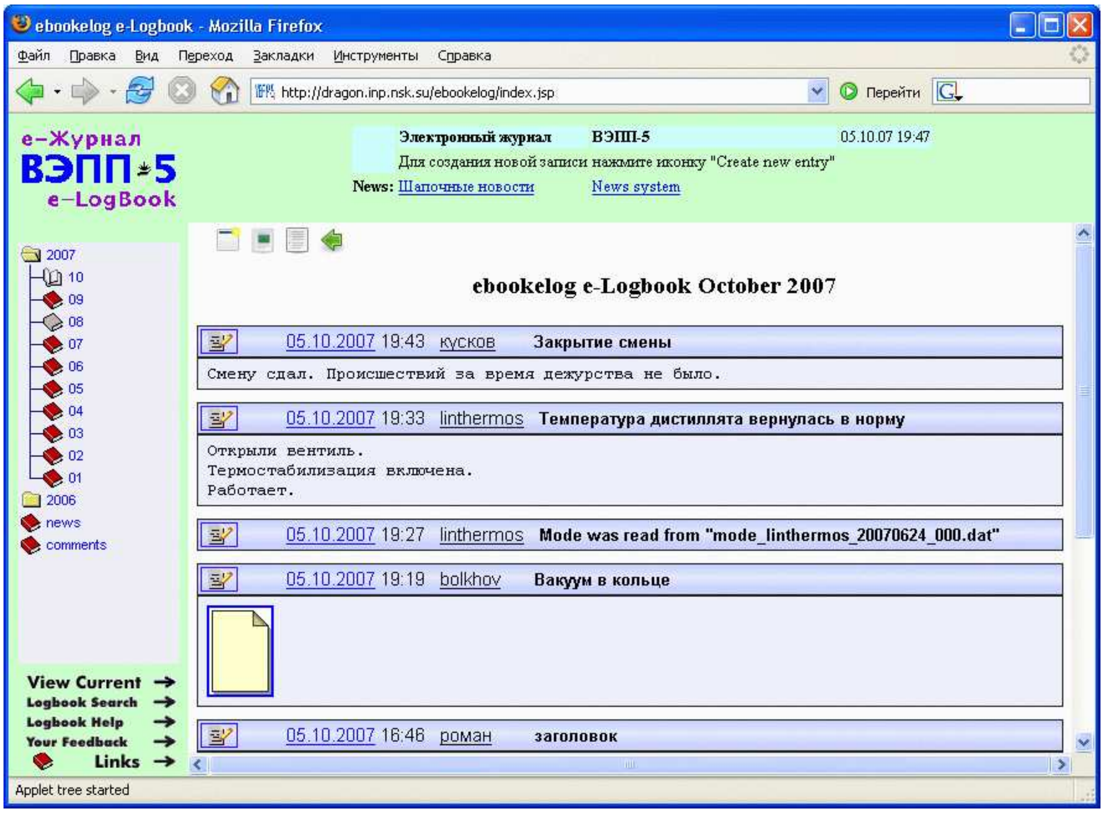

Figure 3: Screenshot of VEPP-5 e-logbook.

PRESENT STATUS

E-Logbook appearance was slightly changed for VEPP-5 to distinguish from DOOCS (see Fig.3).

It is currently used in test operation, to gain experience what other VEPP-5 specific changes are required. At least the repertoire of fields should be slightly changed, to better suit VEPP-5 needs.

Since GIS is being introduced at VEPP-5, the natural thing to do is to integrate E-Logbook with GIS — so that while viewing some event one could see its position at the facility "map". At present time the most natural solution seems to be just an addition of one more field — "GIS position". This text field would be an "address" (or a list of addresses), understandable by GIS.

CONCLUSION

As could be predicted, E-Logbook is a welcome addition to VEPP-5 control system's infrastructure.

Most staff hold the opinion that E-Logbook should have been deployed "the day before yesterday".

Choice of DOOCS E-Logbook proved to be a right decision, since it allowed to get a full-featured e-logbook with modest efforts. Plus, DOOCS E-Logbook organization makes adapting it to VEPP-5 needs a simple task.

REFERENCES

[1] P.V.Logatchev et al, "Status of VEPP-5 Injection Complex", Proc. RuPAC-2006 `http://rupac2006.inp.nsk.su/ready/moao10.pdf`

[2] R.Kammering, "An XML based web service for an electronic logbook", Proc. PCaPAC-2002 `http://www.lnf.infn.it/conference/pcapac2002/TALK/WE-02/WE-02_talk.pdf`

[3] R.Kammering et al, "The Electronic Logbook @ the TTF VUV-FEL - Making the next step" `http://conference.kek.jp/pcapac2005/paper/TUB4.pdf`

[4] T.Larrieu et al, "Beyond an Electronic Logbook" `http://conference.kek.jp/pcapac2005/paper/TUB5.pdf`

[5] Comparison of several E-logbooks `http://docdb.fnal.gov/ILC/DocDB/0002/000292/005/Elogs.pdf`

[6] "Russian letters and more..." (in russian) `http://www.jug.ru/servlets/index?page=rus`