Strigi
Searching files in KDE4
Jos van den Oever
So, list your "Crazy Ideas" here: (http://strigi.sf.net)

- give me all e-mails that have more than one xsl attachment and are overall bigger than 1 MB
- give me all e-mails from user xyz about music but without any music file attached
- show me all music files rated better than 90 % (in amarok) which I played in March
- show me all kopete/IM talks with contact xyz with links to kde.org
- give me all documents related to a scientific reference e.g. "A. Manz, J. C. T. Eijkel, Pure Appl. Chem. 2001, 73, 1555-1561"
- display all files larger than 5kB that I have downloaded in march
- give me all documents related to a specific chemical compound AND a specific author
- find all my social-bookmarked pages on strigi (eg. on del.icio.us or connotea.org)
- give me from all music-related rss-feed posts those containing artist names which are also in my amarok collection
- show me all konqueror-visited locations (local/remote/http/whatever).
What is searching about?

A search interface should

- show the user files or parts of files that match the query,
- match the current context
- and open entries from the search result in the right program

Strigi  Nepomuk
Java has nice streaming base class

```java
public StreamDemo(URL url) throws IOException {
    InputStream filestream = url.openStream();
    ZipInputStream zipstream = new ZipInputStream(filestream);
    ZipEntry entry = zipstream.getNextEntry();
    while (entry != null) {
        handleEntry(zipstream, entry);
    }
}
```
class StreamBase<T> {
    ...

public:
    ...

    virtual int32_t read(const T*& start, int32_t min, int32_t max) = 0;
    virtual int64_t reset(int64_t pos) = 0;
    ...
};

void readdemo() {
    int32_t nread;
    const char* data;
    nread = jstream->read(data, 1, 0); // read at least 1 byte
    jstream->reset(0); // reset to start of stream
    nread = jstream->read(data, 3, 3); // read exactly 3 bytes
}
BufferedStream<char>

Stream with a buffer

- most common use case
- implement one simple function
- called when the buffer is empty

class BufferedStream<T> {
    ...
public:
    ...
    virtual int32_t fillBuffer(T* start, int32_t space) = 0;
    ...
};

Examples

- FileInputStream
- BZ2InputStream
- GZipInputStream
- InputStreamReader
- ProcessInputStream
Streams without buffer

**SubInputStream**
- a size limited version of another stream

```cpp
class SubInputStream<T> {
    ...
public:
    SubInputStream(
        StreamBase<char>* input,
        int32_t size);
    ...
};
```

**StringTerminatedSubStream**
- a size limited version of another stream

```cpp
class SubInputStream<T> {
    ...
public:
    SubInputStream(
        StreamBase<char>* input,
        int32_t size);
    ...
};
```
class SubStreamProvider {
  ...
public:
  SubStreamProvider(StreamBase<char>* input);
  virtual StreamBase<char>* nextEntry() = 0;
  const EntryInfo& getEntryInfo() const;
...

Examples
- TarInputStream, ZipInputStream
- ArInputStream, RpmInputStream
- MailInputStream

Split a stream up
- access parts of files
- implement one simple function
- called to get streams one after the other
**Example: TarInputStream**

**Simple SubStreamProvider**

- fixed size blocks
- no additional buffer required
- parse the header into the `EntryInfo` object
- Position the stream at the start of the content and create a `SubInputStream` with the given size of the stream
DeepFind (a better Find)

A simple JStreams program

- normal `find` without arguments just list files and directories
- `deepfind` also lists all files contained in other files

```python
sub listDir(path):
    dir = open(path)
    for entrypath in dir:
        print entrypath
        if isDir(entrypath):
            listDir(entrypath)
        else:
            ssp = openStreamProvider(entrypath)
            if ssp.isOk():
                listStream(ssp)
```

Jos van den Oever
sub listStream(ssp, path):
    stream = ssp.nextEntry()
    while stream:
        entrypath = path + '//' +
        ssp.getEntryInfo().filename
        print entrypath
        ssp = openStreamProvider(entrypath)
    if sspisOk():
        listStream(ssp)

A simple JStreams program

- normal find without arguments just list files and directories
- deepfind also lists all files contained in other files

Improving 'grep -r' is left as an exercise for the audience
Browsing nested files
Browsing nested files
DeepLocate

cd /usr/bin
rm find
ln -s deepfind find

find + jstreams = deepfind
deepfind + locate = deeplocate

'grep -r' + jstreams = deepgrep
deepgrep + X = deepX
What is X?

Kat

Beagle
Four problems when finding X

1. Beagle is designed to index files, not streams
2. Kat is more or less dead
3. JStreams indexes more than one file at once
4. Kat and Beagle are not just indexing text
Solution:
create X and call it Strigi
4) Extracting more than text

One can extract more than just text from files
- subject, author, modification time, sha1, title, links, etc
- multiple analyzers can add to the object at once

Strigi extracts from each file an “Indexable object”
- file path (URL)
- mtime
- size
- mimetype
- key, value metadata
Stream Analyzers

EndStreamAnalyzer
- there can be only one per stream
- reads the stream by pulling (calling read()) on the stream

ThroughStreamAnalyzer
- there can be many
- reads the stream by passing along read() calls and looking at the passing bytes
- loadable modules for both types
- work on Windows and Linux
Analyzer examples

**TextEndAnalyzer**
- splits the text up and passes it to the Indexable

**MimeTypeThroughAnalyzer**
- uses libmagic to determine the mimetype and encoding

**KFileThroughAnalyzer**
- uses KFileMetaInfo to get metadata

**SaxEndAnalyzer**
- extracts the text content from xml files

**MailEndAnalyzer**
- analyzes the mail headers and indexes the attachments
Strigidaemon

Files → EventListener → IndexScheduler → IndexManager → IndexReader → IndexWriter → DBusInterface SocketInterface

CLucene, HyperEstraier, SQLite

Jos van den Oever
Export data as XML

<?xml version='1.0' encoding='UTF-8'?>
<metadata>
  <file uri='testdata/.svn/text-base/all.zip.svn-base/a.zip' mtime='1150927654'>
    <value name='mimetype'>application/x-zip</value>
    <value name='sha1'>25da41e3282f81b8289ed63da8a534c15d9fee9b</value>
    <value name='size'>275</value>
  </file>
    <value name='mimetype'>image/jpeg</value>
    <value name='sha1'>a25141506f894bd6e963283d758d7ff21aee516</value>
    <value name='size'>23771</value>
  </file>
</metadata>
Dbus support using the C API

- ideal for a daemon
- only 1 dependency
- high performance
- not easy to get it running

Simple C++ Dbus code generator

- generate Service class from a C++ header file
- support for introspection
Akademy 2006 will be in Dublin

... From KDE Apps: ... KBcache, a KDE application ... the KDE address ... title="Â½KBcache KDE-Apps.org" href="http://www.kde-apps.org/content/show.php?content=9924">From KDE Apps</a>:</p> ... /data2/kdedev/kdepmilibs/syndication/tests/atom/qtblog.xml - 4k - text/xml

qtblog.xml.expected
Plasmoid and OSX
 GNOME DeskBar

Added a Strigi adaptor
- written in Python
- communicates over DBus
Can we rely on Strigi staying there?

- code core is small
- most of the code is the implementation of various interfaces
- unit tests for implementations of jstreams are easy
- xmlstreamwriter can be used by other indexers
Contributors

Ben van Klinken (CLucene developer)
- ported plugin architecture to Windows
- JStreams testing and discussions
- advises Strigi as the indexer of choice for CLucene

Flavio Castelli
- Inotify support
- Selective filtering by indexing on filename
- Logging framework

Egon Willighagen
- KFileThroughAnalyzer

Fathi Boudra
- .deb packaging
KDE4 Integration ideas

- KMail filter bar filters on entire mails and colours the mail by search score
- Calendar entries can be found in the calendar file and these entries can be opened directly in Kontact
- Search results are displayed on a timeline or on a sizeline
- File dialog filters the directories based on whether the desired mimetype is somewhere in the hierarchy
- Entry of keywords in the file dialog does a search instead of an error message
- The konqueror context menu gets a menu item for finding duplicate files
- Email on an imap server are indexed and can be opened
How to do this?

Integration

- Implement a JStream that can split up your multipart files
- Write a stream analyzer that extracts the data you want to have indexed
- Teach you app how to handle the URL that Strigi gives you (usually jstream:/ will take care of this)

Resources

- #strigi
- http://strigi.sf.net
- trunk/playground/base/strigi
- trunk/playground/base/strigiapplet
A search interface should

- show the user files or parts of files that match the query,
- match the current context
- and open entries from the search result in the right program

Strigi  Nepomuk
Discussion and Brainstorm

- integration into KDE4 (svn, dependencies, releases)
- enable multiple repositories
- enable indexing of remote files like http and imap
- think about metadata standards
- come up with more search ideas
- generalize the jstream:/ kioslave
- write (yet another) backend
- porting analyzers from other indexers (mp3, jpg, ogg, kfilemetadata)