

OpenELIS Monthly Community Call

December 11, 2013

Participants

- Gary Jones
- Reshma Kakkar
- Steve Davis
- Shondra Johnson
- Huy Doan
- Paul Schwartz
- Travis – Iowa
- Mark Mantle
- James Brockman
- Dari Shirazi

AGENDA

1. Roll call
2. Discussion on adding more information to the OpenELIS website – All
 - Gary said would like to do some more work on the website in 2014. Wanted to know what people would find useful.
 - Minutes from meeting are going to be added
 - OpenELIS global has not been keeping up to date with changes in infrastructure. It is on Univ of Washington to update
 - Have opened a github site. So where people would look for code vs. where they would look just for information are different
 - Shondra suggested adding something on electronic messaging
 - Paul said this makes a features list important
 - System requirements
 - Shondra volunteered to help with the website
3. Discussion on Plug-in architecture – Paul Schwartz (if available)
 - Send flat files through OpenELIS
 - Analyzers change frequently and this affects the interface.
 - Does not take much coding time to re-interface but releases are only once a quarter. So the time duration when analyzer changes and the release of OpenELIS may not match up.
 - Needed plug in architecture so would not have to wait on new release
 - Lab analyzers are usually connected to a dedicated PC that is only for the user of those analyzers. The output files from the analyzers sit on those PC's and ITECH needs access to those files.

- Monitor the PC's for a new file. When see a new file capture it. So they do not talk directly to the analyzer.
- The name of the analyzer has to appear in the menu system. Currently there is a file that links to the database and reads the database to populate the menus.
- In Java localization is handled by resource files
- Analyzer needs to be added to the database. Need to reference the analyzer in the database to maintain integrity
- Need to add permissions so based on role, users have permission to see results from analyzer
- Have code that moves files to database
- Goal in developing this architecture was to have minimum artifacts. For e.g. if analyzer is 10 years and no longer exists don't want garbage from that in the analyzer
- Funding model is moving development to in-country. Low complexity means low barrier to do that.
- Have versioning support. Want to support new and old version
- Don't want to do massive re-write of existing analyzers.
- Dari – defined simple format that OE can take. Then can write whatever we want from the instrument in that input format. Dari asked if the Jar file is being used as glue that can do anything in OE.
- Paul – Jar files externalizes what happens in OE
- Dari – do you have an API? Is the code gluing the file to OE do whatever you want or API that has limitation on what it can do
- Paul – yes API. Have clean interface between plug in and what it has access to.
- Dari – have one input format.
- Paul – more complicated. Most analyzers put out a comma separated value file. What is in the file differs widely. Therefore need separate plus in for each analyzer. Format of file is beyond our control
- Dari – have done the same thing for our older systems.
- From user point of view no diff between plug in architecture and what we are currently using.
- Paul – look for flat file that will only be sent by that analyzer. Each file to be named. If not have to hope that it is not in another analyzer file. Once move the plug in arch, this is not easy as someone else may have the same file name.
- Have installers. These repopulate the WAR file. Plug ins will not be in WAR file.
- Need to be restarted before can use. In most countries have to power down at night anyway.
- Chose these analyzers because they are the easiest. These plug-in's can also be used for reports. That was why they selected this option. This makes electronic reports easier.
- 2 things
 - The code expands
 - Have same model for menu
- Showed how analyzer code integrated into OE. Quite simple. Several different services used.

- Target analyzer – sends strings from file.
 - Analyzer inserted takes flat file and inserts into database
 - Integration of menus.
 - Added capacity to stay in plug in directory - explored this.
 - Dari – asked if have to reload app or restart and asked about reload capability
 - Paul - don't want to redeploy. Dari said there is a reload which is diff from redeploy. App should pick up new JAR file. Suggested Paul tries it.
 - Dari – could have the descriptive xml but also have frame work where it can extend the xml. Take interpretive code and extend it. Sometimes we need to extend analyzer and this can be specified in xml files.
 - Dari – would be interested in how to do reports. Jasper has a Jasper server that you can deploy reports through server and can call it up to do things for you. Would be interesting to find out if Jasper server does what you want it to do. Paul will look into it.
 - Dari – looked like writing the interface to a web service.
 - Paul – flow is from analyzer to dedicated PC to proximal PC that just moved files. This is done through http service.
4. OpenELIS Updates from U.S. and Global users
- Dari – have started on the clinical portion of OE and working with Missouri on that.
5. Other Discussion or questions
- Gary will send out calendar for 2014.