2021-09-07 - VIVO Development IG

Date

07 Sep 2021

Call-in Information

Time: 09:00 am, Eastern Time (New York, GMT-04:00)

To join the online meeting:

- Go to: https://lyrasis.zoom.us/my/vivo1?pwd=a2Q3RUVKVkN2dkNHV3FUaFRtLzhGdz09
  - Passcode: 351860
- One tap mobile:
  - US: +16699006833,,9358074182# or +19292056099,,9358074182#
- Or Telephone:
  - US: +1 669 900 6833 or +1 929 205 6099 or 877 853 5257
  - Meeting ID: 935 807 4182
- International numbers available: https://zoom.us/u/aeANHanzED

Slack

- https://vivo-project.slack.com
  - Self-register at: http://bit.ly/vivo-slack

Attendees

⭐ Indicating note-taker

1. Brian Lowe
2. Dragan Ivanovic
3. Benjamin Kampe
4. Matthias Lühr
5. Georgy Litvinov ⭐

Agenda

1. A brief update about VIVO core development
2. Update/questions from implementers
3. Discussion about priorities for further development of VIVO - https://docs.google.com/spreadsheets/d/103P9P4v6yUBSb5BnVaK40NoGx1fIYyL8uaHKUubZWbE/edit?usp=sharing
   a. The topic for this week - How to make easier VIVO installation/customization/upgrade
      i. Improved documentation for existing versions?
      ii. Video demo of installation process?
      iii. Take #2 on binary installation without unpacking data directory?
      iv. Customizing theme and templates without building from source
      v. GUI for system configuration (option to set runtime.properties and applicationSetup.n3 settings via GUI instead of having to edit files on the server)
      vi. Export GUI changes to filesystem for checking into version control
      vii. Publish from VIVO studio to staging/production server?

Notes

1. Moving Jira Issues to Github

Dragan tried migration with his repository. Have a look at https://github.com/chenejac/VIVOTestMigrationJIRA/issues

1. Discussion about priorities for further development of VIVO - https://docs.google.com/spreadsheets/d/103P9P4v6yUBSb5BnVaK40NoGx1fIYyL8uaHKUubZWbE/edit?usp=sharing

Let's discuss the columns.

- Column F. How the task is interesting for new implementers, who use basic VIVO functionality
- Column G for existing VIVO adopters to make life easier.

- Column H for advanced VIVO adopters who try to get the most from VIVO, for technical people who are going deeper under the hood (including VIVO core committers).
- Column J - Does this help further decoupling or hinders it?
- Column K - Does this alleviate or add to technical debt?
- Columns E and I - should be used for prioritization and classifying task to releases - 1.13, 1.14, 2.0, 2.1, etc.
    - Column E - benefit for the community,
    - Column I - should that be part of reengineering or should that be changed immediately.

At the end of some meetings we should have our opinions aligned about marks in those columns.

Brian: If somebody sees something they do not agree on there should be opportunity to comment on issues and their ratings.

Let's try and discuss the rows (tasks) in the first group

- ### Take #2 on binary installation without unpacking the data directory?

Brian: In 1.11 there was a self unpacking war file and then that caused problems for UQAM so that feature was reverted for 1.12. Take #2 is to do the same without breaking users installations. That was too much to change before 1.12 release.

- ### Improved documentation for existing versions?

Dragan: For new implementers it is very important to have good and localized documentation. It would be nice to have improved documentation (and translated to some languages). There is a lot of useful (and accurate) information on wiki, but it could be even better. Whether we should update documentation after reengineering or before that. Probably yes. Part of this task is not strongly linked with architecture, and might be performed before reengineering (installation, customization). However, some parts of technical documentation might be significantly changed after reengineering.

Brian: Generally documentation is reasonably ok. What tends to be missing is little hint how to do that without a certain level of knowledge in it. Tutorials in other words.

- ### Video demo of installation process?

Dragan: Demo video of installation process would be nice to have. It might be useful and make installation process easier for the beginners.

- ### Easier theming / customization via GUI without building from source

Brian: Right now explicit installer copies files to directories and manages the process. Initially in 1.12 a new installer with war was introduced. That was the first problematic piece for some people and the second was to unpack the tar file in the home directory. Initially that seemed fine, but that wasn’t the thing everybody wanted. It is still kind of an unresolved issue.

Everybody would like to have custom logos and other stuff that could be affected by that process. How many people will have an easier process? Everybody has different opinions on this.

Brian: In some institutions custom way of customizations seems kind of weird. There should be a way to make small customizations without recompiling source code. There could be some way to make modifications via GUI to simplify little changes like that.

Dragan: That would be nice for some common cases.

- ### Easier installation, GUI supported install and GUI for system configuration (option to set runtime.properties and applicationSetup.n3 settings via GUI instead of having to edit files on the server)

Dragan: Options to set runtime properties at runtime with GUI. Nice feature, not sure how many people it will attract to adopt VIVO. Not a task of high priority as we have limited resources. Moreover, it might be easier for implementation after reengineering. If we decouple frontend that would be easier to reach that goals.

- ### Export GUI changes to filesystem for checking into version control

High priority.

There is a complicated system when certain files are first time files and behavior in 1.12 is kind of better.

Local customization could be used with git.

- ### Publish from VIVO studio to staging/production server?

Publishing from VIVO Studio to production server. Let's move that for next week so Michel Héon could take part in the discussion.

- ## Defining architecture improvements

Georgy: We can try to define VIVO 2 architecture to have information on what could be simpler achieved with those architecture improvements.

Georgy: We could improve architecture by moving already implemented features in new API service in accordance with new API Ontology to provide users and developers opportunity to describe requests in a declarative style through the GUI, not by creating new java code. That would simplify the development process, reduce VIVO code to support and increase performance.

Dragan: We need much more discussion about that.

Draft notes on Google Drive

Task List

- Dragan Ivanovic to summarize discussion and update the google sheet describing priorities - https://docs.google.com/spreadsheets/d/103P9P4v6yUBSb5BnVaK40NoGx1fIYyL8uaHKUubZWbE/edit?usp=sharing