IUT Sync-Meeting Notes

Date: 2016/3/23

Time: 4:00 p.m.

Attendants:

* Morgan Stanley: Baiyan Huang, Jing Li
* Shanghai Jiao Tong University: Ziyi Lin, Feiyue Yu, Weizhao Yuan

Discussions (in time order):

**About the version control system used in the object projects:**

We assume that the object projects that IUT is applied for use **git** as their version control system.

**About the structure of the repository of IUT:**

The repo of IUT should be divided into 3 parts, which are

1. Jiabin Ye’s work on Java IUT
2. Work on the simplification of Java IUT workflow
3. Work on C++ IUT

**About adding “version info” into the database:**

We all agree that the “sequence” of the version of object projects should be continuous.

Assume in one instance of IUT processing, we have “old version” and “new version” that is to be processed this time, and in the database we have data that is used to be queried. **If the “old version” is not identical to the version of the data in the database, then IUT fails this time.**

For instance, if we have an object project with 4 versions – v1, v2, v3, v4.

If we process IUT between v1 and v2, and then between v2 and v4, the two times of IUT work successfully.

But if we process IUT between v1 and v2, and then between v3 and v4, then in the second time IUT **fails** because **it only works when IUT have already run between v2 and v3 before it runs between v3 and v4**.

Therefore, a table should be added in the database that records the relationship between a project and the version of the its data in the database. As is roughly shown below:

|  |  |
| --- | --- |
| Project | version |
| … | … |
| … | … |

**About instrumentation:**

There are mainly two ways of instrumentation. One is insert log statement in the source code, as Jiabin did in Java IUT. The other possible way is to instrument during runtime, identify the entrance of a function and instrument. Considering the case of **inline** function, we probably will adopt the first way.

**About some specific cases:**

During the development of IUT, we need to identify some specific cases that may have an impact of the implementation of the system, such as **Macro definitions**, **global variables**, **virtual functions** etc.

We also need to analysis how great the impact can be and how to handle such cases.

**About running testcases:**

When we run test cases, we need to identify whether it is a testcase or a function call when a function is called.

**About Documents:**

The Documents are:

* User guide (with 2 demos)
* Design Doc (maintained iteratively)
  + Behavior/Interface Design (What users will use)
  + Tech Design
    - Concern (such as the specific cases mentioned above)
    - Risk
    - Design Decision (such as decisions made after discussions)
  + Milestone (Deliverable)

**About deliverables in the next meeting (16/4/13):**

1. A program that can do instrumentation.
2. Design Doc

**About updating the database:**

The database should be updated dynamically after every instance of IUT.

The “new version” may be instrumented too in order to update the database.

IUT should be processed online, which calls for high performance.