Undo, Redo and Collaboration in Web Applications

Miroslav Cupák, Hao Yan

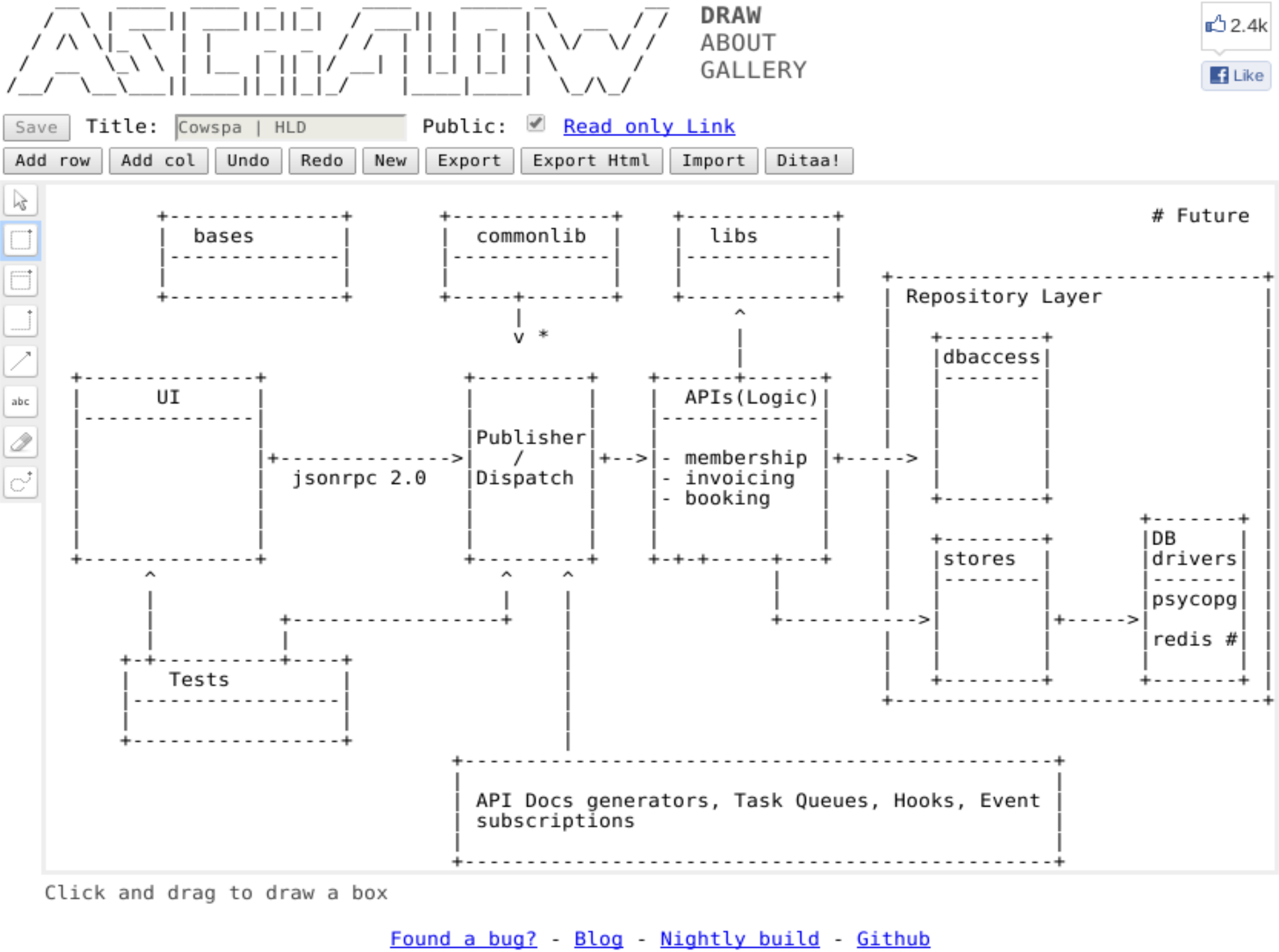
Dec 10, 2012

Introduction

- idea: investigate the use of application-level undo-redo to achieve collaboration
- motivation - collaboration is a crucial feature
    - Google Docs, Zoho Writer, SubEthaEdit...
    - essential for web-based word processing and spreadsheet
- goal: enable collaboration for a web-based diagram drawing software

Background

- Brown, Patterson: Undo for operators: building an undoable e-mail store.
- Lowell, Chandra, Chen: Exploring failure transparency and the limits of generic recovery.
- Edwards, Mynatt: Timewarp: techniques for autonomous collaboration.
- Michael, Towndrow, Wiz: Conditions for successful online document collaboration.



Title: Cowspa | HLD

Public: ☑

```
         +--------------+       +-------------+    +------------+                         # Future
         |  bases       |       |  commonlib  |    |  libs      |
         |--------------|       |-------------|    |------------|
         |              |       |             |    |            |   +------------------------------+
         +--------------+       +-----+-------+    +------------+   | Repository Layer             |
                                      |                 ^           |                              |
                                      v *               |           |   +--------+                 |
                                                        |           |   |dbaccess|                 |
  +--------------+                +---------+    +------+------+    |   |--------|                 |
  |      UI      |                |         |    |  APIs(Logic)|    |   |        |                 |
  |--------------|                |         |    |-------------|    |   |        |                 |
  |              |                |Publisher|    |             |    |   |        |                 |
  |              |+--------------->|   /     |+-->|- membership |+-----> |        |                 |
  |              |  jsonrpc 2.0   |Dispatch |    |- invoicing  |    |   |        |                 |
  |              |                |         |    |- booking    |    |   +--------+                 |
  |              |                |         |    |             |    |                    +-------+ |
  |              |                |         |    |             |    |   +--------+       |DB     | |
  +--------------+                +---------+    +-+-+-----+---+    |   |stores  |       |drivers| |
        ^                            ^    ^                |        |   |--------|       |-------| |
        |                            |    |                |        |   |        |       |psycopg| |
        |          +-----------------+    |                +----------->|        |+----->|       | |
        |          |                      |                         |   |        |       |redis #| |
      +-+----------+----+                 |                         |   |        |       |       | |
      |   Tests         |                 |                         |   +--------+       +-------+ |
      |-----------------|                 |                         +------------------------------+
      |                 |                 |
      +-----------------+                 |
                                 +------------------------------------------------+
                                 |                                                |
                                 | API Docs generators, Task Queues, Hooks, Event |
                                 | subscriptions                                  |
                                 |                                                |
                                 +------------------------------------------------+
```

Click and drag to draw a box

Found a bug? - Blog - Nightly build - Github

Asciiflow

- web application primarily used for diagrams
- based on Java + Google Web Toolkit
  - client-server model with RPC communication
- 20+ operations
- local physical undo and redo
  - maintains a set of pixel-based changes on the client
- no collaboration support
  - local save overwriting external changes

Goals

- allow multiple users to co-author a diagram smoothly
- changes merged and propagated to all the users
- each user is able to perform local undo and redo his own actions
- graceful resolution of conflicts without loss of data

Implementation

- modify RPC communication and persistence
- change model to enable state sharing
  - track the user issuing changes
  - extend information sent to the server
- add autosave capabilities
  - synchronize save after each operation
- adapt the concept to pixel-based operations
- enable manual undo-redo of local operations

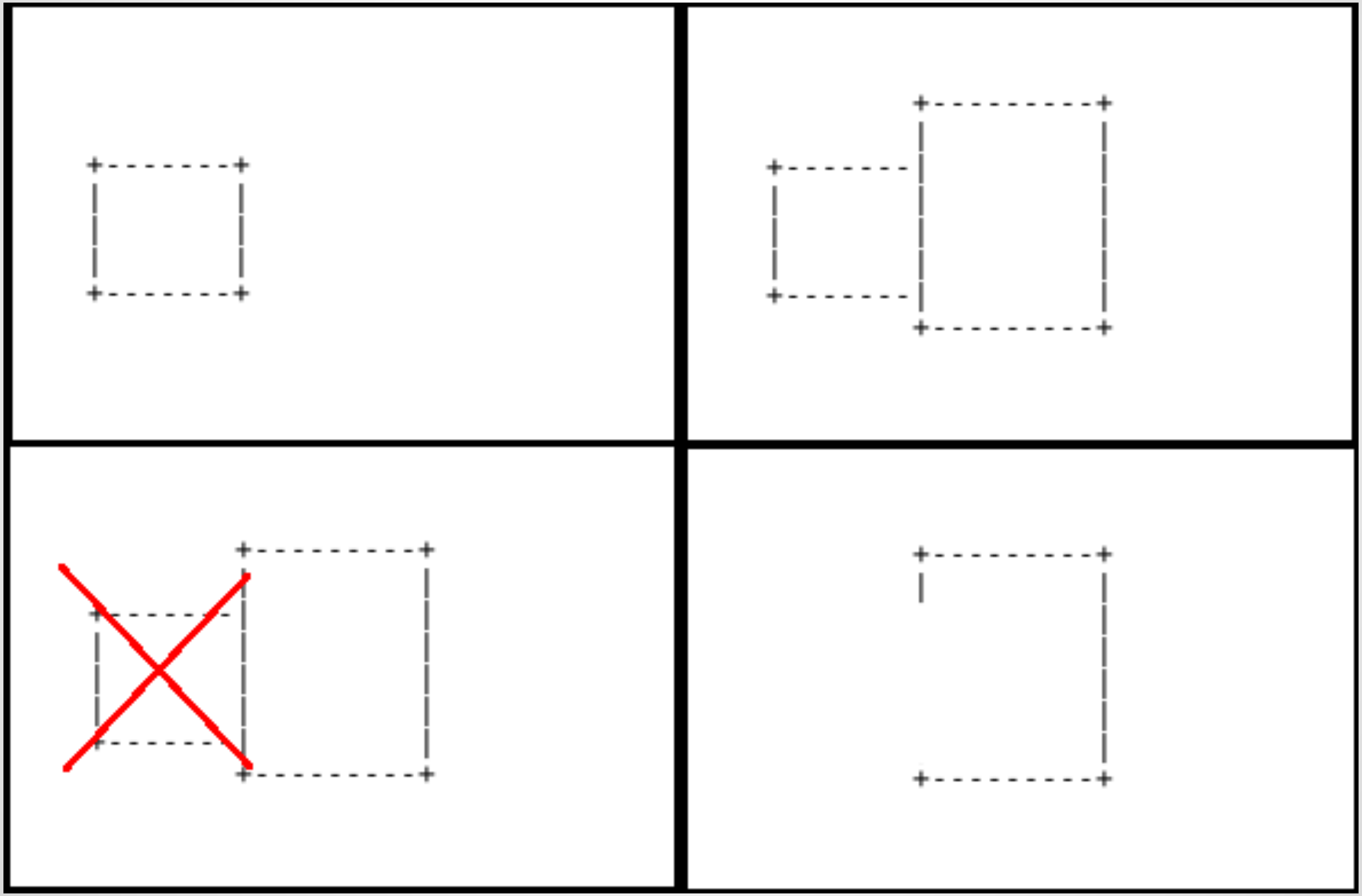
Implementation

- load and save coordination on the server
  - check the owner of the snapshot before saving
  - if owner = current user, proceed with simple save
  - otherwise merge changes before saving
- merging of changes
  - undo local operations
  - load the foreign snapshot
  - redo local operations on the new snapshot
  - save the new state

Conflicts

- save merging might lead to conflicts
  - eg. two players draw two overlapping shapes simultaneously
- most modifications of the same region OK
  - the slightly late change will be applied on top of the earlier change
- conflicts become a problem when undo operation is involved

Inconsistent result

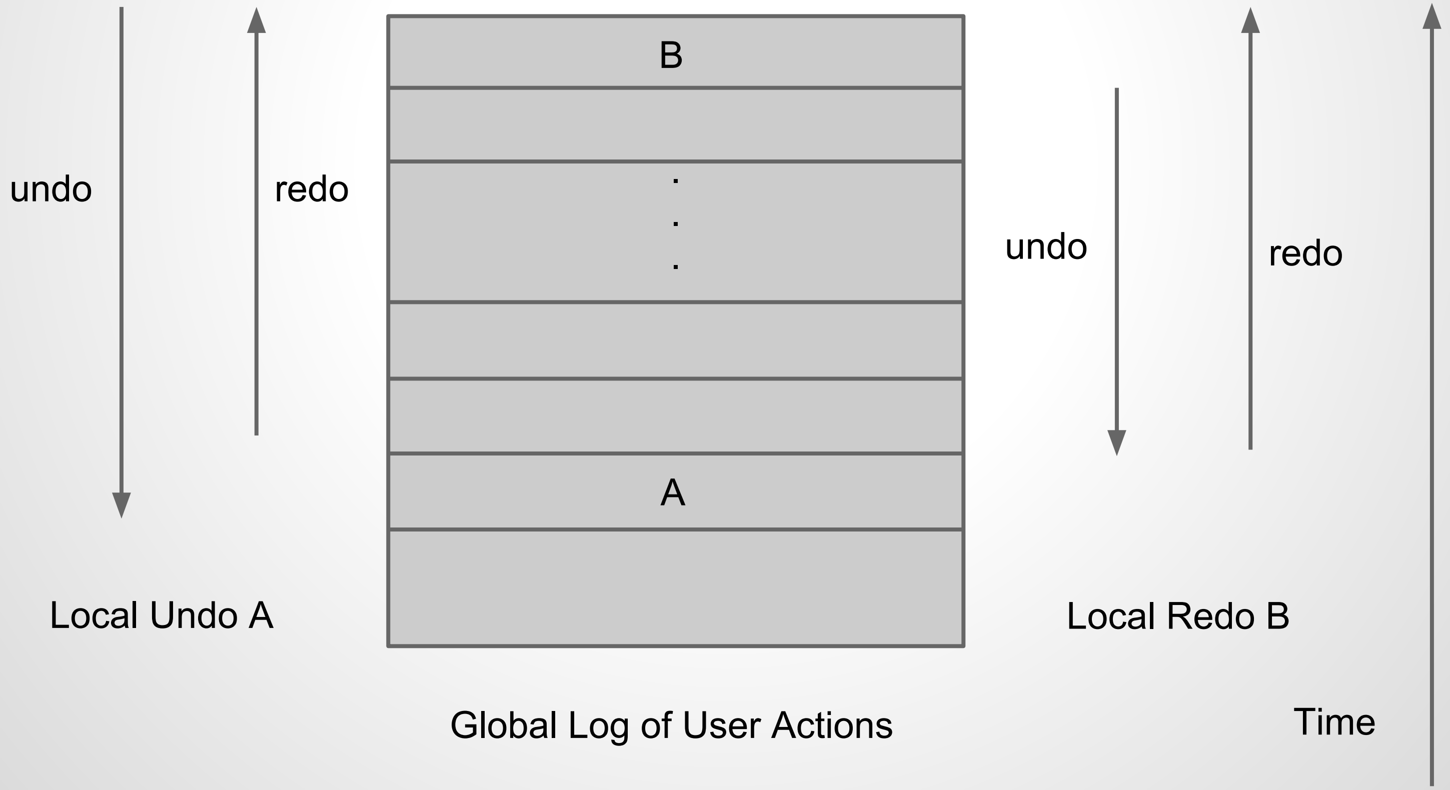


Resolving Conflicts

- all conflicts are associated with undo / redo actions
    - without knowledge of actions of other users, local physical undo / redo always cause consistency problems
- need to save a global log of user actions on server to fundamentally prevent conflicts from happening

Resolving Conflicts

- global action log stored as part of the state on server



Evaluation

- performance overheads of the collaboration features compared to the original application (merge cost, amount of data being sent)

Conclusions

- successful implementation of features that allow collaboration for Asciiflow
- demonstration of application-level undo-redo to achieve collaboration
- may improve performance by sending only the updates to server

Questions?