Discussion on the Project Design

18/June/2015

Project Ippeisoba

Object Oriented Web design using Ruby

Delete events?

- Case Study: “Children are noisy!” - the owner of the Problem wants to delete this problem entry, but some other links are already added to this problem/events. What to do with this?

- Does the system allow the user to remove the events/problems? —> Conclusion: No, In order to maintain the consistency of the discussion, the right to delete record could be done only by the administrator.

Antisocial/malicious writing?

- How to do with antisocial or malicious writings?
- Conclusion: Accept the “removal request,” from users.
- The administrator of the site or the thread manager of the problem, and such will remove the problem/events, together with the links of which the source or the destination of links are that should-be removed events/problems.

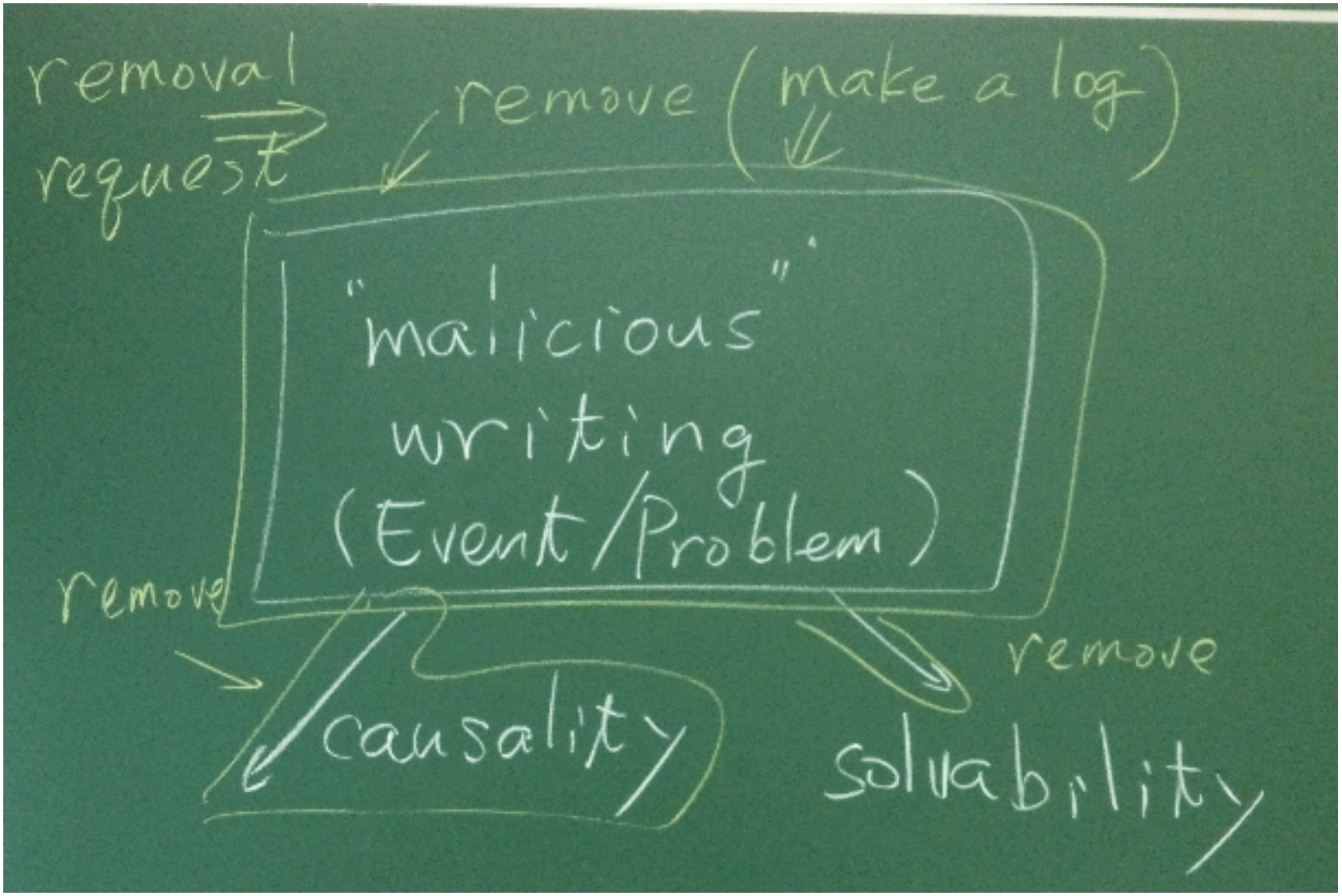
Invalid Flag?

- “kkt753”: Proposal of delete action. Just giving invalid flag and keep the record in the db of such deleted entries. And when the flag is added, the system will not show the contents of events/problems and such.
- Rejected: it may cost run-time speed. We copy the “removed” item into the system log, or “removed items” record, and completely remove from events/such tables.

Need Log?

- Absolutely, we need the system log. These “debate” basis system may cause human troubles, so, even if the writings had been removed within one minute, we should keep the record of who/when/what had been written in the system. (The lecturer’s personal opinion.)
- The lecturer will design and install this part.

malicious writing?



Pros and Cons table

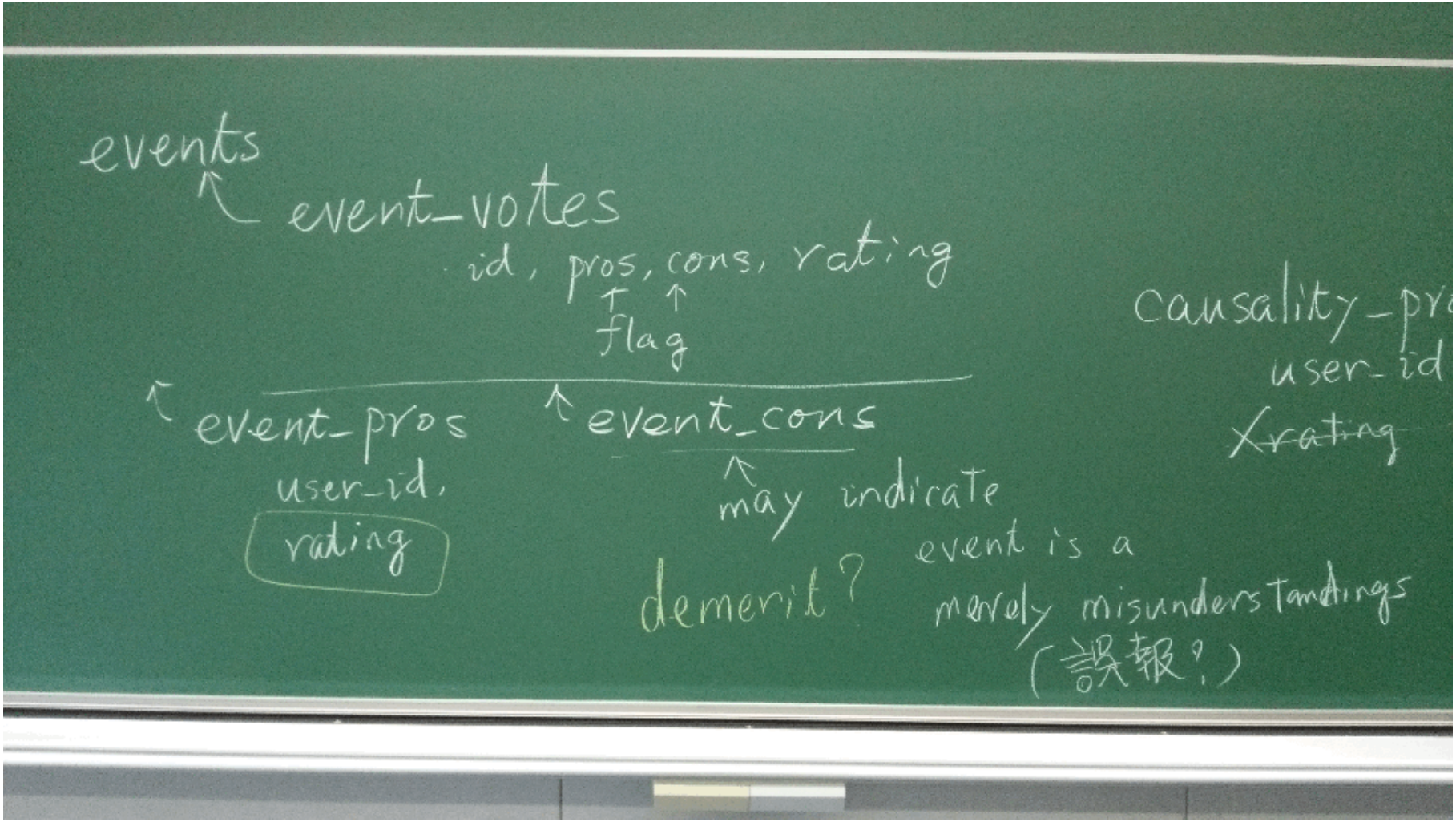
- The proposal was to give “event_votes” for events, “problem_vote” for problems, and such.
- “shun-Higuchi”: Discussion of three weeks ago, we decided that events may need the pros/cons with some ratings, but the cause/solve links may only require counts(numbers) of pros/cons voters. —> Confirmed.

Pros/Cons counter

- Regarding Pros/Cons counter, which do we choose; design A or design B
- A: event_votes table:
  - users_id(voter's), pros/cons flag, rating,
- B: event_pros, and event_cons tables
  - users_id(voter's), and rating, for each.

- We adopt the design B.

Pros and Cons table



Pros/Cons table

- We are going to have;
  - event_pros, event_cons, causality_pros, causality_cons, solvability_pros, solvability_cons
  - tables, for the pros/cons voting record.

Views Design

- Now, the most important screen is the “Events” list view. It will have “is_caused_by,” “causes,” “is_solved_by,” and “solves,” links.
- To keep the operability of the system, the events view screen will be most important.
- This may result that all the entries of “solvability,” and “causality” manipulation would be installed in this event view. Right?

Work Assignment

- At last, we may need the “input screen” of pros vote or cons vote closely related to the causality or solvability tables, in “citrusLov” part or “kkt753” part, not in “Shun-Higuchi”’s part.
- So, everyone may make branch of their program in the git environment, then, merge into master nextweek.

Plan for the next week

- demonstrate each persons' assigned part.
- push/pull/ merge the design to synchronize.

- Now let us think about the "Final form of this WEB system of this semester."
- Plan for the next fiscal year: "Practical Testing Environment."