EnviroInfo 2005 (Brno)
Informatics for Environmental Protection - Networking Environmental Information
Copyright © Masaryk University Brno, Brno 2005. ISBN: 80-210-3780-6

A Web-based Reviewing Process Guidance System for an Ecological Database of Plant Traits

Dirk Ahlers[1], Jens Finke[1], Michael Stadler[2] and Michael Sonnenschein[1]

Abstract

This paper covers the design, development and implementation of a review-based quality management system for an ecological database. We define a quality assurance process similar to that of a peer-review, whereby incoming data is separated into smaller items suitable for individual reviewing based on individual expertise. A simple yet efficient scheduling mechanism is devised that will distribute the trait datasets among the group of reviewers, considering certain particular requirements and conditions.

1. Introduction

The LEDA traitbase project (LEDA Traitbase 2005) is a Europe-wide effort to establish a database of plant characteristics, so called traits, to aid in the conservation and sustainable use of the biodiversity in changing European landscapes. The development is supported by the European Union 5th Framework Programme for Research (EVRI-CT-2002-40022).

This joint effort of 10 partners from european universities and research institutes does not only provide a database and necessary infrastructure, but also tools for data analysis (cf. Stadler et al., 2005). A wide range of sources are used to collect data of about 30 traits (i.e. plant attributes) for use in the database. During the project, new measurements and observations are mainly contributed by project members. However, as non-affiliated scientists - so-called contributors - are also invited to participate in the effort and add their own data, a quality assurance process similar to that of peer-reviewing has to be established to ensure the high quality of the data and adherence to project standards – see also (Stadler et al., 2004). As data is highly detailed and a huge amount of work goes into each measurement, a constantly high quality is mandated. Therefore, we propose a reviewing process guidance system to define a suitable workflow, split incoming data and distribute it to the group of reviewers while optimising the system for a high throughput.

2. Actors

The proposed system calls for several roles to be defined to enable the reviewing process. One necessary role is that of a *contributor*. Contributors are non-affiliated scientists who are interested in adding their own data into the traitbase. Secondly, a group of *reviewers* is needed to conduct the actual reviews of trait data submitted into the traitbase. The reviewers are appointed by the LEDA editorial board based on their in-depth knowledge about specific plant traits. The editorial board is constituted by a group of elected landscape ecologists and biologists from within the project. They are in charge of supervising the LEDA

[1] Carl von Ossietzky University Oldenburg, Department of Computing Science, Environmental Informatics Section, 26111 Oldenburg, Germany. dirk.ahlers|jens.finke|michael.sonnenschein@informatik.uni-oldenburg.de
[2] Oldenburg Research and Development Institute for Information Technology Tools and Systems (OFFIS), Escherweg 2, 26121 Oldenburg, Germany. michael.stadler@offis.de

data standard and assuring an overall high quality of the database contents. A reviewer may gain membership to one or more *trait groups*, i.e groups of reviewers assessing data for a given trait, e.g. leaf size. He can only review data about traits belonging to one of his assigned trait groups.

To ensure the independence of the reviews the specific reviewers are to remain anonymous to the contributor. An *editor* will serve as a single human connector between the reviewing guidance system and the contributor. The editor is in charge of informing contributors about the status of their contribution, which may either be accepted or rejected. In case of rejection the editor will also sum the reviewer's criticisms and make them available to the contributor in order to support revision of the contributed data.

3. Data structure

Each contribution may consist of information about several plant traits. To reflect this, a data structure within the database is defined accordingly.

A contribution is contained within a so-called *batch*, grouping a related set of observations. Within each batch, one or more plant traits can be addressed where each trait can in turn contain one or several records of measurements or observations. The records pertaining to a single trait within a given batch are logically contained within a so-called *subbatch*. The structure is detailed in figure 1. Additional flags are carried within each subbatch to denote its status regarding validity, stage of review etc. The review status of a batch is deduced from the statuses of its subbatches.

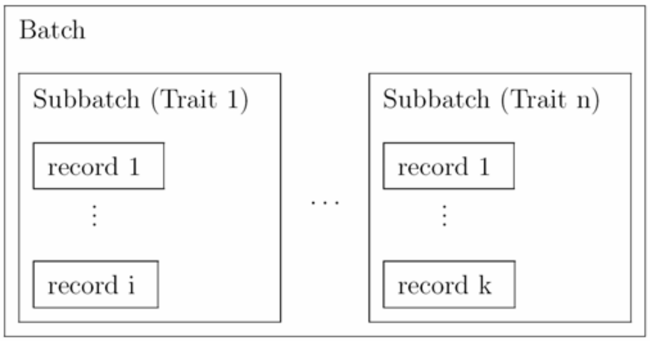

Fig. 1:Logical view of data structure

4. Workflow

The review process is logically situated after the contribution of the data and before it actually being published within the traitbase and used for analysis and reporting. Being subject to the reviewing-process, all data entered by a contributor will be split into its constituent trait records as subbatches, which in turn are individually reviewed by reviewers with matching expertise for a given trait. The review finishes with either acceptance of the data or rejection. On completion of the review for all subbatches of a batch, the batch is presented to the editor who will in turn inform the contributor of the result. In case of a rejection the data can be revised by the contributor and resubmitted as a follow-up. Otherwise the data is published into the LEDA traitbase. The workflow is detailed in figure 2.

One peculiarity of the system is that there is not necessarily a strictly linear sequence of events as in case of a rejection the data is sent back to the contributor who can then revise it. In such a case, the reviewer to receive such a *follow-up* should be the one who rejected it the first time. This ensures that improvements can be assessed best. Prolonged holding time of a subbatch by a reviewer is counteracted by establishing timeouts that will trigger appropriate measures.

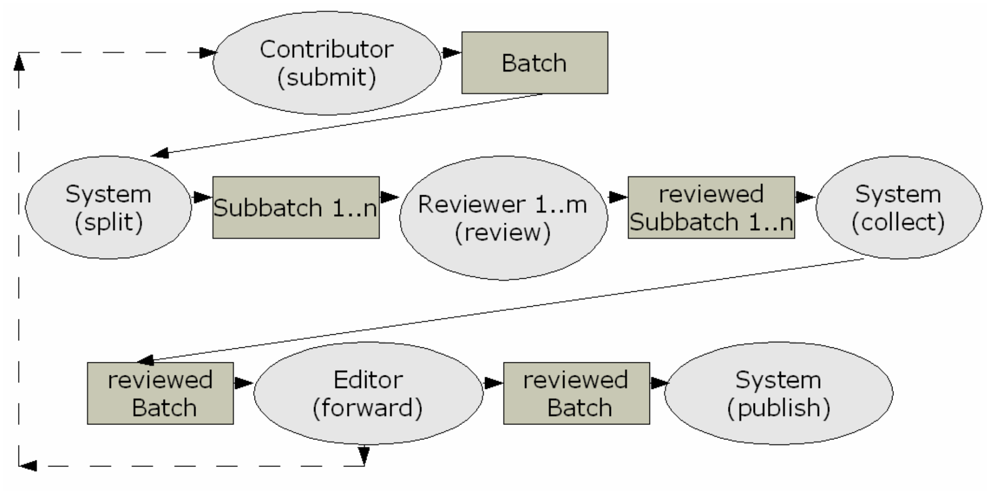

Fig. 2: Workflow

5. Queueing alternatives

One problem that should be examined in greater detail is the designation of a reviewer for a given subbatch. Two complementary feasible alternatives for the system design regarding this issue will be discussed. Findings from Queueing Theory (Bose 2001) show that average handling times in a single-queueing-system are equal or better than in multi-queueing-systems. The proof for this would be beyond the scope of this paper but can be checked in (Waldmann/Stocker 2004), for instance. Here it shall suffice to know that the performance of a queueing system can never be worse in single- than in multi-queueing but can – under certain circumstances possible in this system – be significantly better.

Taken to the system to be developed, one could imagine two assigning strategies: global pooling and individual queues. The pooling strategy would allow a reviewer to freely pick his workload from a set of subbatches that match his expertise and are also visible to every other reviewer with the same expertise. Within such a system, a user would be able to see all subbatches whose traits he is qualified to review. This would be equivalent to a single queue all reviewers can draw from. The alternative of individual queues would be a system that assigns incoming subbatches to individual reviewers, thus creating queues for every single reviewer with their respectively assigned subbatches.

Evaluating these two approaches leads to a prevalence of the assignment approach mainly because of a certain requirement which calls for a way to place each dataset under the direct responsibility of a reviewer to establish a strong responsibility. With the assignment approach, this can be achieved by fixedly tying a reviewer to each subbatch. The more open approach of pooling cannot provide this as easily and would indeed dilute the procedure.

Ease of implementation is not an issue, because basically both approaches would share a lot of business logic. Handling of follow-ups as well as timeouts will be present in both systems. Though with the pooling approach they would not be used for the default data flow, they would nonetheless have to be implemented for person-bound follow-ups. Follow-ups in general are handled in both approaches by first assigning them to the previous reviewer. Then, should timeouts occur, they are treated as normal subbatches, i.e. putting them into the pool or assigning them to a new reviewer, respectively. As stated previously, in terms of handling time a pooling approach would usually serve better than an assignment approach. These shortcomings can be reduced as described further below.

Having evaluated the alternatives, we choose the individual queues approach which ties each subbatch to a reviewer as it inherently satisfies the requirement of direct responsibility and the other factors do not place too high a cost on this choice.

6. Selection of a reviewer

The settings for the scheduling mechanism are as follows: There is no one-to-one mapping of traits to reviewers. Reviewers can be in one or more trait groups and different trait groups can be of varying sizes. Scheduling should in no way judge a reviewer or hold him to certain amounts of work, but simply determine his possible workload and try to assign work accordingly.

As reviewers' capacities may vary widely and unpredictably over time and between different reviewers, a scheduling mechanism assigning datasets to reviewers' queues has to be adaptable to this variety. With no reliable predetermined information about reviewer's behaviour regarding system usage, the scheduling mechanism being devised can solely depend on information gathered by the system during the reviewing process. The choice of a suitable reviewer for a task has to be based on his anticipated ability to handle the assignment quickly and efficiently.

Various properties of the system can be taken into account to choose a reviewer. These include length of reviewer's queue, previous reviewer if any, age of subbatches in the queues, reviewer activity etc. Several of these properties are highly ambivalent regarding their assessment so a decision based on these might be contrary to the desired outcome. One could implement countermeasures for all these eventualities, but this would ultimately lead to an ever-changing and highly complex assignment of subbatches that also cannot be guaranteed to always be suitable and would also be difficult to follow by the users. Therefore, it is better to maintain a slightly suboptimal assignment which in compensation can be clearly understood and has a low complexity.

A scheduling approach only utilizing the simple criterion of a reviewer's queue length can indeed be self-balancing. It would choose the reviewer with the shortest queue for the subsequent assignment. After an initial equal distribution among reviewers, those with a higher work capacity will then have a shorter queue, which will in turn make them more likely to receive the next subbatch to be assigned. Reviewers with less available capacity will, however, shorten their queues slower so they will only receive new subbatches when their queue length is down to the length of other reviewers.

Due to a high number of uncertainties in the prediction of this future possible workload, a corrective is still needed to revise an initial reviewing job distribution in case it becomes unbalanced. A re-scheduling of a subbatch can therefore be initiated by the system after a given time-out kept for each subbatch, hence preventing potentially stalled tasks and overstrained reviewers. One additional parameter is needed though, to allow for a proper handling of this event. The previously assigned reviewer for a subbatch has to be taken into account so that after a re-scheduling he will not receive the same subbatch again. Contrary, a follow-up shall be given to this very same reviewer.

7. Error Situations

As detailed before, the system will balance and – if necessary – reassign the accumulated workload during its regular operation. Additionally, safeguards are put into place to identify potential error situations, react accordingly and return the system to a stable state using error handling and error recovery measures. Fallbacks are established to reduce error situations to few well-defined error states that can be rectified easily.

These system-side escalations assure a timely handling of error conditions. Beyond the check for timeouts on assigned subbatches, these include various integrity checks on the database.

Unassignable subbatches are specially marked and regularly re-checked for assignment. This ensures that subbatches without current reviewer can be re-integrated into the workflow. Thus, the whole reviewing process works with nearly no administration needs and ensures that no contribution is lost. Both aspects were very important for the project, since there is low manpower for administration tasks and the data is very valuable.

8. Technology

The web application, as the rest of the LEDA traitbase system, is implemented using J2EE technology i.e. Java Server Pages, Java-Beans and the Struts framework within the Tomcat application server. The reviewing component was implemented in a modular way featuring a plug-in solution for the actual scheduling algorithm utilising Java's dynamic class loader. Thus, should the need for a different scheduling algorithm ever arise it could easily be incorporated allowing for easy adaptation to changing conditions. In addition to the error handling mentioned above, on the implementation level extensive use is made of database transactions and Java exceptions to detect or deflect possible errors.

Integrating the reviewing component into the LEDA traitbase system, small changes are made to the contributor-component to allow for revision of rejected data as part of a follow-up. Interfaces to this component are provided for invokation of the reviewing process after submission of data by a contributor.

9. Final remarks

As is often the case, special requirements prevent an easy solution. However, using a robust scheduling approach relying on a rather small set of criteria, a self-balancing and self-recovering solution is achieved. The scheduling guarantees to assign a reviewer to a subbatch unless the respective traitgroup is empty, in which case the editor will be notified. Generally, the whole process will continue if there is at least one active reviewer per traitgroup present. Otherwise, a non-recoverable situation is declared which calls for manual assistance, in this case adding new reviewers. Still, the system will continue to work on remaining subbatches. Thus, most problems can be dealt with quite satisfactorily. Remaining issues are identified by the system and resolved automatically, allowing the reviewing system to work just as intended.

Bibliography

LEDA Traitbase (2005): The LEDA Project, http://www.leda-traitbase.org/

Stadler, M., Kunzmann, D., Schlegelmilch, J., Sonnenschein, M. (2004): Data Quality, Abstraction and Aggregation in the Leda Traitbase. In "Sharing" - Proceedings of the 18th International Conference Informatics for Environmental Protection. Part 1, CERN, Geneve, Switzerland, éditions du Tricorne, pp. 515-525

Stadler M., Bekker R.M., Finke J., Kunzmann D. and Sonnenschein M. (2005): Using data mining techniques for exploring the key features of plant dynamics upon a newly built plant trait database. In Proc. EnviroInfo 2005, Brno

Waldmann, Karl-Heinz, Stocker, Ulrike M. (2004): Stochastische Modelle: Eine anwendungsorientierte Einführung, EMILeAstat, Springer-Verlag, Heidelberg

Bose, Sanjay Kumar (2001): An Introduction to Queueing Systems, Kluwer/Plenum Publishers