STUDIES IN LOGIC, GRAMMAR AND RHETORIC 18 (31) 2009

Improving Representation of Knowledge within the Mizar Library*

Adam Grabowski and Christoph Schwarzweller

[1] Institute of Mathematics, University of Białystok
Akademicka 2, 15-267 Białystok, Poland
adam@math.uwb.edu.pl
[2] Department of Computer Science, University of Gdańsk
Wita Stwosza 57, 80-952 Gdańsk, Poland
schwarzw@inf.ug.edu.pl

Abstract. Efficient handling of extensive repositories is one of the major objectives of mathematical knowledge management. It is naturally connected, in order to attract more potential users (both researchers and students), with the need to build large libraries of formalized mathematics, and these two activities should not interfere. This may be achieved by constant enhancement of the quality of digital repositories, in which the proofs can additionally be verified with the help of proof assistants. Based on our experience with the MML we discuss some of the issues concerned with this process, describe mechanisms of revisions which seem to be indispensable to meet the expectations of contemporary mathematicians. We argue that even careful reviewing of contributions cannot cope with the task of keeping a mathematical repository efficient and clearly arranged in the long term.

1 Introduction

The contemporary way of doing mathematics can be substantially enriched by using computers, in comparison to classical pen-and-paper model – and this is what Mathematical Knowledge Management network aims at.

For a researcher, one of the activities here is obviously just doing mathematics for himself – and math assistants known in this areas offer much freedom and flexibility. Certain doubts can appear if it comes to knowledge exchange – incompatible definitions, different notations or more than one proof of the same theorem are something natural in mathematical practice. For repositories of computer-checked knowledge this means that either these various objects should be linked somehow or they should be handled separately, however automatic discovery of such places seems to be hard.

One possibility, of course, is reviewing the submissions. Reviewing improves the quality of knowledge and proofs added to the repository, but we shall illustrate that

* This is an extended and updated version of the paper from MKM 2007.

ISBN 978-83-7431-229-5 ISSN 0860–150X

in the long run it cannot ensure that a mathematical repository meets the expectations. We therefore claim that revisions (as the reorganizations of the Mizar library are usually called) are an essential part of maintaining mathematical repositories: in order to keep it clean and attractive for users, from time to time a "core team" has to check and improve the organization, quality, and proofs of a mathematical repository.

In the following section we describe and discuss the goals and benefits of revisions compared to a straightforward reviewing process. Then, after a brief introduction to the Mizar system [11], we consider the reviewing process of MML submissions in Section 3. We describe reviewing criteria and show which insufficiencies can be handled by reviewing. In contrast, Section 4, is devoted to revisions of MML illustrating on the one hand what kind of improvements reviewing cannot perform, on the other hand the role of revisions in maintaining MML. This is the process done mainly by a human hand as of now, but we also describe existing Mizar utilities devised to facilitate this process; in the next section we discuss some issues concerned with this activity and describe some traps the developers may meet when enhancing the library. Then we conclude drawing some remarks for future.

2 The Motivation

The goal of a revision is to improve the mathematical repository. In contrast to reviewing submissions, however, here the attention is turned to the repository as a whole, not to a single, new part of it. Consequently, the motivation for revisions can be for example:

- keeping the repository as small as possible,
- preserving a clear organization of the repository in order to attract authors,
- establishing "elegant" mathematics, that is e.g. using short definitions (without unnecessary properties) or better proofs.

Note that all these points characterize a qualitative repository and can hardly be achieved by reviewing single submissions. Of course there are different possibilities to achieve the points mentioned. Improving the prover e.g. can shorten proofs and hence – simplify the repository. Reorganizing a mathematical repository probably demands manipulating the whole file structure, not only the files themselves. Therefore we decided to classify revisions based on their occasion, that is on which kind of insufficiency we want to address. Based on our experiences with the Mizar Mathematical Library we distinguish four major occasions for revisions:

1. improving authors' contributions;
2. improving the underlying prover or proof checker;
3. reorganizing the repository;
4. changing the representation of knowledge.

Improving an authors' contributions is the typical task of reviewing and is of course to be recommended for mathematical repositories too: nomenclature can

be polished up to fit to the yet existing one, definitions can be improved, that is e.g. generalized if appropriate. Proofs are also a matter of interest here, especially keeping them as short as possible, yet still understandable is of major concern. In a large, open repository however, authors sometimes may prove and submit theorems or lemmas not being aware that those are already part of the repository. Similarly, special versions of already included theorems can happen to be "resubmitted". It is doubtful that this kind of flaws will be detected by ordinary reviewing.

Strengthening the underlying prover or proof checker has also an impact on the repository. Proofs can be shortened or rewritten in a more clear fashion, both being fundamental properties of attractive mathematical repositories. Even more, theorems in such a collection may become superfluous, because the improved prover accepts and applies them automatically. A typical example here is the additional inclusion of decision procedures.

Reorganizing the repository deals with the fact that a repository is built up by a large number of contributors. For their new developments, authors (should) use already existing theories as a basis. To establish their main results, however, they often have to prove additional theorems or lemmas just because the theory used does not provide them yet. So, these additional facts have to be put in the right place of the repository. Otherwise, it will be hard for other authors to detect them or at least searching the repository becomes less comfortable. The building of monographs goes in the same direction: a frequently used theory should be handled with extra care. Not only should all related theorems be collected in a distinguished place, but also still lacking theorems be complemented, in order to ease the work for further authors. These tasks can only be accomplished when considering the repository as a whole, that is by revisions.

The last point concerns the development of a repository in the long term. What if after a while it turns out that another definition or representation of mathematical objects would serve our purposes better than the one chosen? Should it be changed? Note that a lot of authors already could have used these objects in their proofs, that is changing the definition or representation would imply changing all these proofs – and of course one cannot force authors to redo all their proofs. On the other hand, including both definitions or representations leads to an unbalance: the theory of the new preferred version is much less developed than the one of the old version, so authors hardly will base their developments on the new one. Again, the solution is a revision: in the best case definitions and representations are changed by the "core team", so that ordinary users can further use all theorems without even noticing they have been changed.

In the following sections we will illustrate these considerations by examples taken from the Mizar Mathematical Library and in particular show how revisions maintain mathematical repositories.

3 Reviewing MML Submissions

Reviews of submissions to the MML – as reviews of ordinary submissions for conferences or journals – have the overall goal to check whether a submission should be accepted (for inclusion into the MML) and simultaneously improve the quality of

that submission. For mathematical repositories, however, the criteria for acceptance and improvements are somewhat different.

3.1 The Criteria

Certainly the contents of a submission for a repository should likewise be original and interesting. Original here, of course, means that definitions and theorems presented are not part of the repository, yet. This is easy to check for the main theorem of a submission. For technical lemmas used to establish this main result, however, this task is much more difficult. So, for example, a reviewer will probably neither know nor be willing to check whether a theorem like

```
theorem
  for F,G being FinSequence, k being Nat st
    G = F|(Seg k) & len F = k + 1 holds F = G ^ <*F/.(k+1)*>;
```

is already included in a repository. Even if the textual search via grepping is no longer the only method to find such repetitions since the MML Query by Grzegorz Bancerek [3] is available, even after the volunteer will learn how to use this system, still there is no single automated bunch of tools which removes all repeated theorems effectively. Furthermore, the motivation to check these things in detail will be even decreased, because such a point will not decide between acceptance and rejection.

The question whether a submission is interesting should be handled more liberally. Of course, the usual issues, that is the quality of the main results, apply here too. There is, however, another kind of submissions to repositories: the one that deals with the further development of (basic) theories. This concerns collections of basically simple theorems providing necessary foundations, so that more ambitious developments can be accomplished easier. Usually, these are theorems that easily follow from the definitions, however are used so frequently that repeating the proof over and over again is hardly acceptable. Examples here are the theories of complex numbers or polynomials, where among other things we can find the following theorems.

```
theorem
  for a,b,c,d being complex number st
    a + b = c - d holds a + d = c - b;

theorem
  for n being Ordinal,
      L being add-associative right_complementable add-left-cancelable
          right_zeroed left-distributive (non empty doubleLoopStr),
      p being Series of n,L holds
   0_(n,L) *' p = 0_(n,L);
```

Though hardly interesting from a mathematical point of view, such theorems are important for the development of a repository and should therefore be considered as interesting, too.

Improvements of a submission are a more difficult issue. Firstly, we can consider definitions and notations contained in the submission. Can they be arranged more sparsely, i.e. can the results be established based on fewer axioms? Is it possible or reasonable (in the actual repository) to generalize the definitions? This also applies to theorems. Note that the above theorem, though applicable to polynomials, is in fact stated for power series. Again to address these issues, a good knowledge of the repository by the reviewers is necessary.

Secondly, when it comes to proofs, there are hardly any guidelines, because proving in particular is a matter of style. We can hardly force an author to change his (finished) proof into another one using completely different proof techniques. What we can do, is to suggest improvements for the presented proof. We can, for example, propose a more accurate use of the proof language to get more elegant or better readable proofs. Or we can give pointers to other theorems in the repository that allow to shorten the proof.

3.2 The Evaluation

Based on these considerations the reviewing process for Mizar articles, that is for submissions to the Mizar Mathematical Library, has been introduced. Using basically the commonly used scheme accept/revise/reject (and apart from its descriptive grade) the rating of a submission can be[3]

A. accept
   requires editorial changes only, which can be done by the editors
B. accept
   requires changes by the author to be approved by the editors
C. revise
   substantial author's revisions necessary, resubmission for another review
D. decision delayed
   revision of MML necessary
E. reject
   no hope of getting anything valuable

The most important issue here, of course, is the question whether an article should be included in MML. Note that there are two grades (A and B) for acceptance. The reason is that accepted articles should be included in the MML as soon as possible to avoid duplication of results during the reviewing phase.[4] So, while submissions rated B or C need feedback from the authors, submissions rated A can be added to MML without further delays.

The most interesting point is D. Note that here already the problem of a revision of the whole repository is addressed. Reviewing can point out that – albeit the

[3] There has been and is still going on an email discussion about these options, especially the last grade is questionable since virtually in any paper one can find useful parts. E is kept for articles which cover already formalized topics, blocks of theorems trivial for the checker or just meaningless definitions and their consequences.

[4] There even has been the proposition of making public submissions before reviewing to avoid this problem, but we are not aware of a definite decision concerning this point.

author has proved his main results – the way Mizar and MML support establishing the presented results is not optimal and should be improved.

As the most notable example here, we can cite the newly submitted definition of a kind of a norm for the elements of the real Euclidean plane, which are defined in the Mizar library just as finite sequences of real numbers.

```
definition let n be Nat, f be Element of TOP-REAL n;
  func |. f .| -> Real means
    ex g being FinSequence of REAL st
      g = f & it = |. g .|;
end;
```

where `|. g .|` is a usual Euclidean norm which was introduced in the MML before as

```
definition let f be FinSequence of REAL;
  func |. f .| -> Real equals
:: EUCLID:def 5
    sqrt Sum sqr f;
end;
```

After the change of the loci type (the submission obtained grade D, of course) from `FinSequence of REAL` into `real-yielding FinSequence` in the `EUCLID` article the earlier definition was no longer needed which helped to simplify the structure of notions in this new submission. We can (and eventually did so in December 2007) go even further: it is reasonable to separate all operations on functions fulfilling the same scheme into new library item; that is the way two articles from `VALUED` started – devoted to various functions with numbers as values and their properties.

The possible flattening of the net of notions seems to be highly desirable – another illustrative example can be here the notion of the absolute value, defined originally for real numbers as

```
definition let x be real number;
  func abs (x) -> real number equals
:: ABSVALUE:def 1
    x if 0 <= x
    otherwise -x;
end;
```

and the other, corresponding but fully independent, for complex numbers:

```
definition let z be complex number;
  func |.z.| -> complex number equals
:: COMPLEX1:def 16
   sqrt ((Re z)^2 + (Im z)^2);
end;
```

Because all reals are also complex numbers (it is automatically obtained via the attributes and clusters mechanism), the following modification (such flattening or linearization is called a redefinition) was done:

```
definition let x be real number;
  redefine func |. x .| equals
:: ABSVALUE:def 1
  x if 0 <= x otherwise -x;
end;
```

which needed of course the proof of the equivalence of both notions for reals; `abs` is still kept as a synonym.

3.3 First Impression

The decision is not a typical result of majority voting, because referees giving the C grade point out possible improvements, so usually the lowest grade counts (luckily, in case of E marks, all three referees agreed).

To summarize the grades for 2006, let us look at Table 1.

Table 1. Number of submissions to the MML and their grades in 2006

	all	A	B	C	D	E
items	39	6	4	20	6	3
% of total	100	15.4	10.2	51.3	15.4	7.7

Basically, all ten submissions graded A and B were included into the MML, and among C and D candidate articles, which were returned to authors, other 15 were accepted; in total there were 25 Mizar articles accepted in 2006, the first year when the reviewing procedure as described above was introduced.

All in all we have seen that reviewing MML submissions indeed addresses only the first point mentioned in Section 2. Of course a thorough reviewing process will improve the quality of MML articles and may even guide authors into the direction of a good style of "Mizar writing". As we can conclude from Table 1, this is the case of the majority of submissions because the authors should enhance the articles according to the referees' suggestions. However there remain situations in which the MML as a whole should be improved; in the long term mere reviewing of submissions cannot avoid this. Here even carefully reviewing of Mizar articles – as already indicated by rate D above – can only help to detect the need for such revisions.

3.4 Further Development

In 2007, the policy of reviewing showed its usefulness – all articles with A and B grades were immediately (those with B – after suggested corrections) incorporated into the MML, as well as four among the others. Four articles still wait for a revision of the library. Interesting to see, comparing to the previous year the number of positive grades is granted to nearly 70% submissions. The reasons can be twofold:

Table 2. Number of submissions to the MML and their grades in 2007

	all	A	B	C	D	E
items	50	14	20	10	5	1
% of total	100	28	40	20	10	2

first of all, assuming that majority of authors are active on a long term basis, at least longer than one year, by applying referees' suggestions they worked out their better proof style for their future articles (remembering some of the submissions are revised C-grades from 2006, so the total numbers are bigger). The other reason is that articles can just be written better – thanks to revisions (especially generalizations, removing repetitions, and a better organization of knowledge).

Table 3. Number of submissions to the MML and their grades in 2008

	all	A	B	C	D	E
items	55	8	37	9	0	1
% of total	100	14	68	16	0	2

In the subsequent year, actual D grades were suggested, but they were followed by the detailed description of which changes of the MML should be done, hence appropriate revisions were done immediately. The percentage of A grades is lower; B ones were usually corrected by authors. What is worth noting here is that many C-grades from this year were not improved by the authors and resubmitted, few cases are students who just graduated. The only E was the case of unfortunate repetition of virtually all theorems from an article already accepted to the MML (of course the author did not know this).

4 Improving the Library

The Library Committee has been established on November 11, 1989. Its main aim is to collect Mizar articles and to organize them into a repository – the MML. Recently, from this agenda a new additional one was created – the Development Committee, which takes care of the quality of the library as a whole.

4.1 Types of Revisions

For the reasons we tried to point out before, the Mizar Mathematical Library is continuously revised. Roughly speaking, there are three different kinds of revisions:

– an authored revision – consists of small changes in some articles in the library when somebody writing a new article notices a theorem or a definition in an old

article that can be generalized. This is also the case of D grades as described above. To do this generalization, sometimes it is necessary to change (improve) some older articles that depend on the change. As a rule, a small part of the library is affected, but if it is not the case, usually it is hard to handle more such revisions at a time, otherwise the Library Committee has a lot of work with the synchronization of the patches (some sort of revision control system could be useful here).
- an automatic revision – takes place frequently whenever either a new revision software is developed (e.g. software for checking equivalence of theorems, which enables to remove one or two equivalent theorems) or the Mizar verifier is strengthened and existing revision programs can use it to simplify articles.
- a massive reorganization of the library – although was very rare before, as of now it happens rather frequently. It consists in changing the order of processing articles when the Mizar data base is created. Its main steering force is the division of the MML into the concrete and abstract parts.

4.2 MML Versions

Apart from the Mizar version numbering, the MML has also a separate indexing scheme. As of the time of writing, the latest official distribution of the MML was numbered as 4.128.1063.

As a rule, the last number, currently 1063 shows how many articles there are in the library (this number can be sometimes different because 36 items were removed so far from the MML, but some additional items such as EMM articles, and "Addenda" which do not count as regular submissions, were added). The second number (128) changes if a bigger revision is finished and the version is made official. Although it is relatively small comparing with the age of the library, the changes are much more frequent. To give some numbers, at the beginning of 2006, this value was set to 51, so a revision happens approximately every two-three weeks (or, to be more precise, this is how often it is made public).

4.3 Some Statistics

The policy of the head of Library Committee – to accept virtually all submissions from the developers and, if needed, enhance it by himself, was then very liberal. For these nearly twenty years there were only three persons taking the chair of the head of the committee (Edmund Woronowicz, Czesław Byliński, and currently Adam Grabowski); their decisions were usually consulted with other members of the committee, though.

Such an openness of the repository was justified: in the early years of the Mizar project the policy "to visit Białystok and get acquainted with the system straight from its designers" resulted in the situation that all authors knew each other personally, now the situation changed.

The MML evolved from the project, frankly speaking, considered rather an experiment of how to model mathematics to allow many users benefit from a kind of parallel development. Now, when the role of the library is to be much closer to

the reality and the MML itself is just one among many mathematical repositories, the situation is significantly different.

Table 4. Number of accepted submissions to the MML by year

Year	1989	90	91	92	93	94	95	96	97	98	99
Articles	65	136	46	48	33	33	35	57	39	47	65

Year	2000	01	02	03	04	05	06	07	08	09
Articles	54	33	42	54	80	48	25	40	47	17

As it can be seen, the first two years were extremely fruitful. No doubt, the first one was most influential, when the fundamentals, such as basic properties of sets, relations, and functions, the arithmetic, and vector spaces were established – to enumerate among many these most important. Some articles from that time were more or less straight translations from those written in older dialects of the Mizar language (Mizar PC, Mizar-2 etc., see [9]). Especially the subsequent year – 1990, when many authors could benefit from introducing the basics, hence they were able to work on various topics in parallel, brought into the Mizar Mathematical Library the biggest number of submissions so far (136 per year), then the number stabilized.

4.4 Towards Concrete and Abstract Mathematics

As it was announced in 2001 [12], the MML will be gradually divided into two parts. As the library is based on the Tarski-Grothendieck set theory, the part devoted to the set theory (and related objects, as relations, functions, etc.) is indispensable. There is, however, huge amount of knowledge for which set theory is essential, but actually based on the notion of structures by means of the Mizar language.

There are three parts of the Mizar Mathematical Library:

- concrete, which does not use the notion of structures (here of course comes standard set theory, relations, functions, arithmetic and so on);
- abstract, i.e. `STRUCT_0` and its descendants, operating on the level of Mizar structures; both parts are not completely independent – here the concrete part is also reused (abstract algebra, general topology including the proof of the Jordan Curve Theorem, etc.);
- SCM, the part in which the theory of Random Access Turing Machines are modelled, i.e. a mathematical model of a computer is described – from here in fact all lemmas of a more general character are taken to the other two parts. This will obviously never be self-contained but probably the best separated part of the MML.

This division is reflected in the file `mml.lar` in the distribution containing the order of processing of articles (it is especially important when creating the Mizar

data base) – the "concrete" articles go first, at the end we have those devoted to the SCM series. The process of separating these three parts is very stimulating for the quality of the Mizar library – many lemmas are better clustered as a result of this activity. Paradoxically, the more articles the concrete part consists of, the better is the organization of the library, because although structures allow for more feasible formal apparatus, still many useful "concrete" lemmas are contained in the abstract part.

As of the middle of 2009, this division can be summarized in Table 5.

Table 5. The three parts of the MML

Part	Number of articles	% of total
concrete	275	25.87
abstract	733	68.96
SCM	55	5.17
Total	1063	100.00

Note that apart from the revisions suggested by the referees when giving D-grades, any user can suggest changes via the Mizar TWiki site; he may of course also send an improved version via email to the Library Committee; as an example, lemmas needed for the Gödel Completeness Theorem were reformulated to provide its better understanding as a result of the external call.

4.5 Reviewing Software

Most of massive changes of the Mizar repository are done automatically with the help of computer programs. They can be roughly divided into four groups:

- existing, publicly available in every distribution of the system (the author is encouraged to clean his article until all these tools will report no errors, but there are only a few programs of this kind):
  - on the syntactical level – `chklab`, `inacc` (removing unused labels and blocks of text), here also (`renthlab`) can be listed – unifying labels in the article; this of course does not report any errors, but it makes the source more readable as the enumeration of theorems within the Mizar file reflects that of the abstract file (without proofs),
  - on the proof level – `relprem`, `relinfer`, `reliters`, `trivdemo` (suggesting unnecessary premises, proof steps, parts of iterative equality, and nested proofs, respectively),
- with use limited to the Library Committee (here the number of tools is much bigger, including editing versions of the above) – mainly syntactical (`corrolla`, `incassum`, `irrcase`, `irrvar`, `irrpred` – pointing out theorems justified only by library references, unused assumptions, cases in proofs, irrelevant variables and local predicates, to name only a few), also cleaning and sorting the directives in the environment declaration of the article,

– based on Bancerek's MML Query [3],
– developed by Josef Urban, we will list some of them in the next section.

4.6 MML Management

As a first tool of collaborative work on the library we can enumerate Mizar TWiki (`wiki.mizar.org`) which gradually changes its profile from an experimental – and rarely used – forum into the place where suggestions/experiences with the MML can be described. In our opinion, current Wiki for Mizar is far from being attractive (e.g. it lacks the option of remote MML database building to check if the revision can affect bigger part of the library) and the application of the others' work (also based on Wiki, like that of Coq [4]) would be interesting and highly desirable. But without any additional optimizations (there is no `make` for the MML) it would be abuse to call remote processing "interactive" because as of now, the verification of the whole library takes about six hours while the regeneration of the database files is ca. 30 minutes (frankly speaking, too much of this time is just file handling).

A the most important, and probably one of the better known MML tools, we can point out MML Query [3]. It has proved its feasibility in situations when one tries to search not only on the text level (by `grep`-like tools), but to find out items which use restricted set of constructors or notations, and it is how subsequent EMM (Encyclopedia of Mathematics in Mizar) items were created, e.g. `XCMPLX`, `XREAL`, `XXREAL`, and `XBOOLE` series, dealing with complex, real, extended real numbers, and boolean properties of sets, respectively. Also researchers, when writing their Mizar articles, can find it very useful. But usually, a typical author does not care too much if his lemma which takes some ten lines of Mizar code is already present in the library. Actually, searching for such auxiliary facts can take much more time than just proving them. This results in many repetitions in the library MML Query cannot cope with. And although the author can feel uncomfortable with multiple hits of the same fact, annotating such situations and reporting it to the MML developers is usually out of his focus.

This is the area where another tool comes in handy. Potentially very useful for the enhancement of the MML as a whole, MoMM (Most of Mizar Matches) developed by Josef Urban was primarily developed to serve as an assistant during authoring Mizar articles [15]. It is a fast tool for fetching matching theorems, hence existing duplications can be detected and deleted from the MML (according to [15], more than two percent of main Mizar theorems is subsumed by the others). The work with the elimination of these lemmas is still to be done – many of detected repetitions are useful special cases (or important proof steps, as in the Jordan Curve Theorem polygonal case was), so their automatic removal is at least questionable. The authors often want to add straightforward consequences of some theorems from the MML to enrich the theory they develop and to have a complete set of properties in a single place.

Another popular software, MML CVS – the usual concurrent version system for the MML was active for quite some time, but then was abandoned because the changes were too cryptic for the reader due to the lack of proper marking of items. The main obstacle in the current state of the MML is that still there

are no absolute names for definitions and theorems. Albeit the enumeration of theorems within one file is not a big deal, especially if some of the theorems which are before the chosen are moved elsewhere (`canceled` keyword which serves as a placeholder for deleted theorem keeps the numbers right). The real problem is with the movements of bigger parts between the articles, where simple translation of the old library reference into the new one is insufficient; the user is obliged to add some environment declaration items. Then the changes are usually too massive and too technical to find out what really matters (and then it is better to check the differences on the level of the abstract – i.e. without proofs).

5 Which Way to Go?

Contemporary standards of the publishing process open some new possibilities – there are many journals online, Springer also announces their books/proceedings at their webpage. Paper-printed editions have some obvious limitations, vanishing for electronically stored and managed repositories of knowledge. We can notice, as an example, new functionalities of [10] in comparison to (even online) version of Abramowitz and Stegun [1].

5.1 Flexibility

Of course the content, once published on paper, is fixed. We can mind some real-life situations – rough sets as an example of obtaining new results via a kind of revision process (originally considered to be classes of abstraction with respect to some equivalence relation, then some of the properties were dropped to generalize the notion – see [7], [8]); also Robbins algebras and related axiomatizations of algebras are a good example, when a classical problem could be rewritten and reused when solved. In the aforementioned examples these were reasons for writing other papers, within the computerized repository the enhancement (the generalization of results) can be obtained via the revision process. Still, the problem of authorship of such "mixed", in a sense, results remains.

5.2 Distribution

As a rule, building an extensive encyclopedia of knowledge needs some investment; on the one hand, it can be considered by purely financial means as "information wants to be free, people want to be paid" [2]. That is the way Wolfram MathWorld [17] has been raised, as a collection closed in style, in fact authored by one person, Eric Weisstein.

But right after this service has been closed due to the court injunction, it soon appeared that the need to bridge this gap is that strong – many volunteers were working to develop a concurrent service to that of Wolfram's, but of the more open type, based on the mechanism similar to Wiki.

The effort of PlanetMath[5] is now a kind of Wikipedia for mathematics (in fact they even cooperate closely). Adding or editing content by virtually anyone is an

[5] `http://planetmath.org`.

obvious advantage of such sources, but at the same time this also makes them less stable and less reliable. Jimmy Wales, Wikipedia's founder, recalls that it is not always a definitive source of information, and hence is not ideal for academic purposes, if the knowledge is not supported by books, journals, etc. The urgent need for correctness checking becomes critical when mathematical knowledge is taken into account. Here, however proof assistants give the valuable possibility of checking, at least for the correctness of proofs; still the question of whether the definitions are right, i.e. how the encoded version reflects real mathematical objects, is the one only humans can answer to a full extent.

As we can observe based on the HOL Light system, John Harrison is the person who formalized all 74 theorems from Wiedijk's "Top 100 Mathematical Theorems" [16] proven with this proof-assistant. In the case of Mizar, 50 were developed by 36 people (notable exception is Marco Riccardi who during past three years formalized twelve important facts from the list). Archive of Formal Proofs of Isabelle is closer to the MML in this sense. Of course, for the bigger number of developers, high price of lacking homogeneity must be paid. It is important then, to have a group of supervisors (the "core team") of the repository, as is the Library Committee for the MML.

5.3 The Question of Authorship

Even in non-profit projects, like GNU, people may want to get their payment in another form: at least the added annotation such as "This article is owned by...", as in PlanetMath, which can also be considered a kind of motivation to keep higher standards of the encyclopedia since the authorship is not fully anonymous.

In the MML the authorship is fixed (every article is annotated, there are some items of Library Committee), there were however, especially recently, cases when the parts of submissions moved around so that the authorship actually exchanged (as for example, with the formalization of the Zorn Lemma, originally created by Grzegorz Bancerek, and now, after the changes concerned with the move of this article to the concrete part, attributed to Wojciech Trybulec). Of course, the content published in the automatically translated journal *Formalized Mathematics* is authored accordingly with the original, not revised version. In the case of many articles, especially those submitted much earlier, the content which was written by their author is only a small part of the original size due to new language capabilities or the generalizations which caused triviality of theorems proven there. There is over 200 authors of MML items; many of them are no longer active (they left the Mizar project), some were just short-term collaborators (e.g., students who did the work suggested by someone else) and the ownership of such article does not matter so much, at least for the authors.

It is clear that databases of knowledge will not be (so is not the MML as of now) only collections of formalized existing, classical results. The authors who develop their own results should retain their rights. Partly, although not officially stated, it is achieved via dividing the article into two parts: preliminary, or of much more general interest, and the proper submission. The first section is meant to be moved eventually in a more appropriate place in the core of the MML. Usually

the reviewers' opinions reflect this policy – and even if the grade D is not given explicitly, the suggestions are expected to be taken into account by the Library Committee.

6 Final Remarks

To meet the expectations of researchers being potential users of repositories of mathematical knowledge, such collections cannot be frozen. The availability of the contemporary electronic media opens new directions of the development of the new encyclopedias yet unavailable for their paper counterparts. The need of the enhancement stems not only from the fact that there may be some obvious mistakes in the source; the reasons are far more complex.

In the paper, we tried to point out some of the issues connected with the mechanism of revisions performed on the Mizar Mathematical Library, a large repository of computer-verified mathematical knowledge. The dependencies between its items and the environment declaration (notation and especially, constructors) are as of now too complex to freely move a single definition or a theorem between separate articles.

In our opinion, the current itemization of the MML into articles does not fit the needs we expect from the feasible repository of mathematical facts; if we try to keep authors' rights unchanged, there is an emerging need to have some other, smaller items which guarantee the developer's authorship rights, a kind of ownership similar to that used in the PlanetMath project. We propose to keep the authorship for any basic Mizar block/item (theorems, definitions and registrations), having in mind that some of private lemmas which are not exported to the data base, can be significant steps of the proof of a public theorem.

Also the better automation of the MML revision process is strongly desirable. However possible, at least to some extent, due to some difficulties which can be met as we pointed out, the human supervision of such automatic changes will probably always be needed.

Acknowledgments

The first author wants to express his gratitude to Andrzej Trybulec and Artur Korniłowicz for their continuous cooperation on the enhancement of the MML. We express also our gratitude to all the reviewers of the Mizar Mathematical Library (especially Adam Naumowicz) for their great job done.

References

1. Abramowitz, M. and Stegun, I.A.: *Handbook of Mathematical Functions*; National Bureau of Standards, Applied Mathematics Series No. 55, U.S. Government Printing Office, Washington, DC, 1964, see also `http://www.convertit.com/Go/ConvertIt/Reference/AMS55.ASP`.

2. Adams, A.A. and Davenport, J.H.: *Copyright issues for MKM*, in: A. Asperti, G. Bancerek, and A. Trybulec (eds.), Proc. of MKM 2004, LNCS 3119, Springer, pp. 1–16, 2004.
3. Bancerek, G.: *Information retrieval and rendering with MML Query*; in: J.M. Borwein and W.M. Farmer (eds.), Proc. of MKM 2006, Lecture Notes in Artificial Intelligence 4108, pp. 65–80, 2006.
4. Corbineau, P. and Kaliszyk, C.: *Cooperative repositories for formal proofs*; in: M. Kauers, M. Kerber et al. (eds.), *Towards Mechanized Mathematical Assistants,* Proc. of MKM 2007, LNAI 4573, Springer, pp. 221–234, 2007.
5. de Bruijn, N.G.: *The Mathematical Vernacular, A Language for Mathematics with typed sets*; in: P. Dybjer et al. (eds.), Proc. of the Workshop on Programming Languages, Marstrand, Sweden, 1987.
6. Davenport, J.H.: *MKM from book to computer: A case study*; in: A. Asperti, B. Buchberger, and J. Davenport (eds.), Proc. of MKM 2003, LNCS 2594, Springer, pp. 17–29, 2003.
7. Grabowski, A.: *On the computer-assisted reasoning about rough sets*; in: B. Dunin-Kęplicz et al. (eds.), Monitoring, Security, and Rescue Techniques in Multiagent Systems, Advances in Soft Computing, Springer, pp. 215–226, 2005.
8. Grabowski, A. and Schwarzweller, C.: *Rough Concept Analysis – theory development in the Mizar system*; in: A. Asperti, G. Bancerek, and A. Trybulec (eds.), Proc. of MKM 2004, Lecture Notes in Computer Science 3119, pp. 130–144, 2004.
9. Matuszewski, R. and Rudnicki, P.: MIZAR*: the first 30 years*; Mechanized Mathematics and Its Applications, 4(**1**), pp. 3–24, 2005.
10. Miller, B.R. and Youssef, A.: *Technical aspects of the Digital Library of Mathematical Functions*; Annals of Mathematics and Artificial Intelligence, 38, pp. 121–136, 2003.
11. The Mizar Homepage; `http://www.mizar.org/`.
12. Rudnicki, P. and Trybulec, A.: *Mathematical Knowledge Management in Mizar*; in: B. Buchberger and O. Caprotti (eds.), Proc. of MKM 2001, Linz, Austria, 2001.
13. Rudnicki, P. and Trybulec, A.: *On the integrity of a repository of formalized mathematics*; in: A. Asperti, B. Buchberger, and J. Davenport (eds.), Proc. of MKM 2003, Lecture Notes in Computer Science 2594, pp. 162–174, 2003.
14. Sacerdoti Coen, C.: *From proof-asistants to distributed knowledge repositories: tips and pitfalls*; in: A. Asperti, B. Buchberger, and J. Davenport (eds.), Proc. of MKM 2003, Lecture Notes in Computer Science 2594, pp. 30–44, 2003.
15. Urban, J.: *MoMM – fast interreduction and retrieval in large libraries of formalized mathematics,* International Journal on Artificial Intelligence Tools, 15(1), pp. 109–130, 2006.
16. Wiedijk, F.: *Formalizing 100 Theorems,* `http://www.cs.ru.nl/~freek/100/`.
17. Wolfram Mathworld web page; `http://mathworld.wolfram.com/`.