

# Bluenote: feature building for the Planetary system

Joe Corneli  
September 28, 2011

**Executive Summary:** I describe items from the “PlanetMath-Remixed” and “Future” milestones in the Planetary Trac system, emphasising use cases in education, publishing, and software development.

## Plans for further development of the Planetary System in the post-beta phase

Assuming we are able to complete the port of PlanetMath into Planetary, as described earlier, we will be able to take the system live – but the work will not stop there.

**Linus Torvalds:**<sup>1</sup> The first [mistake] is thinking that you can throw things out there and ask people to help. [More realistically, y]ou make it public, and then you assume that you’ll have to do all the work, and ask people to come up with suggestions of what you should do, not what they should do. Maybe they’ll start helping eventually, but you should start off with the assumption that you’re going to be the one maintaining it and ready to do all the work. The other thing—and it’s kind of related—that people seem to get wrong is to think that the code they write is what matters. No, even if you wrote 100% of the code, and even if you are the best programmer in the world and will never need any help with the project at all, the thing that really matters is the users of the code. The code itself is unimportant; the project is only as useful as people actually find it.

Indeed, this is how PlanetMath was developed in the first place: Aaron Krowne was engaged in a continual conversation with users on the #math irc channel on Undernet, where they gave feedback on the features of the site.

So, realistically speaking, the next phase of development is not about the features *we* want to add to the site, but the features that users find missing. Nevertheless, we can opine, a bit, about what sorts of requests we think will be pending. After all, some fairly generic requests have been pending for years.

**(1) Email integration** PlanetMath could be used as a mail server, for discussions between co-authors of papers or co-editors of encyclopedia sections. We could even set things up so that people could request and post edits from the comfort of their email client. Combined with appropriate metadata, this feature would help researchers who are only interested in reading articles about, say, algebraic topology. There is at least some basic support for mail integration in Drupal; see the relevant TICKET.

---

<sup>1</sup>[HTTP://H30565.WWW3.HP.COM/T5/FEATURE-ARTICLES/LINUS-TORVALDS-S-LESSONS-ON-SOFTWARE-DEVELOPMENT-MANAGEMENT/BA-P/440](http://h30565.www3.hp.com/t5/FEATURE-ARTICLES/LINUS-TORVALDS-S-LESSONS-ON-SOFTWARE-DEVELOPMENT-MANAGEMENT/BA-P/440)

**(2) STACK integration** STACK is a program that can do some automated assessment, which would be very useful for creating quizzes and tests, particularly with introductory topics that don't need detailed proofs or diagrams in the answers.

**(3) math-enhanced Etherpad and other real-time interactive mathematical tools**

A math-enhanced version of Etherpad (or the new node.js-based Etherpad Lite) would be very useful for running live, interactive, tutorial sessions (as well as for live research discussions). Combining Etherpad with voice chat, an interactive whiteboard, a shared Geogebra, a shared SAGE prompt, etc., would give a powerful suite of tools for various kinds of mathematical collaboration. Note that one of the nice things about live tutorials is that they come with an implied business model: people can charge money for tutoring (see the next item).

**(4) an open mathematical marketplace** Students would be able to contract for grading services as well as for live tutorials. With permission of users, transcripts of tutorial sessions and advice about problem sets could be shared publicly on the site, perhaps suitably anonymised. Infrastructure to support such service provision would keep track of tutor reputation, as a way to maintain trust and credibility.

**(5) auto-generated PDF and Epub monographs and course packets** It would be great to be able to export objects that have been built using the “book” feature we have built for Planetary in a variety of user-friendly formats. There is another implied business model here, namely print-on-demand publication of user-created PDF books (precedent for this exists in the Padiapress service integrated with Wikipedia).

**(6) distributed revision control** One of the features of PlanetMath is that, unlike on standard wikis, articles are “owned” by the person who created them. This gives significant authorial control to experts. However, it can make helping out harder, since help is offered through a corrections system that doesn't admit line edits. It would be good to integrate a distributed ownership model (to allow variant versions of articles existing alongside the canonical version) for this reason. This approach would also help with creating different versions of articles for different uses (e.g. at varying levels of detail). Presumably such a tool would come with both web and command-like interfaces, which would support different styles of use.

*For specific constituencies of users, we can anticipate certain other requirements. In the education sector, we have to consider (a) complete novice users – beginning students – and, (b) would-be teachers or content authors who do not have great experience in teaching or pedagogy.*

**(7) equation editor, ASCIIMath syntax, and staged examples for writing mathematical text** Beginning students – and, indeed, many mathematical users (like high school teachers), are not going to be instantly comfortable with  $\text{\LaTeX}$ , and would like a point-and-click means of entering content into the site. For users who are comfortable typing, ASCIIMath syntax may be more usable as an input format than  $\text{\LaTeX}$ . Note that

simplified syntax could be implemented as a “front-end” to L<sup>A</sup>T<sub>E</sub>X<sub>ML</sub>, and we could add multiple modes for editing the same content in ASCIIMath, Mediawiki syntax, etc. In addition to tools, we would want to be able to present users with clearly examples showing how to type the kinds of expressions relevant to the subject they are studying.

**(8) guides for writing with pedagogy in mind** PlanetMath is historically an encyclopedia, and many articles are written in a terse style. When problems are introduced, the encyclopedia will be subject to a deeper critique: do the articles contain enough information for users to figure out how to answer questions? An organic increase in quality should follow. However, good approaches to staging content, building progressive complexity, and so forth, apply at another level (not just within articles, but in course packets based on the articles).

**(9) slide-show mode** It would be very useful for lecturers (and presenters) to be able to present material using a slide-show mode.

*In the publishing sector, we would look for tools for reviewing and editing.*

**(10) editorial roles and workflow** In many respects the workflow management would be similar to the services provided by EasyChair. Combining with Epub and print-on-demand services as in (5), above, would provide a nice integrated solution for journal publication in the mathematical sciences.

*Finally, extensions of the software beyond mathematics could begin by using the tool to document itself.*

**(11) Tutorials for extending the tool** Although we don’t expect users to dive in and work on the software immediately (see the Torvalds quote above), contributions are much more likely if the system is well-documented. On a whole, the underlying Drupal system has “OK” documentation, but it could be more detailed. Indeed, the question here is what sorts of documentation are really useful for online communities: we can well imagine building a “PlanetComputing” that addresses these questions in a general way, but we will start with our own online community first.

**Summary/Conclusion** Unlike the notes on the push up to a beta, the ideas I’ve presented here are somewhat open-ended, and the actual trajectory of future development will depend on feedback from actual users. I have certainly not gone into great detail on semantic services that would support the types of interactions I’ve sketched, rather, I’ve focused on high-level use cases (the ones which I think are likely to make a big impact). Once we are done with the beta, the notes here should be reassessed. One thing is clear, however, which is that support, maintenance, and improvement of the system is likely to be a big job.