

## ABSTRACT

### Keywords:

The vision for open source publishing. The current publishing system for engineering and scientific research is growing at an alarming rate as the 'publish or perish' doctrine continues to grip the academic community. There is an ever-growing catalog of journals to which libraries around the world must subscribe. This expense has created a heavy burden on cash-strapped institutions which are constantly trying to evaluate which subscription they can let lapse and which are important. Publishing is a big business that generates an enormous amount of revenue preying on the need for researchers to disseminate their work. While there are certainly expenses associated with publishing material, even if done entirely online. The cyber-infrastructure must be developed and maintained, documents professionally typeset, etc. However, the most difficult part to develop, i.e. the content, comes free. Journals are inundated with researchers from around the world trying to get their work out (and pad their publication list for whatever reason). Engineering journals accept roughly one in three submissions. The burden of keeping the quality high is put on the editors and associate editors who are typically only compensated for their expenses and not their time which is donated for, among many reasons, service to the community. Explaining the academic publishing world to anyone not directly involved with it makes you see how strange the system really is. It is not difficult to see how we got ourselves into this situation as peer-reviewed publications has become an essential part of promotion and tenure. Someone once commented to me that it's like musicians giving their work and all rights to it freely to the music industry and then having producers donate their time to help polish and refine the work. Why would musicians do this? They wouldn't! So why are academics, whom I like to think are a more intelligent group than musicians, willing to do it? For anyone outside looking in, it seems crazy. In fact, part of this system is already starting to collapse under its own weight. Peer review is an essential part of the publishing system that ensures the quality of the work being published and put a stamp of approval on the work that allows administrators unfamiliar with the area to evaluate it. However, reviewing is a job that comes with no real benefit to the reviewer. For tenure, reviewing is a small part of the service section of a dossier that basically consists of a short statement saying that the author served as a reviewer for journals X, Y, and Z. The actual number of papers reviewed is never requested and does not at all help a person's reputation, standing, or promotion success. The fact that reviewers are anonymous to everyone but one or two journal editors exasperates the problem. A very small number of people know whether a person is a good reviewer that is thorough and timely. Thus, to play the game at an optimal level requires the time spend reviewing papers be pushed to zero while the time spent writing them is maximized. It appears that more and more people are realizing this and it is getting harder and harder to get people to provide quality and timely reviews. This disparity in the 'reward' system for writing and reviewing papers continues to grow and, I fear, will soon reach a breaking point where journal editors just cannot get all the papers submitted reviewed. Most editors will now make a decision based on two reviews when the standard used to be three. Even getting two can be difficult as it frequently requires multiple reminders. I must also admit to being slow to review many papers as there always seems to be more pressing issues on my to do list. One of the reasons my colleagues and I have discussed for this continuing problem with getting adequate reviews is the continuing decline of a community spirit among academics. People used to, or so I am told, feel much more part of a community and, thus, were much more willing to contribute to that community by doing behind-the-scenes work, such as paper reviewing, for which they got no direct benefit. While there have always been (and always will be) people that will not contribute without widespread recognition, the lack of a community spirit and an attitude that everyone is competing with everyone else tends push tasks like paper reviewing aside for much more profitable endeavors. I can only speculate on the reasons for this dissolution of academic communities. I suspect that one reason is the easy and anonymous access that we now all have to

---

Further author information:

A.A.A. (to whom to send correspondence): E-mail: [aaa@tbk2.edu](mailto:aaa@tbk2.edu), Telephone: 1 505 123 1234

B.B.A.: E-mail: [bba@cmp.com](mailto:bba@cmp.com), Telephone: +33 (0)1 98 76 54 32

information. While the 'information age' has certainly benefitted academics in profound ways, it has also made research much more anonymous. Before everyone had easy access to online journals (and certainly before photocopying was readily available), the only way to get a copy of a paper was to contact the author directly. This informed an author of the people that were interested in and following his/her work and opened the door for further discussion and collaboration. Today there is basically no way to know who is reading one's papers. The only indication usually occurs years after when an author notices that a number of people with whom s/he has never spoken have cited the paper in their own work. For example, one of my students pointed out that one of my papers with another student was in the list of the journal's most downloaded papers for 2006. I had no idea anyone was even interested in the work and still have no idea who any of the people are that downloaded the paper. I also think the competitive environment in which we work and position ourselves for success hampers the community spirit. With the funding rates for science and engineering so low, academics are viewing others in their field (i.e. community) as competitors and are thus less willing to share ideas or their latest unpublished work with others fearing theft of their intellectual property. This is truly an unfortunate development in the academic community where pursuit and sharing of knowledge should be the top priority. With academic institutions acting more and more like big businesses, it is not hard to understand how this competitive attitude permeates down to the faculty. While competition between researchers is nothing new and generally healthy for advancing science and engineering, the additional pressures felt by today's academics lead to more unhealthy competition as people feel their advancement is tied to their success at 'winning' funding, awards, etc. The internationalization of research and efforts by more and more academic institutions around the world to develop strong research programs is also another likely cause. With so many more academics having their career advancement tied to their research success, academic communities are becoming more like large cities than small towns. Everyone passes each other anonymously and people treat each other with little respect as there is little chance of ever encountering one another again. My PhD used to emphasize that the academic world was still small and not treating people well (i.e. burning bridges as you cross them) would one day likely come back to hurt you. What is a wiki? The word 'wiki' is a Hawaiian-language word for fast and was first used in the context of online collaboration by Ward Cunningham who began developing in 1994 what he termed the 'WikiWikiWeb'. The idea was to have a web site that would allow anyone to edit the site directly in their web browser. A virtual community would then develop as users would collaborate to refine a web page containing some information. Wikis are similar to web logs (blogs) and content management systems (CMS) with which many are familiar and the idea of a wiki predates that of the current blogging concept. The primary difference is who has the authority to edit the site content. With CMSs editing is typically restricted to privileged users that are allowed to add or update content. Many academic 'units' now use CMSs to allow administrators to easily update (and keep current) information on the site. Most editing is done in the browser (or with simple text editors), the information is stored in a database, and the content is generated dynamically by accessing and formatting the information in the database. Similar to the idea of Cascading Style Sheets (CSS), this allows the content to be decoupled from the formatting and page layout. The people that change the content then do not have to understand HTML, database and server-side scripting languages, Javascript, or the Document Object Model (DOM) to be able to change the basic content of the site. A blog is essentially a slightly more open CMS. Each blog typically belongs to a single 'administrator' that can post content, e.g. articles, commentary, reviews, etc. Only the administrator can edit this content. However, anyone else\* can post a comment about the original blog entry. This allows readers of the blog to discuss the material posted by the blogger and the blogger to respond. Lengthy and sometimes contentious debates often ensue. Unfortunately, these debates often do not resolve anything and lead to so-called flame wars<sup>†</sup>. The reason for this is, in my opinion, the same reason people treat each other poorly in big cities, everything is anonymous. It is easy to be rude and provide destructive criticism when a person is hiding behind a pseudo-name or in society when people think they will never again meet the subject of their abuse. However, once a community develops the dialogue becomes much more constructive and good-natured debate ensues<sup>‡</sup>. Requiring people to register with their real name and email address (i.e. not a random Hotmail or Yahoo! account) and/or having a moderator filter the posts reduces this kind of

---

\*Some blogs require users to register.

<sup>†</sup>A term originally used to describe the infantile arguing between people posting in Usenet newsgroups.

<sup>‡</sup>There will always be the occasional person that enjoys antagonizing people. However, if these people are a small percentage their posts are just ignored and they eventually lose interest.

behavior tremendously<sup>§</sup>. The most open of these systems is the wiki which allows, in its purest form, anyone to edit anything. The most well established and frequented wiki is Wikipedia ([www.wikipedia.org](http://www.wikipedia.org)) which was started as an open-source encyclopedia called Nupedia. Not long after starting Nupedia, the founders decided to use a wiki as the underlying management system. This allowed anyone to edit entries in the encyclopedia so that information would hopefully converge to something well written, factually correct, and comprehensive. The key feature of modern wikis that makes this work is the built-in versioning system. Each version of a page is stored in the database and the content can always be reverted back<sup>¶</sup>. Thus, if a page is ‘vandalized’ it is easily restored. The most interesting Wikipedia pages are those discussing topics that are feverishly debated. The page discussing abortion ([http://en.wikipedia.org/wiki/Abortion](http://en.wikipedia.org/wiki/Abortion "http://en.wikipedia.org/wiki/Abortion")) is a good example. After a very rough and contentious begin in which the content oscillated widely between pro-life and pro-choice perspectives, the page converged to something that begins with factual information and then includes the most common thinking from both sides. There is now an entirely separate page discussing the abortion debate and both sides have essentially agreed to disagree<sup>||</sup>. It is interesting to look at the history of pages to see how they have been refined. One will find that pages about popular topics have actually changed very little over the last few hundred edits or few months—mostly wording has been tightened, and references and cross-links added. The most frequently used server platform is the combination of Linux (the OS), Apache (the webserver), MySQL (the database), and PHP (the scripting language, it is not an acronym) which is often referred to as the LAMP platform. The only piece that is currently necessary is PHP which is a popular server-side object-oriented scripting language PHP in which MediaWiki is written. The operating system, webserver, and database requirements have all been abstracted and nearly any compatible combination can be used. The PI has found the combination of Mac OS X (which has Apache and PHP pre-installed) and MySQL a very efficient development environment. A short description of PHP and MySQL are given below for the reviewer unfamiliar with them. The PHP is implemented as an Apache or IIS webserver module. When a request is made for a page with a file name ending in .php (or any other designated extension) the webserver parses the file, interprets any PHP code, and sends the results to the browser. While PHP began as a simple way to dynamically generate web pages based on user input, it like Javascript has grown into a full-featured language with the functionality of ‘traditional’ languages such as C++, Fortran, and Java. Many major corporations and universities use PHP extensively and many web-enabled software packages use it. IBM is now actively supporting its development, putting PHP in the same league as ASP (active server pages) and JSP (java server pages) for delivering dynamic web content. An intricate knowledge of PHP is not required to run most PHP-based web applications\*\* as installers have become more sophisticated, allowing automatic configuration for different platforms. However, a good understanding of PHP is extremely useful as many web applications are open-source and can be readily modified and/or integrated for a particular use. The PI has become proficient with PHP and has written and/or modified many web applications using the language<sup>††</sup> and will be able to work closely with dedicated programmers to modify the MediaWiki software to fit the needs of this effort. While PHP can be used alone to generate web content, its power is really exposed when used in combination with MySQL. The integration of MySQL and PHP is very tight although other databases such as Microsoft SQL Server, Oracle, or PostgreSQL can be used. MySQL (as well as PostgreSQL) is also open-source, freely available, and extremely powerful and scalable<sup>‡‡</sup>. While a knowledge of SQL (an ANSI/ISO standardized language for interacting with databases) is not needed to simply run and use most web applications that use a database, it is required to customize the application as is proposed here. Fortunately, the PI has also become proficient with SQL over the last four years and will be able to work with dedicated programmers to

<sup>§</sup>Most listserv mailing lists are moderated to prevent spam and other inflammatory posts

<sup>¶</sup>Most wiki software allows users to compare versions to readily see what has been changed.

<sup>||</sup>There is an on-going fundamental debate about whether Wikipedia as an encyclopedia should even allow opinions. However, it seems that the community has agreed to allow the inclusion of common opinions and pages about hotly debated topics have some editing restrictions.

<sup>\*\*</sup>No more so than a knowledge of C++ is required to compile a program written in the language.

<sup>††</sup>See [http://www.smallfeats.com](http://www.smallfeats.com "http://www.smallfeats.com") as an example.

<sup>‡‡</sup>Wikipedia uses it to store \*\* GB of data and retrieve \*\* GB/day.

implement changes. Closer to this proposal is the PIs use of wikis for collaborative writing. He has now used wikis to write seven collaborative proposals over the past three years. This way of writing has been extremely productive and he has been contacted by many others interested in the idea. While the PI is not willing to allow the reviewers to see all the 'dirt' behind proposals on which he was not the sole PI, the wiki used to write this proposal is open to all and can be found at [http://dssl.mne.psu.edu/nsfevo](http://dssl.mne.psu.edu/nsfevo "http://dssl.mne.psu.edu/nsfevo"). The wiki layout and work-flow that has evolved is described briefly here because the PI envisions it as a model for the future of academic publishing. The sections (repositories) of the wiki basically divide the proposal preparation process into natural phases. This is somewhat modeled after the software development process used by most developers. There is an Idea Repository that is a page (or pages) of raw ideas related to the proposal. Entries can be anything from random thoughts about ideas in the proposal to discussions for the focus of the proposal. The Idea Repository is used mostly in the planning phase to converge to a unified vision for the proposal. However, it is also used during the writing phase for any new ideas or directions a PI has developed. Others can consider the information for inclusion in the proposal. The Idea Repository is also used to develop an outline for the proposal. This outline then becomes the template for the Text Repository. Here, ideas are integrated and the text is filled in to expand the outline. This is analogous to the alpha phase of software development where new features are being implemented. PIs are typically assigned to be the owner of different sections of the Text Repository based on their contribution to the proposal. Little editing is done by PIs outside of the section(s) to which they have been assigned. However, this is possible and everyone can see everyone else's contribution in (pseudo-)real time. The contents of the Text Repository are typically very disjoint (basically everyone's 'two pages') and rough. Once all the contributions have been made, all the ideas are present, and the focus of the proposal defined, the contents of the Text Repository are moved to the Integration Repository. The Integration Repository is analogous to the beta phase in software development. Everyone is encouraged to edit and refine each other's work so that the proposal becomes a cohesive document. This is also where text is tightened and trimmed since Text Repository version of the proposal is typically (way) too long. The PI has found that this integration phase is where most of the time is spent and the most editing is done. Like with the beta period for software, the longer the proposal has in the Integration Repository the better the result. Because editing in a browser window is not the most efficient environment, users will typically pull text from the browser to edit offline. The text is then dumped back into the wiki and saved. A user will mark the section header to indicate s/he has checked the section out for editing. This prevents two people from editing the same section at the same time. Recall that each version of the content is saved so that one can readily see what has been changed and revert any change to a previous version. This process has been extremely effective in getting a proposal refined in a very short time (as is often necessary). Once the proposal is essentially ready to be submitted (the release candidate phase) it is moved to the Submission Repository where only minor edits are allowed and any final corrections are made. There is a File Repository where figures, references, draft budgets, etc. can be uploaded and a Reference Repository where citations (in BibTeX format) are added. Because wikis basically store content as pure text files, a markup language (ML) is needed to format the content for display. Each wiki package uses a slightly different ML but all are similar to HTML. When content is requested the wiki software requests the page from the database, converts it from its ML to HTML, and sends it to the web server. While one can accomplish essentially any formatting with cascading stylesheets (CSS), the visual and printed results are still very much browser dependent and users are typically unfamiliar with the syntax. As an interim solution until browser page layout converges, the PI has found that a combination of wiki ML (WML) and LaTeX which can be converted to PDF is the best solution. For example, section headers and basic text formatting (bold, italic, etc.) is done in WML to make the content presented in the browser readable and easily navigated. Unfortunately, MathML has not been adopted as rapidly as the developers had hoped. Thus, the PI has resorted to using LaTeX for equations. There is also no convenient way to do dynamic cross-referencing and bibliography compilation with WML or HTML. LaTeX is again used for this purpose. Figures and tables are also entered using LaTeX, although converting the WML for embedding a figure to LaTeX would be very easy to do. Converting WML or HTML tables to LaTeX tables would be much more difficult but certainly possible. When a user wants a typeset version of the proposal, they click on a link which runs a script on the server. This script pulls the proposal body, the references, and LaTeX headers from the database. The body is parsed to convert WML

---

Although it is now an official standard which browsers are beginning to support.

to LaTeX and wrapped with the headers and formatting styles for the particular funding agency. The LaTeX is then converted to PDF which is returned to the user. The biggest hurdle to getting people to use the wiki for collaborative proposal writing is convincing them that LaTeX is not as hard to use as is commonly thought. This has not been a huge hurdle because the users never see all the ugly header information and class styles that intimidate most novice LaTeX users. Users learn the mathematical syntax fairly quickly and learn to appreciate the non-WYSIWYG way of writing, i.e. get the content down and worry about the formatting later. There are also add-ons for most wikis that will parse LaTeX formulas and return them as image files for more convenient reading. The PI is also working on an add-on that will pop up a browser window with buttons and menus much like the equation editor bundled with Word. Using Ajax (discussed later) techniques, the formula is updated in near real-time in the browser to simulate the experience with which most authors are accustomed. An early version of this add-on can be found at [http://dssl.mne.psu.edu/AjTeX](http://dssl.mne.psu.edu/AjTeX "http://dssl.mne.psu.edu/AjTeX"); <http://dssl.mne.psu.edu/AjTeX/a>. The PI is also planning to develop a similar add-on for making tables, something with which even skilled LaTeX users struggle. MediaWiki (MW), the software on which Wikipedia runs, is open-source, runs on an open-source platform, and is the logical choice around which to build this next generation publishing system. The PI has extensive experience with the software, using it as the back-end for all the proposal wikis. While there are many different wiki applications available, MediaWiki is probably the most actively developed and used. Beyond Wikipedia, MediaWiki is being used by many small software developers to have the community help them document their software (something programmers and engineers are not typically good at). Sites dedicated to specific interests and topics have also found wikis a good tool for keeping information current. Educators are also beginning to use wikis; however this has mostly been in the liberal arts where essays can easily be posted and refined on a wiki. The PI uses a wiki in conjunction with a 'take-home' experiment in the large sections of the required vibrations class he teaches. Students are each given a small motor with an unbalanced mass attached to the output shaft (these are the motors that are used in vibrating cell phones). Each motor is wired in series with a variable resistor and the students are asked to attach the motor to things around their apartments to demonstrate resonance and modes. The wiki is used for them to exchange ideas and interesting observations. Two important features of MediaWiki that will be used extensively for reviewing submitted work are the discussion pages and the email notification system. Each 'main' page in MW has an associated 'talk' page (note the 'discussion' tab at the top of any Wikipedia page). These pages are, obviously, used to discuss and debate the content on the main page. The discussion pages work much like blogs where users can post comments and responses. It is this system that will be used to allow reviewers to post comments and suggestions about paper on the main page. The paper authors can then immediately respond to the comments and change the content of the main page as they see fit. The notification system in MW allows users to be notified when changes to pages on their 'watchlist' are made. Thus, users do not have to actively monitor pages they have created or edited. This feature significantly reduces vandalism of pages because a vandal knows that anyone watching the page will be immediately notified of the vandalism. The notification system will also let authors know immediately that a comment or suggestion has been added to the discussion page of one of their papers. By default, any page a user creates or edits is added to his/her watchlist. A user's watchlist can be modified at any time by the user. talk about how this will all work as a reviewing system The PI is interested in turning what has been a hobby into a contribution to the engineering community. Barriers to adaptation Getting the community to accept this new paradigm in publishing is not going to be easy and adaptation will certainly be slow. One point of trepidation for users is the 'stamp of approval' one gets when an article is accepted in a traditional journal. Obviously, the number of accepted publications has a huge weight in any metric used to measure an academic's performance. If an article published online in the Journal of Interesting Stuff does not get the same amount of credit as an article published in a traditional journal, then authors will not want to publish in this venue. Thus, department heads and chairs, deans, and promotion and tenure committees will require some form of 'acceptance' from the community. The acceptance system we currently have seems to basically be working well and this system will be replicated. The system will have two basic sections—submitted and accepted. An editorial board will decide when a paper can be moved from submitted to accepted; although all papers will always be under-review. The accepted section will be for the archiving of articles that have been deemed worthy by the editors. Users will still be able to discuss the paper and suggest revisions, and authors will still be able to make minor corrections.

---

The owner can then revert the page back to the pre-vandalized version.

The submitted section will contain papers that have not yet been sufficiently reviewed by the community. No paper will ever officially be rejected and authors can leave papers in the submitted section indefinitely if they choose. If modifications are made and new reviews added, the editors will be notified to see if the paper can be accepted. Alternatively, a system in which authors must request a decision might reduce the workload on the editors. Once a paper is accepted, a notice will be sent to the authors and the paper will be moved to the accepted section. Thus, a system of peer review and approval will not require promotion and tenure evaluations to change. Users will also have to alter their workflow to some extent. However, the hope is to eventually make the system simple enough to use that even the less computer-savvy user can feel comfortable. Any time you ask people to change how they do something, they will hesitate to make the change because there will be a time commitment to learning something new. However, if people can be convinced that the time spent learning something new will be saved many times over, they are much more willing to overcome the activation energy needed and settle into a lower energy state. Thus, ease of use is something that must be an essential part of the system. A complex system such as this will certainly need to be refined over time with user feedback. Thus, initial beta testing of the system will be limited to the more computer-savvy user by invitation only. Papers not accepted or those used in the beta test period can obviously be removed and/or submitted to traditional journals. Users will be able to 'submit' a paper in any stage of completion and in a variety of formats. The hope is that authors will eventually submit papers in a ML that can be readily parsed to make publication quality PDFs. However, authors will also be able to submit papers in PDF. These will not be editable online by the authors or any other users. Authors will also be able to upload content in LaTeX. I need to make the desire to allow anyone to edit a person's paper seem attractive. Describe how this whole thing will work. Paper Submission Once a user has an account, resolution printing. The natural choice for a ML would seem to be some superset of TeX. Many potential users of the OSP system would

ahref = "http://en.wikipedia.org/wiki/Help:Formula" class = "external free" title = "http://en.wikipedia.org/wiki/Help:Formula" rel = "nofollow" > http://en.wikipedia.org/wiki/Help:Formula </a >. A MW extension called Texvcparsesth source LaTeX to HTML converters. However, the better ones are not written in PHP. Thus, the first effort will be to determine the screen formatting. 2) Integrate BibTeX into MW. One of the great benefits of LaTeX and other word processors is the ability to auto generate a bibliography. In the PI's current implementation, this page is pulled from the MySQL database along with the LaTeX headers and content, and typeset to PDF using pdflatex and bibtex. This works very well for producing results in PDF that can be printed and archived. However, because the WM parser does not recognize the ?ommand, a bibliography is not generated when the page is converted to HTML. This effort will then replicate the functionality of bibtex in the MW engine in such a way that it can be distributed as an extension for anyone to use. Additionally, there is an outstanding MW extension called BibWiki that makes managing BibTeX databases extremely easy. Importing records from a variety of online sources is implemented and the PI has added sources (e.g. Web of Science) he uses extensively. 3) Add cross-referencing and automatic numbering to MW. Currently, cross-referencing can be accomplished in MW through the powerful ParserFunctions extension and the internal template mechanism. However, implementing something that works more like LaTeX (with ?nd ags) would be helpful for new users with some LaTeX experience, although the current mechanism in MW is no more difficult. The bigger problem with the current MW is that there is no automatic way to number figures, tables, and equations. The Cite extension automatically numbers citations and the PI will use it as a starting point to develop similar automatic numbering mechanisms for other objects in the document. The MW parser was designed to be extensible by simply registering a new function and calling the parser's 'setHook' function with the name of the tag and a callback function for processing the tagged content.

So why would anyone want to allow others to edit your paper? This is an obvious question for anyone to ask; especially those not familiar with the success of Wikipedia. Convincing people that this is a good idea will not

---

This system is however going to be forced to change if publishers continue to expand their portfolios to increase income. Even today some journals will basically accept almost anything, putting the assessment of quality on PT committees.

Editing a PDF is similar to changing a closed source software package by disassembling.

There has been talk about developing a 'WikiReader' that would likely use a WML to PDF conversion but nothing appears to have been done.

There has been a great deal of debate about how to include citations in Wikipedia. One side argues that documents with many citations, which is encouraged, are difficult to edit and that the references should be kept in a separate document. The other side wants to keep the complete reference with the content so users can easily edit the reference when editing the content.

be easy, which is why JIT will allow users to upload papers in not-easily-edited formats (e.g. PDF). However, the hope is that users will see the benefit (mostly by example) of allowing the community to edit their work. There were plenty of people skeptical of Wikipedia until they showed that, with minimal policing, the idea could work. There are actually many benefits and few, if any, detriments to allowing others to edit your work. The reader must remember that no edit is permanent, all can be easily undone, and there is a record of whom made the changes as users will be required to sign in to edit. For example, when the PI is reviewing a paper (especially one written by a non-native English speaker) there are typically many grammatical and stylistic mistakes. Identifying where these are in the document and what should be corrected is a major effort. For this reason, most reviewers do not even bother. However, making the change directly in the document is usually a simple matter (most of the time just adding or removing pronouns). Obviously, the authors would appreciate the corrections and if they disagreed with any they could just reject the change. Likewise, reviewers could quickly correct typos and simple mistakes in equations, all with the authors knowledge and ultimate approval. Reviewers could even add entire sections to the paper if they desired. Again, the author could accept or reject the changes, even adding the reviewer as a co- or contributing author if s/he wants. Papers could become a community effort. Because the history of the changes are tracked and the original author known, attribution of credit for both authorship and reviewing will be easily determined. Users could even post paper 'stubs' containing ideas and research on which they are recurrently there have been discussions about how to make an open access business model for publishing and much debate over whether it is possible to run an endeavor in come would be used solely to cover expenses and any profit would be used to reduce fees and/or donation requests in the

JIT - Journal of Interesting Things (also Just In Time) Open Source Publishing (OSP)

## REFERENCES