ある論文誌のための査読支援システム

千葉 滋

日本ソフトウェア科学会の論文誌でもあるコンピュータソフトウェア誌の査読・編集作業では、2017 年より編集作業を支援する web アプリケーションを利用している。これは当時、同誌の編集委員長であった著者が作成したものである。本稿は、国際会議用の既存の査読システムでは論文誌の査読支援には不十分であることを述べ、論文誌の査読支援のために開発した web アプリケーションの設計概要を示す。

1 はじめに

コンピュータ科学の分野では一流とされる国際会議での発表論文が高く評価される傾向がこれまであり、数多くの国際会議が開催されている。これにともない、国際会議で発表される論文を選ぶ査読過程を支援するソフトウェア・システムも無料・安価で使えるものを含めて広く普及している。一方で論文誌の査読を支援するソフトウェア・システムで、広く知られていて無料で使えるものは著者の知る限りない。

本稿は著者が開発した論文誌論文の査読を支援するソフトウェア・システムの設計概要を述べる。これは日本ソフトウェア科学会の論文誌であるコンピュータソフトウェア誌のためのもので、web アプリケーションとして実装されている。

開発したシステムに学術的・技術的新規性はないが、システムが採用するデータベースのスキーマを中心に設計概要を示すことは有用であると考える。これは国際会議論文の査読を支援する既存システムを論文誌論文の査読に流用することが、なぜ好まれないかを検討する中で考案されたものだからである。現在、コンピュータ科学の分野でも論文誌論文を評価する流れができつつあり、今後、論文誌論文の査読を支援するシステムの需要が高まる可能性がある。そのようなシステムを開発する上でも、本稿の内容が役に立つことが期待される。

本稿のシステムの設計にあたり、国際会議論文の査読システムと比して異なる点として重視したのは、(i) 各投稿論文につき原稿が複数ありえる、(ii) 査読の進捗が個別の論文ごとに異なる、(iii) 例外的なワークフローが起こりうる、である。これらの点に留意してシステムの設計をおこなった。

2 査読システム開発の動機

本章ではまず広く普及している国際会議用査読システムについて概観する。次に既存の国際会議用査読システムを用いて論文誌の査読管理をおこなう場合の問題点について述べる。

2.1 国際会議用査読システム

国際会議の査読を web アプリケーションによって支援するというアイデアは、World Wide Web の普及の初期から存在した。例えばプログラミング言語分野の国際会議である European Conf. on Object-Oriented Programming (ECOOP) のために開発された CyberChair [1][2] は 1996 年にオランダの Twente 大学で開発されている。CyberChair は

A Reviewing System for an Journal.

Shigeru Chiba, 東京大学 情報理工学系研究科, Grad. School of Information Science and Technology, The University of Tokyo.

その後、OOPSLA や PLDI、ASE、ICSE など、プログラミング言語やソフトウェア工学の分野の多くの主要国際会議の査読に用いられた。現在も広く使われている EasyChair [3] は 2002 年に国際会議 Int'l Conf. on Logic for Programming Artificial Intelligence and Reasoning (LPAR) と Int'l Conf. on Automated Deduction (CADE) のために開発された。HotCRP [4] は 2006 年に HotNets V workshop のために最初に開発された。

これらの国際会議用の査読システムはプログラム委員長の作業負担を下げることを主な目的として開発されている [1]。Web アプリケーションとして実装された査読システムが普及する前の 1990 年代前半は、著者の経験によれば、論文は複数部紙に印刷したものをプログラム委員長に郵送（国際郵便または DHL や Fedex のような国際宅配便）するものであった。プログラム委員長は、受け取った紙の論文を仕分けし、査読を担当するプログラム委員を決め、それぞれの元へ発送しなければならなかった。論文を電子的に送付する際に現在広く用いられている PDF 形式が発表されたのは 1993 年 [5] であり、1990 年代前半は PostScript 形式が用いられることがあったものの、紙媒体も広く使われていた。

このような時代状況を背景に開発された査読システムは、投稿論文の受領や査読者の選定、採否の決定を支援することに機能が集中している。システムの利用者は、プログラム委員長の他、論文の著者と査読者であるプログラム委員（以下、査読者）である。論文の著者はシステムにログインし、投稿する論文の原稿を提出する他、論文の表題や著者名など、論文のメタ情報を記録する。投稿した論文が採択された場合は、最終の印刷原稿を提出することもできる。査読者は、bidding と呼ばれる過程で、自身の専門分野と査読を希望する投稿論文、利益相反となる投稿論文をシステムに登録する。この bidding の支援は査読システムの重要な機能で、プログラム委員長は登録された情報を元に査読の割り当てを決定し、システムを通じて各査読者に通知する。査読者は期日までに各投稿論文の査読報告をシステムに提出し、プログラム委員会に出席する。プログラム委員長は、システムに記録された査読報告の情報を活用しながらプログラム委員会を開催し、各論文の採否を決定する。決定された採否はシステムに記録され、システムが査読報告とともに採否を論文の著者に通知する。

当時、ほとんどの国際会議のプログラム委員会では、全ての委員が実際に集まって採否の議論をおこなっていた。このため初期の査読システムである CyberChair は、当日の委員会での議論を円滑にするため、査読者による評価点順に並べた投稿論文の一覧など、様々な資料を生成する機能を備えていた。また、プログラム委員長による委員会の司会を支援するため、現在議論されている論文の諸情報を表示する機能、利益相反がある査読者の退室を促す機能なども備えていた。一方、近年ではオンラインで開催されるプログラム委員会が増えており、EasyChair や HotCRP のような査読システムは、査読者が査読報告を提出した後、互いの査読報告を見てオンライン上で議論をするための機能が充実している。査読者が自分の査読報告を提出するまでは他の査読報告を閲覧できないようにするなど、査読の進捗に連動した閲覧制限の機能も用意されている。また査読報告に対して著者が意見を述べる rebuttal を導入する国際会議もあり、オンライン上での議論の一部として rebuttal の意見を取り扱う仕組みを備えている。

投稿論文の受領や査読者の選定、採否の決定の支援は充実しているものの、国際会議用の査読システムによる論文査読の進捗管理は簡単なものにとどまっている。これは国際会議の場合、論文の投稿と査読があらかじめ決まった日程に沿って実施されるからと考えられる。論文投稿の締め切り日は決まっており、全ての論文が一斉にその日までに投稿される。通常、査読報告の締め切り日も決まっており、オンラインのプログラム委員会であれ、実際に集まるプログラム委員会であれ、決められた期間で採否の議論がおこなわれ、議論の結果は一斉に著者に返される。同一の国際会議に投稿された論文は、全て同じ日程で取り扱われるので、個々の投稿論文ごとの高度な進捗管理の機能は不要である。

さらに、国際会議用の査読システムは、投稿される論文の原稿は一つ、あるいは投稿原稿と最終の印

刷原稿の二つだけである前提で原則として設計されている。例えば HotCRP のデータベースのスキーマ定義 [6] を見ると、Paper テーブルには投稿原稿を表すと思われる属性 paperStorageId と、最終の印刷原稿を表すと思われる属性 finalPaperStorageId とがある。任意の個数の改訂原稿を属性としてテーブルに持つようには定義されていない。あらかじめ決まった数の版の原稿が提出される前提で設計されているといえる。国際会議は一般に投稿された論文を迅速に発表することが重視されるため、多くの場合、採録された論文はほぼ投稿時のままの原稿（あるいは著者が自主的に改訂した原稿）で出版される。このため、そのような設計でも問題はない。

2.2 既存システムの限界

国際会議用査読システムはいくつも存在し、EasyChair など比較的容易に利用が可能なものもある。しかしながら、国際会議と論文誌の査読手順の違いにより、国際会議用査読システムを論文誌論文の査読に流用しても、論文誌の編集委員長の仕事を十分に支援できない。

まず、典型的な国際会議用査読システムでは、各投稿論文につき原稿は一つ、あるいは投稿原稿と印刷原稿の二つだけである。このため、そのままでは論文誌の査読に流用できない。論文誌では、採録に至るまで何度か原稿が改訂され、そのたびに査読がおこなわれるため、一つの投稿論文につき初稿から最終印刷稿まで、複数の原稿を扱わなければならないからである。流用する場合は、一回目の査読で採録とならなかった論文は全ていったん不採録になったとし、査読結果に基づいて改訂され再投稿された原稿はまったく新しい異なる論文が投稿されたと扱う等の対応が必要になる。その場合、一回目の査読と二回目以降の査読の関連性がわかりにくくなる。そのような対応を取らない場合は、査読システムを改修して、例えば rebuttal の一種として論文の改訂原稿を提出できるようにし、査読報告もそれぞれの原稿に複数種類提出できるようにする必要がある。

また個別の論文の査読の進捗管理も国際会議用査読システムを流用する場合の問題点である。国際会議用査読システムにおける進捗管理は、主に各査読者が割り当てられた論文の査読を何割程度完了しているか、である。論文の採否を決定するプログラム委員会の日までに全ての査読報告を集めなければならないので、プログラム委員長は各査読者が割り当てられた全ての論文の査読を終えているかを確認し、必要なら督促をおこなわなければならない。国際会議用査読システムの進捗管理の機能は、査読者ごとの査読の進捗管理の機能が重視されている。逆に査読の進捗状況は論文ごとにほとんど変わらないので、論文ごとの査読の進捗管理の機能は重視されていない。

このような理由により、国際会議用査読システムとは別に、論文誌の査読には、それを支援することを主目的とするソフトウェア・システムが必要とされる。しかし、論文誌の査読の支援を主目的とするソフトウェアやサービスのうち、無料ないし安価に利用できるものは著者らが知る限り存在しない。そのようなソフトウェア・システムは新規に開発しなければならない。

3 論文誌向け査読支援システムの開発

日本ソフトウェア科学会の論文誌であるコンピュータソフトウェア誌に投稿された論文の査読管理を支援するために新たに web アプリケーション（以下、査読システム）を開発した。査読システムは論文誌の編集委員長の作業を支援するために開発された。本章ではこの査読システムの設計の経緯と、その技術的要点を述べる。

3.1 査読手順の明確化

査読システムの開発にあたって、最初に取り組まなければならなかったことは査読の規則とそのワークフローの明確化であった。ソフトウェア開発にあたっては、まず要求分析および要件定義が必要である。査読システムの場合、これは論文の査読手順の分析と、そこから導かれる支援内容の定義となる。

コンピュータソフトウェア誌に投稿された論文の査読手順は、文書による詳細な規定があるため、要求分析は当初容易であると思われた。しかし合わせて 5 つの文書によって細部にわたって手順が定められてい

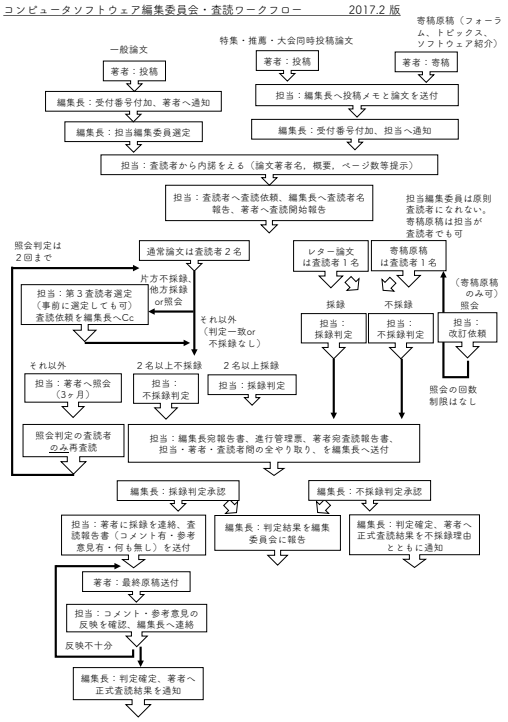

図 1　査読ワークフロー 2017 年 2 月改訂版

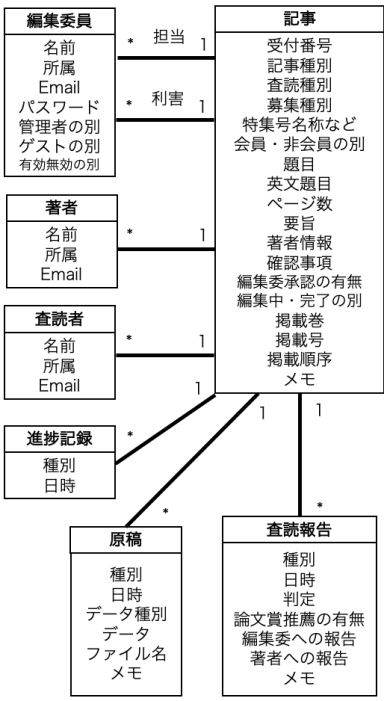

図 2　データベースのスキーマ

る一方、日本語文書による規定のため、一読しただけでは特に例外的な状況での取り扱い手順が自明ではないという問題があった。

そこでこれらの文書を読み込み、査読ワークフローとして図解することとした。作成したワークフローを図 1 に示す。この図は編集委員会で公式なワークフローとして承認され、編集委員が査読の手順を確認する役にも立っている。

図に示すよう査読のワークフローは比較的複雑である。この複雑さは既存の国際会議用査読システムを流用しようとしても上手くいかない原因でもあり、これを機に査読の手順の方を簡素化し、既存システムを流用しやすくする、あるいは新規開発する査読システムの開発を容易にする、ということも考えられた。しかしながら査読の手順は、論文誌の性格を決める重要な部分であり、十分な議論を経ずに安易に簡素化することはためらわれる。査読手順の見直しは実施しなかった。

3.2　データベース

査読システムが取り扱うデータは関係データベースに保存する。データベースのスキーマを図 2 に示す。論文誌の査読支援に適するするシステムとするため、一つの論文に対して原稿が複数存在すること、論文ごとに細かな進捗管理ができること、この 2 点に配慮したスキーマとなっている。なお本スキーマでは、論文の代わりに記事という名称を用いる。コンピュータソフトウェア誌に掲載されるのは論文に限らず、国際会議の参加報告なども掲載されるからである。

一口に査読の進捗管理といっても、進捗に関係する事象は様々である。各記事の進捗に関係する事象の全体像は、その記事に関連づけられた原稿、査読報告、進捗記録の 3 つのエンティティをまとめて時間順に整列することで得られる。例えば記事の投稿や改訂稿の提出は、その記事の進捗記録エンティティではなく、原稿エンティティの日時を調べないとわからないからである。

現実の査読はしばしば例外的なワークフローをとる。例えば、二重投稿の疑いが発生した場合、「採録

のためのコメント」の条件を満たすため著者と担当編集委員の間で複数回の原稿のやり取りがあった場合などである。そのような例外的な状況での事象も、その経過を記録し、必要な文書を合わせて記録できなければならない。

例外的な状況にも柔軟に対応できるよう、そのような状況で起きた事象も、原稿、査読報告、進捗記録の 3 つのエンティティを使って記録するように査読システムのスキーマは設計されている。査読に関係する事象を記録する際、合わせて記録される文書は pdf あるいは zip などのファイルであるか、テキストであるか、あるいはそのような文書が無いか、そのいずれかである。査読システムのスキーマは、その文書の種類ごとに、原稿、査読報告、進捗記録の 3 つのエンティティを対応させている。事象に合わせて記録する文書がファイルであれば原稿エンティティを用い、テキストであれば査読報告エンティティを用い、合わせて記録する文書が無いなら進捗記録エンティティを用いてその事象を記録する。このような場合に備え、それぞれのエンティティには「その他」という種別を選べるように設計されている。

スキーマの設計にあたっては、査読における様々なワークフローを想定して検討し、進捗に関係する事象が、合わせて記録される文書の種類で 3 つに分類できることを見いだした。この分類に対応するエンティティを用意したのが査読システムのスキーマの特徴である。同じ査読報告でも、担当編集委員への報告と編集委員長への報告では書式が少し異なるが同じエンティティを用いるのはこのためである。

このようなスキーマ設計により、例外的な状況にもある程度柔軟に対応可能になっている。データの形式によって 3 種類のエンティティを使い分けなければならないが、基本的には任意のデータを記事に関連づけて記録できるのが特徴である。記録されたデータの中身は、タグとして適切な種別を選ぶことで区別する。この設計の利点は、種別を増やすことでスキーマを容易に拡張できる点である。また利用者が「その他」を種別として選ぶことで、例外的なワークフローにおいても必要なデータを柔軟に査読システムに記録できる。標準的なワークフローに登場するデータの種類に過度に依存して、細かく多数のエンティティを定義してしまうと、標準的なワークフローに登場しないデータの記録に苦労することになる。このため、そのようなスキーマ設計は採用しなかった。

3.3 ユーザインタフェース (UI)

典型的な web アプリケーションは、背後にある関係データベースのテーブルを操作するインタフェースとして作成される。査読システムもその方針で作成されている。具体的には、記事エンティティを格納する記事テーブルと編集委員エンティティを格納する編集委員テーブルを操作し、そのレコードの新規作成、読み出し、更新を可能にするインタフェースを提供する。レコードの削除は実際には可能だが、使いやすいユーザインタフェースは提供していない。査読システムの役割の一つは査読の経緯の履歴を記録することであり、不用意に履歴を削除してしまうことを防止するためである。

3.3.1 記事の一覧

査読システムの利用の中心は、記事テーブルの操作である。記事テーブルの操作を通じて、各記事に関連づけられた原稿や査読報告、進捗記録エンティティの情報も得られる。査読システムは記事テーブルのレコードの一覧を表示するために、主に以下の 4 つのページを提供している。

- 記事一覧：編集中である記事を一覧表示する。採録の場合は記事を掲載した号が発刊されたとき、不採録の場合は編集委員会で査読結果が承認されたときに編集完了となる。編集完了前の記事は全て編集中となる。
- 担当記事一覧：現在ログイン中の編集委員が担当している記事だけを一覧表示する。
- 編集委員会承認待ち一覧：種別が「編集長への報告」である査読報告があり、編集中かつ編集委の承認が無しである記事の一覧を表示する。これは編集委員会で各記事の査読結果を承認する際に用いる。
- 掲載記事の一覧：指定された号に掲載される記事の一覧。掲載順を調整して、編集委員長が次に発刊する号の目次を作成するのに用いる。出版社

は、その号の掲載記事の原稿をこのページを介して受け取る。

3.3.2 進捗管理の支援

国際会議の査読と異なり、論文誌の査読の進捗は個々の記事ごとに管理しなければならない。このため記事を一覧表示するページで個々の記事の査読がどの段階に達しているかを細かく表示するだけでなく、編集委員会承認待ち段階の記事一覧や、指定の号に掲載することが決まった記事の一覧など、査読のワークフローにおける重要な段階にある論文をまとめて一覧表示するページも用意されている。編集委員長は、これらの一覧表示を活用することで、査読の進捗状況の全体像を把握する。例えば次に発刊される号に記事が十分集まっているかなどを簡単に知ることができる。

国際会議用の査読支援システムでも、例えばHotCRP は論文に自由にタグをつけられ、タグの有無を条件に絞り込んだ一覧表示ができる。このため、利用者が適切にタグをつければ、上で示した進捗管理の支援と同様のことが可能である。本稿で述べる査読システムの特徴は、そのようなタグに相当するものを記事エンティティの属性としてあらかじめ組み込んであり、進捗管理に有用な絞り込みをした一覧表示もあらかじめ用意している点である。査読のワークフローを分析し、それに適したシステムを設計しているといえる。

3.4 実装と運用

本稿で述べている査読システムは 2017 年 2 月の編集委員会から稼働を開始した。稼働直後こそ SQL クエリの誤りによる不具合が起きたが [7]、その後、本稿執筆時点まで大きな事故なく稼働している。

査読システムの開発が始まったのは 2016 年 8 月からである。開発にあたり開発言語には Ruby を、フレームワークに Ruby on Rails 5 を採用し、著者 1 名で開発をおこなった。

2020 年 8 月現在、約 4 年間の運用後、データベースに記録されている記事の数は 360、編集委員の数は 65（退任して無効になった委員も含む） である。査読報告の数は 811 である。原稿の数は 664 であり、このうちデータが大きい等の理由でデータベース外のファイルの形で保存されている原稿は 79 である。pg_dump コマンドで作成したデータベースの dump ファイルの大きさは、圧縮なし plain-text 形式の場合約 1.9 GB、圧縮ありのカスタム形式の場合約 1.1 GB である。データベース外のファイルの総量は約 2 GB である。これらは週に 1 回バックアップのために別のマシンに転送され保存される。

4 まとめ

本稿では、コンピュータソフトウェア誌の査読作業を支援するために著者が開発した web アプリケーションの設計概要を述べた。国際会議のプログラム委員長による論文査読の進捗管理ではなく、論文誌の編集委員長による論文査読の進捗管理を指向している点が特徴である。本稿では両者の進捗管理の違いについても述べた。

開発した査読システムは、様々なオープンソースソフトウェアを部品として用いて構築している。このため各部品の保守期間が終わると、査読システムの設計寿命も終わりを迎える。現在、実際に設計寿命は終わりを迎えつつあり、早急な再構築が必要である。本稿はその際の参考情報となることが期待される。

参 考 文 献

[1] Richard R. van de Stadt. Cyberchair – an online submission and reviewing system, or: A program chair's best friend. In *9th Int'l World Wide Web Conference (Poster)*, 2000.

[2] Richard van de Stadt. Cyberchair: A web-based groupware application to facilitate the paper reviewing process. `http://arxiv.org/abs/1206.1833`, 2012.

[3] Andrei Voronkov. Keynote talk: Easychair. In *Proc. of the 29th ACM/IEEE Int'l Conf. on Automated Software Engineering*, page 3–4. ACM, 2014.

[4] Eddie Kohler. Hotcrp conference management software. `https://read.seas.harvard.edu/~kohler/hotcrp`.

[5] Tim Bienz and Richard Cohn (Adobe Systems Inc.). *Portable Document Format Reference Manual*. Addison-Wesley, 1993.

[6] Eddie Kohler. `hotcrp/src/schema.sql`. `https://github.com/kohler/hotcrp/blob/master/src/schema.sql`.

[7] 千葉 滋. 坊主の不信心. **コンピュータソフトウェア**, 34(2):1, 2017.