

名古屋大学における課題採点支援 LMS 連携 Web ツールの開発

樽林 優¹⁾, 鈴木 直樹²⁾, 木迫 璃玖³⁾, 大平 茂輝⁴⁾, 戸田 智基⁴⁾

- 1) 名古屋大学 情報学部
- 2) 名古屋大学 大学院工学研究科
- 3) 名古屋大学 大学院情報学研究科
- 4) 名古屋大学 情報基盤センター

Development of a LMS Cooperative Web Tool for Supporting Assignment Grading in Nagoya University

Yu Kurebayashi¹⁾, Naoki Suzuki²⁾, Riku Kisako³⁾, Shigeki Ohira⁴⁾, Tomoki Toda⁴⁾

- 1) School of Informatics, Nagoya Univ.
- 2) Graduate School of Engineering, Nagoya Univ.
- 3) Graduate School of Informatics, Nagoya Univ.
- 4) Information Technology Center, Nagoya Univ.

概要

昨今の大学教育において、学修支援システム (LMS) の重要性は確固たるものとなっている。名古屋大学においても、Sakai をフォークした LMS の活用が進んでおり、その利便性の改善に向けて、様々な周辺サービスの開発が行われている。しかしながら、依然として利便性に欠ける点は残されており、特に、LMS 上で提出された課題の採点作業においては、要求される操作量が多いなどの問題がある。この問題の解決に向けて、我々は LMS と連携し、課題採点を支援する Web ツールの開発に着手した。本稿では、開発したツールについて紹介するとともに、今後の展望について述べる。

1 はじめに

昨今、学修支援システム (LMS) は大学教育においてなくてはならない存在となっている。実際、名古屋大学では Sakai をフォークした LMS (TACT) の活用が全学的に進んでおり、大部分の授業において、資料配布、課題提出、成績管理、教員学生間の連絡の手段として TACT が用いられている。一方で、操作性や利便性の面で十分とは言い難く、これまでに様々な周辺ツールの開発が行われている [1][2]。

しかしながら、依然として、教員にとっての利便性という側面で課題が残されており、特に、TACT 上で提出された課題の採点作業においては、提出物を表示するたびに別のタブを開く必要があり不便、提出者の切り替えのためにページをスクロールする必要があり

手間がかかるなどの問題がある。

課題の採点は教員にとって重要な日常業務の一つであるため、ツールの利便性の問題により業務負担が生じることは、教員にとって不利益になることはもちろん、組織にとっても、課題の採点その他の業務を圧迫するという点で、学生にとっても、採点に時間がかかることで、課題に対するフィードバックが遅れる要因になりうるという点で不利益である。

この問題に対して、Sakai (およびそのフォークである TACT) の課題ツールには、Sakai Grader という採点を支援する仕組みが提供されている。しかし、本学で利用している Sakai のバージョンにおいて動作が不安定であったこと、繰り返し行う操作に対して要求される動作が多かったことなどにより、問題を解決するには至らなかった。また、Sakai における課題の

閲覧を支援する外部ツール [3] は存在していたものの、LMS とのデータ連携を手動のダウンロード・アップロードによって行う必要がある、採点方式や遅延提出などの情報を活用できないなどの制約があり、改善の余地がある手法であった。

こうした背景のもと、TACT の LTI1.3 対応を契機とし、TACT と連携し、Web ブラウザでの操作で完結する課題採点支援ツール（以下採点支援ツールと呼称）の製作が企画され、学生が主体となって開発を行った。本稿では、採点業務の負担軽減手法の提案として、採点支援ツールの機能および技術的構成を紹介する。

2 採点支援ツールの機能

2.1 機能概要

採点支援ツールは、Sakai Grader をはじめとした従来のツールとの差別化を図るため、「大画面での利用を前提とした高機能なツール」、「繰り返し行われる操作を必要最低限の動作で実現すること」をコンセプトとして開発された。

基本的な機能として、提出者、提出物の一覧表示をサポートする。採点対象の提出者、提出物は一覧から選択することができ、切り替えが行いやすい仕様となっている。ここで選択された提出物はページ内に表示され、別のタブが開くことはない。選択された提出者についても、遅延提出の有無などの情報を含め、ページ内に表示される。また、課題の採点方式に従って適切な採点フォームが表示され、ページ内で採点を行うことができる。こうして採点された内容はページ内に設置されたボタンから LMS に反映することができる。

その他に実現された機能として、以下のようなものがある。

- 課題評価の分布の表示
- 一覧表示された提出者のフィルタリング
- 参考資料を表示しておくことができるクリップ
- 一定の基準に基づいた評価を可能にする評価項目
- 評価項目による採点を Sakai における評価やコメントに変換するテンプレート
- マウスによる直感的な操作とホットキーによる効率的な操作の両立

2.2 機能詳細

採点支援ツールには、存在する課題の一覧からなる「課題一覧ページ」と実際に課題の採点を行う「課題採点ページ」が存在する。課題一覧ページで課題を選択し、課題採点ページで選択した課題の採点を行うのが、採点支援ツールの基本的な利用の流れとなる。

図 1 は課題採点ページである。課題採点ページの左側には、下記の 2 つが含まれる。

- ① 提出者の絞り込みや並べ替えを行うためのフィルタ設定ボタン
- ② 選択可能な提出者と提出物の一覧

中央左側の領域はコンソールとメインビューであり、具体的にはそれぞれ下記の内容が含まれる。

- ③ 「課題一覧ページへ遷移する」、「課題の評価分布を確認する」、「LMS 側にコメント、評価、再提出許可に関する情報（フィードバック情報）を反映する」などの、特定の提出者によらない機能を集められたコンソール
- ④ 提出物一覧で選択された提出物を表示するメインビュー

中央右側の領域はクリップビューと評価項目パネルであり、それぞれ下記の内容が表示される。

- ⑤ 参考資料を表示するクリップビューおよびその操作盤
- ⑥ 一定の基準にしたがって課題を採点するための評価項目

右側には、採点時に入力するフォームを含む以下の内容が並んでいる。

- ⑦ 選択中の提出者の遅延提出の有無などを含む情報
- ⑧ フィードバック情報の確認と、直接編集ができるフォーム
- ⑨ 評価項目によって算出された採点支援ツール独自の点数を、Sakai における評価に変換する「テンプレート」の設定ボタン

なお、フィルタ設定画面やテンプレート設定画面の開閉、提出者や提出物の選択、評価項目の作成は、デフォルトでマウス操作が可能となっているが、キーボ

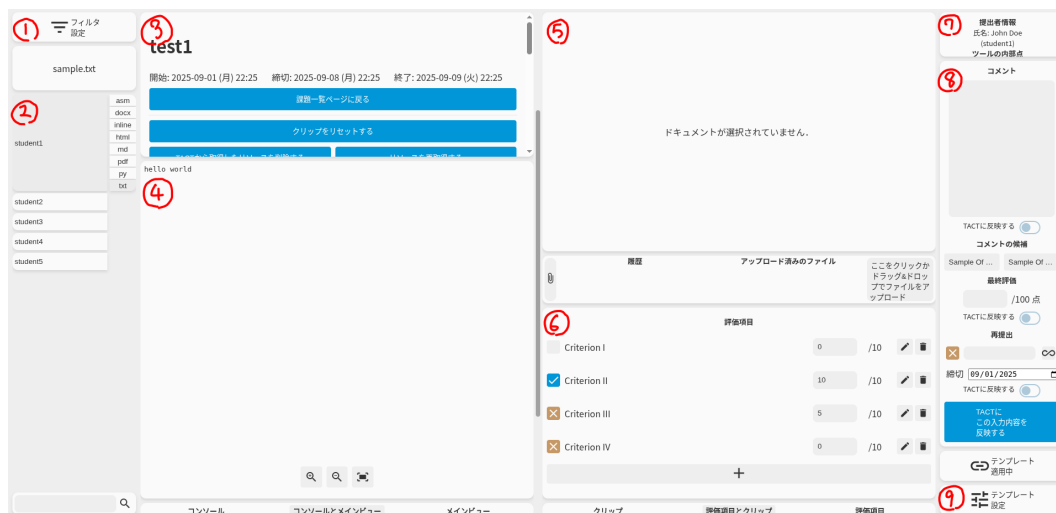


図1 採点支援ツールの課題採点画面

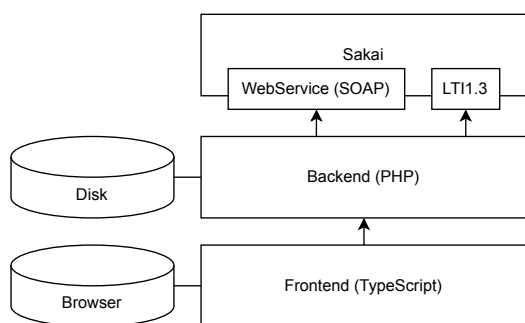


図2 採点支援ツールの概略. 実線の矢印は依存の方向を示す.

ードを用いて操作することもできる.

3 採点支援ツールの技術的構成

3.1 全体概略

図 2は採点支援ツールの構成の概略を示したものである. まず, フロントエンドとバックエンドは明確に分離している. それぞれ使用している言語が異なり, バックエンドでは PHP, フロントエンドでは TypeScript を採用している.

採点支援ツールの開発にあたって, TACT の WebService が提供する SOAP API や LTI1.3 実装には一部独自改修が加えられている.

バックエンドは, 以下の実現を主要な目的としている.

- 採点支援ツールを LTI ツールとして利用可能にすること
- LTI を利用して SSO をサポートすること
- Sakai の WebService と LTI をラップしてフロントエンドに提供すること

バックエンドで利用するフレームワークとしては, Laravel を採用している. 既存のプロジェクト [1] との一貫性, およびインフラの再利用のためである.

フロントエンドでは Laravel の機能である blade テンプレートはほとんど用いられおらず, TypeScript のライブラリ (SolidJS) を用いて DOM が組み立てられている. もともとは, blade テンプレートを活用してサーバサイドで HTML が生成されていたが,

- 要素のイベントを監視したり, 更新したりするスクリプトと要素との対応を, id 属性などをリテラルで指定して取ることになるため, 対応の確認方法が煩雑
- HTML+CSS と blade テンプレートを利用したコンポーネントの共通化では, 利用側のポイラプレートが多く煩雑
- 動的に DOM 要素を追加したり削除したりする場合, スクリプトとの紐付けも動的に行う必要があり煩雑

といった問題のため, 追加や変更のコストが高くなってしまっていた. JSX 記法がこうした問題を解消する

と考え、JSX 記法をサポートする SolidJS によるリブレイスが行われた。

また、サーバリソースが十分でなく、可能な限りユーザ情報をサーバに蓄積しないというポリシーに基づき、フロントエンド側がリッチな構成になっている。

3.2 バックエンドの構成

バックエンドのコンポーネント構成は、Laravel における標準的な構成と大きく離れるものではない。特筆すべき点としては、以下の 2 点が挙げられる。

- LTI1.3 と Webservice の SOAP API のそれぞれに対してラッパークラスがあり、更にその 2 つのラッパークラスを合成したクラスを定義
- ディスクアクセスにあたって、ディレクトリ構造を PHP のコードによって規定したクラスを利用

外部サービスとの境界にラッパークラスを設けることには、アクセスのためのコードを共通化できることはもちろん、仕様変更への追従の容易化による頑健性の向上、今後の使用状況検証の容易化など、様々な利点がある。特に、Webservice が提供する SOAP API の使用に際して、WSDL 文書で形式的に説明されているものの、詳細な使用方法が一見して分からない問題があった。自己文書化されたラッパークラスを設けることは、この問題を解決し、後継の開発者の動作理解を容易にする。

ディレクトリ構造を PHP のコードによって規定したことは、リテラルの誤字による誤動作、ディレクトリ構成の不整合による不具合、ファイルの非存在による誤動作、名前の衝突といったリスクを集約し、対処を容易にする。

3.3 フロントエンドの構成

図 3 は、フロントエンドの構成の概略図である。

採点支援ツールにおける最もコアな部分は Service と Model であり、これらがドメインの知識を表現する。すなわち、課題の採点に特有なデータや知識、処理がここに定義されている。Service は基本的にシングルトンであって、「処理」をおこなう。一方、Model は基本的にイミュータブルなオブジェクトであり、「データ構造」の表現を担う。

Util と Model にはプロジェクトのあらゆるところから依存されるようなデータ構造や機能が定義され

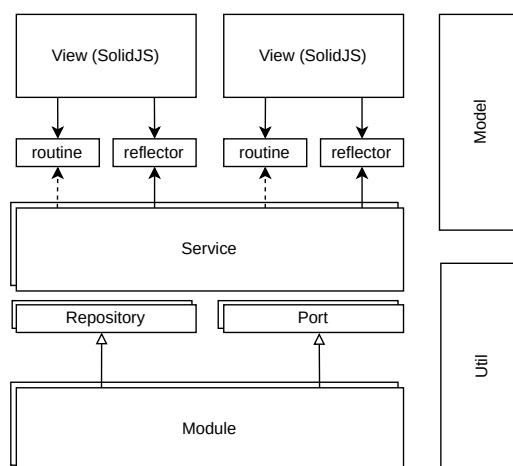


図3 今回開発された課題採点支援ツールのフロントエンド構成の概略図

ている。Util にはより一般的な機能やオブジェクトが定義され、Model にはドメインに特有な、すなわち Sakai における課題採点に特有なデータ構造や、操作上の制約が定義されている。

Module は Service が公開するインタフェース (Repository や Port) を実装する。これによって依存性を逆転し、Service から外部への依存をなくし、ソフトウェアの頑健性を高める。

View は見た目を定義するものである。

Routine は View からクリックなどのイベントの通知を受け、登録された Service を呼び出すことで、イベントドリブンなデータ処理を実現する。Routine は DI コンテナのようなもので、View で Service のインスタンスをバケツリレーする煩わしさを避けるために導入されている。

一方、Reflector は表示すべきデータが更新されたとき、View に更新を要請することで、データドリブンな View 更新を実現する。View と Service を直接つなぐのではなく、間に Reflector を噛ませているのは、View とコアな部分 (Service) との結合を疎に保ち、View (SolidJS) を捨てやすくするためである。

3.4 フロントエンドでの使用技術

フロントエンドでは、View の実現のために主に SolidJS が用いられているが、これ以外にも以下に列挙したライブラリ群などが用いられている。

- PDF.js: PDF をブラウザで表示
- chart.js: 評価分布をグラフとして表示
- marked: Markdown を HTML に変換
- mammoth: Word 文書を HTML に変換
- highlight.js: ソースコードをハイライト

3.5 技術的な工夫点

3.5.1 Changable 値の導入

図 4は、採点支援ツールにおいて最も複雑な機能の一つとして、採点支援ツールの提出物情報を反映する機能をアクティビティ図として表現したものである。SubmissionService がツールの情報が紐付けられた提出物情報のリストを提供するには、このアクティビティを実行する必要がある。しかしながら、フローの開始地点が複数存在する上、多数のシグナルにも対応しなければならず、通常の関数呼び出しによる制御ではこれを実現するのは至難である。

イベントリスナを登録すれば実現できそうだが、その方針ではボイラプレートが多く、処理の本筋が曖昧となり可読性が下がる。さらに、依存元で変更の監視が必要な場合と、即座に値が必要な場合で、依存先の作りを変える必要があり、最悪の場合にはその両方が必要なため、本質的には同じ処理を2つの異なる方法で実現するという状態になる。

Changable<T> は、後に変更の可能性がある T 型の値を意味する。関数の返り値を Changable<T> とすることで、通常関数呼び出しのような形でありながら、呼び出し先が呼び出し元の変更を駆動することが可能となる。

また、Changable<T> から T に変換するにあたって、即時に値を取得する方法と、コールバックを受け取って随時監視する方法を提供することで、呼び出し元が値の扱いを選択できるようになる。これによって、呼び出し先は呼び出し元の影響を受けずに実装を一本化することができる。

3.5.2 エンティティオブジェクト用データ構造の用意

採点支援ツールでは、原則としてすべてのドメインオブジェクトをイミュータブルとして扱っている。この方針には、予期せぬバグの発生を防ぐ、SolidJS の変更検知の仕組みと相性がよい、というメリットがあるものの、同一エンティティを表現するオブジェクトが複数発生してしまい、管理が困難になる問題があった。

そこで、エンティティオブジェクトを格納するデータ構造を用意した。連想配列のように、id を指定してオブジェクトを取得することができる。連想配列との決定的な違いは、該当エンティティが存在しない場合は格納されるまで待つ、という動作をサポートすることにある。

エンティティを利用する側は本データ構造からエンティティを取得し、エンティティを用意する側は初期化が終わり次第データ構造に追加していくという運用によって、利用側は初期化が完了するまで自然に待機できる。また、エンティティを更新する際は、データ構造から該当エンティティを取り出してから変更を加えて再び戻す。該当エンティティが存在せず、取り出せない場合は、取り出せるようになるまで待機するという動作によって、自然な排他制御が可能となる。

4 採点支援ツールの利用方法

採点支援ツールを利用するには、事前に TACT 管理者が LTI1.3 に基づく外部ツールとして本ツールを登録しておく必要がある。各サイトの Instructor ロールのユーザ（多くの場合、教職員）が、サイトの外部ツールとして本ツールを有効化することにより利用可能となる。

本ツールが有効になっていると、図 5のように、TACT のツール一覧に採点支援ツールが表示される。採点支援ツールをクリックすると、図 6のような課題一覧ページが表示される。課題一覧ページには、採点支援ツールを起動したサイトに存在する課題が一覧で表示される。ここで、課題名部分をクリックすると、該当課題の課題採点ページ (図 1) に遷移し、採点を開始できる。

採点支援ツールは、提出物をツール上に表示するため、課題採点ページに遷移した時点で TACT から該当課題に対する提出物をすべて取得する。取得されたデータは一時的にバックエンドに保存されるが、一定時間利用されないと自動的に破棄される。

課題採点ページでは、左に表示される提出物一覧で提出者と提出物を選択できる。右側のフォームでは、選択された提出者に対してコメント、成績、再提出許可の有無を設定できる。このように設定した内容は、「TACT にこの入力内容を反映する」ボタンや、コン

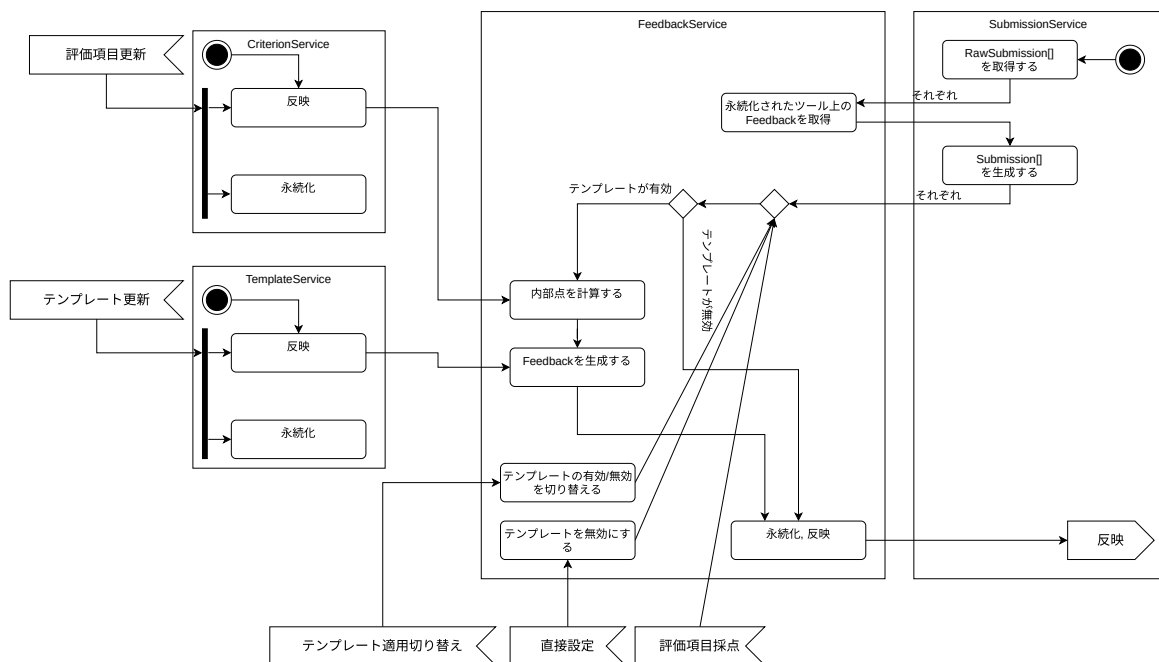


図4 採点支援ツールの提出物情報を反映する機能のアクティビティ図



図5 TACT上のツール一覧。採点支援ツールは「課題採点支援」というツール名で登録されている。

ソールの「採点結果をまとめて TACT に反映する...」ボタンを用いて TACT に反映させることができる。

採点情報は採点者のブラウザに保存され、次回以降ツールを起動した際には前回の状態が引き継がれる。

また、ブラウザ上のデータを消去する際や不慮の事態で採点中のデータが消えてしまった場合に備えて、バックアップ機能が実装されている。コンソール下部には、「採点結果を TACT ホームに一時保存する」、「保存された採点状況を復元する」というボタンがあり、「採点結果を TACT ホームに一時保存する」ボタンを押すと、採点状態がツール起動者（教員）個人の TACT のリソースツールに保存される。「保存された採点状況を復元する」を押すと、前述の操作によって保存された採点状態を復元する。

5 今後の展望

現在、さらなる機能実装として次の事項が検討されている。

- 評価項目などの設定の CSV/JSON 等による入出力機能
- PDF.js の書き込み機能やキーボード操作などのサポート
- 返却ファイルの添付
- キーボード操作の見直し、拡大
- コメント、成績、再提出許可の一括編集



図6 採点支援ツールを起動して最初に表示される課題一覧ページ

こうした機能の実装によって、利用者や利用シーンが拡大する可能性がある。一方で、採点支援ツールは、未だ十分な動作検証を受けていない。今後は、徐々に利用者を増やしながら、フィードバックを活かしてツールの改善に努める必要がある。安定的な運用に移行するためにも、柔軟な開発体制を構築することが重要な課題と考えられる。

参考文献

- [1] 澤田桂都, 山本実央, 木迫璃玖, 大平茂輝, 後藤明史, 戸田智基, "名古屋大学における学生の利便性向上を目指した LMS 連携システムの開発", 大学 ICT 推進協議会 2022 年度年次大会論文集 (オンライン), 15AM1B-1, pp. 427-432, 2022.
- [2] 大平茂輝, 清谷峻也, 伊藤瑠哉, 岡本康佑, 谷川右京, 出口大輔, 戸田智基, "LMS 経由で手書きレポートを返却する Web サービス「かみレポ」の開発・評価", 情報処理学会論文誌：教育とコンピュータ, vol. 6, no. 1, pp. 52-68, 2020.
- [3] 青木学聡, 喜多一, "利用者側での CMS の運用改善 - Sakai における課題ツールを例に", 大学 ICT 推進協議会 2016 年度年次大会論文集 (オンライン), TP27, 4 pages, 2016.