

ETL ツールを活用した Open LRW への Caliper 標準形式によるデータ集積の拡張の試み

中野 裕司^{1,2,3,4,5,a)} 喜多 敏博^{3,2,1} 戸田 真志^{1,2,3,4,5} 久保田 真一郎^{1,3,2,4} 右田 雅裕^{1,3,4,2}
杉谷 賢一^{1,4,3,2}

概要：ETL(Extract Transform Load) ツールはデータの変換に有用なツールで、データベース、WebAPI、SSH、ファイル等様々なデータソースから入ったデータを変換・統合・加工し、他のデータベースへ書き込んだり、他の WebAPI で提供したり、様々なデータの変換・融合が可能である。その機能を活用し、Moodle の IMS Caliper 対応プラグインの Caliper log store から REST で取得した Caliper 標準形式のデータに、他のデータベースから取得した学生番号、科目コード等の情報を extensions として追加し、Open LRW に REST で保存、蓄積した。実現に関する注意点、有利な点、問題点等について報告する。

1. はじめに

近年、Learning Analytics (LA) が学習支援や IR (Institutional Research) において注目を集めており、学生のドロップアウトの早期発見や学習状況の詳細な把握、データに基づくフィードバック等に応用され、今後さらに発展していくと思われる [1, 2]。

大学における学習支援システムは LMS が中心であり、そこに残されている学習ログは LA にとっても中心的存在であるが、それ以外の e ポートフォリオ等の各種システムのアクセス状況データも総合的に扱う必要がある。そのためには学習データの標準化とシステム連携が重用となると考える。

標準化に関して、オープンソース LMS である Moodle の学習履歴を、その国際標準の 1 つである IMS Caliper [3] 形式による蓄積を、Caliper log store プラグイン [4] と OpenLRW [5] を用いて実装し、実運用している Moodle

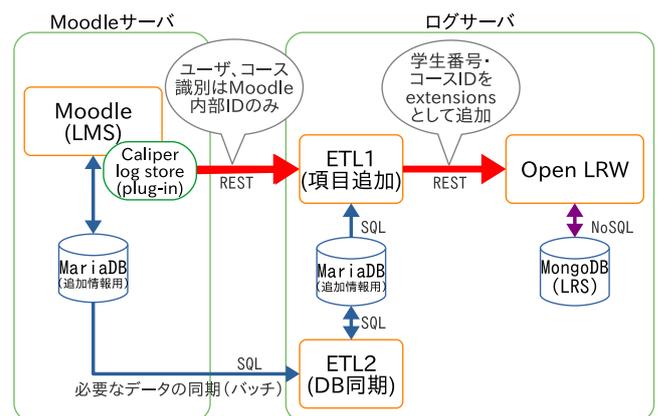


図 1 システムの概要

サーバでテストした結果を報告した [6]。本研究では、その際問題となった、記録データが Moodle の内部 ID で記録されており実際の学生番号や科目コード等が記録されていないことを解決するために、ETL(Extract Transform Load) ツール [7] を用いたシステム連携に関して報告する。

ETL ツールとは、種々のシステムからデータを抽出 (Extract) し、必要に応じて変換 (Transform) し、利用先システムで読み込む (Load) ためのミドルウェアである [7]。その用途は、各種システム連携 [7, 8] にとどまらず、データアナリティクスにおけるデータの前処理 [9] 等にも広く活用されている。大学のシステム連携にも応用されており、シェルスクリプト等を利用したデータ同期と比較して、属人性を減らすことができる [8]。

本研究で構築したシステムの概要を図 1 に示す。ETL ツールにより 2 つのアプリケーション ETL1, ETL2 を

¹ 熊本大学 総合情報統括センター
Center for Management of Information Technologies, Kumamoto University, Kumamoto 860-8555, Japan
² 熊本大学 大学院社会文化科学教育部 教授システム学専攻
Graduate School of Instructional Systems, Kumamoto University, Kumamoto 860-8555, Japan
³ 熊本大学 教授システム学研究センター
Research Center for Instructional Systems, Kumamoto University, Kumamoto 860-8555, Japan
⁴ 熊本大学 大学院自然科学教育部 情報電気工学専攻
Department of Computer Science and Electrical Engineering, Graduate School of Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan
⁵ 熊本大学 大学学情分析室
a) nakano@cc.kumamoto-u.ac.jp

開発した。ETL2 は、夜間バッチで、Moodle の内部データベース (MariaDB) から変換に必要なデータのみをログサーバ上のローカルデータベース (MariaDB) に同期している。ETL1 は、Moodle のプラグイン Caliper log store [4] から OpenLRW に成り代わって Caliper データを WebAPI(REST) で受け、その JSON データ内にある Moodle の学生や科目を識別する内部コードから、ローカル「データベース参照により実際の学生番号や科目コードを導出し、Caliper 規格の extension として追加し、OpenLRW の WebAPI に渡すことで、OpenLRW は本来の学生番号、科目コード等を追加された形で Caliper データを蓄積することができる。

2. 動作環境

ここでは、利用した Moodle サーバと ETL ツール及び OpenLRW を導入したログサーバのスペック等を示す。Caliper log store と OpenLRW 導入時と既存部分も弱冠異なるが、詳細は [6] を参照して欲しい。

2.1 Moodle サーバ

- CentOS Version 7.6
- Moodle Version 3.4.7
- Caliper log store 3.2.3
 - Event Store URL:
http(s)://ログサーバ名:9965/key/caliper
 - API key :
db.mongoOrg.find().pretty() で得られた apiKey
 - Send statements immediately to event store? : Yes
 - Batch size : 200

2.2 ログサーバ

ここでは ETL ツールとして、オープンソースで利用できる Talend Open Studio [8, 10] の ESB 版を使っており、GUI エディタ上で処理ブロックの組み合わせで機能を確認しながら Java アプリケーションを構築できる。

- CentOS Version 7.5
- ETL ツール: Talend Open Studio ESB Version 7.1.1
- OpenLRW Version
- MongoDB Version 3.6.3

3. システム構築と動作検証

3.1 Moodle データベース部分同期 (ETL2)

Moodle の内部 ID から学生番号や科目コードを導出するために、Moodle のデータベースの一部データが必要になるため、図 1 の ETL2 にあたるアプリケーションを ETL ツールで開発した。

コースメタリンクを使用していない場合は、基本的に Moodle の mdl_user テーブルで学生番号等 user 情報を、

```
select a.id userid , a.username , a.firstname ,
       a.lastname , x.id courseid ,
       y.fullname , y.shortname , y.idnumber
from mdl_user_sub a ,
( select customint1 as id from mdl_enrol_sub b
  left join mdl_context_sub c
    on b.customint1=c.instanceid
  left join mdl_role_assignments_sub d
    on c.id=d.contextid
  where
    b.courseid={cid} and c.contextlevel=50
    and d.modifierid=2 and d.userid={uid}
  union
  select id from mdl_course_sub where id={cid}
) x
left join mdl_course_sub y on y.id = x.id
where a.id={uid}
```

図 2 学生番号、科目コード等を導出する SQL。(ただし、{uid}、{cid} は、各々、Moodle 内部のユーザ ID、科目 ID)

mdl_course テーブルで科目コード等の科目情報を取得できる (もちろん登録してあればであるが)。ただし、一つの科目を複数科目として扱うことのできるコースメタリンクを使用していると、どの子科目にユーザが所属しているかによって科目名や科目コード等の情報が変化する。この機能は場合によっては非常に便利なこともあり、今回対象としている本学の Moodle サーバではかなり使われている。

そこで、今回はコースメタリンクに対応することとした。そのためには、Moodle の mdl_enrol、mdl_context、mdl_role_assignments の各テーブルの一部が必要になる。よって、Moodle から一部カラムを同期する必要のあるテーブルは 5 つで、そのテーブルと必要なカラムを表 1 に示す。

また、これらのテーブルから、学生番号や科目コード等の情報を取り出す SQL を図 2 に示す。コースメタリンクを使用する場合とそうでない場合では導出方法が異なるため、union 結合を用いている。caliper の extensions への追加の際の基本となる SQL コードではあるが、詳細は後述する。

Moodle のデータベースとログサーバのデータベースを同期するという事は、Moodle のデータベースからデータを読み込み、ログサーバのデータベースを更新することになる。このプロセスを ETL ツールの Talend で行うには、tDBInput モジュールと tDBOutput モジュールを Flow(row) で接続するだけで実現できる。これを、図 3 に示すように、5 つ作成することで全てのデータベースの同期をとるアプリケーションを作成できる [10]。このままだと、1 個 1 個順番にテーブルを同期することになるが、並列で処理することも可能である。今回は、深夜に一度ではあるが、動作中の Moodle サーバへの負荷を考え、並列にはしなかった。なお、tLogRow モジュールは、途中のデータを確認するために設置しているが、図のように無効化する

表 1 Moodle と同期するテーブル

Moodle 上のテーブル名	ログサーバ上のテーブル名	同期するカラム名
mdl_user	mdl_user_sub	username, firstname, lastname
mdl_course	mdl_course_sub	id, fullname, shortname, idnumber
mdl_enrol	mdl_enrol_sub	id, courseid, customint1
mdl_context	mdl_context_sub	id, contextlevel, instanceid
mdl_role_assignments	mdl_role_assignments_sub	id, contextid, userid, modifierid

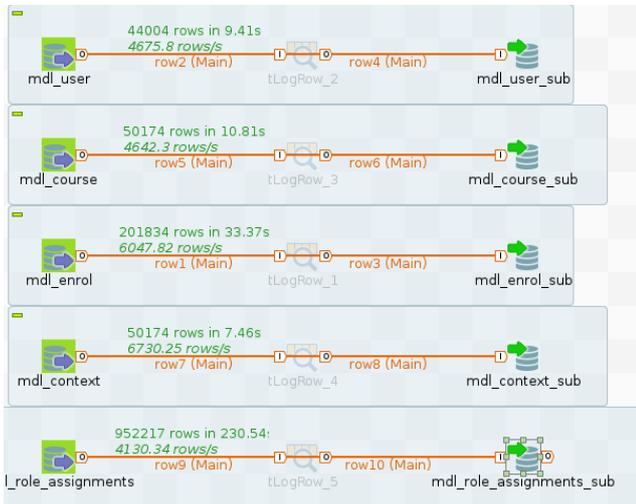


図 3 ETL2(Moodle のデータベースの一部を同期する)のブロック編集画面

ることも可能である。

3.2 WebAPI の中継と extension の追加 (ETL1)

Moodle の Caliper log store から REST でデータ受信し、対応する学生番号等データを追加して Open LRW に REST でデータを渡すために、図 1 の ETL1 にあたるアプリケーションを ETL ツール Talend で開発した。Talend 上のブロックダイアグラムを図 4 に示す。

tLogRow モジュールは途中のデータを確認するために設置しているものなので動作には無関係だが、それを除いても結構な処理ブロックの連結となった。主な処理ブロックの役割を図 5 に示す。

Talend の処理ブロックの接続方法には数通りあるが、データを渡したい場合、スキーマとよばれるデータ構造を渡すことができる flow 接続 (row) を用いる。tExtractJSONFields ブロックでは、指定従って JSON の各オブジェクトをスキーマに変換し、次のブロックに渡す。変換設定の様子を図 6 に示す。

読み込んだ JSON データには、Moodle のユーザ、科目を表す内部 ID が含まれ、前のステップでデータとして利用可能となった。ここで、前節で示した ETL2 で同期したローカルデータベースにその ID を問い合わせ、学生番号や科目コード等を取得したいのであるが、データベースからデータを読み込むブロック tDBInput は flow 接続ができないようであり、このままでは、REST で取得した

Moodle 内部 ID を用いたデータベースクエリができない。そこで、一旦、tFlowToIterate ブロックを用いて、それまでのスキーマとして保存されていたデータを一旦、global 変数に保存した上で、ローカルデータベースへのクエリを行った。

少しトリッキーではあるが、図 7 に示すような SQL を用いることで、学生番号等の問合せを行うと同時に global 変数に保存された変数もスキーマにコピーすることができた。

“select” 以下、最初の 3 行は、global 変数をスキーマに保存するため、その後の 3 行は、新たに問合せで得た学生番号等の情報で、最後の方は、SQL の where 句に、Moodle の内部 ID を使っているところである。ここも弱冠問題があり、REST で送られてきた「https://{Moodle の FQDN}/user/XXXXXX」の「XXXXXX」を取り出すために MySQL の substr と instr コマンドを用いている。最後の limit 1 は、コースメタリンクを利用した場合、学生には 1 つの科目だが、担当教員には複数の科目が表示されてしまうため、今回は、学生番号等も同時に取得しているため、表示科目コードを 1 つに絞っている。

ここで得られた学生番号や科目コードを extensions として加えて、Caliper 形式の JSON テキストに再構成するのが、tWriteJSONField ブロックで、その設定の一部を図 8 に示す。

最後の tReplace ブロックは、上記 tWriteJSONField ブロックは「@」の含まれる変数名が使えないため、代わりに「_AT_」としておき、このブロックで「@」に戻している。

以上のように extensions に学生番号や科目コード等の付加情報を加えた JSON テキストを、最後の tRESTClient ブロックで、OpenLRW の 9966 ポートに REST 形式で送っている。

4. 動作検証と考察

本システムで extension に学生番号や科目コード等を extensions に追加した例の一部を図 9 に示す。この JSON を OpenLRW に渡したものを MongoDB から読み出すと、extensions が加わった形で蓄積されていることも確認した (図は省略)。

ただし、実行速度等に関してはまだ測定しておらず、今後の課題としたい。

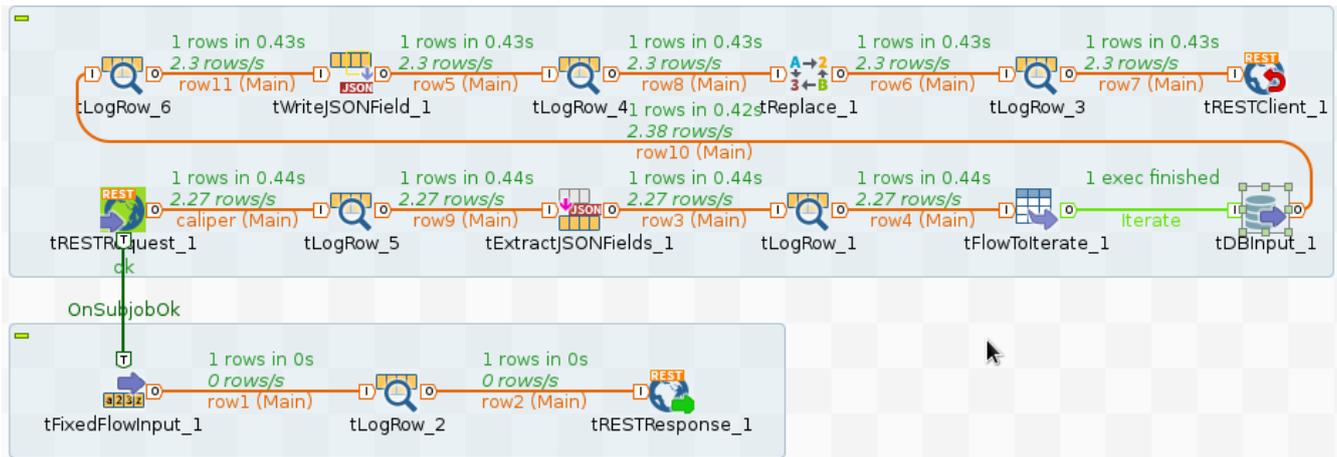


図 4 ETL1WebAPI の中継と extension の追加する) の Talend 上のブロック編集画面

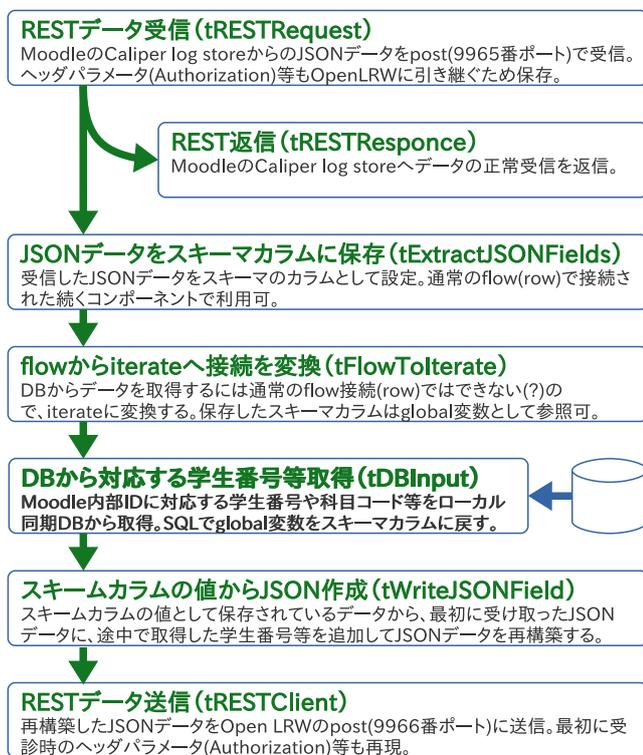


図 5 ETL1(Moodle の Caliper log store から REST でデータ受信し、対応する学生番号等データを追加して Open LRW に REST でデータを渡す) 仕組み

ETL ツール Talend を使ってみた印象であるが、WebAPI の中継を行い、その途中で他のデータベースに問合せ得た項目を追加するといった処理を、プログラムのソースコードを全く書かずに実現できるところは十分評価できると思う。ただし、初めて利用したことも原因として考えられるだろうが、ツール自体の使い方がわからなかったり、どのような処理ブロックがあるのか、どのような組み合わせでできるのか等、それなりに悩む時間も多、場合によってはプログラムを書いたほうが早いかもしれないこともあった。しかし、ブロックダイアグラムによる処理の見える化、明確化、ソースコードをなくすことによる属人性

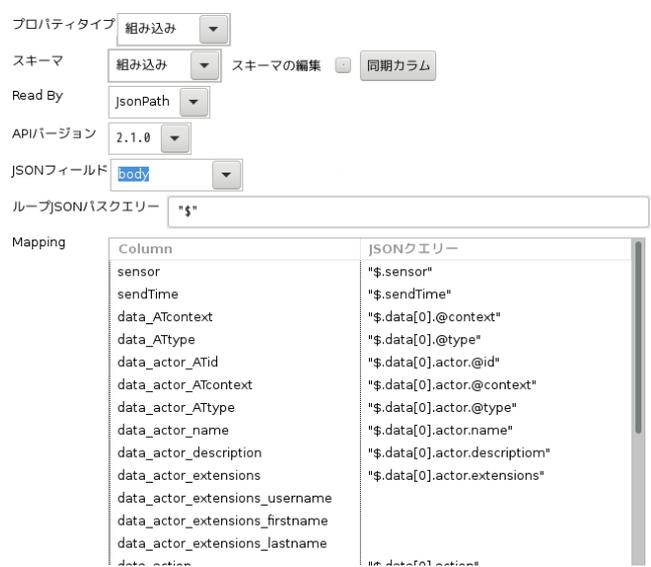


図 6 tExtractFields の変換設定の一部

を減らす効果等を総合的に考えると、かなり効果的なツールであるといえるのかもしれない。

また、一つの目的のために設計、構築されたシステムではないので、修正や変更が容易な点もあげられる。例えば、今回のシステムの入力部分を REST ではなく、別の LMS や e ポートフォリオのデータベースからの入力に変更するだけで、それらのシステムの学習履歴を Caliper に準拠した形式で蓄積できそうである。このような柔軟性があるところも利点であろう。今後、他のシステムの Caliper 形式によるデータ蓄積にも挑戦したい。

謝辞

本研究は JSPS 科研費 15H02795 の助成を受けたものです。また、ETL ツールのアイデアをいただいた京都工芸繊維大学情報科学センターの永井孝幸先生に深く感謝します。

```

" select "
+"'" +(String)globalMap.get("row4.sensor")+"'" as sensor,"
+"'" +(String)globalMap.get("row4.sendTime")+"'" as sendTime,"
+"'" +(String)globalMap.get("row4.data_ATcontext")+"'" as data_ATcontext,"
.....
+"a.username as data_actor_extensions_username,"
+"a.firstname as data_actor_extensions_firstname,"
+"a.lastname as data_actor_extensions_lastname,"
.....
+" where a.id=substr("'" +(String)globalMap.get("row4.data_actor_ATid")+"'",instr("'" +(String)globalMap
.get("row4.data_actor_ATid")+"'",'user/')+5,char_length("'" +(String)globalMap.get("row4.
data_actor_ATid")+"'))"
+" limit 1"

```

図 7 tDBInput ブロックの SQL(一部)

The screenshot shows the 'tExtractFields' tool interface. On the left, a 'スキーマリスト' (Schema List) contains various fields like 'sensor', 'sendTime', 'data_ATcontext', etc. On the right, an 'XMLツリー' (XML Tree) shows a hierarchical structure with nodes like 'rootTag', 'sensor', 'data', '@class', 'element', '@class', and 'actor'. A table on the right maps these fields to '関連するカラム' (Related Columns), 'ノードステータス' (Node Status), and 'スタティック値' (Static Value). For example, 'sensor' maps to 'sensor' column, 'ノードステータス' is '-', and 'スタティック値' is 'array'. 'data_actor_ATid' maps to 'data_actor_ATid' column, 'ノードステータス' is '-', and 'スタティック値' is 'object'.

図 8 tExtractFields の変換設定の一部

参考文献

[1] N Sclater, A Peasgood, J Mullan : Learning analytics in higher education, Jisc. Accessed February, 2016.

[2] 緒方 広明, 殷 成久, 毛利 考佑, 大井 京, 島田 敬士, 大久保 文哉, 山田 政寛, 小島 健太郎 : 教育ビッグデータの活用に向けた学習ログの蓄積と分析, 教育システム情報学会誌, vol. 33, no. 2, pp. 58—66 (2016).

[3] IMS Caliper : <https://www.imsglobal.org/activity/caliper> (2019年2月確認)

[4] Caliper log store : https://moodle.org/plugins/logstore_caliper (2019年2月確認)

[5] OpenLRW : <https://github.com/Apereio-Learning-Analytics-Initiative/OpenLRW> (2019年2月確認)

[6] 中野 裕司, 榎原 竜之輔, 喜多 敏博, 戸田 真志, 久保田 真一郎, 右田 雅裕, 杉谷 賢一 : Caliper log store と OpenLRW を用いた Moodle 上の学習履歴のログサーバへの Caliper 標準形式による集積の取り組み. 情報処理学会研究会報告, Vol.2018-CLE-25, No.11, 北海道大学, pp.1-5 (2018).

[7] 田中 哲雄, 湯本 真樹, 齋 礼 : 企業情報システムにおける連携技術, 電気学会論文誌C, Vol. 124, No. 5, pp. 1051-1057 (2004).

[8] 永井 孝幸, 山岡 裕美, 榎田 秀夫 : , 京都工芸繊維大学における利用者原簿管理基盤の強化と連携サービスの構築, 情報処理学会研究会報告, Vol.2018-CLE-25, No.9, 北海道大学, pp.1-8 (2018).

[9] 浅原 彰規, 森田 秀和, 海野 英一郎, 池本 理, 小野 寛太 : 非順序実行型データベースにより多次元配列データを管理するマテリアルズ・インフォマティクス・プラットフォームの開発, 情報処理学会 デジタルプラクティス, Vol. 10, No. 1, pp. 224-243 (2019).

[10] Talend, Inc. : Talend Components Reference Guide 7.1.1 (Creative Commons Public License), November 12, 2018.

```
{
  "sensor": "https://md.kumamoto-u.ac.jp",
  "sendTime": "2019-02-25T02:34:36.518Z",
  "data": [
    {
      "@context": "http://purl.imsglobal.org/ctx/caliper/v1/Context",
      "@type": "http://purl.imsglobal.org/caliper/v1/ViewEvent",
      "actor": {
        "@id": "https://md.kumamoto-u.ac.jp/user/88888",
        "@context": "http://purl.imsglobal.org/ctx/caliper/v1/Context",
        "@type": "http://purl.imsglobal.org/caliper/v1/lis/Person",
        "name": "太郎 熊本",
        "description": "No NULL!",
        "extensions": {
          "username": "12345678",
          "firstname": "太郎",
          "lastname": "熊本"
        }
      },
      "group": {
        "@id": "https://md.kumamoto-u.ac.jp/course/99999",
        "@context": "http://purl.imsglobal.org/ctx/caliper/v1/Context",
        "@type": "http://purl.imsglobal.org/caliper/v1/lis/CourseSection",
        "name": "情報基礎 B2018",
        "description": "A Moodle course",
        "extensions": {
          "fullname": "情報基礎 B 17(2018-58-A0157)",
          "shortname": "2018-58-A0157",
          "idnumber": "2018-58-A0157"
        }
      }
    }
  ]
}
```

図 9 最終的に得られた Caliper データ (一部、氏名、学生番号、内部 ID 等は仮名)