

oTree を用いた配属システム：学生希望登録システム

吉田 真聖人* 島田 夏美* 澤 亮治†

2020 年 4 月 15 日

1 サーバーの手続き

この手引きでは Windows 10 をサーバーとし、Python 3.7 と oTree バージョン 2.3 を使用するケースを例に説明する。まず、IP アドレス等を学内ネットワークの管理者に発行してもらう。サーバーとなる Windows のコントロールパネル「ネットワークと共有センター」から使用するネットワークのプロパティを開き、インターネットプロトコルバージョン 4 のプロパティを開く。ここで発行してもらった IP アドレス、サブネットマスク、デフォルトゲートウェイと DNS サーバーの設定をする。ここで設定した IP アドレスは、サーバー自身や他の PC からサーバーへアクセスする際の住所として使うイメージである*1。

ネットワークに接続したのち、サーバーへ必要なソフトウェアをインストールする。以下の説明では、oTree ホームページの oTree documentation を参照している。oTree のバージョン変更に伴って使用できるコマンドは変化するため最新版の oTree documentation を参照すること*2。

1.1 Python のインストール

1. oTree documentation のインストール用ページ(<https://otree.readthedocs.io/en/latest/install.html>) から Windows をえらび、oTree 対応最新バージョンの Python をダウンロードする。ダウンロードリンクがない場合は、対応バージョンを確認したのち <https://www.python.org/downloads/windows/> からダウンロードする (Windows 64bit なら Windows x86-64 web-based installer を選択)。
2. ダウンロードした exe ファイルを実行する。環境変数を設定したいので、“Add Python 3.x to PATH” はチェックする。
3. Windows キー +R から cmd と入力する。コマンドプロンプトが開くので、`python --version` と入力する。正しくインストールされていれば、インストールしたバージョンが表示される。

* 〒305-8573 茨城県つくば市天王台 1-1-1 筑波大学大学院システム情報工学研究科

† 〒305-8573 茨城県つくば市天王台 1-1-1 筑波大学システム情報系社会工学域

*1 本プロジェクトでは、セキュリティを考慮してネットワークには学内のみで接続が可能なイントラネットの IP を発行して頂いた。ネットワークの接続については学校や組織の管理者に問い合わせ、必要であればハブなども設定する。この文書では、組織内のイントラネットのことを学内ネットワークと呼ぶことにする。

*2 oTree バージョン 2.5 以降では以下の説明にある `runserver` コマンドが削除されているため、たとえばコマンド `runprodserver` を代用する。その場合、以下の説明では `otree runserver` を `otree runprodserver` に読み替えて実行する。

1.2 PyCharm のインストール

Python のエディタとして PyCharm をインストールする。

1. <https://www.jetbrains.com/pycharm/> からコミュニティ版をダウンロードしてインストール (2020.3.31 時点で無料)。
2. PyCharm を起動し、File>NewProject>Location の untitled を適当に変更して Create ボタンを押す。
3. 新規プロジェクト上で右クリックして、New>File>test.py と入力し、OK をえらぶ。
4. test.py に `print('hello, world!')` と書いて test.py 上で右クリックして Run をえらぶ。
5. PyCharm の下のほうに `hello, world!` と表示されれば正しく動いている。

1.3 oTree のインストール

oTree は Chen et al. (2016) が作成した Python ベースの経済実験用ソフトウェアである。

1. コマンドプロンプトを開き、`pip3 install -U otree` を実行。
2. `otree startproject oTree` を実行すると oTree ディレクトリを作成しスタートアッププロジェクトが入る。囚人のジレンマなどのサンプルもダウンロード可能である。
3. コマンドプロンプトで oTree ディレクトリに移動する (`cd otree`)。初回であれば、`otree resetdb` でデータベースをリセットする。
4. `otree runserver` で localhost で oTree が起動される。
5. chrome などのブラウザから <http://localhost:8000/> を開いて oTree の画面ができればインストール成功。
6. 起動した oTree を終了するにはコマンドプロンプトで `ctrl+C` を押すか、コマンドプロンプトを閉じる。

oTree を起動するとネットワーク上で oTree の管理インターフェースへアクセスできるようになる (localhost で起動する場合を除く) ため、学内ネットワーク上で oTree サーバーを動かす前に管理者パスワードを設定しておく。システムのプロパティの詳細設定タブから環境変数を開き、システム環境変数で以下を編集する*3。

変数名: `OTREE_AUTH_LEVEL`, 値: `STUDY`

変数名: `OTREE_ADMIN_PASSWORD`, 値: `****` (好きなパスワード)

編集したのち一度コマンドプロンプトを閉じて再起動する。その後、コマンドプロンプトから

1. `cd otree`
2. `otree resetdb`
3. `otree runserver`

で oTree を起動し、管理インターフェースから設定したパスワードでログインができることを確認する。

*3 この手引きでは “study” モードに設定する。詳しくは <https://otree.readthedocs.io/en/latest/admin.html> を参照。

1.4 Postgres と pgAdmin のインストール

デフォルトのデータベースはサーバーとしての使用に適していないため、oTree で推奨されているように^{*4}、Postgres をデータベースとして設定し、pgAdmin にて接続ができるようにする。

1. PostgreSQL を
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads#windows>
から環境に合わせてダウンロードし、インストールする。
2. pgAdmin を <https://www.pgadmin.org/download/pgadmin-4-windows/> からダウンロードし、インストールする。
3. C:/Program Files/PostgreSQL/にある pg_hba.conf を開く。IPv4 と IPv6 の METHOD 列の md5 を trust に変更する。その後サービスから postgresql... を再起動する。
4. pgAdmin を開き、新規データベースを作成、名前を django_db にする。
5. 環境変数に以下を追加する。
変数名 : DATABASE_URL, 値 : postgres://postgres@localhost/django_db
6. コマンドプロンプトで pip3 install psycopg2 を実行。
7. otree resetdb を行う。
8. oTree プログラムのサンプル (quiz など) を選んで起動し、何か入力してみる。プログラムを動かすには settings.py を編集する (<https://otree.readthedocs.io/en/latest/tutorial/intro.html> の “for PyCharm user” を参照)。
9. pgAdmin を起動して djangodb>schemas>Tables で入力を確認する。
10. 入力内容を適当に変更して保存したあと、oTree 管理インターフェースの Data からダウンロードしてデータの変更を確認する。

1.5 追加の準備

- <https://otree.readthedocs.io/en/latest/server/server-windows.html> を参照し、Redis をインストールする。コマンドプロンプトで redis-cli ping と打ち、PONG と返ってくるとインストール成功。
- Windows のファイアウォールを開き、受信の規則から新しい規則をクリックして「ポート」、「特定のローカルポート : (80, 8000 を入力)」、「接続を許可する」を選択し、「oTree」などの名前をつけておく。

2 希望登録用プログラム (shakom) の設定

以下では oTree 用プログラムである shakom プログラムを説明する。筑波大学社会工学commons <http://commons.sk.tsukuba.ac.jp/data> からダウンロードしたファイルから shakom フォルダをコピーし、すでに作成した oTree フォルダに oTree/shakom で配置する。つぎに oTree フォルダ下の settings.py ファイルを開き、SESSION_CONFIGS と ROOMS の記述がある箇所を shakom プログラムを起

^{*4} <https://otree.readthedocs.io/en/latest/server/server-windows.html> を参照。

動できるように編集する。参考として、oTree バージョン 2.3 では以下のように設定する。

```
SESSION_CONFIGS = [
    dict(
        name='shakom',
        display_name="shakom",
        num_demo_participants=1,
        app_sequence=['shakom']
    )
]

ROOMS = [
    dict(
        name='shakom2019',
        display_name='shakom2019',
        participant_label_file='_rooms/econ101.txt'
    )
]
```

shakom プログラムのセッションは1) ログイン2) 希望の入力3) 入力の確認 の順で構成され、ログイン時の参加者ラベル（後述する登録キー）と、研究室希望順位をデータベースの変数として取得して Postgres へ記録するものになる。希望する研究室数の最大値（初期値は 10）を変更したいときは、shakom フォルダの models.py を PyCharm などを開き、class Constants にある `number_of_choices=10`; を好きな数に編集する。仕組みとしては、oTree プログラムのグローバル変数となる Constants クラスに希望研究室数の最大値 `number_of_choices` を用意しており、HTML ページでその変数を参照して入力欄を生成している。また、oTree 上で用意する変数の数は `number_of_choices` に合わせることができないため、希望順位はひとつの変数（`t1,t4,t13` のような文字列）として取得している。以下ではこれを希望登録用システムとよぶ。

3 oTree 上の操作

Windows キー + R から cmd でコマンドプロンプトを開き、oTree ディレクトリに移動する (`cd otree`) ことで oTree を操作できる。まず、oTree のプログラムやディレクトリ下のファイル、サーバーの環境変数を書き換えた場合は oTree のデータベースをリセットする必要があるため、`otree resetdb` を行う。データベースをリセットすると、データベースに記録された情報は消えるので注意する。

次に oTree セッションを実行する準備をする。希望登録用ページの URL を作成するには、管理インターフェースから Rooms をえらび、参加者の最大値に余裕を持たせ、希望登録用システムのセッション（shakom）を選んでルームを作成する。ルームとは同一の oTree セッションに複数人があらかじめ決められた参加者ラベルを用いて参加できるようにする仕組みのことで、このシステムでは参加者のログインに相当する機能として用いる。ルームの参加者ラベルは後述の登録キーとして参加者に配布しておく。参加者ラベルの一覧は oTree ディレクトリの `_rooms` 下に `econ101.txt`（1 行にひとつの参加者ラベルを書いたファイル）として置いておき、oTree ディレクトリ下の `settings.py` を開いて `ROOMS` とかかれたところに `participant_label_file='_rooms/econ101.txt'` が記述されていることを確認する。`ROOMS` からは

- `name` : ルームの名前
- `display_name` : 管理インターフェース上で表示される名前
- `participant_label_file` : 参加者ラベルの一覧が置いてある場所

が設定できるので、`name` と `display_name` を適切な名前に変更しておく。ルームの名前はアクセスする際の URL の一部になることに留意する。

ルームを作成したあとは、oTree を実行する。発行してもらった ip アドレスが 10.0.1.3 のときは `otree runserver 10.0.1.3:8000` を実行する。以上の手続きにより、参加者はルームの URL (10.0.1.3:8000/room/[ルームの名前]) から oTree ルームへアクセスできる。管理インターフェースへのアクセスは、ブラウザから 10.0.1.3:8000 を開き、前述の `OTREE_ADMIN_PASSWORD` で設定したパスワードでログインする。

希望登録用システムの運用中に oTree を実行しているサーバーが停止したときには、もう一度 oTree を実行する。このときデータベースをリセットしないことに注意する。万が一のためデータベースは運用期間中に定期的にバックアップしておく。データベースの内容は管理インターフェースの Data もしくは pgAdmin から取得できる。

4 pgAdmin 上の操作とデータベースの読み方

pgAdmin は Postgres を直観的に操作しやすくするためのインターフェースである。開くには

1. pgAdmin を起動する (既に起動している場合、windows の右下のアイコンを展開、Postgres の象のロゴを右クリックして New pgAdmin window... をえらぶ)。
2. ブラウザに pgAdmin のウィンドウが開きパスワードを聞かれるので、インストール時に設定したパスワードを打ち込む。
3. Servers をクリックして一番下の階層
Servers/PostgreSQL/Databases/django_db/Schemas/public/Tables
から `shakom_player` を右クリックして View/Edit Data>All Rows をえらぶ。
4. ここからデータを閲覧、編集できる。データを書き換えることもできるので注意する。
5. 上のメニューの Download as CSV でダウンロードできる。

行はそれぞれの学生、列の変数が学生の情報を表している。変数の意味は

- `reg_key` : 登録キー
- `belongs` : 所属する専攻^{*5}
- `student_choices` : 登録した希望情報

となる。

oTree の管理インターフェースの Data から csv ないし excel ファイルでダウンロードした場合は `player.reg_key`、`player.belongs`、`player.student_choices` が同じものに相当する。

^{*5} 学生がもつ“専攻”の属性は Abdulkadiroğlu and Sönmez (2003) で提案されたタイプ別定員のある DA メカニズムで使用する“タイプ”を想定しており、タイプごとに定員を設定したマッチングを行う場合にはこの変数を使用する。本プロジェクトを運用した筑波大学社会工学類では学類の中で3つの専攻があり、学生はどれかの専攻に所属している。研究室の定員に専攻についての上限を設けなければならないケースが発生したため、この専攻の変数を用いた。

5 運用面の手続き

運用面の手続きに関して必要な事項を順に説明する。作業としての流れは次節を参照のこと。

5.1 ID への変換

ID はマッチングを実行する上で個人に割り当てた識別子である。oTree の入力面では教員の名前が選択できるが、データベース上では各教員の ID (t_1, t_2, \dots) で記録される。これには教員名と教員 ID の対応関係が必要であり、`shakom/teachers.csv` から編集する。学生の ID には登録キーと学生番号を用いる。学生番号は学生名を簡単な数字などに置き換えたものであり、教員側入力ファイルで学生を表す ID として用いる（教員用入力ファイルの手引きを参照）。一方で登録キーは学生を特定しないための匿名化された ID となる。

5.2 登録キー

登録キーは oTree のルームに参加するときの参加者ごとのラベルであり、パスワードも兼ねている。あらかじめ参加学生ごとにランダムかつ一意な 10 桁の文字列を用意しておく。学生はあらかじめ URL を配布した希望登録用システムから個人の登録キーを用いてログインできる。以降、サーバー上での学生の入力データは登録キーによって匿名化される。oTree 側では学生が登録キーを用いてルームに参加するとき専攻も同時に取得するようにしている。これには登録キーと所属専攻の対応関係が必要であり、`shakom/users.csv` で編集する。一方で登録キーから学生名への対応関係がわかるものは oTree サーバーには保存しない。よってサーバー管理者およびマッチングプログラムの実行者は登録キーから学生を特定することができない。一方で学生番号には学籍番号などを用いることもでき、こちらは個人を特定できるため、管理上の区別に注意する。

5.3 対応表の作成

学生の識別は、oTree サーバー上では登録キーで、教員用入力ファイルでは学生番号で表現されている。これを学生の名前と紐づけた excel ファイルを対応表とよぶ。参加学生の名前と学生番号（たとえば s_1, s_2, \dots ）の一覧表を用意し、ランダムかつ一意な 10 桁の文字列を excel 関数で生成して登録キーを発行する。対応表は学生名と学生番号と登録キーを相互に変換できる。学生の個人情報となるため、運営者が責任をもって管理する。また、すべての参加学生に加えて架空の参加者をいくつか作成しておくといよい。これは学生が入力内容を間違えた際に別の登録キーを発行するときや、運営者のテストログイン用として使用する。

5.4 登録キーの配布

研究室配属に参加する学生へ期間中に事務室を通して登録用紙を配布する。登録用紙には学生の名前（学籍番号）、登録キー、そして希望登録用システムの URL が書いてあり、受付期間内であれば学内の PC からログインと入力ができる。登録キーは学生番号と登録キーの対応表となる excel 上でランダムかつ重複がないように生成したのち、VBA を用いて 1) 配布資料のテンプレート（word ファイル）を人数分複製し、2) 個人名、登録キーを含む個別の情報は excel から転写する。

5.5 個人データの管理

個人情報管理の上で、理想としては選好の集計、マッチング、サーバー管理のすべてのプロセスを少数ないし1名で行うことができればよい。一方で正しい手法をとっているかのチェックや役割分担によって複数の運営者が必要となることも想定される。その場合対応表の管理者を決めておくことで提出された希望順位を知っている運営者を限定し、個人情報が流出する可能性を最小限にできる。対応表の管理者は、データを専用のusbで保存し、使わないときは金庫にいられておくなど、責任をもって管理する。

本プロジェクトでは2019年度についてはマッチングの実行と対応表の管理は別の人物が行ったため、選好の集計後から実際のマッチング結果の確定までは次の手順で行った。1) 対応表(登録キーと学生番号)管理者は教員側の選好と学生側の選好をすべて取得し、それらを教員IDと学生ID(登録キー)へ変換し、マッチング担当者へ渡す^{*6}。2) マッチング担当者はプログラムでマッチング結果を計算し、それを対応表管理者へ再び送る。そして3) 対応表管理者が最後に教員名と学生名のマッチに変換し、学生および教員への最終的な通知を行う。この場合、サーバーの管理者とマッチングの実行者は各教員ごとの学生からの希望数はわかるものの、どの学生がどの研究室に希望をだしたかはわからない。

5.6 教員への excel データの配布・回収

教員には excel で選好を提出してもらった。筑波大学で教員支援ツールとして授業などで使用している manaba から配布、回収した^{*7}。教員に渡す excel シートは学生を入力する入力用シートと、入力結果をあとでマッチングを行うプログラムが処理できる形式に変換する統合フォーマットシートを用意して後の作業を簡単化した。収集した教員側選好は登録キーの対応表の管理者が管理し、マッチング実行のときには学生を学生IDに変換した。詳細については教員用入力ファイルの手引きを参照のこと。

6 ID 発行関連作業のまとめ

これまで oTree 上の操作、pgAdmin 上の操作、そして運営面の手続きにまたがって書いてきた情報を、実際の作業手順に沿ってまとめておく。

1. (対応表の作成) 学生ごとに学生名、学生番号、登録キーを記入した excel ファイルを作成する。このとき予備用のアカウントをいくつか用意しておく。
2. (学生番号と登録キーを反映) 教員用入力ファイルの id に学生番号を記入し、サーバーの oTree/_rooms/econ101.txt に登録キーの一覧を記入する。
3. (プログラム内情報の編集) oTree/shakom/teachers.csv を、教員名とその ID (たとえば t1,t2,... と決める) となるように編集する。oTree/shakom/users.csv を、reg_key が登録キー、belongs が専攻を表す番号となるように編集する(登録キーは econ101.txt と一致させることに注意)。

^{*6} 学生IDを学生番号の方に統一してしまうと学生が匿名化されず、マッチング実行者はサーバーの情報をもとに学生および教員のすべての選好を知ることができる。よって必ず登録キーを使用する。ただし登録キーは10ケタの文字列なので、使い勝手が悪いこともある。その場合、対応表上で別途マッチング用ID(u1,u2,...)を学生ごとにランダムに発行して、学生IDをマッチング用IDに変更して渡すなどする。

^{*7} shakom プログラムを改変すれば教員側希望登録も同じ形式で運用できる。本プロジェクトの運用では学生名のリストをプロジェクト用のサーバーで保管することに関してセキュリティリスクの判断を避けるため別の手段を利用した。

4. (oTree ルーム名の編集) `oTree/settings.py` を開き、ROOMS にある `name` と `display_name` を好きな名前に編集する。`name` はシステムの URL に使われることに留意する。
5. (登録用紙の発行) 対応表をもとに、学生名、登録キー、希望登録用システムの URL などを書いた word ファイルを作成する。
6. (希望登録用システムの運用) `shakom` プログラムを実行するルームを設定して、学内ネットワーク上で oTree を起動する。
7. (マッチングの準備と実行) 希望登録用システムで収集した情報のうち、`reg_key` (登録キー)、`belongs` (専攻)、`student_choices` (選好) を取り出し、マッチングプログラムの学生側入力形式に合わせて編集する。教員側についても、教員用入力ファイルの統合フォーマットシートをひとつのファイルにまとめて教員側入力形式とする。対応表管理者は両形式で登録キーと学生番号が混在しているためどちらかに統一し、専攻の番号も統一されていることを確認してマッチングを実行する。他の運営者 (マッチング実行者等) に渡すときには学生 ID は学生番号を使わず、登録キーに統一して渡す。登録キーだと扱いづらい場合は別途対応表上でマッチング用 ID (たとえば `u1,u2,...`) をランダムに発行し、それに統一する。
8. (結果の確定) マッチングの出力後、対応表管理者が学生 ID と教員 ID から学生名と教員名に変換してマッチ結果を周知する。

7 こんなときはどうする？

Q. oTree を起動し続けているのに、学内ネットワーク上の端末からシステムにつながらないことがある。

A. つぎの作業を行う。

- PC の電源設定で自動的なスリープが設定されていればそれを切る。
- ネットワークの節電機能もオフにする。デバイスマネージャを開き、ネットワークアダプタのプロパティから電力の管理を選び、省電力イーサネットのチェックを外す。
- システムのプロパティ>詳細設定>パフォーマンスを開き、「パフォーマンスを優先する」をチェックして OK を選ぶ。

設定を終えたあと PC を再起動する。解決しない場合はシステムの管理者の方に相談する。

Q. `otree runserver` をするとエラーになる。

A. まず現在のバージョンでサポートされている実行コマンドを oTree documentation から確認する。また、プログラムを編集したあとなどにそのまま実行するとエラーが発生するときがある。必要であればデータベースの情報を保存し、一度コマンドプロンプトを閉じて再起動したあと `otree resetdb` を行う。それでもエラーが発生する場合は `shakom` を再度ダウンロードして上書き保存、`settings.py` を再編集したあと、`otree resetdb` をする (プログラムと設定を元に戻す)。

Q. 登録期間中にサーバーが落ちてしまった。

A. もう一度サーバーを立ち上げ、`otree runserver [IP アドレス]:8000` で oTree を起動すれば同じルームが立ち上がる。このときデータが保存されていることを確認する。もしも消失しておりバックアップもない場合はすべての学生にもう一度 URL から登録してもらうよう周知する。

Q. 学生から、入力を間違えたと連絡があった。

A. 予備の登録キーをその学生の新しい登録キーにして、対応表を更新し、もう一度入力してもらう。

もしくは現在の登録キーから変えずに、pgAdmin から手動でデータベースを修正する。

Q. 現在大学にいない学生が研究室配属に参加する場合どうすればよいか。

A. その学生についてもあらかじめ登録キーを発行し、メールなどで希望を送ってもらい、システムに代わりに入力する。

Q. 書いてある手順通りのメニューや言葉が見当たらない。

A. インストールしたソフトウェアはバージョンによって少しずつ操作方法や見た目が違う可能性があります。PC の操作に慣れていそうな人に聞いてみてください。

付録 A 使用したサーバーの詳細

概要	OS : Windows 10 Home 64bit, CPU : Core i5-8400 (2.8GHz), Memory : 16GB (8GB×2)
ソフトウェア	Python, oTree, Postgres, pgAdmin

参考文献

Abdulkadiroğlu, Atila and Tayfun Sönmez (2003) “School choice: A mechanism design approach,” *American economic review*, Vol. 93, No. 3, pp. 729–747.

Chen, Daniel L, Martin Schonger, and Chris Wickens (2016) “oTree-An open-source platform for laboratory, online, and field experiments,” *Journal of Behavioral and Experimental Finance*, Vol. 9, pp. 88–97.

oTree documentation, URL: <https://otree.readthedocs.io/en/latest/>, accessed on 2020-03-31.