

研究室配属プログラムの開発と運用

加 藤 暢†

学生を研究室に配属する際、多くの大学では学生の希望を優先し、研究室の定員以上の学生が集中した場合、席次等の成績情報を利用して学生を選抜し、選ばれなかった学生は第2志望以降の研究室へ配属されるという方法がとられている。この方法は、特に学生数の多い私立大学では作業が非常に複雑になり、手作業で行うと、数人がかりで3~5時間かかってしまう。本報告では、配属作業を自動化し、10数分で完了させることのできる配属プログラムの詳細と、運用の際に生じた問題等について述べる。

An Assignment System for Students to Laboratories and its Effective Use

TORU KATO†,‡

In many Japanese universities, junior or senior students are usually assigned to laboratories according to their requests so far as the capacity of the laboratories permit. When the number of applicants of a laboratory exceeds its limit, we have to choose the students who are accepted to the laboratory according to some criterion (in many case, their grade) and assign the rest of them to next requested laboratories and so on. This process is, however, very complicated and it takes long time when doing by hand. We present an automatic assignment system for this purpose which can do it in about ten minutes and show the problems produced in the case of employment.

1. 序 論

卒研究生の研究室配属は、どの大学どの学科でも、程度の差こそあれ手間のかかる問題である。近畿大学情報学科の前身である経営工学科では、4年生220~240名に対し研究室数は17であり、本稿で紹介する配属システムを開発する以前は手作業で配属作業を行っていた。作業はまず、学生に第1~3配属希望を記入した調査票を提出させ、17個の研究室箱を用意し、調査票を第1志望通りに箱に入れる。定員以上になった研究室箱の中の調査票と席次表(もちろん非公開)を見比べながらそこに配属される学生を選抜し、残りの学生の調査票を第2志望の箱に移すというものであった。この作業は煩雑で、3人の教員で5時間程度を要した。さらに、この作業で配属が決まらなかった学生を後日集め、定員に満たない研究室の中からもう一度配属希望を調査し、同様の作業を行なった。最初の段階で第1~17志望まで書かせればこの二度手間が生じないが、できるだけ学生の希望に添った形で配属できるよう、このような方法をとっていた。勿論

より効率の良い方法はあり得るが、どのような方法でも、公平性と学生の希望というある意味相反する要素を尊重すれば、人手による作業では多くの時間を費やすことは明らかである。作業終了後、さらに配属情報を計算機に打込みデータ化する時間もかかる。

本稿の目的は、このような配属作業を自動化するプログラム(3節で詳述)を紹介することではあるが、プログラム自体の解説よりも、むしろ、4節で述べるように、その運用方法が重要であると考え、実際にこのプログラムで2001年度~2005年度の5回の配属作業を行ってみて、如何に学生にとって不公平感を抱かせない配属方法をとるか、また、配属結果に不信感を持った学生を如何に納得させるかという運用方法が重要であることを痛感した。そこで4節では、実際に生じた問題とその解決方法について述べる。

2. 配属方針例

配属システムの詳細を述べる前に、本学科の研究室配属方針を例として紹介しておく。配属プログラムはこの方針に従って動作するが、プログラム内のパラメータを変化させることで、様々な方針に対応可能である。なお、現在は学生数200~220名、研究室数25という環境で本システムを運用している。以下で述べ

† 近畿大学理工学部情報学科
Department of Informatics Kinki University

る配属方針例は、過去5回の配属作業を通して、様々な問題を解決する中で(4節参照)得た、一つの解だと見ることができる。

第3志望までは席次順位優先配属

例えばA研究室(定員9名)に10名以上の学生が第1～第3志望した場合、席次上位の者から順に配属される。この段階で配属が確定した学生に対しては、以下の配属作業の結果は全く影響しない。

第4志望からは希望順位優先配属

B研究室(定員9名)に第3志望までの学生が8名しかおらず、あと1名が配属可能な場合、希望順位の高い学生が配属される。例えば、甲君(席次25位)がB研究室を第6志望し、乙君(席次63位)がB研究室を第4志望した場合、乙君がB研究室に配属される。しかし甲君も乙君もB研究室を第4志望した場合、甲君がB研究室に配属される。

つまり、第1～第3志望までは完全に席次順、それ以降は希望を優先した席次順になっている。このような配属方針をとるようになった経緯については、4節の運用時の問題点の中で詳述する。

志望順位のシフト

丙君が第3志望までの研究室に入らず、第4志望以降を次のような順番で希望していたとする: 第4志望A研、第5志望B研、第6志望C研、第7志望D研、第8志望E研、第9志望F研、…。しかし、A研、B研、C研、がすでに第3志望までの学生で満員だった場合、丙君が不利になるのを避けるため、丙君の志望研究室順を以下のように変更して希望順位優先の配属を行う: 第4志望D研、第5志望E研、第6志望F研、…。

このシフト操作により、第1～第3志望だけの調査による配属作業を行ない、後日未配属の学生を集め、定員に満たない研究室のみを対象とした配属作業を行うこととはほぼ同じ効果を持つことになる。

3. 配属システム

本節では、我々が開発し実際に使用している、研究室配属システムの詳細について述べる。本システムは大きく二つの部分に分かれる。学生の希望をデータ化する部分と、そのデータを入力として、2節で説明した配属方針に従い実際に配属作業を行なう配属プログラムである。

3.1 学生の希望のデータ化

本システムでは図1に示すようなマークシートによる調査票を学生に記入してもらい、各自の希望を収集する。この調査票はエクセルにより作成し、普通

図1 希望調査票

Fig. 1 A questionnaire for assignment

03001	14	23	21	24	2	15	...
03002	4	8	24	16	1	22	...
03003	21	24	3	6	9	2	...
...							

図2 配属希望調査結果データ

Fig. 2 The data of student's request for assignment

のコピー機で学生数分 A4 の普通紙に印刷するだけで使用できる。図中の A 研, B 研の部分には、加藤研など実在する研究室名が記入されている。また確認のため、希望研究室番号のマーク欄の左には、手書きで希望研究室名を記入してもらっている。この調査票をフィーダー付のスキナーで取り込み、マルチ tiff 形式でファイルに出力する。このファイルを、市販のマークシート読み取りソフトで読み取り、図2に示すような、学籍番号と希望研究室番号列からなるデータとしてテキストファイルを得る。図2の1行目は、学籍番号03001の学生の第1志望が14(N研究室)、第2志望23(W研究室)、…、であることを示している。

我々は読み取りソフトとしてハンモック社のRemark Office OMR ver. 5.5を使用している。調査票の読み

1	A 研	9	0
2	B 研	9	0
:	:	:	:
25	Y 研	5	0

図 3 研究室データファイル
Fig. 3 The data of laboratories

08002	近大 1 朗	63
08003	近大 2 朗	218
08004	近大 3 朗	111
:	:	:

図 4 学生席次データ
Fig. 4 The data of student's marks

08002	近大 1 朗	第 1(1) 希望研究室:	2: B 研
08003	近大 2 朗	第 10(20) 希望研究室:	14: N 研
08004	近大 3 朗	第 2(2) 希望研究室:	15: O 研
:	:	:	:

図 5 配属先データ
Fig. 5 The data of student's labs

取り時間は、使用するスキャナの性能に大きく依存するが、フィード付のベッド型の普及機 (EPSON ES-9000H) で 220 人分を読み取るのにおよそ 15 分程度を要した。読み取りソフトを用いて、tiff ファイルからデータを抽出するために要した時間は、30 秒程度であった。

このような調査票ではなく、希望調査の専用ページを学内に立ち上げ、オンラインで入力する方法も検討したが、学生による手書きの調査票を配属結果の証拠として残すことの意義を考え、今回紹介したような調査方法を採用した。

3.2 配属プログラムの入出力データファイル

配属プログラムは、図 2 で示した学生の希望調査データファイルと、図 3, 4 で示す、各研究室の定員データファイル、及び学生の席次を示すデータファイル (いずれも text 形式) を入力とする。そして、各学生の配属先を示す配属結果ファイル (図 5) と、各研究室毎の卒研生名簿ファイル (図 6)、欠席者名簿ファイル、そして、各学生毎に、自分が希望したにも関わらず配属されなかった研究室の情報を示すファイル (図 7) を出力する。図 7 の第 9(19) 志望... の部分は、実際の志望は第 19 志望だが、席次満席シフトで第 9 志望として配属作業を行ったことを示している。図 7 の詳しい説明は、4.1 項にて行なう。

:	:
B 研	現役生 9 名
08002	近大 1 朗
08017	近大 14 朗
:	:
N 研	現役生 9 名
08003	近大 2 朗
08033	近大 28 朗
:	:
O 研	現役生 9 名
08004	近大 3 朗
08008	近大 5 朗
:	:

図 6 研究室配属結果
Fig. 6 The results of the assignment

08003	近大 2 朗 (席次 218) を Reject した研究室
第 1(1) 志望	研究室番号: 15 研究室名: O 研
08004	近大 3 朗 希望: 2 席次: 111
08008	近大 5 朗 希望: 1 席次: 33
:	:
第 2(2) 志望	研究室番号: 16 研究室名: P 研
08016	近大 13 朗 希望: 1 席次: 12
08272	近大 212 朗 希望: 1 席次: 17
:	:
第 9(19) 志望	研究室番号: 25 研究室名: Y 研
08012	近大 9 朗 希望: 3 席次: 168
08014	近大 11 朗 希望: 4 席次: 220
:	:

図 7 学生の配属経過
Fig. 7 The trace of the assignment of a student

3.3 配属プログラム

配属プログラムは、Java 言語により実装した。本プログラムは大小 10 個程、合計 1000 行ほどのプログラムである。その中の主なクラスは図 8 に示す、Student.class と Lab.class であり、それぞれ学生と研究室を表している。それ以外に入出力を受持つ Main クラス (Haizoku.class) 等が存在する。

実行時にはまず、3.1 項で述べた各データを入力するための窓が開く (図 9)。

ここからプログラムを走らせると、本学科の 25 あ

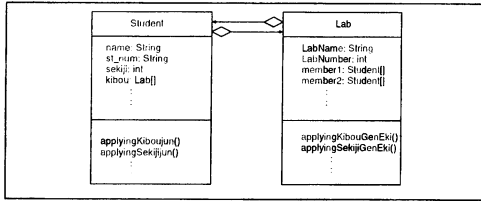


図 8 クラス図
Fig. 8 Class Diagram

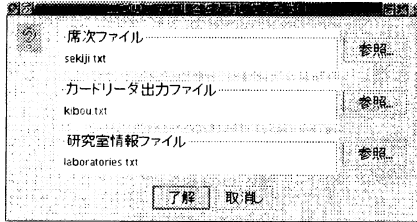


図 9 データ入力窓
Fig. 9 Window for Data Input

る研究室のそれぞれの名前と定員を持ったオブジェクトが Lab.class から、3 年生 220 余名の学籍番号、名前、希望研究室リスト、席次等の情報を持ったオブジェクトが Student.class からそれぞれ生成される。学生オブジェクトはまず、自分の第 1 志望の研究室に対し applySekijun メソッドを実行し、配属を試みる。研究室オブジェクトは、自分の定員に満たない状態までは無条件に学生オブジェクトを受け入れる。定員をオーバーしている場合は、2 節で述べた席次優先方針に従って、学生の受け入れを判断する。第 1 志望研究室に配属を許可されなかった学生オブジェクトは、順に第 2 志望、第 3 志望の研究室に向かう。

第 3 志望までの研究室に配属されなかった学生オブジェクトに対し、それぞれの希望リストの中から既に定員に達した研究室を削除し、定員に空きのある研究室のみの希望リストを自動的に作成する。それらの学生オブジェクト達は、今度は applyKiboujun メソッドを実行し、第 4 志望以降の研究室に対し配属を試みる。各研究室オブジェクトは、2 節で述べた希望優先方針に従って、学生の受け入れを判断する。実行中の様子を図 10 に示す。

配属を希望した学生数と、配属が終わった(未配属と認識された学生も含むが)学生数の整合性を確認し、3.2 項で述べた、各種ファイルを出力したのち本プログラムは停止する。

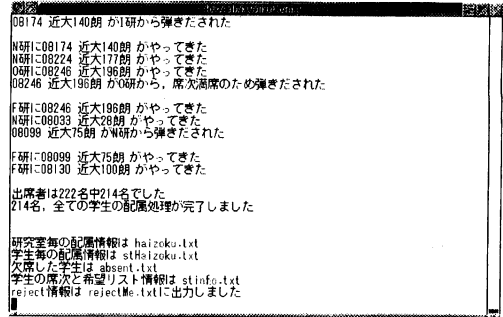


図 10 実行状況
Fig. 10 Window for Execution

4. 運用上の問題点

本学科では本システムを使用して、2001 年度から 2005 年度まで、5 回の研究室配属を行ってきた。本節ではこれまでに生じた問題点等を述べる。

4.1 配属結果に異議を申し立てる学生への対応

本システムに限らず、配属作業終了後に必ずといっていいほど起る問題が、希望通りに配属されなかった学生からのクレームである。手作業で行なう配属作業では配属経過等が残りやすく、異議を申し立ててきた学生に筋道を立てて説明することが困難であった。しかしここで納得の行く説明を受けられなかった学生は後々まで感情的なわだかまりを残し、配属先の教員に対しても配属変更を言い続けるケースが見られた。

配属のルールを明文化することはもとより、そのルールに厳密にのっとった作業であったことを示すため、本システムでは異議を申し立ててきた学生に対し配属作業責任者が、図 7 に示す資料を見せながら配属経過を説明できるようにしてある。この資料は、自分が希望したにも関わらず配属されなかった全研究室の、配属済の学生の席次と希望順位を示したものである。この資料と学生が提出した希望票とを見比べながら、なぜその学生が、これらの研究室に配属されなかったか、筋道を立てて説明できるようになったため、苦情を言い続ける学生はいなくなった。図 7 では、席次 218 位の近大 2 朗君が希望したにも関わらず配属されなかった、O 研、P 研、…、Y 研所属の学生の情報が示されている。この情報によると、第 1 志望の O 研所属の学生は全て近大 2 朗君より席次が上位である。また第 9 志望の Y 研には近大 2 朗君より席次下位の近大 11 朗君が所属しているが、第 4 志望以降は希望順位優先配属というルールがあるので、やはり近大 2 朗君は配属されなかったことが分かる。なお、図 7 中

の第9(19)志望という記述は、本来近大2朗君はY研を第19志望していたのだが、第1～第3志望までの席次優先配属操作の後、定員に達している研究室を近大2朗君の志望リストの中から削除し、残りの研究室を志望リストの上位にシフトした結果、Y研を第9志望したとして配属作業を行なったことを示している。

またこの資料は、配属作業が正しく行なわれたかどうかを担当者が目で確認するためにも有効である。勿論全ての学生の結果を追跡することは困難だが、学生10名ほどをサンプルとして抽出し、この資料を元に配属過程を追跡することが可能である。

4.2 配属ルール見直しの経緯

希望優先配属だけの場合

2001、2002年の配属作業では、「第3志望までは席次優先配属」というルールはなかった。これは席次優先をルールに入れると、人気のある研究室に席次上位の学生が集中してしまうためであった。しかしその結果席次が中位の学生が最終志望(またはそれに近い)研究室に配属されるケースが頻出した。前項で述べた異議を申し立ててくるほぼ全ての学生がこのケースに相当する。これは、第1志望研究室には配属されず、第2志望研究室に作業が移ったとき、既にその研究室も定員に達しており、しかもその学生より席次が低い学生が配属されていたとしても、希望優先のため第2志望研究室にも配属できないからである。これと同じことが第3以降の希望研究室でも起こり、結局希望下位の研究室に配属されることになる。

2段階の希望優先配属

この問題に対する一つの解決策としては、配属作業を2段階に分け、例えば第5希望までを最初に調査し配属作業を行ない、未配属の学生に対し、定員に達していない研究室を対象にもう一度配属作業を行なうことがあげられる。これを行なったのが2003年度であった。これで上記のような問題は軽減したが、全く無くなったわけではなかった。また、配属作業を2段階にした結果、作業が煩雑化、長期化し、本システムを採用した利点が半減してしまった。席次が中位の学生からも「安心して第2志望、第3志望を調査票に書くことができない」等の意見が寄せられた。

希望優先だけをういた配属ルールのもう一つの問題として、このルールは席次下位の学生に非常に有利に働く可能性があることが実証された。2002年度と2003年度の配属作業において、ある研究室で、席次がほぼ最下位の学生が定員の9割を占めるという現象が見られた。これらの学生はいずれもこの研究室を第1志望しており、席次上位から下位のほぼ全ての学生

が、この研究室を第2志望以降にしていたためである。2002年度は、こういう偶然もあるかとの観測で終わったが、2003年度も全く同じ現象が同じ研究室で生じたため偶然では済まされなくなり、席次上位～中位の学生に有利になるような規則の改訂が求められた。

席次優先配属後の希望優先配属

そこで2004年度から、席次最下位付近の学生が有利に配属されることの無いよう、2節で述べた方針、すなわち第3志望までは席次優先配属し、第4志望からは希望順位優先配属するという配属ルールに改訂して運用している。また、第4志望からの希望順位優先配属の際にも、各学生の希望をできるだけ尊重するよう、各学生の希望リストのなかから無効な研究室名(つまりすでに第3志望の学生までで定員に達している研究室名)を削除する作業も配属プログラムの中に取り入れている。これにより、席次が中位程度以上の学生はほぼ全員第3志望以内の研究室に配属されるようになった。その反面、席次が下位付近の学生の多くが、第20～25志望の研究室に配属されるという極端な結果が見られるようになった。また、これら席次下位の学生の多くが特定の研究室に集中するという新たな問題が生じている。

5. 結論・今後の課題

本報告では、研究室配属を自動化する一つの手法として、本学科で開発、運用している配属システムについて紹介した。当然のことではあるが、手作業での配属と比較した場合の労力は実際に激減(3名で5時間かかっていたものが、1名で15分程度)している。また、配属結果を検証するための情報も配属作業中に出力されるため、配属結果の検証、配属結果に異議を申し立てる学生への説得といった、作業後に起る手のかかる問題への対応の効率化も同時に実現している。また、今後配属規則が変化した際も、柔軟に対応できるよう整備されている。例えば、何らかの理由である学生をある研究室に優先配属する場合、単にその学生の席次情報を1番にするだけで望みの結果が得られる。また、席次優先配属を第2志望までにする場合も、単にパラメータを変化させるだけで対応できる。それ以外に、現役生と留年した学生を、別々の基準で一度に配属することも可能である。これは、留年した学生が一つの研究室に集中した場合、その研究室の教員の負担が増えるためである。

4.2項の最後で述べたように、席次最下位付近の学生が、特定の研究室に集中するという問題が残っている。これはシステムの問題と言うよりも、配属作業そ

のものの持つ問題である。これを解決するためには、例えば席次上～中位、中～下位の2グループに分け、各研究室も定員枠を2つ設け、各々のグループで配属作業を行なうと言った方法が挙げられる。これは先に述べた留年生を扱うための方法を適用すればすぐにも実現出来るが、中位の学生が、下位の学生への対応のために、所属できるはずだった研究室に所属できなくなるなど、新たな問題が生じる。

また、本システムは本学科のみで利用しているため、今だ5回の運用実績しかない。今後多くの局面で利用を試み、問題点の洗い出しを行なう必要がある。また、類似のシステムとの比較等も行なっていく必要がある。

ところで本配属プログラムは、各学生を表すオブジェクトからスレッドを作り、それぞれ独自に希望研究室オブジェクトを訪れるマルチスレッドプログラムである。これは、実在する人間の各学生が一斉に各研究室に訪問し、そこで他の学生と席次や希望順位を比べ合うことで配属先が決定されていくという現実世界の並行動作を抽象化して設計したためである。勿論現実世界でこのような作業が出来れば理想的だが(我々教員の手間が一切かからないので)、プライバシーの保護、正確性、時間的問題、配属経過の保存など様々な問題から非現実的である。

マルチスレッドを利用した並行動作をさせることにより、制御構造を使って各学生オブジェクト(のスレッド)の動作順序を逐一制御する煩雑さから解放されている。本プログラムの動作検証としては、現在は図7に示した reject 情報を利用して、人間の目で最終結果を確認しているが、この単純な検査作業は自動化が容易であると考えられる。

それとは別に、このような並行プログラムの動作検証を自動的に行なうために、現在我々はプロセス代数を用いた研究を行なっている⁶⁾。並行に動作するスレッドを生成する Java のソースプログラムを π 計算³⁾ のプロセス式に変換し、MWB 等の検証ツールを利用して等価性の調査などが可能になることが期待できる。また、本配属プログラムの特徴は、学生エージェントプロセス達が、研究室エージェント内に入ったり出たりを繰り返すという単純なモデルとして捕らえることができ、このようなシステムのモデル化に最適な言語として Ambient Calculus¹⁾ がある。我々は Ambient Calculus に対する等価性理論に関する研究⁵⁾、実装に関する研究等を行なっている⁴⁾。本配属プログラムを Ambient Calculus のプロセス式に変換し、様相論理の一種である Ambient Logic²⁾ を用いて、性質の検証

等が可能になることが期待できる。

参 考 文 献

- 1) Cardelli, L. and Gordon, A.D.: Mobile Ambients, *LNCS*, Vol.1378, pp.140-155 (1998).
- 2) Cardelli, L. and Gordon, A.D.: Any time Anywhere Modal logics for Mobile Ambients, *POPL'00*, *POPL'00*. pp.365-377 (2000).
- 3) J.Parrow: An Introduction to the π -Calculus. *HANDBOOK OF PROCESS ALGEBRA* (J.A.Bergstra, A.Ponse and S.A.smolka(eds.)), NORTH-HOLLAND (2001).
- 4) Kato, T.: The Implementation of Ambient Calculus with HORB for Mobile Agents, *Proc. of The 7th World Multiconference of Systemics, Cybernetics and Informatics*, Vol. II, pp.367-372 (2003).
- 5) Kato, T.: An Equational Relation for Ambient Calculus, *Trans of Information Processing Society of Japan*, Vol.46, No.12, p.to appear in (2005).
- 6) 山口将志, 加藤暢, 樋口昌宏: 動的な接続関係を持つ Java プログラムの一記述法と π 計算への変換, *FIT2005 第4回情報科学技術フォーラム*, A-015 (2005).