

Abstract – At graduate research, although the process is more important than the results, most students don't notice it. Because the guild system is nice to learn the process, the graduate research possesses a kind of relationship between a mentor and a padawan learner.

On this project, we are developing a system for noticing importance of learning process, ornb, whose specifications and the connections to a static web system, jekyll,

Keywords : keyword 1, keyword 2, keyword 3, keyword 4, keyword 5

1. Introduction

1.1 背景 I - 大学の学び

今日の大学教育は、フンボルト理念を基礎としている。その通説を

近代大学の出発点は1810年に創設されたベルリン大学である。この大学の基本構想を作ったのは、ヴィルヘルム・フォン・フンボルトであり、近代大学はこのフンボルト理念から始まった。フンボルト理念の中核は研究主義にある。つまり、大学は教育の場である以上に研究の場であるという考え方は、このフンボルトから始まった。これがドイツばかりでなく、世界の大学を変えた。

と潮木はまとめている。

- アルカディア学報（教育学術新聞掲載コラム）,No.246

この後、潮木はその著書でフンボルト理念が単なる「神話だった」と力説している。

日本の工学を中心とする学部においては、研究至上主義としてのフンボルト理念が、大規模講義形式の授業への出席、レポートや期末試験を経て単位を取得する仕組みに繋がっている。

しかし、学生は、意欲的に授業に参加し、自らが参加者となり授業に出席することで、知識を得ようとする姿勢が見受けられない。「単位さえあればいい」あるいは「卒業して就職さえできればいい」という本音と、「眠い」という実感によって、大学で授かるべき知識について魅力を感じなくなっている。あるいは社会に出て、すぐに実用できるスキル実用できるスキルの習得、例えば、英会話や論文作成術を身につけようとする。

したがって、授業の内容の重要性に気づくことなく、レポートや期末試験の前にレジュメを確認しその内容

を提出する。つまり、内容や過程を重要視せず、結果のみを重要としている。

つまり、現在学生の知識習得に役立つと思って課されているレポートや試験は、結果のみを求めているという誤ったメッセージとして、学生が受け取っている可能性がある。

1.2 背景 II - 大学生は徒弟制を否定しがち

ところが、最新の教育を行うと看板を掲げている大学に通う今日の学生は、大学は研究の場であるという認識が薄く、卒業研究や、研究室におけるゼミへの参加にどのようなスタンスで取り組みがいいか明確には理解できない。ある意味、学生には、研究室とは、「大学システムが切り捨てようとして来た、徒弟制的な制度である」ということは全く脳裏にない。

1989年にグロスハンスの指摘によれば、西ヨーロッパとほかでもない米国において中略

徒弟制は最も価値のある、しかも最も力のない労働者を統制するための伝統的な形態だと長い間みなされていた。LaveWenger[p.41]

これが、徒弟制に対する一般的な意識であり、大学生の多くもそのように認識していると考えられる。

1.3 目的

次節で示す通り、このような旧態然とした徒弟制も現代的な視点で見直しが進められている。本プロジェクトで提供しようとするシステムは、

- 研究室は徒弟制
- 学生はそれを知らない

という前提のもとで、徒弟制を新たな学習形態として提供することを目的としている。

次節では、どのような経緯で徒弟制が見直されて来たか、また、学習がAM/PMという視点によってどのように捉えられているかを明らかにする。その上で、近代的な徒弟制を研究室活動に導入するのに必要とな

る仕様を列挙し、それに基づいた実装デザインを示す。

2. 徒弟制の見直し

2.1 状況に埋め込まれた学習

1991年にレイヴとウェンガーによって、「状況に埋め込まれた学習」あるいは「正統的周辺参加」という学習形態・概念が提案された。彼らは、アフリカの仕立て職人や助産婦の育成法を社会学的に詳しく調査した結果、徒弟制のなかに学びの本質があると指摘した。少し複雑ではあるが、その概念をもっとも短くまとめたと思われる箇所を以下にそのまま書き写す。

学習はいわば参加という枠組で生じる過程であり、個人の頭の中ではないのである。このことは、とりもなおさず、共同参加者の間での異なった見え方の違いによって学習が媒介されるということである。この定義では「学ぶ」のは共同体である、あるいは少なくとも、学習の流れ(context)に参加している人たち、といえよう。学習はいわば、共同参加者間にわかれ持たれているのであり、一人の人間の行為ではない。生産過程では徒弟(見習い)が益々増大していく参加によってきわめてドラマティックに変容していくものではあるが、この変容の発生の場合と発生条件は、さらに広範囲の過程そのものである。徒弟の親方たち自身が共同学習者としてふるまうことを通しどれほど変化するか、したがって、習熟されている技能でもその過程でどれほど変化するか。実践者の共同体がより大きくなると、徒弟の形成によって共同体は自らを再生させるが、同時に変容もすると考えられる。LaveWenger[pp.8-9]

中略

また新参加者を親方、ボス、あるいは管理者と深く対立する関係に陥らせる、参加させるよりも非自発的に隷従させるなど、これらの条件は実践における学習の可能性を部分的に、もしくは完全に、歪めてしまうと唱えた。LaveWenger[p.42]

と記している。

2.2 AM/PM

1998年数学者のSfardは、Lave and Wengerの考えを受け、学習者、教授者、研究者のあり方についてAM(Acquisition Metaphor)とPM(Participation Metaphor)と名付けた。学習に対する従来の考え方であるAMは、個人が知識を習得することを目標とし、「学習」とは何かを獲得することであった。また、

「知る」は個人が所有するものであるとしていた。一方で学習に対する新しい考えであるPMは、学習の目標は共同体の構築であり、「学習」とは参加者となることである。学習者は、徒弟であり、教授者は、有識の参加者と定義した。つまり、個人ではなく、教授者、学習者がチームとして、また徒弟制を築くことでお互いお互いの知識構築が捗る仕組みとなっている。

表1

2.3 PMの実践例と学生の受け止め方

2.3.1 ペア評価の意図

関西学院大学理工学部・情報科学科で西谷が参加型学習の試みとして数式処理演習という3年生向けの演習がある。学生は好きなもの同士がペアを組み、授業中課題や期末試験をペアで受け、ペアの点数は全く同じとなる。ペアで「相方の足を引っ張らないように」という思考に至り、互いが怠けることなく、授業や課題に意欲的に取り組む。その結果、互いに高め合い、知識の定着につながる。「共同体の構築」、「参加する」というPMの実践を意図している。しかし、実際には知識の定着に至らない学生が多数いる。

2.3.2 学生の見え方

教育の非対称性、

- 教える側はどう役にたつかを知っているが、
 - 教わる側は、知識を獲得するまでわからない。
- ：ダンス：京大アメフトの水野監督：運動神経がない人に教えるときは、喋れ。体で覚えるのは、偶然の失敗。

- サボりたい
- どこで役にたつかわからない
- 役に立たない

数学、英語全てやくに立たない。ならば、単位取得で縛りましょう。

- 一つ目の要因はペアによる演習のため、一人が作業すれば課題をクリアできる点である。つまり、問題毎に役割を振り分け片方が問題を解いている時、もう片方は携帯を見るなど考える事を完全にやめることがある。一緒に考えることをせず、「休憩」の時間を作ることで知識定着を目的とするのではなく、課題達成、単位習得の事のみを考えた結果である。
- 二つ目は、ペアで課題を一つ提出することが、いわゆる出席点となるため一人が授業を欠席しても、点数が減点されることがない点である。これは、一つ目に述べた要因より酷い例であり、日にち毎に出席する担当を決めることで授業に出席、参加すらしなない場合があった。

結果的に見るとこの授業は、PMといった考え方を気づかせる授業ではなく学生にとって、「授業に出なくても良い楽に単位を取れる授業」という風に見受けら

表1 Acquisition metaphor と Participation metaphor の比較.

Acquisition metaphor	要素	Participation metaphor
個人を豊かにする	学習の目標	共同体の構築
何かを獲得する	学習するとは	参加者となる
受容者, 再構築者	学習者	周辺参加者, 徒弟
供給者, 促進者, 仲裁人	教授者	有識の参加者
資産, 所有物, 一般商品	知識, コンセプト	実践, 論考, 活動の一側面
持つ, 所有する	知るとは	所属する, 参加する, コミュニケーションをとる

れた.

また, 個人での学習よりも互いに高め合い, 知識・スキルを習得するといったペアは一部であり, PM を理解した生徒, 優秀である生徒とそうでない生徒がペアを組んだ場合, 前者がほとんどの課題をこなし, 後者はほとんど考えないというパターンも存在した.

- 参加型学習の実践の失敗,
- 何が悪いかを分析

3. 卒論・ゼミ

前提条件として, 西谷けんにきた学生は

- サボりたい
- どこで役にたつかわからない
- 役に立たない

としようよ.

それでも, 西谷研に参加したいと思わずには, どうすればいい?

- やりたいことを聞いて, それをさせる

ではない.

どうすればいい??,

- 自信が持てるような活動
- そういうことをしているということに気づいてもらう?
- coding ができるようになってきているから.
- 実感させるシステム?
- 周りに反応してもらう,

知識, 反応, 行動が「強化される」.

客観的な事実の記述

- merit 見えない, demerit 時間が取られる
- 西谷研究室では, 生徒の出席率が悪い. この原因は, 他人任せであることが一番大きい. 「誰かが出席していれば, なんとかしてくれる」, 「欠席による出席者より理解が遅れるという危機感の無さ」, 「プライベートとゼミとの優先順位がおかしい」等が挙げられる.

卒業研究やゼミにおいても, 教授や先輩が後輩に計算機の使い方や, プログラミング, レポートの書き方を教える. この時, 後輩は自らの意思で参加するという考え方であるべきだ. 西谷研究室

では, 後輩が参加者となり先輩から学ぶという風潮が見受けられない. 後輩は, 卒業研究を発表すること, 結果のみを考えており, 卒業研究を発表するまでの過程の重要性に気づいていない.

4. 構築システムのアイデア

以上の考察に従って, 研究室運用システムとして

- 日々の個人活動を構成員に公開する
- ペアによる個別指導
- 欠席者のフォロー

を提供することを当初の目標とした.

4.1 これらを運用するために

- まず初めに, レポートを作成し, 日々の個人活動を構成員に公開する方法として, my_help, Jekyll, GitHub Pages を用いたシステムを開発した. my_help は Rubygems で提供されている, 直交補空間を実現した知識構築を補助するツールである. ディレクトリに拘束されることなく, メモやレポートを作成・管理できるという利点があるため, どこからでもアクセスできる. my_help では Blog という形で文書を作成し, 構成員に日報を伝える.

my_help で Blog は org-mode で作成されるが, Jekyll では Markdown が Blog 記事となるため, org ファイルを md ファイルに変換し, Jekyll で公開する記事次に, my_help で作成された Blog を Jekyll と GitHub Pages を用いて公開する. Jekyll は, Rubygems で提供される静的サイトジェネレーターである. また, GitHub Pages と連携し公に公開する.

4.2 デザイン (つもり)

- ペアの活動がスムーズに進むように環境を提供する.

今は, ゼミを中心とした個別の時間調整で行っている. 遠隔でも実践できるようなシステム.

- 欠席者の知識フォローを強制するシステム

これから, Blog の作成, 公開のみならず, ペアの活動や欠席者の遅れをフォローするシステムを考案する.

現在、西谷研究室ではゼミを中心とした、個別の時間調整を行っている。しかし、1週間学校に来れない人もいるため、それを援助すべく遠隔でもペアプロや知識の共有をするため、Skypeを用いる。また、構成員は教授者と学習者の両方に成り得るものとし、欠席者はゼミ出席者を教授者としゼミの内容や課題を教えてもらう学習者とする。課題に取り組むとともに、Blogを作成し、知識定着をはかる。また、達成後は教授者が学習者にスタンプを押す。そのスタンプが、課題達成の証明となり、卒業するまでの必須過程とする。また、ゼミ欠席者でも教授者にスタンプを押してもらうと、教授者としての資格を獲得し、他の欠席者に教える事ができ、課題達成後はスタンプを押す。

4.3 新しい徒弟制と古い徒弟制、西谷研でどうするよ？

- このような参加型の学習を西谷研で実践するのに何が必要か？
 - 徒弟制の本質
 - 親方と徒弟の関係
 - － 古いよね
 - － 見習い？
 - － 習慣
 - 先輩の仕事
 - 硬い上下関係ではなく、参加して学ぶ
 - 徒弟のモチベーション、参加意欲
 - 教える方も、確固とした知識として身につけることができる
 - 今まで何で、教えなかった
 - ダンスはあるけど、
 - 後輩と勉強する機会がない
 - そういう機会がないから
 - 優先順位がおかしい
 - 共同体への参加というのを気づいていなかった。
 - 同じやろ
 - 声かけ
 - 目に見える形にして共有する
- 学べる人はどこでも学べる？

4.4 [o] 日報

- 自分がやったことの記録
- blogはそのつもり
- 見た目って大事
- 手軽にできる
- org
- my_help

4.5 [o] ペアプロ

- ゼミ？ テキスト？
- 徒弟制と大学、スキルと記述的記憶の違い
- スキルの伝達をチェックする？

- free rider が増える
- コピペはダメ！
- ペアでの作業
- ペア評価
- 責任、サボれない

ただ始めること。これがたぶん生産性の鍵なのだ。ペアプロが機能する理由は、「相方とペアプロ作業を予定する」ことで、「作業を始めることをお互いが強制する」からに違いない(原文より訳出)。Joel Spolsky 著、青木靖訳「Joel on software」(オーム社、2005)p.133。

- チェックにならない。
- だから、skype とかが必要。
- scheduling
- 予定の強制、スタンプ集め

4.6 [o] スタンプ集め

例えば

- 全員 jekyll を入れて、blog を晒す
- というゼミで実行した課題があるとする。そいつを全員が実行したかどうかを、教えた方がチェックする。手順は以下の通り、
- 欠席者が出席者に聞く
 - 出席者がスタンプを押す
 - それが埋まってなかったら卒業なし。

これを自動化するシステム。

いっぺん聞いたら他の人に教えるのはあり。そうすると、教えることによる記憶強化の可能性が高まる。また、不明瞭な点のあぶり出しが可能になる。

4.7 [o] org-mode = 便利な mark down

org-mode は、Emacs 上で動作するアウトライナーでありプレーンテキストの文書作成環境である。ノートの保存、TODO リストの管理、スケジュールや時間の管理、また発表原稿やスライドの作成など様々な用途に対応している。また、コードの実行はもちろん、リンク付け、テーブル表記の入力、図や表の表示、ライブ計算、HTML や $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ への変換等の機能も兼ね備えている。今回のレポートとなる文書の作成するために、org-mode を用いる。

4.8 [o] my_help == 直交補空間

ファイル構造において、メモやレポートが増えれば増えるほど chunking する。chunking することにより、構造が深くなる。その結果、レポートやメモの場所が把握できなくなる。

directory ってのは知識のマップ。知識が大きくなると、chunking する。深くなる。迷子になる。my_help ってのは直交補空間に置かれている。いつでも access できて便利。

4.9 [o] jekyll == 晒すと何がいい？

jekyll は Rubygems で提供されている静的サイトジェネレーターである。テーマや構成を変更することができ、好みのサイトを作成できる。今回の文書の公開を jekyll で行う。

- 文章，文を気にする，
- 共有しやすい，
- 形になる，
 - report
 - 他人事だと思っているから
 - 自分が学んでいることとの関連性を自覚する
 - 深く理解する
 - 経験知識に変える，説明する，議論する
 - * 徒弟制ではない，大学システム手に職を，中世のシステム

5. ornb の仕様

5.1 スタンプ集め

例えば

- 全員 jekyll を入れて，blog を晒す
- というゼミで実行した課題があるとする。そいつを全員が実行したかどうかを，教えた方がチェックする。手順は以下の通り，
- 欠席者が出席者に聞く
 - 出席者がスタンプを押す
 - それが埋まってなかったら卒業なし。
- これを自動化するシステム。

いっぺん聞いたら他の人に教えるのはあり。そうすると，教えることによる記憶強化の可能性が高まる。また，不明瞭な点のあぶり出しが可能になる。

5.2 レポートの作成と公開

まずはじめにレポート作成から公開までの流れは図1のようになる。

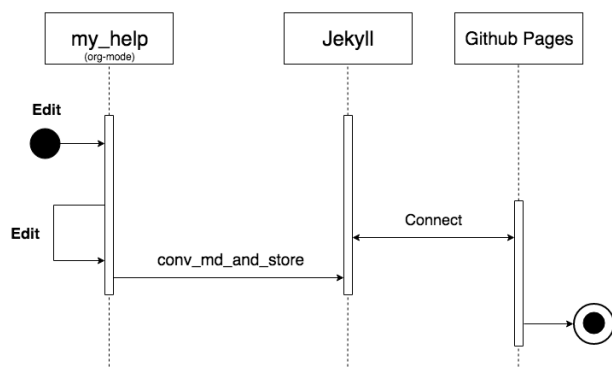


図1 レポート作成から公開までの流れ

次に，レポートの作成について述べる。レポートの作成は Rubygemse で提供されている，my_help を用いる。

my_help を以下の方法でインストールする。

```
gem install my_help
```

次に，my_help に blog という list を作成するために，以下のコマンドを実行する。

```
my_help new blog
```

これで新しい list が作成される。次に blog の内容を編集するため，以下を実行する。

```
my_help edit blog
```

編集画面では，タイトルをつけた後，タイトルの横に図2のような日付を付けることとする。

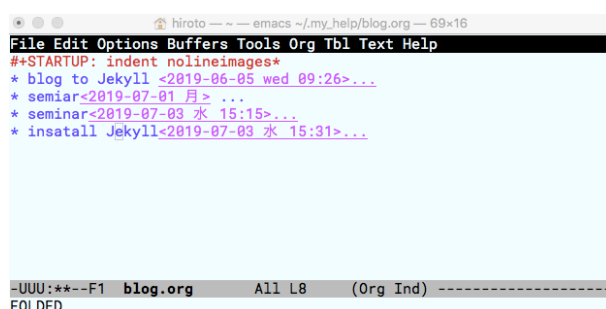


図2 日付を付ける例

また，以下のコマンドで日付を付ける事が出来る。

```
Control-u Control-c .
```

あとは，Blog の内容を書き込んでいく。

5.3 Jekyll のインストールと実行

Blog の作成と公開は Jekyll と Github Pages を用いる。以下にその手順を記述する。jekyll の install と GitHub Pages との連携ははじめに，Jekyll をインストールする。

```
gem install bundler jekyll
```

次にローカルサーバを立ち上げる。

```
jekyll new my-awesome-site
cd my-awesome-site
bundle exec jekyll serve
```