知識構築システム ornb を利用した学習過程の考察

河野大登 *1　　福森聡 *2　　西谷滋人 *2

hoge hoge

Kono Hiroto*1,　Satoshi Fukumori*2　and Shigeto R. Nishitani*2

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 reseach 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 ゼミナール

卒業研究などのゼミナール (以下、ゼミ) は，フンボルト理念を基礎としているとされる．その通説を潮木は次のようにまとめている．

> 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中心主義にある。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。[1]

と． また,「大学が伝えるべきことは，いかにして新たな知識を発見するか，いかにして知識を進歩させるか，そのための技法である」との観点から,「研究する学生」を志向したとしている [2].

今日の学生が，このような深遠な思想を理解して，長い受動的学習の最終段階として，ゼミに参加している気概は感じられない．また、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」という認識はなく，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むべきか迷っている．研究のための予備知識，スキルの習得に役立つとして課されているレポートや試験などは，新しい発見につながる途中経過を求めているにもかかわらす，既存の知識の暗記という結果のみを求めているという誤ったメッセージとして受け取られている可能性がある．実践的な課題を，project based learning [3] や active learning として取り入れる試みがなされているが, [4], [5] かける時間の割に，成果をあげるのが難しいとされている

「研究する学生」を育てるには，個人に適合した指導が欠かせない．これはある意味，メンターとメンティ，あるいはマスターとパダワンのような関係となる．しかし，ゼミナールへやってくる学生は，研究指導が徒弟制的な制度であるとは全く予想していない．

1.2 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，徒弟制を現代的で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洗い出し，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 [7]. 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

*1: 関西学院大学大学院　理工学研究科
*2: 関西学院大学　理工学部
*1: Graduate school of Science and Technology, Kwansei Gakuin Univ.
*2: Department of Informatics, Kwansei Gakuin Univ.

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ち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と発生の条件は，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．[7][pp.8-9]
>
> 中略
>
> また新参者を親方，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．[7][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，研究者の知識に対する心持ちを AM(Acquisition Metaphor) と PM(Participation Metaphor) と名付けて分類した [8]．表 1に示した通り，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」と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．本プロジェクトでは、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，学習者が共同体（チーム）として，また徒弟制を築くことでお互いの知識構築がはかどる仕組みとなっている．

2.3 PM の実践例と学生の受け止め方

このような徒弟制の見直しは，単なる概念として語られるだけでなく，実践としてシステム化されつつある．イギリスの一部の大学では 2015 年に Degree Apprenticeships という徒弟制度を取り入れた実践的技能を身につける教育形態が開始されている [9]．

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，PM，すなわち参加型学習の試みとして数式処理演習で実践している．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．この授業への取り組みの根底にあるのが,「共同体の構築・参加」であり、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

なぜ,

- 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，参加すらしない場合があった．

結果的に学生的視点から見ると、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れた．

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，優秀な学生と，そうでない学生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．

これら一連のなぜを考えると、次の疑問が湧いてくる．はたして、PM を理解して共同体を作ろうとまで考える学生はいたのだろうか？

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,

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

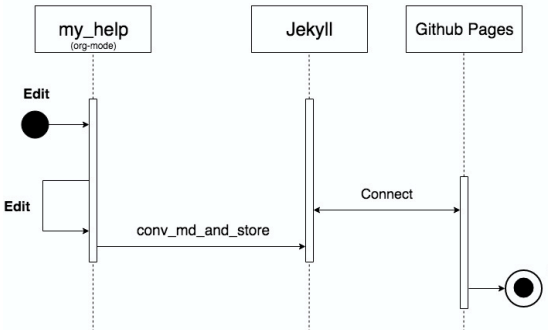

図 1　blog 作成から公開までの流れ.

- 教わる側は，知識を獲得するまでわからない.
- サボりたい
- 教わることはどこで役にたつかわからない
- 役に立たない

ならば，単位取得で縛りましょう．となるが，それでは伝統的な徒弟制となんら変わりはない．どうすれば，役に立つことに学生が気づくのか.

3.　構築システムのアイデア

「新しい徒弟制」という視点に立って，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 (blog システム)
- ペアによる個別指導 (遠隔ペアプロ)
- 欠席者のフォロー (スタンプ集め)

という機能提供することを当初の目標とした.

3.1　blog システム

学生個人のゼミ活動はゼミ発表などであるが，より細かな活動を記録することが望ましい．実験系の研究室では，研究室ノートや実験ノートなどで日々の活動を記録する習慣をつけるが，プログラミングを主体とする研究室においては何行書いたを聞く指導教員もいる．より良いプログラマの習慣は，活動記録として blog の公開が推奨されている．これによって，

- どのようなアイデアで
- どこで何を調べて
- どこで挫折したか

などを公開することで，coding がうまくいかなかったとしても，思考の過程を記録し，先輩や同僚の code review を通じて，programming skill の向上が期待される．また会社においては，これは日報にも通じている.

研究室内の活動においても実践可能な blog システムを目指して，

- my_help
- org-mode
- Jekyll

のそれぞれの機能を利用して実装した．blog の作成から公開までの流れは図 1のようになる.

3.1.1　my_help = 直交補空間

my_help は Rubygems で提供されている，ファイル構造において，メモやレポートが増えれば chunking の必要が出てくる．ところが，chunking することにより，ディレクトリ構造が深くなる．その結果，レポートやメモの場所が把握できなくなる．これに対して，my_help は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事なく，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.

3.1.2　org-mode = 便利な mark down

org-mode は，Emacs 上で動作するプレーンテキストの文書作成環境である．HTML や LaTeX への変換等が可能であり，ノートやメモの作成，TODO リストの管理，発表資料，スライドの作成など様々な用途に対応している．また，コードの実行の他，テーブル表記の入力，図や表の表示，ライブ計算，機能も兼ね備えている [6] ．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.

3.1.3　Jekyll = 晒すと何がいい？

my_help は emacs の org-mode を利用しているが，個人での使用を前提としており，公開するためのシステムが存在しない．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．Github には，Jekyll で作成されたサイトを公開する

GitHub Pages というサービスが用意されている．

my_help で作成された Blog を，Jekyll に連携することで，local においてその完成度を確認することができる．また，git push するだけで github pages で world wide に公開される．

3.2 遠隔ペアプロ

ペアの活動や欠席者の遅れをフォローするシステムを提供する．現在，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．

ペアプロが機能する理由は，

> 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.

であるとされており，空間を共有する必要はない．

キャンパスが郊外にあるため，効率的にゼミナールの研究を進めるためには，遠隔での共同作業が不可欠である．そこで，いくつかの環境を使って実際の作業を試行して，結果を収集する．

そのような環境で必要となる仕様は，新しいタイプの徒弟制の視点に立って，1． 先輩と後輩によるペアプロ 1． コードのリアルタイム共有 1． 音声，ポインタなどによる指示 1． 作業記録，振り返りなどが効率的に行えることである．

これらを踏まえ，遠隔でもリアルタイムで共同編集できる環境として，Teletype for Atom がある．Teletype for Atom は，ホストが Key を発行し，その Key によってメンバーが，参加できる．参加すると，メンバーのカーソルが表示され，リアルタイムでコードの編集やコメントを書く事ができる．また，メンバーが退出しても，ホスト側に記録が残るようになっている [10]．また，Visual Studio Live Share も遠隔で複数人がリアルタイムにコード編集，ブレークポイント，デバッガ操作を共有できる環境である [11]．Visual Studio Live Share と Teletype for Atom の違いは，編集できるファイルの数である．Type for Atom は，ホストが開いた 1 つのファイルのみ編集が可能であるが，Visual Studio Live Share はフォルダ全体をリアルタイム編集可能である．また，Key の発行が share ボタン一つで発行できる点や，ブレークポイントを設定し，コードの実行を共有できる点は Visual Studio Live Share の特徴である [?]．

3.3 スタンプ集め

ゼミに欠席した学生のフォローシステムである．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．

欠席学生は，出席学生に教えてもらいながら，課題に取り組むとともに，Blog を作成し，知識定着をはかる．課題達成後は指導学生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，欠席者でも指導者にスタンプを押してもらうと，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．

ゼミ毎にスタンプを用意し，全てのスタンプの取得が卒業の必須項目とする．

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. 今後の課題

参考文献

[1] 潮木守一: アルカディア学報（教育学術新聞掲載コラム），No．246.

[2] 潮木守一: フンボルト理念の終焉？現代大学の新次元: 東信堂, 2008.

[3] S．Bell: ‘Project-Based Learning for the 21st Century: Skills for the Future ’ ; The Clearing House: A Journal of Educational Strategies， Issues and Ideas， vol．83， no．2， pp．39–43， (2010).

[4] B．Settles: ‘ Active Learning Literature Survey ’ ; University of Wisconsin-Madison Department of Computer Sciences， Technical Report， (2009).

[5] 溝上慎一: アクティブ・ラーニング導入の実践的課題; 名古屋高等教育研究，第 7 号，pp．269–287（2007）

[6] Org mode for Emacs: あなたの生活をプレーンテキストで， https://orgmode.org/ja/ (accessed on 10 Feb 2019).

[7] シーン・レイフ, エティエンヌ・ウェンカー, 佐伯胖訳, 福島正人解説: ’ 状況に埋め込まれた学習, 正統的周辺参加’ : 産業図書, 1993.

[8] A. Sfard: ’On Two Metaphors for Learning and the Dangers of Choosing Just One’: Educational Researcher, 27(1998), 413.

[9] press release: ’Government rolls-out flagship Degree Apprenticeships’:

https://www.gov.uk/government/news/government-rolls-out-flagship-degree-apprenticeships, (accessed on 8 July 2019).
[10] 【リアルタイム共同編集】Atom で出来るようになったってよ - Qiita; https://qiita.com/k-waragai/items/a372800c262f56fe688a (accessed on 7 July 2019).
[11] ［速報］「Visual Studio Live Share」発表。複数のプログラマがリアルタイムにコードの編集、ブレークポイント、デバッガ操作などを共有。Connect(); 2017; https://www.publickey1.jp/blog/17/visual_studio_live_shareconnect_2017.html (accessed on 7 July 2019).
[12] リアルタイムでコードの共同編集ができる、Visual Studio Live Share を使ってみた（パブリックプレビュー版）; http://yfp5521.hatenablog.com/entry/vscode-liveshare (accessed on 7 July 2019).