

# Chap 10 : Tools to assist learning by doing: Achieving and assessing efficient technology for learning

Alan Lesgold, Martin Nahenmov

## 1. Introduction

過去数十年にわたり、学習の認知科学が発展

その中で、Cognitive Engineers が出現

教育に関するデザイン問題を明らかにし、認知科学の知見を適用

過去、複数の学習方法が開発

learning by doing, didactic presentation, peer collaboration and argumentation, drill  
方法の状況への対応付け(コストと利点を明示)

単純な算数ならドリルが効率的。しかし、電磁気のことを学ぶのにドリルは適さない

同様に、learning by doing についても効率的な状況とそうでない状況

## 2. The economics of training system development

過去、複数の企業、政府、準行政的組織と共同研究を行う中で、コストがトレーニングシステムの開発を制約することを学ぶ。コストの制約は2レベルにおよぶ。

- レベル1：コストと利益の単純な関係。コストを上回る具体的な利益(工場の改善、雇用の質が向上するなど)が見こめなければ、開発に協力してくれない。
- レベル2：組織構造、伝統的なトレーニング方法の問題。従来の企業は教育(研修)に金をかけることなど考えていない。それに対して、著者らのシステム(知的徒弟的研修環境)の開発にはコストがかかる(プロジェクトの初期で\$2000000。今では数十万ドル)。研修担当者に勇気ある決断を迫る。いくらその利益が高くても、受け入れられるのには時間がかかる。

## 3. Learning by doing

inert knowledge

Whitehead：アイデア間で十分な結合がない、十分に深められていない、概念の指示的意味が十分に獲得されていない状況

inert knowledge 排除の方法

Whitehead：生徒と同じコミュニティに生活する教師が経験に基づいて教訓的教示を行うことにより、経験と知識の結合ができる。

著者：現在では、経験の多元化が進行。Learning by doing によって、経験を通した原理の獲得が必要。

本章で Learning by doing を教育・トレーニングにおける強力なスキーマとするためのツールを紹介。

### 3.1. Three levels of facility

Learning by doing は Rasmussen(1994)が提唱した 3 レベルの知識を発達させる

知識のタイプ	定義	トリガー	例
スキル	知覚的再認と行動的活動との直接的結合	知覚的特徴	工場内の危険な状況の再認と安全な状況へ戻すための標準的な行為
ルール	エキスパートの習慣的産物 (プロダクションルール)	スキルの失敗	トレーニングされた方法で対処できるときに、装置を診断し修理する
概念的知識	領域知識、問題状況へ写像可能な知識 (weak methods)	ルールの失敗 + 問題表現と関連知識の結合	トレーニングされた方法が失敗したときに、装置を診断し、修理する

エキスパートは 3 レベルの知識を結合させる

- 問題を解決するルールを数多く知っている
- 状況を再認し、コンパイルされたルールを発動させることができる
- 実際の課題遂行に概念的知識を使わないが、必要に応じて、概念的コメントができる(研修生は概念的知識の助けを必要とし、さらには、知識を適用する方法でもアシストが必要な場合がある)

Anderson のモデルトレーニングアプローチ = ルールベースの知識に焦点を絞る

問題点は概念的知識の発達をアシストできないこと

概念駆動の問題解決に焦点をあてた課題分析に基づいて、システムを構築する必要

しかし、どこまで概念的知識を考慮すれば良いのか(きりが無い...)

エキスパートは自分の概念的知識がどのように現実的(ルールベース)知識と結合しているか知らない 理論とドリルが分離されて教育されてきた背景

考えなければならない問題

- トレーニングコースの目標は何にすべきか
- どのようにして目標の達成に必要な知識をエキスパートから引き出すか
- 強固なトレーニングシステムを構築するために、知識をどのように使うか

### 3.2. Knowing a fixed algorithm versus knowing a generative algorithm

Boshuizen and van de Wiel(1999)が複数レベルの知識結合に関する研究をしている

診断において利用される概念と説明、患者への処方へのリンクは強固に結びつく概念クラスターの利用により強められる。クラスターを要約したカプセル化された概念は、直接的に、強固に臨床的特徴、他のカプセル化された概念、病気の原因とリンクした時点で、それらのクラスターの中心として獲得された概念ではなくなる。習慣的な事例は兆候と疾患を結合する概念のカプセル化を進行させる。より困難な事例においては、カプセル化のパスは直ちに進行するわけではなく、生理学的知識が役割を果たす推論をガイドする詳細な概念の活性化により進行される。

固定されたアルゴリズムを単純に教えて、その利用経験を積ませるのは不十分

例：フレボトミスト（採血しかしない人）はほとんどルーティンの仕事だが、突然の事態にそなえなければならない

ルールベースの活動と概念的知識を利用した推論とを結合する生成的アルゴリズムの必要

- 問題を表象するスキーマ
- 問題の解を推論する概念的知識
- 解を実行する固定ルール

#### 4. Problems of multiple representation levels

エキスパートがもつ 3 レベルの知識を特定することは困難

##### 1. 各レベルにおける知識の均質性

イオンビーム装置の修理を行う人々は量子、磁力、形状、化学の知識が区別できなくなっている。カテゴリー化されていない。知識間の関係は家族的類似によって結びついているよう。

##### 2. 単一の知識フォームに対する違和感

エキスパートの知識は分散され、冗長である。現場での要求に応じる形で知識の構成言語化が困難。概念を教えることによる教育が困難

##### 3. ルールベースの知識と概念的知識を結合する必然性が不足

概念的知識の重要性は状況によりけり

例：アスピリンはメカニズムの理解と臨床的使用が分離されて発達

ウィンターグリーン アセチル酸への改善 アスピリンの開発 開発からかなりの時間を経て、アスピリンのメカニズムが理解

ほとんどの薬剤師はアスピリンのメカニズムを理解していない。アスピリンに関する知識はルールベース（心臓発作のときにアスピリンを投薬する）

しかし、アスピリンのメカニズムを理解するエキスパートはルールベースの適用が間違っていたときに、概念的知識を適用することで問題を解決

#### 4.1.A Hypothesis

特定の問題に取り組んだ経験が 3 レベルの知識を結合させる

問題に固有のルール、断片化された概念知識(問題遂行中にチューターが投げかけたアドバイス、頭に思い浮かんだアイデア)

Boshuizen & Schmidt (1992)は困難な問題の遂行そのものが知識の結合を促すと議論

複雑な課題に必要なとされる知識の塊とはルールベースの知識と概念駆動の表象とを結合することである。生理薬学的知識を臨床で繰り返し適用することで、低次の詳細な生理学的命題が高次の臨床的命題に支えられたものになる

#### 4.2.The encapsulation (Schematization) Processes

認知的徒弟制の研究(Boshuizen & Schmidt, 1992; Brown & Collins, & Duguid,1991;

Collins, Brown, & Newman, 1989)が提案するカプセル化(スキーマ化)のプラン

- 明示的な概念的枠組みの発達
- 現実の課題状況で、状況付けられたルール駆動の活動をリフレクションする
- 状況付けられた知識とグローバルな枠組みとの結合を洗練させること

チューターのすべきことは？

- 研修員の新たな経験をガイドするラフな概念的枠組みを提示
- 特定の問題・状況におけるリフレクションの支援
- 成功した活動とその活動を成功させた知識とのリンクを強める

#### 4.3. Designing Learning Systems to Support Encapsulation

概念的知識とルールの結合を強めるアプローチは複数存在

ANDES (Gertner, Contai, & VanLehn, 1998)

- 力学の問題解決に基づいて、課題に必要とされる知識、生徒に不足している知識を特定
- エキスパートシステムが問題を解決するプロセスを自己説明、コーチ

Albacete のアプローチ

- 生徒の誤りのタイプと関連する概念を提示

前者のアプローチは非明示的、後者は明示的に状況と概念の結合を強める。しかし、後者のアプローチも問題状況に特化してしまっていて、他の状況の知識との結合を強めることはできない。

#### 5. Forms of Knowledge Needed to Support Learning by Doing

システムに知識を実装するために、3レベルの知識を考慮に入れる必要がある。特に焦点をあてるのは弱解法に基づく概念とエキスパートの洗練されたルールの結合。

技術者は自動化された状況の再認によって習慣的に作業 失敗したら領域固有のルールを適用 弱解法(一般的な問題解決ストラテジー)に移行

Learning by doing システムに必要な知識

1. 技術者の仕事環境のシミュレーション  
仕事環境を表現する
2. 技術者が扱っている問題の範囲を特定  
生徒が解くべき問題を決定する
3. トレーニングのために利用される全ての問題を解決する機構(ルール)  
生徒が完全に行き詰まったときには、トレーニングシステムが生徒に解を示す
4. 概念的原理とトラブルシューティングルールとの結合をサポート  
Rasmussen の提唱した 3 レベルの知識を結合する

## 5.1. The Work Environment

### 5.1.1. Function

作業環境をシミュレートするために、

- 環境デザインの種類、マニュアルから個々のオブジェクトを引き出す
- エキスパートがどのように、作業環境を機能的なサブシステムに分割しているか、組織化の方法を知る

エキスパートがもつ環境の表象を反映した作業環境を実装

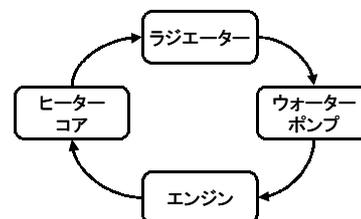
エキスパートの表象を反映した階層的ディスプレイ、メニュースキーマ  
研修生がエキスパートと同じ表象を構築することを助ける

### 5.1.2. Classification and Flows

材料、エネルギー、情報の流れに関する知識を実装することが重要

例：冷却水の流れ

ウォーターポンプが流水率の入力 ラジエータの水の流れを見て、ウォーターポンプが機能していることを知る  
他のコンポーネントに関する観察から、特定のコンポーネントの作動を知るためには関係的知识の実装が必要



このようなコンポーネント間の結合はマニュアルを見ることで実装できるかもしれない  
しかし、エキスパートに聞き込みすることも重要  
マニュアルには余分な知識が掲載、それらを除外することが必要

### 5.1.3. The Date That Signal a Problem

問題を決定する情報(Signal)が必要

ウォーターポンプの例

ベアリングやウォーターパドルの破損に起因すると思われる異音を伴うオーバーヒートの兆候によって、ウォーターポンプ問題が決定

このような情報を実装するために、Figure10.2のような装置をインテルと共同で開発  
問題の詳細とウエハー基盤の表面を表示

タブをクリックすると問題と関連する他のディスプレイが表示

多数の知識フォームを実装することで作業環境を再現

スタティックな知識：マシンの挙動のログ

ダイナミックなログ：研修生が行う装置への操作

## 5.2. Expert Diagnostic Knowledge

### 5.2.1. Rules

どのような領域の専門家であってもルール駆動の活動が主、

システムに実装するルールはプロトコルやインタビューから抽出

### 5.2.2. Expectations about Actions

エキスパートはルール適用の結果を予測している

エキスパートのルールは以下のように表される

IF <condition> THEN do <action> and EXPECT <outcome>

予測がルールに付け加わることで

現在の状況が修復を必要とすることをエキスパートに知らせる

また、予測に合致しない結果を入力とするいくつかのルールをエキスパートは持っており、問題解決の次の段階に進む

さらに、予測に合致しない結果はルール駆動の活動から概念駆動の活動へ連結する

### 5.3. Expert Conceptual Knowledge

問題の特徴と結合した概念知識を適切な状況で想起することが重要

しかし、それを獲得することが困難であることは、過去の教育研究により示されている

状況付けられていない物理学概念(その日の授業以外で斜面の問題を解くことがない)

例：シリコンウエハーをコンピュータチップに載せる機械

チャンバーにチップを入れ、適切に操作したガスを注入し、プラズマを利用して、ウエハーをロボットが載せる

このような因果知識を課題状況の出力にマッピングするには？

問題の出力と機能的モデルを結合する行列により定式化

$$N_i = P_{ij} \times \{F_j\}$$

$P_{ij}$ ：知識工学者の仮定する写像行列、 $F_j$ ：(サブ)システムに関する機能のベクトル、 $N_i$ ：問題の出力

しかし、このような定式化は現実の技術者の知識写像を捉えきれない

$$F_j = M_{ij} \times \{i\}$$

$i$ ：逸脱した出力、写像の行列  $M_{ij}$

より広くいえば、 $P_{ij}$  はエンジニアの世界観、 $M_{ij}$  は技術者の世界観を表す

$M_{ij}$  を学習者に教えることがシステムの目的だが、 $P_{ij}$  と  $M_{ij}$  を一致させることが困難  
 $M_{ij}$  を教育するために開発されたシステムが process explorer (Fig. 10.3.)

- $M_{ij}$  のサブシステムを研修生に示す
- 各セルには  $i$  と  $F_j$  の関係を示すアイコン(例：熱が上がるとチリが減るは )と解説へのリンク

問題空間の適切な探索をサポートするために全ての行列を示さない = 関連のあるサブシステムを示す

- 列には問題と強固に関連した概念
- 行には概念と関係のある問題の特徴(出力)

行と列の構成はエキスパートの知識を反映 エキスパートの概念的知識を研修生が身に付

ける

#### 5.4. The Problem Set

エキスパートが直面し得るトラブルシューティング問題を積む

資源が許すなら全ての知識カテゴリーに対応した問題を積む =  $M_{ij}$  のできるだけ多くのセルをカバーする問題を積むことが理想

問題セットができたら、もう一度、ルールと概念を見なおすことが重要

繰り返し知識を追加することで知識ベースが完成

### 6. Tools for Building Learning Systems

#### 6.1. Initial Domain Constraints

知識を実装するための具体的な手段を議論

実際には何らかの課題分析をした後でなければ、知識を実装するプランが見えてこないが、準備段階の議論をすることは有用

#### 6.2. The Knowledge Gathering Process

知識工学の典型的な手法 = プロトコル分析

発話プロトコルからルールの推定 問題へのルールの適用 ルールに対するエキスパートからの批評 批評を説明するルールの修正

Learning by doing システムの構築にこの手法は不十分 PARI(Gott, 1987; Gott et al., 1986; Hall, et al., 1995; Means & Gott, 1988)

発話プロトコルに加え、個々の行為に関する質問を受ける

- Precursor: 次のアクションに移る前に何をしましたか？
- Action: どのようなアクションが合理的ですか？なぜあなたはそのようなアクションをとったのですか？
- Result: あなたのアクションによってどのような結果が有り得ましたか？
- Interpretation: 実際に得られた結果からどのようなことを学びましたか？

発話プロトコルとの組み合わせにより、実装すべき知識、エキスパートの組織化を知る

Precursor からの回答はルールの実装に役立つ

もし、エキスパートが特定の行為  $i$  をするまえに、 $1, 2, 3, 4, 5$  を言及したのなら、IF  $1, 2, 3, 4, 5$  THEN  $i$  というルールを作ることができる

Result からの回答により、ルールへの予測の付加ができる

フラッシュライトが点かなくて、バッテリーを診断したエキスパートが受け取り得る結果 正常か異常 正常ならバルブを疑う、異常なら正常なバッテリーに取りかえる

IF battery-voltage-level < 1.2V, THEN replace battery

IF battery-voltage-level = 1.5V, THEN assume-battery-good AND assess-lightbulb

### 6.3. Deciding on Conceptual Knowledge and Connecting It to Problem Contexts

関連知識を特定(列)し，関連している領域(行)とリンクする

大雑把な方法 = プロトコルと PARI から推定する

さらなる推定

列の推定：学問的な特徴を参考にする(例：ガスの法則を問題に則して分解し適用 異常なガスの濃度，エネルギーの供給，ロボットの運送 セルの数値を決定するアルゴリズム)

行の推定：様々な方法が考えられる(エキスパートの言及からウエハーのチリ，ウエハーの濃度，表面の圧力などを推定，さらに，バルブの問題，ロボットの問題，電気の問題) 行と列のカテゴリーを決定することは1回ではできず，繰り返しによって決定

- PARI からマッピング行列を推定
- カテゴリカルに列と行を構成
- 行列と関連する他の問題を構成
- その問題を解いたエキスパートのデータを付け加える

### 6.4. Some Available Tools

システムの開発にかかるコストは知識収集に関する形式的な手法が不在であること

著者らはデータの取りなおしで \$5000 ~ 20000 を使った

データ収集の定式化はツールに依存するので例として PC PACK (Epistemics 社)を紹介  
オブジェクトベースのシステム = オブジェクトを定義すると，他の場面でもオブジェクトが使用できる

#### 6.4.1. Protocol Editor (Fig. 10.4.)

プロトコルに複数のタグを書きこめる(概念，値，関係，ルールなど)

#### 6.4.2. Card Sort Tool (Fig. 10.5.)

カードソート = オブジェクトに属性・値を割り振るために，エキスパートがカードを分類  
カードソートをコンピュータ上で = 概念を表すカードをドラックして適切な欄に移動  
例：

インタビューでエキスパートが複雑なシステムを “ エネルギーに関連したもの ” “ ガスに関連したもの ” “ ロボティクスに関連したもの ” に分けていることが示唆された  
エキスパート自身にシステムに関連する概念をソートしてもらった  
概念はプロトコルエディタで構成されたものが自動的に用意される

#### 6.4.3. Laddering Tool (Fig. 10.6.)

概念をドラックして概念間の階層関係を定義できる

下位概念は上位概念の属性を継承する

下位概念を定義することで上位概念の属性を書きかえる ( Fig. 10.7. )

#### 6.4.4. Rule Inference and Rule-Editing Tools

普通のルールエディタであるが、他の場所で定義された概念がそのまま使用される  
ルールと概念を結合するシステムを構築する上で非常に有効な機能

### 7. Assessment in The Contexts of Learning by Doing Systems

Learning by doing が有効な領域を切り出すための方法

#### 7.1. Scoring Performances Using Policy-Capturing Methodology

パフォーマンスの評価のための方法 = Policy-Capturing Approach (Gott)

著者らのシステム (Sherlock intelligent coached apprenticeship system) を評価

1. 口頭でトラブルシューティングに関する複雑で難しい現実的な問題を提示
2. 問題遂行後にインタビュー (PARI)
3. エキスパートが研修生のプロセスをトレース, 評点
4. 研修生のプロセスをコンピュータ上にシミュレートされた作業環境に入力し, うまく推論が成り立つが判断

##### 7.1.1. Policy-Capturing Approach

Policy-capturing schema を構成する方法

1. 複数の人間のパフォーマンスをエキスパートに提示し, 順序付け
2. 順序付けの理由を特徴に言及して言う 評価に関わる特徴セットの集合を構成
3. 順序を従属変数とし, 特徴セットを独立変数とした回帰分析
4. 他の人間のパフォーマンスを評価する基準が出来上がる

著者らが試した Policy-capturing

- 完全に口頭で研修生に問題を提示する方法 (エキスパートが検査につきそう)  
...コストが高い
- 質問機能を備えた Learning by doing システムを使う方法  
...システムになれているだけで, パフォーマンスが上がってしまう

##### 7.1.2. Mapping Competence to Value

コストに則したシステムの評価 = 研修生がいかにコストを減らす行為を取ったか  
トラブルシューティングの手法にはコストがつきもの (時間, 材料, 他の技術者の援助)  
コストを含めた得点化を行うことが重要

#### 7.2. Questions Focused On Explanation and Understanding

1. トラブルシューティングに関する行為の記述を研修生に提示
2. 研修生は提示された行為に関する説明をする

### 3. 説明の質を評価

多肢選択項目などを用意すればエキスパート以外でも評価を実施できる

しかし、自分でトラブルシューティングすることと他人のプロセスを説明することは別の能力なので、濫用は禁物

### 7.3. Scoring With Neural Networks

ニューラルネットがプレテストとポストテストの成績を区別できるか

判別分析などの手法は満たさなければならない条件が厳しすぎる

ノンパラメトリックな判別分析のようなものとしてニューラルネットを使用

方法

入力層：可能なアクションユニット（検証，分類，部分的な交換）

出力層：プレとポストの得点を表すユニット

個々のプロトコルを使い，ニューラルネットのトレーニング

インテルと共同で作成したトレーニングシステムに関するデータで計算したのが下の表

研修生はプレとポストで複数の問題をとくとき，そのうち，問題 3，7，8，12 を入力として計算．他の問題はニューラルネットとのトレーニングに使用．0 がプレ，1 がポストを示すようにバックプロパゲーションにより調整．

問題	プレテストにおけるアウトプットの平均	ポストテストにおけるアウトプットの平均
3	.026	.704
7	.102	.863
8	.016	.907
12	.063	.845
平均	.052	.830

ニューラルネットはプレとポストのパターンを学習 トレーニングシステムの有効性

## 8. Conclusions

Learning by doing システムに要求される知識を示した

今まで見過ごされていたルールと概念を結合することの重要性を示した  
知識を集める方法を示した

有効性はまだ検証していないが，効率的にシステムを構築できるだろう  
システムの開発に使えるツールを示した

しかし，今だ十分なシステムは構築されていない

Learning by doing システムを評価するスキーマを示した

自分たちの研究をデータに評価方法を試した

システムの有効性を示す評価方法のさらなる定式化が望まれる