

# Exploring the Assistance Dilemma in Experiments with Cognitive Tutors

Kenneth R. Koedinger(Human-Computer Interaction Institute and Psychology Department, Carnegie Mellon University)

Vincent Aleven(Human-Computer Interaction Institute, Carnegie Mellon University)

Educ Psychol Rev (2007) 19:239–264

## Abstract

Intelligent tutoring system (ITS) は高いインタラクティブも持った学習環境である。Cognitive Tutor は ITS を認知科学の問題解決と学習の理論に基づいたシステムである。Cognitive Tutor は以下を使用者に提供する

1. ステップごとのフィードバック
  2. エラーの詳細メッセージ
  3. オンデマンドの教示ヒント
- この論文ではインタラクティブの手法の比較、そしてそれらの結果の再解釈を行う。
    - どのようにして我々は情報やアシストの授与や保留のバランスを取るべきか
    - どれくらいが教示科学において基本問題
  - 学習環境デザインについて
    - ビデオやテキストはインタラクティブではない(cf., Schwartz and Bransford 1998)
    - Doing by learning(e.g., Duch et al. 2001; Krajcik and Starr 2001)や自己説明 (Chi et al. 1994)はインタラクティブである。
    - 人間対人間のチュートリングはとても効果的である(Bloom 1984)
    - 人間の教師を模倣したインタラクティブな学習は高い効果があることが実際の教室で確認された (VanLehn 2006)

## Framework: Our View on What Counts as Interactivity

教示方法とイベントを2次元で表1に示す。

### ◇ 表1の1列目: Passive

テキストの記述や加工された例題

### ◇ 表1の2列目: Active

問題解決について訪ねる事や、加工された例題の自己説明 (Chi et al. 1989)

- passive と異なり、多くの情報が差し控えられ、生徒はそれを埋める事を要求される。

### ◇ 表1の3列目: Interactive

システムの反応、生徒の反応のやりとりの往復

Active のように情報は保留された状態。インタラクティブな教示的イベントでの追加の情報は学習者がその情報を組み立てられなかったら与えられる。

☆ 表1の1行目: **Explicit, verbal generalization** 明確な口頭の今日イベント

一般的な表現、概念や原理の説明や表現するルール

☆ 表1の2行目: **Implicit, instances** 暗黙またはインスタンスベースの教示

例や練習活動→概念や原理の適応物（暗黙的に示している）

- 教示とは通常は表1を行と列で組み合わせられたものである

例: 教科書=Passive explicit description と Active 問題をとくこと

- 表2に情報の授与と情報の保留についての利益やコストの要約を示す。

この表2がアシスタンスジレンマのことである。この表2より以下のことを考慮する。

☆ 情報を統合できる機会は一時的なものなのか永続的なものなのか？

☆ 情報をどの程度提供すればよいのか、最も適切なタイミングは何なのか？

☆ フィードバックは迅速であるべきなのか？

☆ 教示的ヒントはさらに詳細、広義的にすべきか？

- 教示中、困難なパフォーマンスにしていくことは学習を改善する(Schmidt and Bjork 1992)

➤ Paas and Van Merriënboer (1994)は加工された例題は教示中のパフォーマンスをよい学習に導く一方で、多様な紹介や問題はパフォーマンスを難しくするだけでなく学習をよい結果に導くことを発見した。

worked 事例は学習から気をそらせる”無関係な”認知負荷を低減するのに対し、variability は学習を強調させる”密接な”認知負荷を誘

しかしながら、この説明は”extraneous” VS ”germane”の負荷を生み出す教示の形はなんなのかという疑問をさがむ。アシスタンスジレンマの解決ではないのです。

- 数々の著者は、もっとinformation/assistanceを与えることを主張(Kirschner et al. 2006;Klahr and Nigam 2004)

➤ これらの論文での実験ではアシスタンスジレンマの対処にむけた進展を表した。

➤ これらの研究はどのような状況下でどれくらいの情報とアシスタンスを提供したのか十分であり明確でも正確でもなかった。

➤ 効果的な情報提供と情報自粛の明確な境界を定義していない

- 極端な no-dilemma (情報を与えることはいつも良い) は全ての加工された例題は交互的である加工された例題や問題解決よりもいいということを予期することになる。

➤ 交互的効果 (Trafton and Reiser's 1993)、助言の逆効果 ((Kalyuga et al. 2001)、世代的影響(Slamecka and Graf 1978).の結果のような比較を行った例はない

- この論文ではアシスタンスジレンマを Cognitive Tutors と関連して考えた。

例：高校数学(Koedinger et al. 1997)やLispプログラミング(Anderson et al. 1995)

- Cognitive Tutor は以下の 2つの理由から興味深いケーススタディを作った。
- 1. 設計者はさまざまなアシスタンスジレンマに直面し、それらを解決するための多くの手法を提案したこと。
  - それらの手法は認知科学的、学習的 ACT-R 理論の応用を通して多くの理論的裏付けがある。学習という範囲での経験的支持、効果、特徴の両方を評価した。
- 2. Cognitive Tutor はアシスタンスジレンマに関する研究のための適切なテストベッドであること。

## Brief Overview of Cognitive Tutors

すべての Cognitive Tutor は表 3 の要素を共有している。

- 表 3 : Cognitive Tutors の主なインタラクティブリティの要素
  - 1. 問題解決の環境、インタラクティブなツール
  - 2. チュートリアルガイダンス
    - a. 暗黙的な yes/no のステップごとのフィードバック
    - b. エラーメッセージのフィードバック
    - c. 次の問題の選別
  - 3. 生徒の問題解決能力に基づく適応問題選別
- 
- 図 1 に代数学の Cognitive Tutor を示す。
  - 図 2 に生徒がシナリオに沿って入力したときの GUI の表示を示す
    - エラーは “!” マークで知らせてくれる
    - おかしいときは “?” マークで知らせてクリックすると詳細を表示してくれる
  - 図 3 に生徒がヒントを要求したときの画面を示す。ボトムアップヒント
- ヒントは段階的なヒントで、まず原理的なものが説明される。次第に解答により近いヒントが与えられる。最終的には「 $2.5t+67.0$  を入力しないさい」と表示される
- 
- オンデマンドなヒントに加えて自由に閲覧できるオンラインの用語集を提供 (図 4)
  - システムは生徒ごとに問題を選別してくれる機能がある
    - Knowledge Component という知識や概念を管理して、これを基に教示を行う。
    - Knowledge Component の確率を見積もる。その確率は問題選択に利用される。
    - 学習によって獲得した Knowledge Component の評価は “スキルバー” としてシステムに表示される。完全に習得したら金色に光る。

- **Cognitive Tutor** は ACT-R 理論の体系である (Anderson 1993)。複雑な認知ドメインは独立してプロダクショナルルールと呼ばれる **Knowledge Component** の小さな単位として理解されている。
  - それゆえ、**Cognitive Tutor** は生徒の要求を助けるターゲットの能力をはっきりと表現するプロダクショナルルールを所有している。
- プロダクショナルルールについて
  - 発火する状態とそのときの行動で、ゴールやサブゴールを結んでいく行動の単位
  - ドメイン的な知識を提供するもので、生徒が問題解決するために伺われるものと同じ種類の問題を解決することが可能
- 指導的知識は二つのアルゴリズムから由来している。
  1. **Model Tracing**  
 生徒の行動を解釈するために生徒の戦略の後追い。  
 Model Tracing の結果は正解やフィードバック、生徒の推論や方略選択の教示的アドバイスを了される。
  2. **Knowledge Tracing**
    - 個々の生徒へ問題選別
    - 生徒がカリキュラムユニットですべてのコンセプトやスキル **knowledge components** をマスターしたときの決定に用いられる
- Cognitive Tutorを含んだ研究が手続き的知識としての単位がプロダクショナルルールというACT-R理論を裏付けている(Anderson 1993; Anderson et al. 1995)
  - 特に学習曲線を説明するために重要なもの。
  - 生徒はプロダクショナルルールのサイズで学習している。
  - 最初の時はプロダクショナルルールを獲得していないためもがくが、のちにプロダクショナルルールと同じ行動をとって成功的な改善を示す。これはプロダクショナルルールのサイズで学習している証拠である。

## **Cognitive Tutor** でのインタラクティブなサポートのアルゴリズム

- **Model Tracing** と **Knowledge Tracing** の 2 種類
- **Model Tracing** アルゴリズムは生徒のヒント要求と同じ形式で行われる
- **Model Tracing** の技術の ITS 応用例
  1. the Andes tutor for physics (VanLehn et al. 2005)
  2. 皮膚病の診断写真の解釈 SlideTutor(Crowleyetal. 2005; Crowley and Medvedeva 2006)
- **Knowledge Tracing**
  - プロダクショナルルールから生徒がそれらの **Knowledge Component** を持っているかの確率を見積もる。
  - 問題解決中に生徒が正しいステップを実行したか、エラーを出したか、ヒントを

要求したかを基に Knowledge Component の評価はアップデートされる

- 生徒がプロダクションルールを知っているか知らないかの単純なベイズで表現
  - ◇ ベイズは以下3つの関数をもとに Knowledge Component を知っているかを表現
    1. 問題において Knowledge Component と遭遇したかどうか。
    2. Knowledge Component を体得してないときでさえ正しい推測を行ったかの確率
    3. Knowledge Component を体得していても正しいステップを取っていない確率

### Cognitive Tutor のインタラクティブな要素の特徴化

- Cognitive Tutor はテキストの問題を解くことや、計算機での支援助けを用いた問題解決よりインタラクティブな重要性があると明らかになる。
  - step-by-step の詳細なヒントや、適応問題選別。インタラクティブティにおいてこの操作のループの単位でよく分析される。
- 人間の Tutor はより生徒とともにフレキシブルなやり取りをする能力がある
  - Cognitive Tutor より広い範囲での教示やモチベーションな戦略をもっている (Lepper and Malone 1987) が、Cognitive Tutor は問題を通して人間のチューターと同様の方法のものを持ち合わせている (Merrill et al. 1992)。

### Cognitive Tutor のインタラクティブティに対する実験的裏付け

#### 全体的な効果

幾何学の tutor (図4) で、早期的にクラスルームで使用すると、大きな学習収穫があった。

- 生徒同士で tutor を使う事での利益は観察されない、協同学習よりもより効果的な個人的な学習サポートをしているということを指し示すことは言えなかった。
  - Lisp Tutor は標準的な Lisp プログラミングの環境と比較して 30-43% の高い学習収穫や 30-64% の高い学習効果があった
- 代数科目の Cognitive Tutor の効果を評定したいくつかの研究がある。

ピッツバーグやミルウォーキーでの研究で、代数の Cognitive Tutor を利用したカリキュラムは標準的な代数カリキュラムと比較された

  - 生徒は平均して 15-25% 高いスコアをだす。
  - 問題解決や表象の利用を含むテストにおいて 50-100% 高いスコアを出した。
- ピッツバーグから集めたデータだと Cognitive Tutor を利用した生徒は 30% ほど TIMS (国際的な学力テスト) のスコアが良くなった。伝統的なコースを履修した学校の生徒よりも 227% 実世界での問題解決を含むタスクでよくなった
- Cognitive Tutor を用いた中学数学のカリキュラムの評価もまた、ほかのカリキュラムの生徒よりも優れていたことを示した (Koedinger 2002)。

### インタラクティブィを4つの要素に分けて議論

### Interactivity 1: 即時的な Yes/No フィードバックについて

- 良い教示=適切な情報量。良いバランスは実験的な課題である
- Cognitive Tutor のフィードバックの方略をデザインについて
  - ACT-R 理論では生徒が行き詰ることを最小化すべきとされている。
  - Cognitive Tutor はすぐに生徒に正しい解決ステップを教える。が、生徒が解答に取り組んでいるとき情報を授与することの潜在的利点を認めるかどうか。
  - フィードバックを先延ばしにしての情報を保留することの効果について裏付けはされない Corbett and Anderson (1995)
- Lisp Tutor を用いて、4つのフィードバックを比較
  1. Immediate Feedback
  2. Flagged Feedback (tutor はエラーを通知するが、フィードバックがないメッセージを生徒がメッセージについて聞いてくるまで)
  3. On Demand Feedback (フィードバックを要求するまでエラーはしられない)
  4. Non-Feedback (最後だけフィードバック)
  - 3つのフィードバックは速い学習をもたらす。
  - しかし、問題を完全達成する浪費時間の合計には違いが出た。
  - On Demand Feedback 条件では、生徒はほとんどにフィードバックを要求しなかった。
  - プログラミングの90%において彼らはとりあえずの解答に到達するまでフィードバックを要求しなかった。
- 即時的な Yes/no フィードバックについて保留してしまう事は逆効果である。すぐにフィードバックを返したほうが良い

### Interactivity 2: feedback content : フィードバックコンテンツ

- Yes/No フィードバックを即時に返すことは有効である
- では、さらに説明的な情報を付加させてやれば行き詰るのを防げるのではないか?
- Geometry Proof Tutor を2つの次元でのフィードバックを用いた研究(McKendree (1990) )
  1. ゴールの情報を含むフィードバック (間違った選択をしたらサブゴールを指摘)
  2. 違反的なことを指摘するフィードバック (幾何学の定義や定理の適応におけるエラーを指摘)
- ポストテストのエラーやトレーニング中のエラー率で有意差が見られた。
- 説明フィードバックを受け取った生徒は後の試行でエラーをより修正する可能性
- ポストテストでのよい成績は説明付加されたフィードバックを返すことはより明確な学習プロセスへ導くであろう (See also Anderson et al. 1995, p. 191.)
- 現在の問題に関して生徒が試み続けるのを助ける方法についてよい推測が得られたとき、フィードバックを返すことの効果的な戦略であるということである。

- 問題解決のエラーにおいて yes/no フィードバックだけでなく、より説明的な情報を与えることが支援的だということである。
  - とくにこれは学習をより明確にし、行き詰ることを低減する

### Interactivity 3: hint content and timing : ヒント内容とタイミング

- Cognitive Tutor のヒント提供についてのアプローチに関する 2つの疑問がある
  1. 次のステップを単に提供するヒントを超えて原理的な説明までを含むヒントは効果的なのか? → 効果的である
  2. 要求されたときだけヒントを与えることはいいことなのか? → よくない
- Tutor での学習手法の体得効果やフィードバックやヒントでの説明的なコンテンツの効果の評価した研究 Anderson et al. (1989)
  - 説明的フィードバックメッセージは学習を早めるが良いものではない。システムが答えを見せた後、生徒は解答の説明をものすごく余分に時間をかけて生成する
  - Aleven et al. (2003) は Schworm and Renkl (2002) はオンデマンド式のヒントや例による学習をシステムでやるのは良い方向に学習を導くと述べた
- 原理的な説明を与えることは効果的だが、Cognitive Tutor は生徒の要求においてヒントを与えるため、生徒自身が情報の受け取りや生成することのバランス考えなければならぬ (e.g., Aleven and Koedinger 2000a; Aleven et al. 2006, 2003; Baker et al. 2004; Koedinger and Anderson 1993).
- 生徒はタイムリーな時にアシスタンスや情報を探すことが得意ではない。
  - 例えば、幾何学のチューター (図 1) のログデータから、生徒は答えに近いボトムアウトヒントをよく使う
  - help seeking の疎さで、他に例えば多重なエラーの後でさえも、ヘルプを要求しないことがある。
- Cognitive Tutor のデザインで 2つ異なった手法が存在する
  1. チューターが良い情報の保留とヒントの提供の間の良いバランスを達成するためのデザインがある。たとえば問題解決次のエラーのあとに積極的な情報のさらなる提供を作成する
  2. ドメインスペシフィックな knowledge Components を表現するだけでなく help seeking のスキルを要求する Tutoring を提供するためにシステムを改良すること。
    - メタ認知教示は生徒自身に情報やアシスタンスの対立を探すときのバランスを学ばせるためにある。

### Interactivity 4: ナレッジコンポーネント評価と学習体得

- Cognitive Tutor は生徒がすべての Knowledge Component を習得できたという十分な証拠が積み重なるまで個人別問題選択を行う。

以下3つは tutor のメタなレベルでの学習法の獲得の効果の証拠を示した。

1. モデルトレーニングやナレッジトレーニングが同時に働くアルゴリズムは正確に生徒のスキルを評価しているかどうか、Tutor の問題選択アルゴリズムが適切に問題を選択しているかどうかである
  2. Lisp の Cognitive Tutor の研究は Tutor のマスターラーニングメソッドは学習の改善を導いているとあらわした
    - 精通した状態の生徒は重大なより高い学習収穫をもっていた。Tutor は重複していない問題にアサインすることに成功していることを確認
  3. 生徒の消費する時間を考慮したこと。マスターラーニングメソッドは少ない時間で大きな学習効果をもたらした (Corbett 2001)
- マスタリーレベルの基準を設けることは” gaming” 戦略を無駄なものさせる。
  - スキルバーを見せることはモチベーションや学習利益を提供しているか？
  - 問題選択や学習のペース配分のためのナレッジトレーニングよりも利益を得ているか

## Implications

情報を与えることと保留することでは、yes/no feedback, feedback content, and hint timing のこの研究の結果は一貫して問題解決のステップ後に情報を与えることは保留することよりも良いと示した。

- 情報は非常に慎重に差し控えされるべきである。教師ありの問題解決を改善する研究が情報保留する手法より、より多くの情報を授与する手法に焦点を当てたならばより成功的になる。

## Enhancing Cognitive Tutors: What Does and Does Not Work

多くの研究は加工された例題を一人で問題解決することの利益を示している。(Atkinson et al. 2000; Renkl 2002; Sweller and Cooper 1985; Trafton and Reiser 1993; Zhu and Simon 1987)

- Mathan は Excel プログラミングの Cognitive Tutor の中で改良された例題を追加することの調査を始めた。
  - 加工された例題の性質 (表 1 のセル 4 passive/implicit) は生徒がステップを行うようなよりアクティブな形。それは以下を含んでいる。
    1. それらのステップの推論についての問題に答えること
    2. ステップ自身を試すこと
  - 4つの比較のうち2つに有意差が見られた。

- McLaren らは化学の Cognitive Tutor での教師有問題の間に Worked example を取り入れた。
  - 例題の利点が薄れ、生徒の専門性を磨くような “逆転の効果” の結論はない。
  - 問題解決の活動は指導され、インタラクティブなのだ。
  - 先行研究において Worked example や問題解決（表1のセル4セル5）は情報を与えること（examples）や保留すること（practice）の良いミックスを表現するため改善する（セル5アクティブ・暗黙）
  - 問題解決の実践は単独で多くの情報保留することのコストを所有している（表2）
  - エラーや行き詰まりに対するコストはエラーを起こしたとき即座に合図されるので減少した。
  - 教師無し問題解決は勉強を始めた学習者に十分な情報提供がないことに悩まされ、インタラクティブな tutoring 問題解決は十分な即座なステップごとのフィードバックや生徒がいま必要な情報を提供する。
  - 教師有り問題解決に関しては、交互的な加工された例題の学習において、情報を与えることは学習開始時には必要以上のことで、学習の後半であると多すぎる
- McLaren らの結果とは対照的に、Scwonke らは Worked examples を幾何学の C T に追加する利点を発見した。
  - ” faded ” example という形で解決する例の勉強から問題解決への段階的移行を実装した (Renkl et al. 2004)。
    - 生徒は初めに問題のすべてのステップの解答が与えられた例題を見る。つづく例題の答えは徐々に無くなる
    - 生徒は実行したステップや説明フィードバックを受け取ることを尋ねられる。
    - faded-example の生徒は手続き的知識においての等しい結果や宣言的知識におけるポストテスト結果をよくすることを少ない教示で大いに取得し効率的に学習した。

## Exploring self-explanation in cognitive tutors

効果的な学習である複雑な推論や問題解決において” self-explanation ” の仕方

- Chi らは乏しい学習者が worked example をテキストブックから吸い出し、浅い類推をするのに対し、良い学習者は次の一歩から自分自身を説明し、問題解決の練習中では深い類推をしていることを発見

表1で、implicit example-based induction は学習において協力である（セル4、受け身／暗黙的）、より明確的な自己説明を背景にルールベースの推論によって協調されることができる（セル2、能動的／明確）

- VanLehn らは良い対悪い学習者の計算機モデルを用意した。良い学習者が直感的なステップを埋めるために既存や連鎖的な知識による例において、ステップ間のギャップを

埋めることに挑戦する傾向があると示した。

- 自己説明のサポートはクラスルームでの使用で効果的になる。” gone to scale” 幾何学授業の Cognitive Tutor の一部として実装された (Aleven and Koedinger 2002)
- フィードバックありの自己説明とフィードバックなしの自己説明を対照的にする。
  - 過去の研究のほとんどは、自己説明が正確であるかどうかのフィードバックを学生に提供することなく、自己説明の入力を求めている。
  - この設定は自己説明に取り組むように要求に応じる生徒のために社会的な要望を生み出した。フィードバックがない自己説明は実験者の存在や暗黙的な”重要特性”のためこの研究の設定で動く。
  -
- 筆者らはジオメトリCTにおいて新しい自己説明のバージョンを開発。
  - 用語集における理由を参照するかわりに原理の自己説明（例えば、三角形の内角の合計は $180^\circ$ です）を促される。
- 最初の高校でのパイロットスタディがある (Aleven and Koedinger 2000b)
  - 自己説明に対してフィードバックがないと、理由づけられた試みをめったにしかない（10%以下）
  - 不適応な説明”だめだったらだめ”みたいなオフタスクなレスポンスをする。
- これらの研究はフィードバックなしでは自己説明を約束するようなモチベーション、自己鍛錬、メタ認知能力を失うことを発見。
- 初めから最適な学習をする生徒はテキストのような記述や例題をやるような教示の形式（表1のセル1やセル4）で無理なく効率的に学習を行う。
  - 乏しい学生をより直接的なメタ教示によってよい学習者にさせることができるのだろうか？？

## Other cognitive tutor studies of metacognitive support

- *Improving learning through error self-detection and correction*
  - いままでに紹介した Lisp の Tutor における即時的なフィードバックの成功的なものに関わらず、多くは Tutor において即時のフィードバックの意見の批判を持っていた。
  - 多くの批判は Cognitive Tutor を情報やアシストを与えることと保留することの単純な2分化として評価した。
  - Tutor における問題解決中に即座なフィードバックはテキストや加工された例題よりもより情報を差し控えるのではないかという評価。

それにも拘わらず、即座なフィードバックは生徒自身のエラーやそれらから学ぶことの機会を生徒から奪う可能性がある (Schmidt and Bjork 1992)。

- 遅れたフィードバックは生徒の学習を改善するかどうかを検討した (Mathan and Koedinger 2005)

- 生徒の反応と tutor のフィードバックの間の遅延について”望まれたパフォーマンスのモデル”というゴールの教示としてフィードバックを再解釈した。
- ” 知的なノービス” をプロデュースする。
  - 知的なノービスとは最初にエラーを起こすかもしれないが、エラーを調査し、修正することができるものである。
- 生徒は知的なノービスの Tutor からもっと学ぶ。
  - ” 評価” というメタ認知スキルの向上によるものではない。
  - 学習曲線分析ではフィードバックを体験する最初の機会のあとに利益がでた
  - 加えて評価スキルにおける向上は
    - ✧ ポストテストの結果のパターンと合致
    - ✧ ロバストな学習の尺度、長いタームの記憶や移動のようなものを分割。
  - 優秀な生徒は明確に、どのように最初の試みがエラーにつながったのか、どのように彼らが望まれるべき出力に対して修正したのかについて説明。
    - ✧ この解釈は Siegler’ s (2002)によって堅実になった。自己説明は正しい解決だけでなく、間違っただけの解決もそれらの学習を改善させる。
    - ✧ 学習者は正しい knowledge component を強化したり獲得するだけでなく、間違っただけの知識を弱めたり除く。
  - Nathan (1998)は生徒がエラーの意味のある最終に近い結果を参照するフィードバックを提供する利益を実践し
  - Ohlsson(1996)は詳細な理論を提唱し、エラーによる学習の計算機モデルを提供した。
- *Cognitive Tutors for improving help-seeking skills and reducing ‘gaming’*
  - help-seeking のスキルを改善させることや学習の振る舞いの非生産性を減少させることを調査。
  - ヒントを尋ねたり推測することを繰り返すことで考えを放棄。このような振る舞いを”ゲーミングシステム”という (Baker et al. 2004)。
- 生徒は、必要時のヘルプシーキングをしないことによって情報授与を避ける。
  - 早期的にヘルプシーキングを改善し、ゲーミングを減少させる Cognitive Tutor を作ることは多くの利益を出す。
  - 筆者らは、help-seeking tutor を開発し、幾何学 Cognitive Tutor にそれを統合した。
  - システムは生徒の help seeking の振る舞いのフィードバックを提供した。
  - 多くの教示中の乏しいヘルプシーキングの振る舞いを減少することを示した
  - しかし、幾何学の学習では向上がなかった(Rollet al. 2006)。

- データ分析 Cognitive Tutor は gaming behavior を検出したら、アニメーションのエージェントの感情ディスプレイや関連するマテリアルの補充問題を提供することで反応した。

gaming tutor の生徒は全体的な gaming behavior 現象さえ、受け取った多くの補充問題は良い学習者に結び付けられた。

しかしながら、ドメイン学習における違いは統計的には信頼性がなく、有害なゲーミングはおそらく生徒のごく一部（10%）で存在しているのである。

Crowley and Medvedeva (2006) はインテリジェント tutor を使うことが医学部生のサブセットがカリキュラムの早期の問題中では gaming-like な振る舞いに従事し、後半では問題において成功の証拠立てをした。

## Conclusion

- 情報の保留やアシスタンスの利点は多く存在。
  - 知識の構築、効果を生むメリット、ぼけっとすることの低減、長期記憶への従事、知っている考えをしないこと防ぐ知識のチェックを提供する。

- コストも多く存在する。

Cognitive Tutor でどのように実験するかの実例により、アシスタンスを与えることや控えることのトレードオフを発見した。

- システムは相互作用的に情報を加えていき、必要な時だけ、yes/no フィードバック、説明的ヒント、ダイナミックな問題選択をする。
- このような情報の授与と保留のバランスや個人個人でのインタラクティブな要素にたいしてアプローチしている裏付け先行研究が行ったが、この結果はジェネラルな結果としては捉えてはいけない。理想的なバランスに対して主張していないのである。
- faded example（わずかな情報を与える形式）や知的なノービスモデルによるフィードバック（わずかな情報保留の形式）のような要素は既存のアプローチよりも効果的になるだろうと発見。
- 理想的なエラーの確率の閾値は何？が今後の課題でもある。
  - Pavlik (2007) は実践的実験の間での遅延の長さを決定するために 5-25%の理想的エラー率を提案。
- 定性的な状況の特徴づけにむけた努力や教示デザイン、インストラクターがよい決断をすることが出来る定性的な閾値の基準を実験的研究や理論的統合によって達成するために必要なのである。