2005. 9. 6
森田純哉

Modeling Parallelization and Flexibility Improvement in Skill Acquisition: From Dual Task to Complex Dynamic Skills

Niels Taatgen
Cognitive Science 29 (2005) 421–455

概要

1. Introduction: 検討課題・タスク領域・レビュー
2. Model of perfect time sharing: 単純なタスクを対象としたスキル獲得
3. Skill acquisition in complex dynamic tasks: 複雑なタスクを対象としたスキル獲得
4. Conclusion: まとめとインプリケーション

Keywords: Dual tasking, Psychology, Cognitive architecture, Complex systems, Human–computer interaction, Instruction, Learning, Situated cognition, Skill acquisition and learning, Knowledge representation, Symbolic computational modeling

1. Introduction（検討課題・タスク領域・レビュー）

- ルールベースモデル（プロダクションシステム）
  - 人間の認知の伝統的な表現
  - 特に推論プロセスを正確に説明

- ルールベースモデルへの批判
  - 批判 1: 問題解決に必要な知識の詰め込み
    - パフォーマンスとプロセスに関する説明のみで，知識獲得に関する説明なし
  - 批判 2: 脆さ（柔軟性・適応性の表現が困難）
    - 保持するルールに限定されたモデルの行動
    - 現実場面（予測不可能なことがおきる状況）への適用が困難
      *ニューラルネットは適応性に優れるが，教示や背景知識の表現が困難

ルール表現そのものではなく，ルールベースモデルの多くが採用する仮定が問題

- 従来の仮定と本稿で提案するアプローチ
  - 仮定 1: 心理実験の教示を直接的にルールに翻訳
    - 被験者は教示をただちに理解できるのか？実験の中で徐々に理解するのでは？
    →モデルを動かす中で教示を徐々にルールに翻訳
  - 仮定 2: 階層的に分割されたタスク構造の実装
    - トップダウン的な行動の統制→脆さの原因
      - 世界の状況が変化してもモデルはプランを貫く
        *多くのタスクでは内的表象が必要なため，ボトムアップのみも困難
    →最小統制の原則 (minimal control principle)に基づくモデリング
      - 統制状態 (control states) の数を最小化
        *統制状態＝目標状態・副目標
      - 基本はボトムアップ（環境から駆動されるルールの実行）
      - トップダウンは最小限（ボトムアップが行き詰ったときのみ）

- 提案アプローチを適用するタスク領域: マルチタスクにおけるスキル獲得
  - マルチタスク:

- 並列行動の生起
  *並列行動の階層的スケジューリングが困難
- スキル獲得:
  - 認知モデル研究の中心的トピックだが，速度の向上に関する研究のみで，プロセスレベルの変化を説明した研究少

→提案アプローチと従来アプローチの比較に適したタスク領域

1.1. Cognitive models of skill acquisition（スキル獲得のモデル）

- スキル獲得のモデル 1 [ACT: Anderson, 1982]
  - スキル獲得とは
    - 宣言的知識から手続き知識へ移行すること
      - ノービス: 宣言的知識を利用した問題解決
      - エキスパート: 手続き知識を利用した問題解決
  - 学習メカニズム
    - 宣言的知識のタイプ
      - 事実: 3 + 4 = 7
      - 過去経験: "クローゼットのスイッチが壊れた"
      - 教示: "ヤカンに水を注ぐ"→"沸騰するまでストーブに乗せる"→"ティーポットに茶葉を入れる"→"ティーポットにお湯を注ぐ"→"3-5 分待つ"（お茶の入れ方）
        *本稿では“教示”に注目
    - 宣言的知識の利点
      - 宣言的知識は単一アイテムとして貯蔵（簡易な貯蔵）
        *宣言的知識の獲得がスキル獲得の第 1 段階である理由
    - 宣言的知識の難点
      - 環境の変化に対して間接的（ルールに介在）→速度の遅延
      - 並列行動に関わる記載なし→系列的なプロセス
    - 宣言的知識から手続き知識（ルール）への移行
      - ACT*: 手続き化・組織化・特別化・一般化
      - ACT-R: コンパイル（複数ルールのまとめ上げ）

- スキル獲得のモデル 2 [SOAR: Newell & Rosenbloom, 1981]
  - スキル獲得とは
    - 一般的ルールから領域固有のルールが学習されること
      - ノービス: 一般的なルール・領域知識を利用した問題解決
      - エキスパート: 領域に特化してチャンク化されたルールを利用した問題解決
  - 学習のメカニズム
    - インパス中の行動をトレース
    - トレースされたルール系列からチャンクの構成（ショートカットの作成）

- スキル獲得の認知モデル 3 [Instance Theory: Logan, 1988]
  - スキル獲得とは
    - 事例の蓄積により事例ベースの行動に移行すること
      - ノービス: 一般的アルゴリズムで問題解決
      - エキスパート: 事例検索によって問題解決
  - 学習のメカニズム
    - 問題解決事例を貯蔵
    - 類似問題に対する直接的な解の当てはめ

- スキル獲得のモデル 4 [著者の先行研究: Taatgen & Lee, 2002] （ACT-R を利用）
  - スキル獲得とは
    - 教示から徐々に手続き知識を学習すること
      - ノービス: 教示の検索に基づく問題解決
      - エキスパート: ルールの実行に基づく問題解決
  - シミュレーション
    - タスク領域: Kanfer-Ackermann Air Traffic Controller task
    - 大雑把な結果: 遂行時間・得点をよく再現
    - 問題点: 階層的なタスク表現→並列行動の再現ができない

- スキル獲得のモデル: 総括
  - 共通点
    - パフォーマンスの向上・エラーの低減
    - 一般的な方略から領域固有の方略へ
  - 相違点
    - 学習メカニズム（コンパイル・チャンキング・事例）

1.2. Models of parallel behavior（並列行動のモデル）

- 並列行動のモデル化: 非同期的に複数モジュールが並列・独立に動作
  - 並列行動のモデル 1 [CPM-GOMS: Gray, et. al., 1993]
    - 認知・知覚・運動が並列モジュール・モジュール内では系列
  - 並列行動のモデル 2 [EPIC: Meyer & Kieras, 1997]
    - CPM-GOMS を引き継ぐが，認知モジュールは内部で並列
  - 並列行動のモデル 3 [ACT-R: Byrne & Anderson, 2001]
    - EPIC のモジュールを使用するが，認知モジュール内部は系列

- 並列行動の実験的検討
  - 並列行動の実験的検討 1 [デュアルタスクパラダイム: e.g., Pashler, 1994]
    - 実験状況
      - 2 つの刺激を同時に提示→両方の刺激にできるだけ早く反応
      - 一方のタスクにプライオリティを置く教示
    - 結果
      - 第 1 タスク: デュアルタスクコストなし（単一課題時と遂行時間の差なし）
      - 第 2 タスク: デュアルタスクコストあり（単一課題時より遂行時間が増加）
  - 並列行動の実験的検討 2 [Schumacher et al., 2001]
    - 問題意識
      - デュアルタスクコストは教示のアーティファクト？
      - プライオリティに関する教示がデュアルタスクコストを増加させた？
      - 第 1 タスクの反応を確認するまでは第 2 タスクの反応を確定しなかった？
    - 実験
      - プライオリティなしのデュアルタスク
    - 結果
      - スキル獲得の初期段階: デュアルタスクコストあり
      - スキル獲得の後期段階: デュアルタスクコストなし
        独立に行ったときと同じ速度で 2 つのタスクを実行可能に

      →2 章におけるタスク領域

- 単一タスク内での並列行動 (Within-task multitasking)
  - 単一タスクであっても，複数のステップが並列に進むことで効率的に
    e.g., 沸騰を待ちながら，茶葉を入れる
    →3 章におけるタスク領域

1.3. Models of learning parallel behavior（並列行動の学習モデル）

- 並列行動の学習モデル 1 [Instance theory: Logan, 1988]
  - 事例ベースの問題解決が注意を減らし並列行動が生起
    - 事例ベース: 視覚刺激のエンコードと事例検索のサイクル（注意必要なし）

- 並列行動の学習モデル 2 [EPIC: Meyer & Kieras, 1997]
  - 5 段階の学習を経て並列行動が生起
    - ステージ 0: 宣言的知識の検索と解釈が必要
    - ステージ 1: 中央実行系によるマルチタスクのスケジューリング
    - ステージ 2-4: タスク固有のスケジューラ・中央実行系の関与が減少

- 並列行動の学習モデル 3 [EPIC-SOAR: Chong & Larid, 1997]
  - EPIC の知覚-運動能力・SOAR の問題解決能力
  - 2 つの視覚刺激が競合するマルチタスクの実施
  - タスク間の競合解消に使用されるルールがチャンキングされることで並列行動が生起

- 並列行動の学習モデル: 総括
  - Instance Theory: 記憶検索のモジュールのみ
  - EPIC: 学習メカニズムの実装なし
  - EPIC-SOAR: 競合解消に学習メカニズムが限定

2. A model of perfect time-sharing（単純なタスクでのスキル獲得）

2.1. The central production system, modules, and buffers（アーキテクチャ・タスク領域）

- ACT-R の概要 (Fig. 1)
  - 伝統的プロダクションシステムとの差異
    - fMRI データとの結びつき (Anderson, 2005)
    - 外界（実験ソフトウェア）とのインタラクションが可能 (EPIC からの Perceptual-Motor module の移植)
    - ゴールスタックの削除（心理データとの非一貫性を理由）
  - アーキテクチャの概要
    - 中央認知系
      - プロダクションシステム
        - モジュール間のスイッチングボード
        - 各バッファの状態とマッチするルールを選択，発火（バッファの書き換え）
      - Declarative Module
        - 宣言的知識の保持
        - ルールからの知識検索の要求
          *宣言的知識の一部をキューにした検索 (i.e., 3 + 4 = ?)
        - Retrieval Buffer 内での知識の一時的な保持（ルールによって利用）
      - Goal state Buffer

        - 統制状態（課題構造における現在位置）を保持し，ルール選択のコントロール
          *統制状態の可能な値の数が最小→最小統制
    - 知覚・運動系 (Perceptual-Motor Module)
        - Visual Module
            - スクリーン変化への気づき（初期知覚）
            - スクリーン中の特定位置へ注目（眼球運動）
            - 注目位置内の情報を Visual Buffer へ転送（感覚記憶へのエンコード）
        - Manual Module
            - Manual Buffer にルールからの要求を一時保持
            - 実験ソフトウェアに対するキー押し
        - Aural Module（Fig. 1 では省略）
            - 実験ソフトウェアから出力される音声の知覚
            - Aural buffer へのエンコード
        - Vocal Module（Fig. 1 では省略）
            - Vocal Buffer にルールからの要求を一時保持
            - 発声
- モジュール間の関係
    - 中央認知（Goal・プロダクション・宣言的知識）・知覚モジュール・運動モジュールは独立・並列に動作
    - モジュール内の動作は系列的
        - 一度に一つのルール・ゴール状態・検索知識・一度に 1 つの注目領域
    - 各動作の実行速度はそれぞれのモジュールの質に依存
        - 1 つのルールの発火は 50msec

- タスク領域: Schumacher et al., 2001
    - 2 つのタスクの並列
        - Visual-Manual Task
            - ○が画面の右・中央・左のいずれかに出現
            - ○の位置に応じたキー押し
              *[○－－]→左のキー，[－○－]→中央のキー，[－－○]→右のキー
        - Aural-Vocal Task
            - 220Hz, 880Hz, 3520Hz のトーンの提示（40msec の持続時間）
            - トーンの高さに応じて発声
              *220Hz→1 と発声，880Hz→2 と発声，3520Hz→3 と発声
    - 実験試行
        - 単一ブロック
            - Visual-Manual Task，Aural-Vocal Task のいずれかのみを繰り返し
        - 複合ブロック
            - デュアルタスクとシングルタスクが混在
                - Dual task trial: 同時に 2 つの刺激が提示 ("Visual" and "Aural")
                - Single task trial: 単一の刺激のみ提示 ("Visual" or "Aural")

          *どちらのトライアル・タスクが提示されるか予測不可能

- Fig. 2: エキスパートモデル
    - デュアルタスクコストなしの並列行動＝完全な時分割処理 (Perfect time sharing) の達成
        - 第 1 ステップ (0 秒)

- Visual Module
  - スクリーン変化の認識（Visual buffer の更新）

- 第 2 ステップ
  - *Compiled-attention-rule* の発火
    - Visual buffer の更新を条件
    - Visual buffer へ刺激の注目・エンコードを要求
  - トーンの認識（Aural buffer の更新）
- 第 3 ステップ
  - *Attend aural* の発火
    - Aural buffer の更新を条件
    - Aural buffer へトーンのエンコードを要求
- 第 4 ステップ
  - Visual module
    - 視覚刺激への注目・エンコードの完了
    - Visual buffer の更新 [o - -]
- 第 5 ステップ
  - *Extract-index-finger* の発火
    - Visual buffer の更新を条件
    - Manual buffer へ[左キー]を押すことを要求
- 第 6 ステップ
  - Manual module
    - キー押しの開始
- 第 7 ステップ
  - Aural module
    - トーンのエンコードが完了
    - Aural buffer の更新 [Low pitch]
- 第 8 ステップ
  - *Low-pitch-rule* の発火
    - Vocal buffer の更新を条件
    - Vocal buffer へ[1]と発声することを要求
- 第 9 ステップ
  - Vocal module
    - 発声の開始
- 第 10 ステップ
  - *Ready* の発火
    - Manual と Vocal の完了を条件

2.2. The role of declarative memory（ノービスのプロセス）

- Fig. 3: ノービスモデル
  - デュアルタスクコストあり: 太線/細線のプロセスのギャップ（待ち時間）
    - 原因: 宣言的知識（教示）の検索・解釈・実行の関与
      - プロダクションルールによる要求に応じて受動的に検索
      - 解釈・実行が環境の変化に対して間接的

- 教示の実装（[ ]内は Fig. 3 と対応）
  1. 聴覚刺激に注意を向けなさい [*Attend any aural stimulus*]
  2. 視覚刺激に注意を向けなさい [*Attend any visual stimulus*]

3. 聴覚刺激に注意が向いているのなら,
   刺激に対する反応を検索 [*Retrieve Response*]
   刺激に対する反応を生成 [*Low Pitch="One", Middle Pitch="Two", High Pitch="Three"*]
4. 視覚刺激に注意を向いているのなら,
   刺激に対する反応を検索 [*Extract location*]
   *視覚刺激の位置とキー位置が対応するため，反応を生成する宣言的知識なし

- 宣言的知識（教示）の検索・解釈・実行
  - 最小統制の原則
    - ゴール構造による教示検索の統制なし
    - 環境の変化に応じた利用
      *トライアルによって利用される教示の差異

- 宣言的知識（教示）の検索と実行を媒介するルール
  - 宣言的知識の検索に関わるルール（Fig. 3: Declarative への矢印）
    - *Get-next-visual-location-instruction-rule*
      IF Goal buffer に"タスク"が入っている
      AND Visual buffer に"刺激"が入っている
      THEN "タスク"と"刺激"に関係した教示を Retrieval buffer へ要求
    - その他，可能なイベントタイプに応じた 3 種類のルール
      *Visual buffer に"エンコード済みの刺激"・Aural buffer に"刺激"・Aural buffer に"エンコード済みの刺激"
  - 宣言的知識の解釈・実行に関わるルール（Fig. 3: Declarative からの矢印）
    - *Attend-visual-stimulus*
      IF Goal buffer に"タスク"が入っている
      AND Retrieval buffer に[*Attend any visual stimulus*]が入っている
      AND Visual buffer に"刺激の位置"が入っている
      THEN "刺激の位置"に注意を向けるよう Visual buffer へ要求
    - その他のルール（括弧内は Fig. 3 と対応）
      - 聴覚刺激へ注意をむけるためのルール [*Attend Aural*]
      - 音声と単語を結びつけるためのルール [*Retrieve Response*]
      - 検索された単語を発声するためのルール [*Say Retrieve*]
      - 視覚刺激 [○- -] と対応するキー [左] を押すルール [*Extract index finger*]
      - 視覚刺激 [-○-] と対応するキー [中央] を押すルール [*Extract index finger*]
      - 視覚刺激 [- -○] と対応するキー [右] を押すルール [*Extract index finger*]

    *視覚刺激の位置とキー位置は対応するため直接的なルールを用意

- 宣言的知識の利用に関連したパラメータ
  - トーンの有無の判断に要する時間: 50msec
  - トーンのピッチの判断に要する時間: 200msec (デフォルト 280)
  - 視覚刺激に注意を向けるまでの時間: 100msec (デフォルト 85)
  - 発声と運動時間: デフォルト

2.3. From declarative to procedural knowledge: Production compilation

- ACT-R におけるルールの学習 = コンパイル
  - 時間的に連続した 2 つのルールの統合
    - 第 1 ルールの条件部に第 2 ルールの実行部

- 宣言的知識の検索・解釈・実行がスキップ
  - 環境から直接的に駆動され，直接的に働きかけるルールが構成
- 知覚モジュール・運動モジュールはスキップされない
  *外界の状況に応じてアウトプットが変化するため
- 宣言的知識の検索が全てスキップ→Fig. 2

- コンパイルの例 1
  - *Get-next-visual-location-instruction-rule* + *Attend-visual-stimulus* → *Compiled-attention-rule*
    IF　Goal buffer に[Schumacher タスク]が入っている
    AND　Visual buffer に"刺激"が入っている
    THEN　"刺激の位置"に注意を向けるよう Visual buffer へ要求

- コンパイルの例 2
  - 2 つのルールのコンパイル→コンパイルされたルールと他のルールのコンパイル
    Learned-Low-Pitch-rule
    IF　Goal buffer に[Schumacher タスク]が入っている
    AND　Aural buffer に[Low-pith]が入っている
    THEN　[1]と発声するように Vocal buffer へ要求

2.4. Gradual learning of production rules

- ACT-R におけるサブシンボリックな学習
  - 漸進的な学習を実現
    *コンパイルのみでは飛躍的にエキスパートへ
  - ユーティリティ: 競合解消に用いられるパラメータ
    - 要素 1: "ルール"を利用して"ゴール"到達にかかるコスト
    - 要素 2: "ルール"を利用して"ゴール"到達に至る確率
      *[期待値] = [成功確率][定数] – [コスト] ± [ノイズ]（2005/03/09 寺井さん）
  - ACT-R における競合解消
    - 最もユーティリティの高いルールが最も高い選択確率
    - 最もユーティリティの高いルールが必ず選択されるわけではない
      *競合解消のノイズによってエラーと発達が生起

- コンパイルされた新ルールのユーティティ
  - 始めは最低のユーティリティ
  - コンパイルの繰り返しによってユーティリティが漸進的に向上

- 新ルールのユーティティに関する形式的定義

$$U_p = \frac{m*U_{p,\,prior} + experiences*U_{experienced}}{m + experiences}$$

m:固定値 10（デフォルト）
$U_{p,prior}$: 親ルールのユーティリティから引き継ぐ要素（初期値-20 から徐々に親ルールのユーティリティに近づく）
experiences: 新ルールが選択された回数（新ルールが利用されるまで 0）
$U_{expericences}$: これまでの各選択において新ルールが得たユーティリティの平均

*新ルールのユーティリティは親ルールのユーティリティと自身のユーティリティを要素
*初期段階では *experiences* が 0 で，$U_{p,prior}$ の影響が強い
*一旦，新ルールが利用されはじめると $U_{expericences}$ の影響が表れる

- 親ルールの影響について

$$U_{p,prior}(t+1) = U_{p,prior}(t) + \alpha[U_{parent}(t) - U_{p,prior}(t)]$$

$U_{p,prior}(t+1)$: 次回コンパイル時の $U_{p,prior}$
$U_{parent}(t)$: 親ルールのユーティリティ
α: 学習速度を決定する少数．デフォルトは 0.05

* U_{parent} と $U_{p,prior}$ の差分が次回の $U_{p,prior}$ に加算
*$[U_{parent}(t) - U_{p,prior}(t)]$は t の増加に伴い減少するため，漸進的に $U_{p,prior}$ が U_{parent} に近づく

- コンパイル回数(x)とユーティリティ(y)の 関係 (Fig. 4)
  - x 軸と平行な細い直線: 親ルールのユーティリティ U_{parent}
    - U_{parent} = 10 に固定
  - 漸進的に上昇する太い曲線: 新ルールのユーティリティ U_p
    - 初期値-20（グラフ外）から，漸進的に新ルールの真のユーティリティに近づく
    - 新ルールの真のユーティリティは 10.1 ($U_{experiences}$ 10.1)
      *親ルールに比べ新ルールは平均 100msec の短縮 (ルール実行 + 宣言的知識の検索)
  - S 字カーブ (x-x-x-x-x-x-x): 新ルールの選択確率
    - 親ルールの発火ごとに新ルールが生成されるが，U_p が低いうちは利用されない
    - U_p が U_{parent} に近づくに従い，新ルールの選択確率が生起（100 回目付近）
    - 一旦，選択が開始すると U_p が上昇（$U_{p,prior}$ を上回る）
    - それを受けて選択確率が上昇（相互作用的に影響）
    - 新ルールの選択確率が 50%に至った時点で U_{parent} を上回る（125 回目付近）

2.5. Experiment 1 (Schumacher et al., 2001)

- ブロックの構成
  - 単一ブロック: 48 試行
  - 複合ブロック: 48 試行（Dual task trial×18; Single task trial×30）

- 実験の流れ
  - 1 日目: 単一ブロック×6（Visual-Manual を 6 ブロック，Aural-Vocal を 6 ブロック）
  - 2 日目: 単一ブロック×6 + 複合ブロック×8
  - 3-5 日目: 単一ブロック×6 + 複合ブロック×10

- 結果 (Fig. 5)
  - 知見 1
    - 初期段階では Visual-Manual と Aural-Vocal の速度差が大きい（グラフのスケール）
      →Visual-Manual は容易（刺激位置とキー位置が対応）
  - 知見 2
    - 初期段階では Visual-Manual での Dual-Single 差が Aural-Vocal での Dual-Single 差に

比べて大きい（■-■, □-□と◆-◆, ◇-◇の違い）
→初期段階でのデュアルタスクコストの差の違い

- 知見 3
  - 後期段階では Visual-Manual・Aural-Vocal ともに Dual-Single 差がなくなる
    →後期段階でのデュアルタスクコストの消滅（完全な並列行動の達成）
- 知見 4
  - モデルとデータがよく一致
    →5 日目のモデルのトレースはほぼ Fig. 2
    ルールのコンパイル・完全な時分割処理としてパフォーマンスを説明

2.6. Experiment 2 (Schumacher et al., 2001)

- 実験の目的
  - タスク間でのプライオリティがある場合を検討
    *p3. 1.2 節における仮説を検討した実験

- 被験者
  - 実験 1 の後に実験 1 と同じ被験者に対して Dual task trial を実施

- 実験 1 との違い
  - Vocal の後に Motor を出力するように明示的に教示
  - Aural の後に Visual を提示 (SOA: 50msec, 150msec, 250msec, 500msec, 1000msec)
    *SOA の増加に応じて Single task trial の状況に近づく

- モデルの調整
  - Vocal の後に Manual を実行するようにトップダウンな制約を導入
    *ゴールバッファに格納されるゴール構造を変更

- 実験結果 (Fig. 6)
  - Aural-Vocal task（第 1 タスク）
    - SOA による変化なし（Fig. 6 の◆-◆, ◇-◇）
  - Visual-Vocal task
    - SOA の増加 (Single task trial に近づく) に応じ遂行時間が減少（Fig. 6 の■-■, □-□）

    →教示によってデュアルタスクコストが生起
    デュアルタスクコストは教示のアーティファクト

2.7. Experiment 3 (Schumacher et al., 2001)

- 中央認知の並列性を検討
  - Visual-Manual task は Aural-Vocal task に比べ容易
    * Visual-Manual task はほとんど中央認知（ルール）に負荷をかけない
    *アーキテクチャとして中央認知の並列性（EPIC）・系列性（ACT-R）の区別がつかない状況
  - Visual-Motor task を困難にしたときに中央認知のボトルネックが生じるか
    *中央認知のボトルネック→系列性（ACT-R）を示す

- 実験の方法
  - 刺激提示位置を 4 つに増やす
  - キーの位置が刺激提示位置に対して逆転（対応の非一貫性）

*[○－－－]→右端のキー，[－○－－]→右から 2 番目のキー…
- 実験の流れは実験 1 と基本的に同様（6 日目を追加）

- モデルの調整
  - 刺激提示位置とキー位置の対応に関する教示を宣言的知識に追加
  - Manual の実行ルールの発火に遅延を設定（0.05 から 0.2）
    *対応の非一貫性に伴う遅延効果に対応

- 結果 (Fig7)
  - Aural-Vocal Task でデュアルタスクコスト（Fig. 7a における 6 日目）
    →完全な時分割処理を獲得できなかった
    *中央認知のボトルネックが生じた？中央認知の系列性？

- 追加分析 (Fig. 8)
  - 最終日にデュアルタスクコストが存在した被験者（5 名）と存在しなかった被験者（4 名）
  - デュアルタスクコストの存在しなかった被験者に対応するモデル
    - Manual の開始遅延時間を 0.2 から 0.1 に設定
  - デュアルタスクコストの存在した被験者に対応するモデル
    - Manual の開始遅延時間を 0.35 に設定

    →ルール実行の遅延時間のパラメータがデュアルタスクコストの個人差を説明
    中央認知の系列性？更なる検討が必要

2.8. Discussion
- 2 章で示したモデルの特徴
  - 最小統制の原則に基づく
    - 教示順序の階層化なし
    - 単一の統制状態（課題遂行中のゴールの更新なし）
      *実験 2 を除く
    - 状況に独立な教示から環境から直接的に駆動されるルールが学習
  - 最小統制の原則に基づく順序の独立性が並列行動を導いた
    *環境の変化に敏感に反応しなければならないためトップダウンなモデルは困難

3. Skill acquisition in complex dynamic tasks（複雑なタスクでのスキル獲得）

- タスク領域: CMU-ASP task (Georgia Tech Aegis Simulation Program (Hodge et. al., 1995) の発展)
  - 課題
    - イージス艦（防空巡洋艦）のオペレータの役割
    - 航空機を制御するレーダを使用し，航空機の分類
  - インタフェース (Fig. 9)
    - レーダ中央の○: 母艦
    - レーダ中の□: 航空機（直線: 速度・移動方向）
    - 画面左上: 選択中の航空機に関する情報（ALT: 高度，SPEED: 速度）
    - 画面下側: ファンクションキーに対応（F6→[Track Manager]の起動）
      - どのような機能が利用できるかは状況によって異なる
        *キー押し後に新たな機能が表示
  - 操作系列
    - ステップ 1: 航空機の選択

- マウスで選択
  *基本的に自由だが，優先すべき航空機に関する教示を与える
  ＝母艦に近い・速度が速い・母艦に向かう航空機を優先
- ステップ 2: 航空機の同定
  - ステップ 2-1: 画面左上の情報を参照
    *速度と高度が一定以内→一般機
  - ステップ 2-2: [EWS キー] (F10) の利用
    *レーダープロファイルの起動
    *航空機の種類を自動的に同定（失敗する場合あり）
  - 航空機の同定に失敗した場合
    →他の航空機を選択
- ステップ 3: 航空機の入力
  - ファンクションキーの系列を利用した入力
    [Track-Manager]→[Update hooked track]→[Class/Amp]→[Primary ID]→[Friend]→[Air ID AMP]→[NonMilitary]→[Save Changes]

- 得点（画面中央左）
  - 正しい情報を正しく入力できた場合にポイントが加算
  - 「母艦に近い・速度が速い・母艦に向かう航空機」を素早く同定したら高得点

- 心理実験
  - Anderson, et. al., (2004) で実施した実験
    - 20 シナリオ（各 6 分）を 2 日にわたって実施
    - 結果 (Fig. 8): パフォーマンスと速度の向上

3.1. From a strong to a weak hierarchy（モデル構築の目的）

- Anderson, et al., (2004) によるモデル
  - モデルの構成
    - 宣言的知識に教示が格納
    - ゴール構造が明示され，教示検索の順序が完全に統制
      →完全なトップダウンモデル
    - 宣言的知識がルールによって検索・解釈・実行されることで行動
    - 宣言的知識の利用をコンパイルすることで学習が進行
  - 結果
    - 速度・眼球運動の再現に成功したものの，柔軟性と並列性は再現していない

- 本研究の検討項目: 以下を並列行動の生起から説明
  - 検討 1: 航空機の選択
    - 徐々に航空機の選択の質が向上
      *ノービスは航空機の種類に注意を払わず低得点
      *エキスパートは優先すべき航空機を選択し高得点
  - 検討 2: ファンクションキーの選択
    - 徐々にファンクションキーのラベルに注意を払わなくなる
      *見なくても何を押せばよいのかが分かるようになる
  - 検討 3: 手の動きの学習
    - 徐々に効率的な手の動きを学習
      *ノービス: キーボード操作とマウス操作のバッティング
      *エキスパート: 予測によって予め手を移動/左手と右手の使い分け

3.2. Minimizing control（モデル構築の指針: 最小統制の原則）

- 緩やかな課題構造の表現
  - 完全なトップダウン (Anderson, 2004) ではなく，完全なボトムアップ (2 章) でもない

- 教示の表現
  - 教示の貯蔵
    - ユニットタスク単位の貯蔵
      - 単一のタスク構造ではなく，ユニットタスク（教示セット）に分割
    - 教示セットに複数の連続した教示が貯蔵
      - セット内での教示の連鎖を連想強度によって表現（弱い連鎖）

      *教示を読むことで学習が成立したと仮定
  - 教示の検索
    - 活性度に基づく検索（緩やかな教示順序の統制）
      - 教示セット内の順序に即した検索（教示 1 の実行後，教示 2 の活性度が上昇）
      - 知覚刺激などによって，影響が加えられることがある
  - 教示の実行
    - 教示を検索後，行動の実行可能性（外界からの制約）をチェックするルールを用意

      *教示中には実行の際の制約は記載されない
    - 航空機の選択での例
      - 手がマウスに置かれていなければ，手をマウスに移動
      - 手がマウスに置かれ，マウスポインタが航空機の上にあれば，直ちにクリック

3.3. The CMU-ASP model and global results（モデルの実装と大雑把な結果）

- 教示セットの実装 (Fig. 10)
  - 教示セット全体での条件部 + 連想強度によって結合された各教示
  - 各時系列で単一のユニットのみがアクティブ（ゴールバッファに記載）
  - 一旦，教示セットがアクティブになったら，途中で他のルールセットにスイッチしない

    *ルールセット実行中のスイッチは原理的にはあり得るが, 今回のタスクではあり得ない

- 教示セットの内容（メインユニット (4) + 補足ユニット (1)）
  - *Find-and-click-plane*: ステップ 1 に対応
    - 条件部: 飛行機が選択されていない状態
    - *Find a plane*: 飛行機を見つける
    - *Click a plane*: 飛行機をクリック
  - *Check-commercial-profile*: ステップ 2-1 に対応
    - 条件部: 飛行機が選択されていて，かつ，クラスが同定されていない
    - *Look for alt*: 高度をチェック
    - *Look for speed*: スピードをチェック
    - *Check range*: 高度/スピードの範囲をチェック
    - *Decide commercial*: 一般機かどうかを判断
  - *Check-radar-profile*: ステップ 2-2 に対応
    - 条件部: 飛行機が選択されていて，かつ，クラスが同定されていない
    - *Select key*: "Select key"を呼び出す
    - *Read Air ID*: レーダープロファイルを読む
    - *Decide Class*: 航空機のクラスを判断
  - *Enter-Classification*: ステップ 3 に対応

  - 条件部: クラスが決定済み
  - *Select key*: *Select key* を呼び出す
- *Select key*:
  - 条件部: *Select key* が要求されている
  - *Find Label*: ラベルを見つける
  - *Determine F key*: ファンクションキーを決定する
  - *Press Key*: キー押しを実行する

- 教示セットの実行
  - 検索ルールによる各教示の検索 (Retrieval buffer への一時的な貯蔵)
  - 解釈・実行ルールによって外界の状況と教示の制約とのチェックがなされる
  - コンパイルによって教示の検索ルール，教示の解釈・実行ルールが統合
    →タスク固有のルールの構成

- Anderson, et. al のデータとモデルの対応 (Fig11)
  - Fig. 11a: パフォーマンスの向上（正確に同定された航空機の数）
  - Fig. 11b: 各ステップの所要時間の短縮
  - フィッティングはよいが，Anderson のモデルと比べて特によいというわけでもない

3.4. Finding and clicking a plane（検討 1）

- 教示セット *Find-and-click-plane* の構造
  - 第 1 段階: 教示 *Find a plane* の検索と実行
    - 1-1: オブジェクト位置の取得 *Find-location*
    - 1-2: オブジェクトへの眼球運動・エンコード *Find-object-attend*
    - 1-3: オブジェクトが選択対象（未同定）なら，オブジェクトを貯蔵 *Store object*
  - 第 2 段階: *Click a plane* の検索と実行
    - 2-1: マウス上に手がなければ，マウスに手を移動 *Hand-to-mo*use
    - 2-2: マウスをオブジェクトに移動 *Mouse-to-object*
    - 2-3: マウスのクリック

- 教示の検索・実行に関するルール *Find-object-attend* の検索と実行
  - *Retrieve-instruction*
    IF　　Goal buffer に"教示セット"が入っている
    THEN　"教示セット"から"教示"を Retrieval buffer へ格納
    *全ての教示へ適用可能な汎用的ルール．教示順序は連想によって表現
  - *Implement-attend-location*
    IF　　Goal buffer に"ルールセット"が入っている
    AND　Retrieval buffer に「"視覚刺激"に注意を向けよ」という教示が入っている
    AND　Visual buffer に"視覚刺激"が入っている
    THEN　眼球を"視覚刺激"へ向け，Visual buffer にエンコード

- ノービスモデルの動作 (Fig. 12)
  - 教示の検索・実行をサイクル
    - 一つの段階が完了するごとに教示の検索→同じ教示を何度も検索
  - 系列的な教示の検索・実行（太線のプロセスと細線のプロセスが重ならない）
    - 第 1 段階: *Find-a-plane* の検索→第 2 段階: *Click-a- plane* の検索
      *Perceptual と Manual が分離

- トップダウン的な行動系列
  - 1-1 から 2-3 までが段階的に実行
    *教示単位の統制状態と振る舞いは完全に同一

- 教示検索ルールと教示実行ルールのコンパイル
  - *Retrieve-instruction* + *Implement-attend-location*→
    Learned-attend-rule
    IF　Goal buffer に教示セット[*Find-and-click-plane*]が入っている
    AND　Visual buffer に[□]の位置が入っている
    THEN　眼球を[□]の位置に向け，Visual buffer にエンコード
  - *Retrieve-instruction* + *Hand-to-mouse*→
    Leaned-hand-to-mouse-rule
    IF　Goal buffer に教示セット[*Find-and-click-plane*]が入っている
    AND　Manual buffer の手がマウス上にない
    THEN　手をマウスに動かす

  *コンパイル後のルールは共通の条件部（*Find-and-click-plane*）
  *教示検索なし・ルール固有の制約に基づいた実行

- エキスパートモデルの動作 (Fig. 13)
  - 宣言的知識の検索がスキップ
    - 完全な時分割処理・待ち時間なし
    - 教示系列からの逸脱・ボトムアップな動作
  - ルール固有の制約に基づいた並列的な動作
    - *Find-and-click-plane* のセット直後に Visual と Manual が同時に開始
      *マウスを予めマウス上へ移動
    - Manual が手をマウスへ動かす間，Visual module が 2 つの飛行機を見つける
      *Manual のビジー状態を有効活用
      *得点の高い飛行機を選択
    - Manual がマウスを航空機へ動かす間，4 番目の飛行機を見つける
      *位置を修正した上で改めてマウスを動かす

- 航空機選択の質の向上 (Fig. 14)
  - モデル: 12 回の実行/被験者: 16 名
  - モデルと被験者はともにシナリオが進むにつれて質が向上

3.5. Learning the location of function keys（検討 2）

- 教示セット *Select key*
  - 教示 *Find label*
    1-1. スクリーン中のテキストラベルの認識・エンコード
    1-2. Visual buffer へのテキスト位置の格納
  - 教示 *Determine-F-key*
    2-1. テキストラベルのスクリーン位置とファンクションキーとの対応 (472, 715)→F6
  - 教示 *Press-key*
    3-1. 同定されたファンクションキーを押す

- 教示検索ルールと教示実行ルールのコンパイル
  - *Learned-Find-Track-Manager-rule*

IF "Goal buffer"に[Track Manager の選択]が入っている
THEN Visual buffer を[472, 715]に設定し，眼球を動かす

- *Learned-Determine-F6-rule*
  IF Goal buffer に *Select key* が入っている
  AND Visual buffer の注目位置が[472, *]に設定されている
  THEN キー押しを F6 に設定せよ (Manual buffer)
  *最終的には 2 つのルールのみでキー選択がなされる
  **Find label* の実行途中で *Press-key* の実行が可能

- スキル獲得へ向けて
  - コンパイル・ユーティリティの更新に時間がかかる

3.6. Learning to optimize hand movements（検討 3）

- 教示 *Press-key* における実行ルール
  - 左手でキー押し
  - 右手でキー押し

- キー押しの効率性
  - キーボード左側のファンクションキーは常に左手でキー押し
  - キーボード右側はマウス操作と競合する場合アリ

- Fig. 15: キー押しに右手を利用した頻度
  - 異なるキー押しパターンを学習した 3 回のモデル実行を比較
    - Left hand only: 左手を利用して右側キーを押すことを学習した実行
    - Both hands: 左手と右手の両方を利用してキー押しをした実行
    - Mixed: 右手がマウス上にあるときに限って左手でキー押しをした実行
  - 学習されたスタイルに多様性（どのスタイルが優れているということはない）
  - はじめはランダムだが，徐々に習熟したスタイルへの偏りが現れる

3.7. Conclusions

- 従来のルールベースモデルの仮定と本稿で提案したアプローチ
  - 仮定 1: 心理実験の教示を直接的にルールに翻訳
    →モデルを動かす中で徐々に教示をルールに翻訳
  - 仮定 2: 階層的に分割されたタスク構造の実装
    →最小統制の原則 (minimal control principle)に基づくモデリング

- 提案アプローチと構築されたモデル
  - Schumacher 課題
    - 教示からのルールの学習
    - 常に単一の統制状態
    →タスク間の完全な並列行動
  - CMU-ASP 課題
    - 教示セットからのルールの学習
    - 初期から後期にかけた統制状態の減少
    →ステップ間の並列行動

- 先行するモデルとの比較

- EPIC
  - タスク間の協調をスケジューラによって説明
    *ノービス: 系列的なタスクのスケジュールが強制的に割り振られる
    *エキスパート: 課題固有の並列的なスケジューラが発達
  - 並列行動は初期段階ではブロックされ，後期でブロックが緩和
  - EPIC ではブロックの緩和が何を要因として起きるのかに関する理論がない
- SOAR
  - 統制状態によってオペレータが適用（最小統制の原則に従わなない
  - 学習に従って統制状態が増加する

- インプリケーション
  - 熟達者のモデル
    - 熟達者の優れたパフォーマンスを形式的に表現することに焦点を当てた領域
    - 学習メカニズムを考慮する必要がある
    - 熟達者のモデルを作る上でも最小統制の原則に従うことは重要

- リミテーション
  - 教示からの学習のみ（事例や類推からの学習は捉えていない）
  - ルールセットへの恣意的な分割：モデル自身がどのようにルールセットの区分を学習できるのかを検討するべき
  - 教示の分割の仕方によって並列行動が出現するか否かが変わってしまう
  - 教示が絶対的（連想に基づかない）で，各ステップにおける制約が良く分からない状況で，モデルは並列行動を学習できない

  →貧弱な教示は統制状態を必要以上に増加

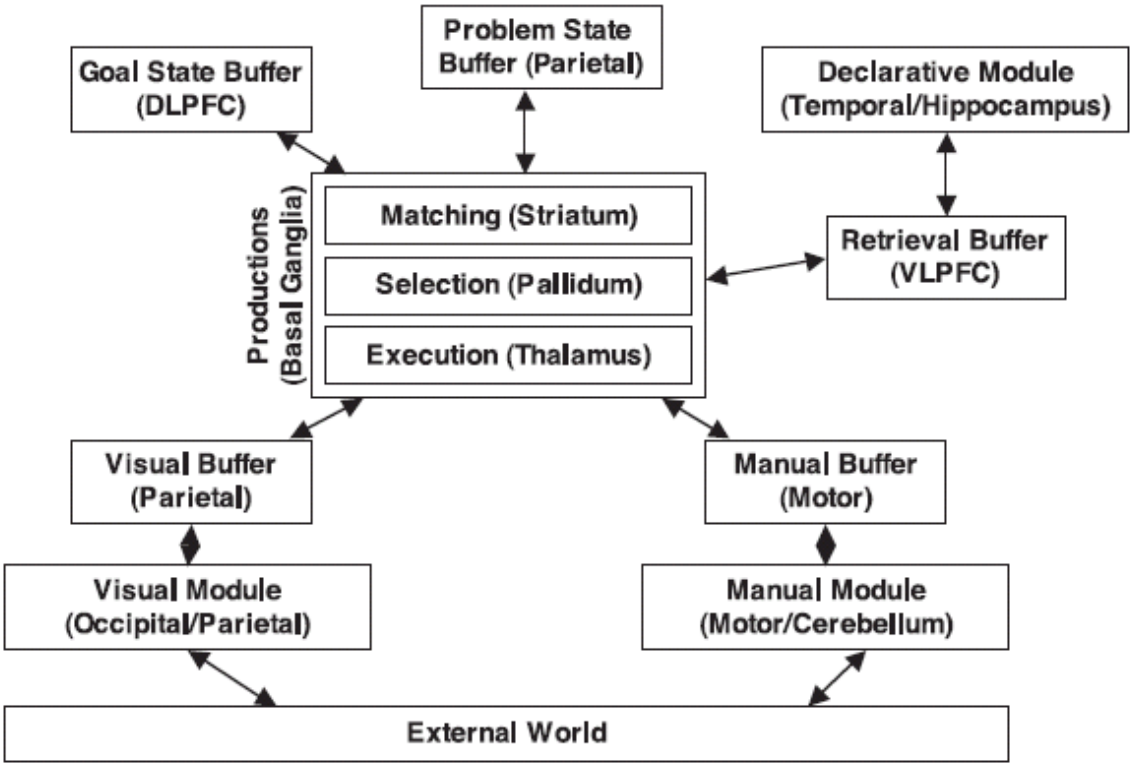


Fig. 1. Overview of the ACT–R architecture (adapted from Anderson et al., 2004).

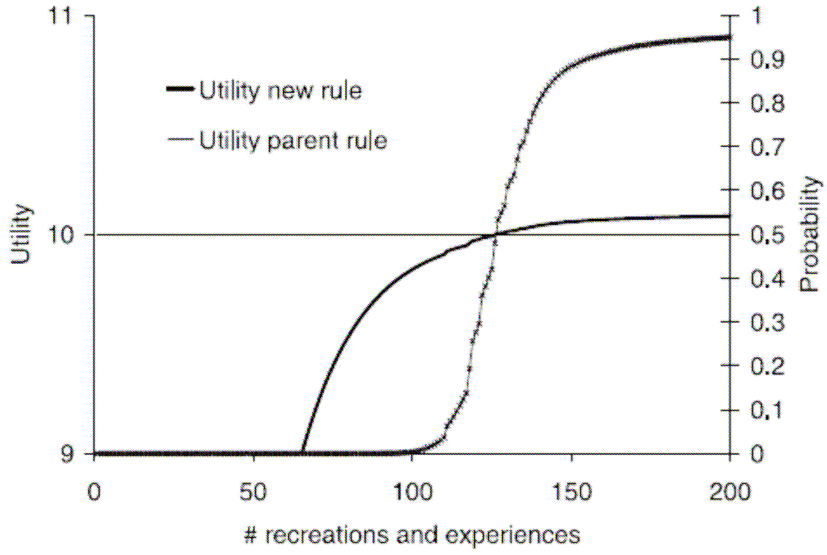
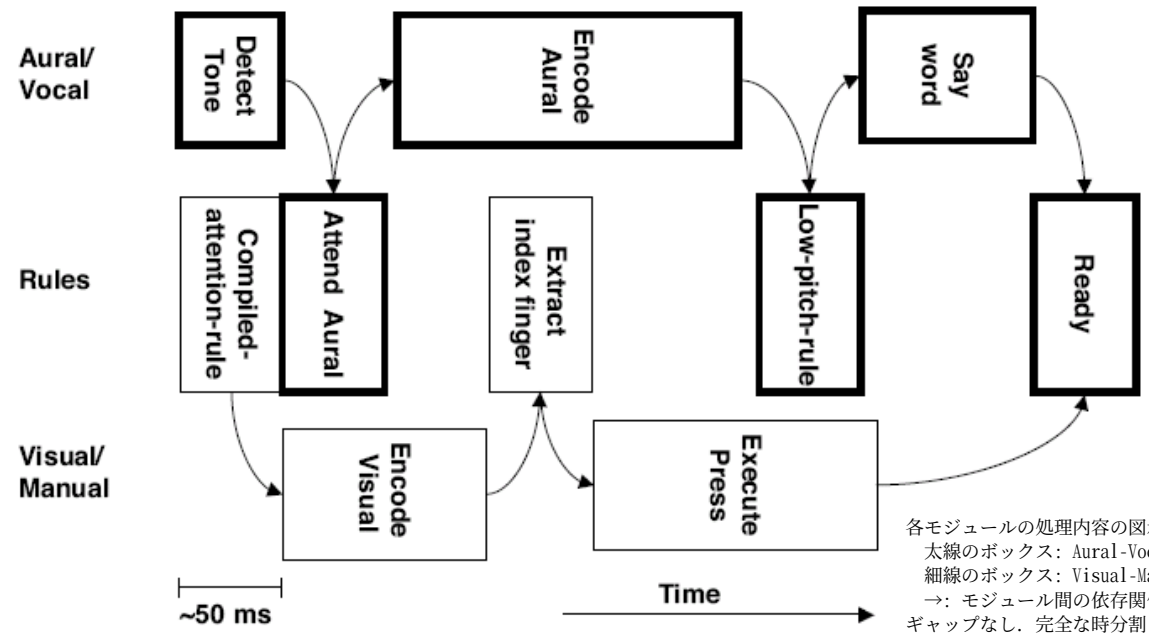


Fig. 4. Example of gradual introduction of a new production rule. The thin line represents the utility of the parent rule, and is fixed on 10. The new rule starts with a utility of −20 represented by the thick line (which is initially outside the graph's scale), but eventually approaches its true value of 10.1. The S-shaped curve represents the probability (with its axes on the right side of the diagram) that the new rule will win from the parent rule.



各モジュールの処理内容の図示
太線のボックス: Aural-Vocal task
細線のボックス: Visual-Manual task
→: モジュール間の依存関係
ギャップなし．完全な時分割

Fig. 2. Time diagram of expert-level parallel behavior in the Schumacher et al. (2001) Experiment 1 according to the ACT–R model. Arrows indicate dependencies between steps. Thin-bordered boxes represent steps in the visual–motor task and thick-bordered boxes in the aural–vocal task.

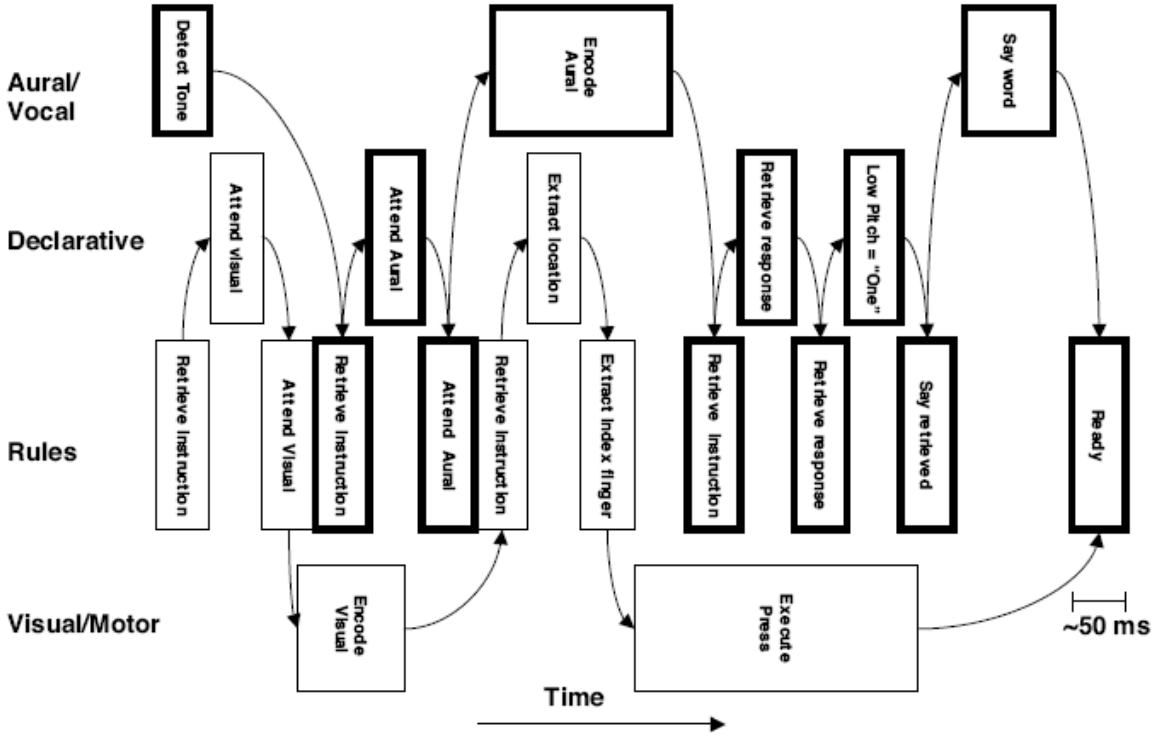


Fig. 3. Time diagram of a novice in the Schumacher et al. (2001) Experiment 1 according to the ACT–R model.

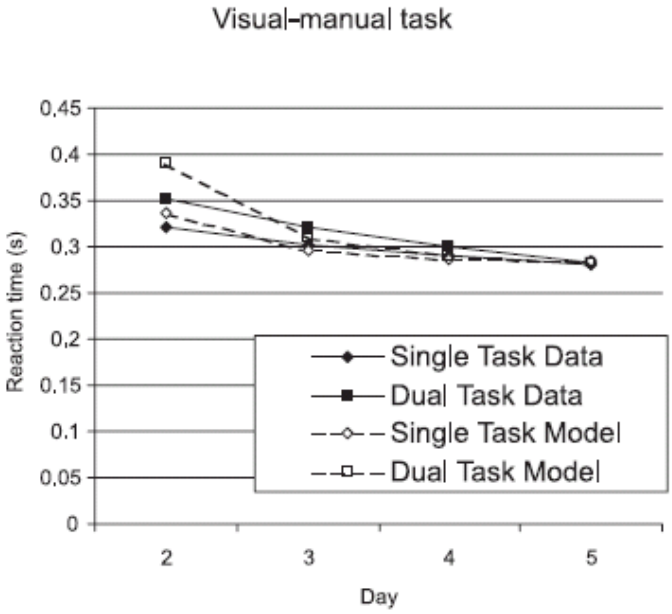


Fig. 5a. Results of Experiment 1 of Schumacher et al. (2001), model and data, visual-manual task. Single-task trials are averages of homogeneous and heterogeneous trials.

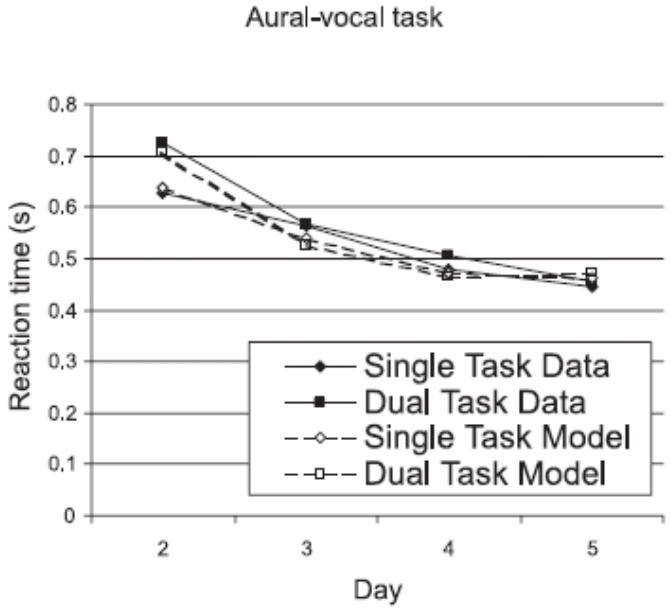


Fig. 5b. Results of Experiment 1 of Schumacher et al. (2001), model and data, aural-vocal task. Single-task trials are averages of homogeneous and heterogeneous trials.

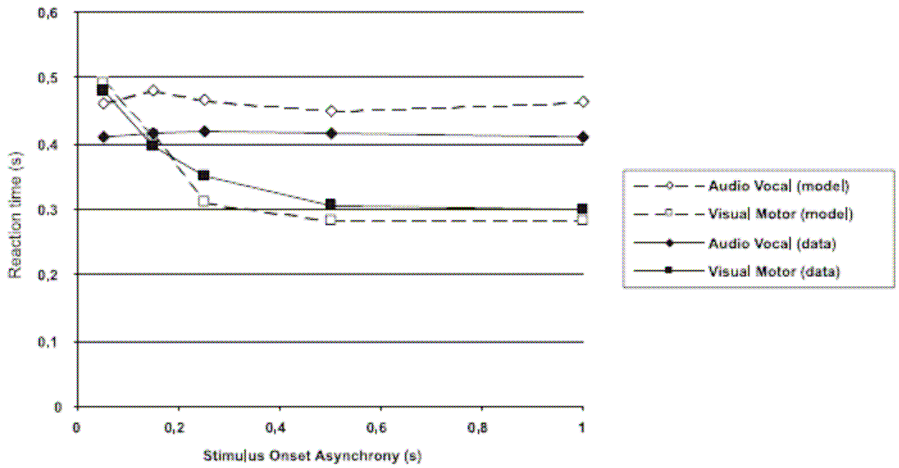


Fig. 6. Results from Experiment 2 of Schumacher et al. (2001), model and data.

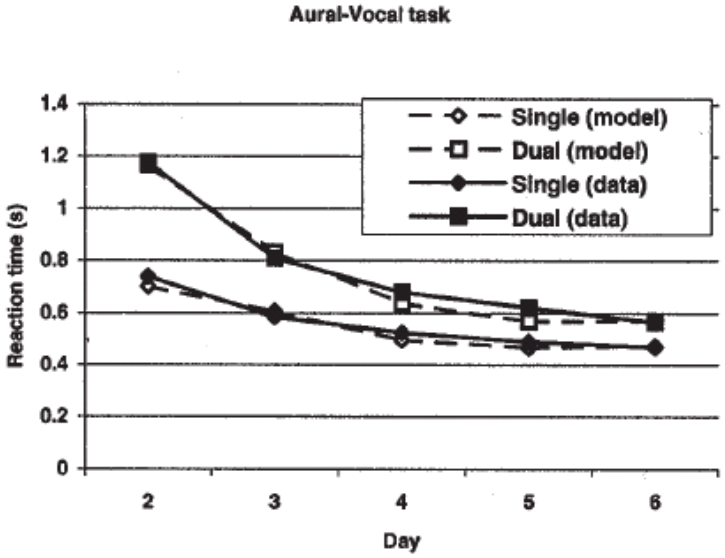


Fig. 7a. Results of Experiment 3, model and data, aural-vocal task. Single-task trials are averages of homogeneous and heterogeneous trials.

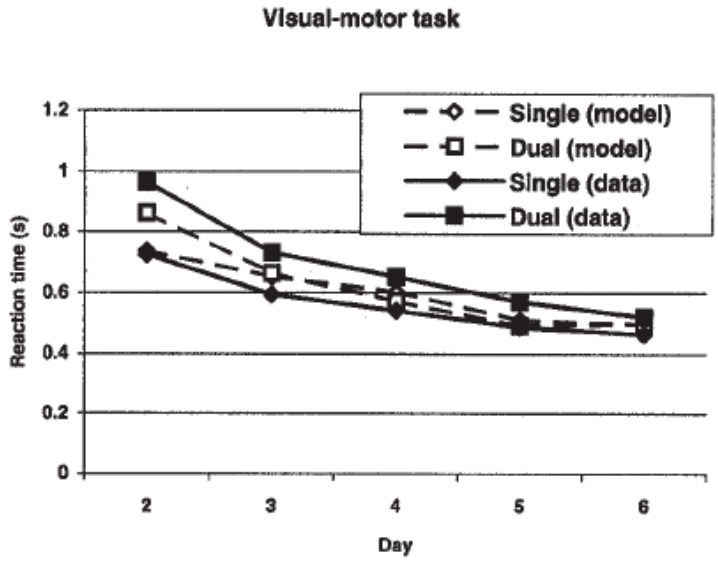


Fig. 7b. Results of Experiment 3, model and data, visual-motor task. Single-task trials are averages of homogeneous and heterogeneous trials.

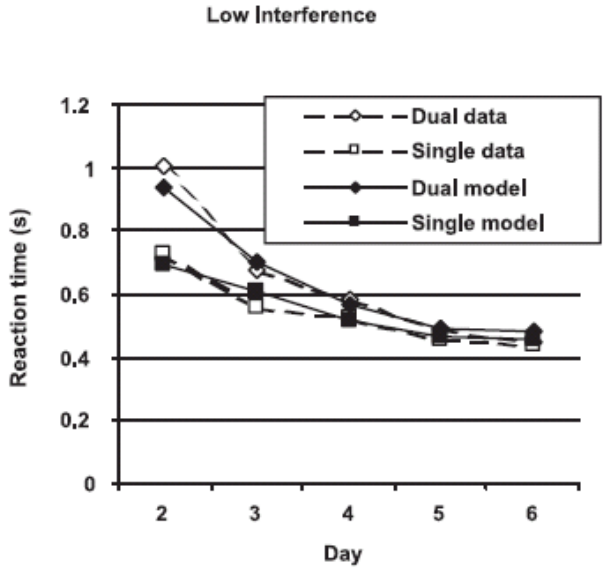


Fig. 8a. Data and model for 5 participants with no dual-task costs (low interference). Results are averaged over the two tasks and the two types of single-task trials.

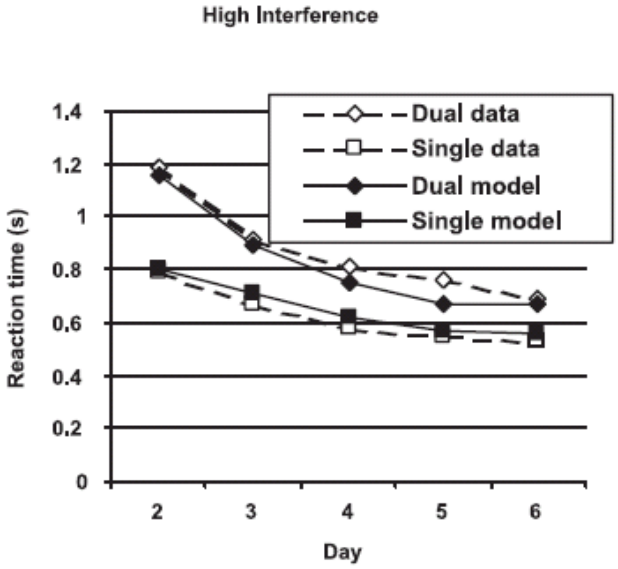


Fig. 8b. Data and model for 4 participants with high dual-task costs (high interference). Results are averaged over the two tasks and the two types of single-task trials.

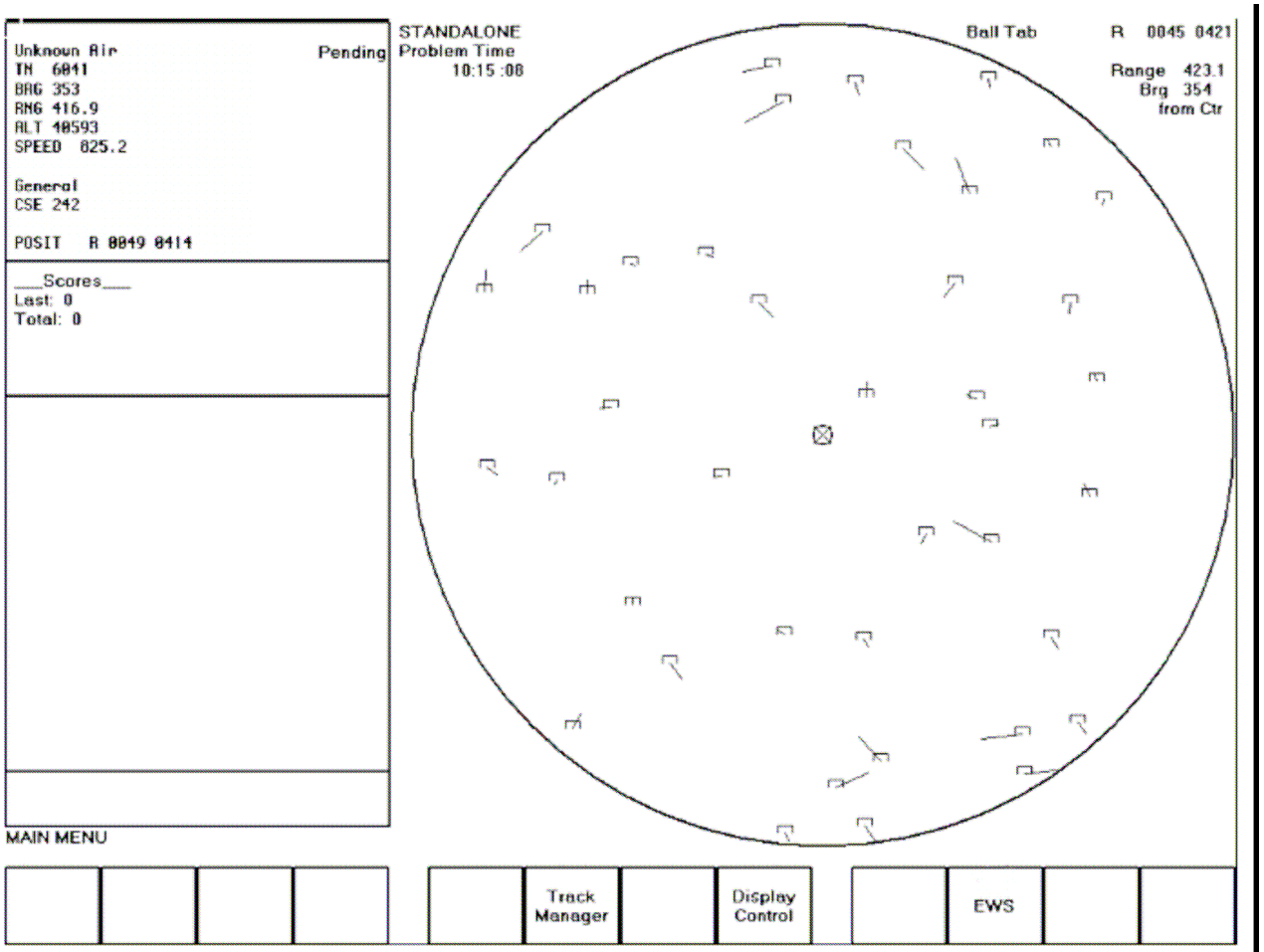


Fig. 9. The CMU-ASP task.

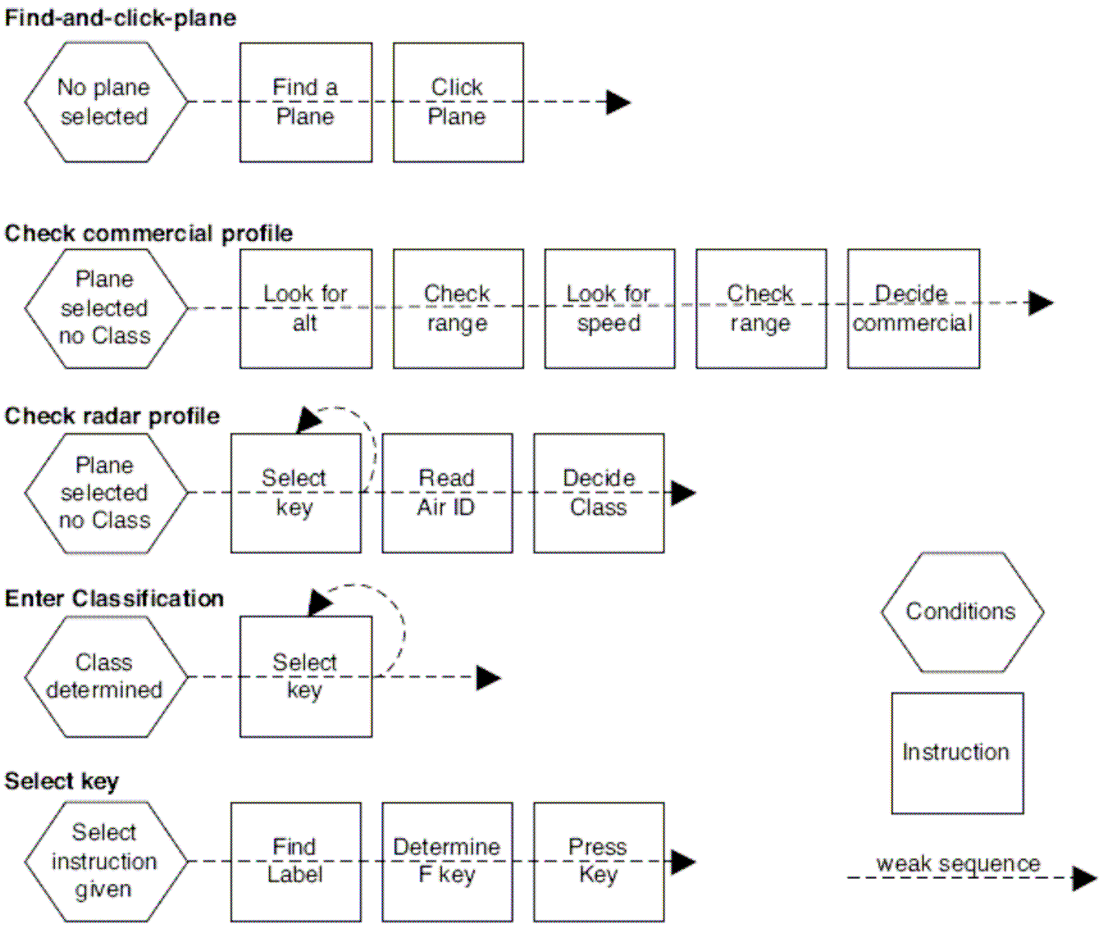


Fig. 10. Schematic representation of the instructions for CMU-ASP. Each row of boxes represents a rule set.

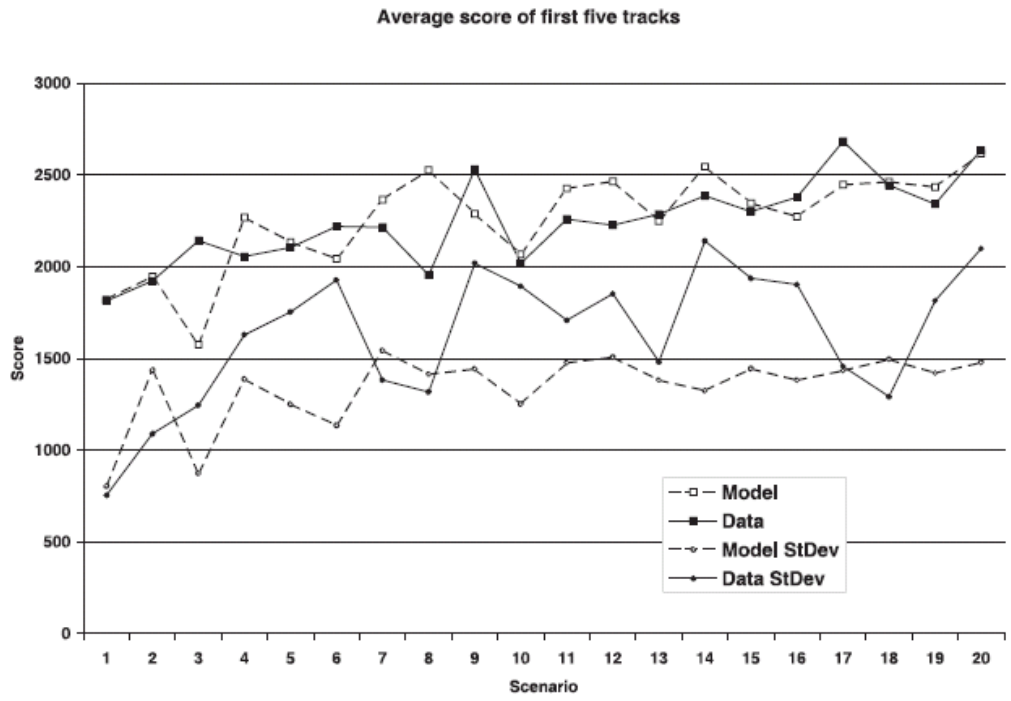


Fig. 14. Average en standard deviation of the point scores for the first five planes.

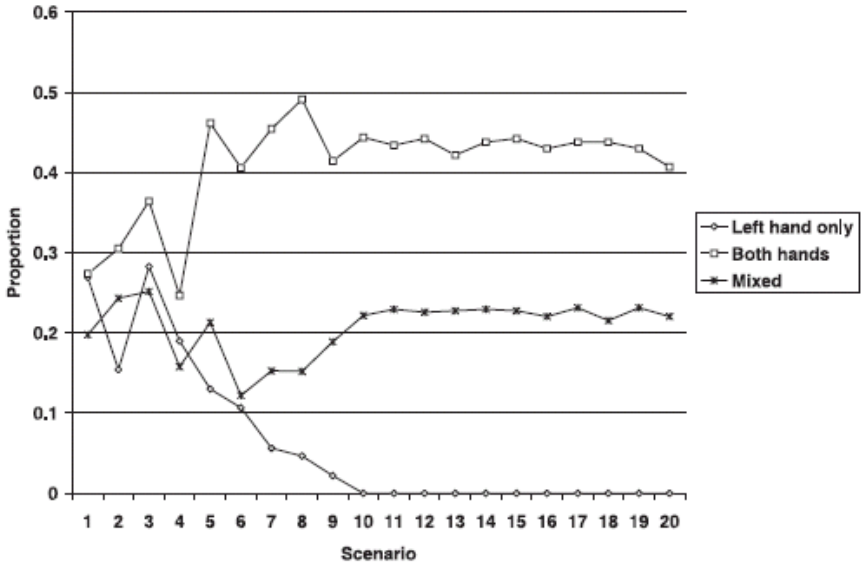


Fig. 15. Proportion of right-hand use for function keys. The function keys on the left side of the keyboard are also included, which are always pressed with the left hand.