

学位論文

インストラクショナルデザインの自動化を志向した
教材シェルの開発

市川 尚

熊本大学大学院
社会文化科学研究科
文化学専攻

博士（学術）

2009年3月

論文要旨

本研究では、インストラクショナルデザイン (ID) の自動化を志向した教材シェルを 2 種類開発した。インストラクショナルデザインの自動化 (Automating / Automated Instructional Design; AID) は、ID プロセスの一部または全部をコンピュータによって自動化することを指す。また、AID を志向した教材シェル (以降、AID 教材シェル) は、あらかじめ ID 理論に従った方略をアルゴリズムとして組み込んでおき、知識をデータとして登録することで、効果的な学習および効率的な開発を目指すシステムである。本研究では、特に内容の専門家 (Subject Matter Expert ; SME) でも利用可能な AID 教材シェルの開発を行った。

第 1 章では、近年の日本における e ラーニングや ID の現状について概観し、AID ツールの社会的必要性和本研究の目的を示した。インターネットの普及が進み、誰もが教育コンテンツの発信者となるような基盤が整備されつつある状況を想定すると、インストラクショナルデザイナーを養成するだけでなく、教材作成ツールや、e ラーニングプラットフォームに ID を組み込むことも考えられる。また、e ラーニングにおける教授法の質が低いという状況から、SME でも利用可能な ID の原理を内蔵したツールの必要性も指摘されている。このような研究は、欧米において、AID という名称で進められてきた。AID 研究の知見は、e ラーニングの質を高めるための手段として、今後ますます重要になってくると思われる。そこで本研究では、AID ツールに着目し、その研究のレビューと AID 教材シェルの開発を行うことにした。

第 2 章では、AID ツールの全体像と AID 教材シェルの位置づけを明らかにした。ID 理論 (モデル) の研究では、モデル自体の構築や改善と平行して、モデルの使い勝手に関する研究も行われており、AID 研究は特に後者に該当する。AID 研究は、1980 年代後半からはじまり、1990 年代を中心に最も研究され、現在も継続しているが、日本にはまったく紹介されていないのが現状である。本研究では、AID ツールのレビュー研究に着目し、どのような分類で AID ツールが整理されているかを示すことで、AID ツールの全体像を明らかにした。また、先行研究を参考に AID ツールを整理していくための、計 12 項目から成る独自の枠組みを作成した。特に AID ツールは、設計レベルの ID 理論に基づいて

いることが必須であるとの視点に立ち、枠組みには ID 理論という項目を設けた。また、著名な ID 理論に基づく異なる種類の AID ツールの 5 事例をこの枠組みにあてはめ、多様な AID ツールの特徴を明らかにし、(1)ID 理論の適用支援、(2) ID 理論の学習支援、(3) 標準化と再利用の 3 つの観点で考察を試みた。さらに、AID ツールへの批判や社会的背景から、SME でも質の高い教材を作成できるようなツールの必要性を示し、SME が利用可能な AID 教材シェルの開発を本論文の研究課題とした。AID 教材シェルは、言語情報（第 3 章）と知的技能（第 4・5 章）に対応した 2 種類を開発対象とし、AID 教材シェルの構造上の特徴から認知的方略についても検討を行った。

第 3 章では、e ラーニングにおけるドリル型教材の作成支援を目的として、統合型ドリルシェル「ドリル工房」の開発について述べた。これまでに ID の分野では複数のドリル制御構造が提案されてきたが、既存のツールにはドリル制御構造が組み込まれておらず、効果的な練習を支援しているとは言い難い状況であった。そこで、ドリル制御構造をシステム上に統合し、簡単に効果的なドリルを開発できるようにした。ドリル制御構造は複数提案されていたため、状況に応じて望ましいドリル制御構造を適用できるようにした。また、ドリル制御構造を作成者が学習できるようにし、可視化モードとしてドリル実行中にアイテムの状態遷移がわかるようにした。さらにドリル制御構造を制御要素の組み合わせとして整理することで、各ドリル制御構造の違いを明確にしてわかりやすくするだけでなく、制御構造自体のカスタマイズも可能にした。評価の結果、ドリルは簡単に作成できたこと、ドリル制御構造を学習できたことは確認された。また、他の学習課題への適用の可能性を検討するために、知的技能への対応も検討し、下位ドリル群のプロトタイプ開発を試みて、動作することを確認した。この開発を通して、AID 教材シェルが、SME でも効果的な教材を生成できること、逆に ID 理論を知りたい人には効果的な学習環境となる（学習者自身が開発する場合は認知的方略の学習を促進することにもつながる）ことを示すことができた。

第 4 章と第 5 章では、Merrill の教授トランザクション理論 (Instructional Transaction Theory ; ITT) に基づいた教材シェルの開発について述べた。第 4 章では、そのサブシステムの 1 つである学習環境について取り上げた。ITT は AID 教材シェル構築のための理論であり、AID 教材シェルを検討する上では欠かすことのできない理論である。あらかじめ教授方略をアルゴリズムとして内蔵しておき、知識をデータとして処理することで、効果的な学習環境が提供されることを目指している。アルゴリズム（方略）が教授トランザクション、データ（知識）がナレッジオブジェクトに相当する。教授トランザクションは、構成要素、抽象、連合という 3 つの大分類のなかに計 13 種類が特定されているが、その実装は、構成要素のトランザクションを中心に、一部しか行われてこなかった。本研究では、構成要素のトランザクションのである IDENTIFY（部品の名前や位置を同定できる）、EXECUTE（手続きを実行できる）、INTERPRET（なぜそうなるのか

を説明できる)のすべてを実装することにした。特にこれまで検討段階に留まっていた INTERPRET の練習については、Merrill の提案を踏まえて、予測とトラブルシュートの設計と実装を行った。また、提案されているすべての方略を実装することは、メニュー項目が増えて学習者が混乱するとの判断から、トランザクションの種類によって提供する方略を厳選するように工夫した。AID 教材シェルは複数の教材を同じ枠組み(方略)で提供することから、方略を整理して明示的に学習者に示すことが認知的方略の促進にもつながると考えられる。形成的評価の結果、学習環境を利用して学習効果があることは示唆された。また、さまざまなテーマに適用できることも確認した。さらに抽象のトランザクションの JUDGE (順位付けができる)と CLASSIFY (分類できる)について、設計と簡単なプロトタイプの開発を行い、動作することを確認した。

第 5 章では、ITT 教材シェルのオーサリング環境のプロトタイピングについて述べた。ITT 関連のオーサリングシステムはこれまでに数種類が開発されてきたが、現在利用可能なものは無い。従来のオーサリングシステムは、リソースを画面上に配置し、リソースに関連づけながら、ナレッジオブジェクトの情報を 1 つずつウィンドウ上で登録していく、上級者用(ソースレベルの内容を理解している人のため)のインタフェースであった。本研究では、筆者らの学習環境のコンテンツ開発を簡便化するために、まず先行研究とほぼ同様のシステムとなる上級者用インタフェースの開発を行った。また、それを SME でも利用できるようにするためのプロトタイプと位置づけて評価を行い、システムの要件を検討した。その結果として、入力の手間を少なくするために、特に入力の行き来が多いエンティティとプロパティ、アクティビティとプロセスの各ナレッジオブジェクトのペアについて、同時に入力できるようにした。また、学習環境ではトランザクションごとにメニューを厳選したが、オーサリング環境においては、トランザクションに応じて、必要な入力項目をシステム側で判断し、入力欄を厳選する機能を設けた。さらに、本研究で実装した学習環境はシミュレーションの構築が中心であるため、PEAnet 構造と呼ばれるナレッジオブジェクト間の関係性に着目したモデル化が必要となるが、リソースを配置してナレッジオブジェクトを登録している際に、ナレッジオブジェクト間のつながりをリンク線で明示する機能を設けた。これにより、SME が利用できるオーサリングツールにしていくための基盤となる環境を整えることができた。

第 6 章では、AID ツール研究のレビューや AID 教材シェルの開発を通して、研究課題であった SME の利用と認知的方略の対応について考察し、今後の展開を整理した。SME の利用については、ドリルシェルは SME でも利用可能な状態となったこと、ITT 教材シェルについてはそのための基盤となるシステムを開発できたことを示した。また、AID 教材シェルの開発の困難さについても議論した。認知的方略については、ドリルシェルは学習者が自らの学習のために教材を作成する場合を特に想定し、オーサリング環境において ID 理論が学習できるようにし、効果が期待されることを示した。一方で ITT 教材シェ

ルは、AID 教材シェルの特性上、同じ学習環境の枠組みで様々な教材が提供できることに着目し、方略を学習者にわかる形で提供することによって認知的方略の学習を促進する可能性を示した。さらに AID 教材シェルの今後の展開として、ITT の批判的検討、学習理論（パラダイム）からのアプローチ、AID 教材シェルの構成要素間の共有可能性や、コース管理等のマクロ的な AID ツールとの連携の可能性、シェル上でのテンプレート提供について述べた。

以上から、次の 3 点について特に AID 研究に貢献できたと考えている。また、これらの貢献を通して、AID 教材シェルの構造が、認知的方略の学習に効果的であることも示唆した。

- 日本に紹介されてこなかった AID 研究のレビューを行い、AID 研究の動向を明らかにした。
- 既存のドリル制御構造（ID 理論）を制御要素の組み合わせで表現することによりシステム上に統合し、簡単に効果的なドリルを作成できるだけでなく、ドリル制御構造自体の学習やドリル制御構造のカスタマイズを可能にした。
- ITT の先行研究で提案に留まっていた部分を含めて、構成要素のトランザクションにおけるすべての方略を実装することで、理論で提案されていた内容を具体化した。さらに実装を通して、学習環境では、方略の多さによる混乱の回避や方略をわかりやすく提示するために、トランザクションごとに方略を整理して提供するインタフェースを提案し、オーサリング環境においては、SME でも利用可能なインタフェースとして入力の簡便化を行う方法を提案した。

実際に開発した AID 教材シェル（ドリルシェル）は Web 上で公開を開始している。
<http://ichi.et.soft.iwate-pu.ac.jp/df/>

目次

第1章	序論	1
1.1	研究の背景	1
1.2	研究目的	3
1.3	本論文の構成	3
第2章	インストラクショナルデザイン自動化ツール	5
2.1	はじめに	5
2.2	AID 研究	5
2.3	本研究における定義と事例整理の枠組み	7
2.4	AID ツールの事例	10
2.5	事例整理を通しての考察	12
2.6	SME 用の AID ツール	17
2.7	AID 教材シェル	20
2.8	研究課題	22
2.9	まとめ	24
第3章	統合型ドリルシェル「ドリル工房」の開発	25
3.1	はじめに	25
3.2	先行研究	26
3.3	ドリル制御構造の整理	29
3.4	システム概要	34
3.5	学習環境	34
3.6	オーサリング環境	36
3.7	評価	42
3.8	知的技能への対応	46
3.9	まとめ	50

第 4 章	教授トランザクション理論に基づく教材シェルの学習環境の開発	53
4.1	はじめに	53
4.2	教授トランザクション理論	54
4.3	先行研究	59
4.4	構成要素のトランザクションの設計	62
4.5	学習目標 (トランザクション) の選択による制御	64
4.6	IDENTIFY と EXECUTE の実装	67
4.7	INTERPRET の実装	71
4.8	テストモード	78
4.9	評価	78
4.10	抽象のトランザクションの検討	85
4.11	まとめ	88
第 5 章	教授トランザクション理論に基づく教材シェルのオーサリング環境の開発	89
5.1	はじめに	89
5.2	先行研究	90
5.3	上級者用インタフェースの開発	92
5.4	上級者用インタフェースの評価	100
5.5	SME 用インタフェースの開発	104
5.6	まとめ	111
第 6 章	考察	113
6.1	はじめに	113
6.2	SME の利用について	113
6.3	認知的方略について	114
6.4	ID 理論の実装について	115
6.5	AID 教材シェルの展開	116
6.6	まとめ	119
第 7 章	結論	121
7.1	本研究の成果	121
7.2	今後の課題	123

付録 A	ドリル制御構造	125
付録 B	教授トランザクション理論の展開	131
付録 C	ナレッジオブジェクトの XML 仕様	134
謝辞		137
発表論文		139
参考文献		141

第 1 章

序論

1.1 研究の背景

インストラクショナルデザイン (ID) とは、教育活動の効果・効率・魅力を高めるための手法を集大成したモデルや研究分野、またはそれらを応用して学習支援環境を実現するプロセスを指す (鈴木, 2005a)。ID は米国では 40 年以上も前から教育工学の中核領域として研究されてきた。教育工学とは、「人間の学習のあらゆる面に含まれる諸問題を分析し、これらの問題の解決法を考案し、実行し、評価し、運営するための、人、手だて、考え、道具、組織を含む複雑な統合過程である」と定義される学際的な研究分野である (日本教育工学会, 2000)。

一方で、欧米に対して日本では ID が浸透しておらず、2000 年頃からの e ラーニング普及に伴い、e ラーニングの実効性を高めるための手法として注目が集まり出した (鈴木, 2006)。e ラーニングの設計原理には、ID モデルや理論として数多く提唱されている知見を応用することができると言われている (鈴木, 2005a)。

ID は日本ではまだ普及し始めた段階であり、教育の質を左右するインストラクショナルデザイナー (IDer) の養成も始まったばかりである。IDer とは、ID を用いて、ニーズ調査などの分析を行い、その結果を使って、教育プログラムを設計、評価する専門家のことである (玉木ほか, 2006)。近年においては、2006 年度から熊本大学大学院社会文化科学研究科教授システム学専攻 (大森, 2008) や青山学院大学総合研究所 e ラーニング人材育成研究センター (2008) が IDer の養成を行っている。日本 e ラーニングコンソーシアム (2008) は eLP (e ラーニングプロフェッショナル) 資格制度を開始しており、e ラーニングの文脈の中で ID を身につける環境が整いつつある。しかしながら、現状において IDer の絶対数は少なく (鈴木, 2004)、専門家養成の成果が出てくるにはまだかなりの期間がかかると思われる。また、e ラーニングの設計は、IDer が教授内容の専門家である SME (Subject Matter Expert) と協力しながら行うことが望ましいとされるが (玉木ほ

か, 2006), IDer が少ない現状においては, 実現が難しい. よって, e ラーニングコンテンツ開発は, 開発担当者や SME が経験則に基づいて行うことになる. 大学教員が e ラーニングコンテンツを開発する事例もあるが (不破ほか, 2003), 大学教員はほとんどが SME であり, IDer ではない.

Clark and Mayer (2007) は, 今日 e ラーニングと呼ばれるものの多くは 30 年前からある CBT (Computer-Based Training) と同じであり, 多くの e ラーニング教材は画面上に本を再現したものに過ぎないと酷評しており (鈴木, 2005b), 既存の e ラーニングコンテンツの質は, 必ずしも高いとは言えない. e ラーニング白書 (経済産業省商務情報政策局情報処理振興課, 2007) では, e ラーニングにおける今後のビジネス展開において重要と思われる事項の調査について, コンテンツの質の向上が最も高い結果となっている.

また, インターネットの普及が進み, 誰もが情報の発信者となるような基盤が整備されつつある. 近年では Web2.0 という概念が登場し, 常磐 (2007) は Web2.0 の e ラーニングに関わる要素を, アプリケーションの利用 (BLOG など), システム連携への利用 (RSS など), 考え方の応用 (集合知の利用など) に整理している. 仲林 (2007) も, 教育の観点から Web2.0 について特に注目することの 1 つに集合知を挙げ, 学習者の多様なニーズに対応した双方向コミュニケーションの場において, 自らも知識や知恵を提供して集合知の形成に貢献しながら学習を継続するという方向性があるとしている. これは, ネット上で相互に作用しながら協調的に学習を進めていくというだけでなく, 誰もが教材提供者となる可能性があるということも示している. 例えば, クイズ (検定) などは, 実際に Web 上で皆が持ち寄って賑わっている事例もある (廣済堂, 2008). 教材の共有や再利用を行うための仕組みとして, 検索を効果的にするためにメタデータを付加するような学習オブジェクト (学習者が学習する際の対象のこと) に関する研究も進められている (清水, 2007).

IDer が少ないだけでなく, 誰もが発信者となるという状況を想定すると, IDer を養成するだけでなく, 教材作成ツールや, LMS (Learning Management System) や CMS (Course Management System) と呼ばれる e ラーニングプラットフォーム (梶田ほか, 2007; 田中ほか, 2005) に ID を組み込むことも考えられる. Merrill は e ラーニングにおける教授法の質が低いという指摘から, インストラクショナルデザイン (ID) の原理を内蔵したツールの必要性を述べている (鈴木, 2006). ID をツール化することで, IDer が支援できない状況においても, ある程度品質の高い教育を提供できるようになる. こういった研究は, 欧米において 1990 年代を中心に ID の自動化 (AID; Automating / Automated Instructional Design) という名称で進められてきた (Spector and Ohrazda, 2004). AID ツールは, ID プロセスの一部または全体を支援するコンピュータ上のツール (ソフトウェア) のことである. AID 研究の知見は日本ではほとんど紹介されてこなかったが, e ラーニングの質を高めるための手段として, 今後重要になってくると思われる.

また, これまでの AID ツールは IDer の作業を効率化するために提供されている場合も

多いが、上記で議論した日本の現状やインターネットの普及を考慮すると、ID の素養のない SME でも利用可能な AID ツールの必要性が増していると考えられる。さらに、ID が普及していない日本の現状において、ツールに ID 理論を埋め込むことによって ID に触れる機会を作ることによって、学習者は自らの学習方法を拡張したり、開発者は ID の素養を身につけていくことも、少なからず必要であろう。

1.2 研究目的

本研究の目的は、SME でも利用可能な AID ツールを開発することにある。それには AID ツールの中でも、あらかじめ ID 理論に基づく教授方略を実装しておき、知識（データ）を登録するだけで、教材が生成されるような教材シェルの形式が有効であると考えられる。教授方略があらかじめ実装されているため、ID を知らない SME であっても、一定品質の教材作成が期待できる。データを入れ替えることで複数の教材を作成可能であり、プログラミングの知識も必要としない。また、教材シェルを通して学習者が認知的方略として学習方法を学ぶ側面についても期待できる。本研究の目的は、以下の 3 点に集約される。

第 1 に、AID ツールを開発する前段階として、これまでに行われてきた AID ツール研究を概観するとともに、AID ツールの中でも SME にとって教材シェルの形式が有効であることを明らかにする。また、教材シェルは認知的方略の学習にも有効であることを示す。AID ツールは日本ではほとんど紹介されてこなかったため、AID ツールのレビューは日本の教育工学研究に寄与できると考えられる。

第 2 に、ドリルにおいて複数提案されている ID 理論を統合し、SME が簡単により良いドリルを構築できるみならず、ID 理論自体の学習を行うことができるドリル型の教材シェル（ドリルシェル）を開発する。統合においては、提案されているドリルの制御に関する ID 理論を理解しやすい形に構造化することで、ID 理論自体の学習や制御のカスタマイズにも対応する。

第 3 に、AID を志向した ID 理論である教授トランザクション理論（Instructional Transaction Theory ; ITT）に基づいた教材シェルを開発する。ITT は、理論としては提案されているが、システムの実現は途上段階にあるため、これまでに実装されていない部分を含めて、SME が利用可能な教材シェルを開発する。

1.3 本論文の構成

本論文は、全 7 章から構成される。

第 1 章（序論）では、本研究の社会的背景と研究目的を述べ、e ラーニングの普及と

ID の必要性，AID ツール（教材シェル）の必要性を示した．

第2章では，AID ツール研究の動向を把握するために，AID ツールの先行研究を整理・分析し，AID を志向した教材シェル（AID 教材シェル）を定義づけ，その特徴を明らかにする．また，本研究の課題として，SME が利用できるツールの開発と，認知的方略への対応を示す．

第3章は，これまでに複数提案されているドリル制御構造（ID 理論）をシステム上に統合し，理論自体の学習も可能とする，統合型ドリルシェル「ドリル工房」の開発と評価について述べる．

第4章は，ITT に基づく教材シェルの学習環境について，構成要素のトランザクションの実装と，トランザクション（学習目標）ごとに方略を厳選して学習者に提示する工夫を中心に述べる．

第5章は，ITT に基づく教材シェルのオーサリング環境について，上級者用インタフェースと SME 用のインタフェースの開発を試みた結果について述べる．

第6章は，2種類の教材シェルの開発を踏まえながら，研究課題である SME の利用や認知的方略への考察，および AID 教材シェルの展望について述べる．

最後に，第7章で，本研究で得られた成果と今後の課題について述べる．

第 2 章

インストラクショナルデザイン自動化ツール

2.1 はじめに

インストラクショナルデザイン (ID) 活動を支援するツールの開発は、これまでに ID 自動化 (Automating / Automated Instructional Design ; AID) の研究として行われてきたが、日本ではほとんど紹介されてこなかった領域である。海外では、欧米を中心として、1980 年代後半から研究が進められている。近年の e ラーニングのコンテンツ開発において、汎用的な商用ツールから独自開発の専用ツールまで様々に利用されているが、コンテンツの品質向上のためには AID ツールの視点が必要と考えられる。

本章の前半では、AID ツールに関する研究の動向を概観した。また、先行研究を参考に AID ツールの分類枠を独自に作成し、ID 理論に基づく AID ツールの 5 事例をあてはめて整理した。また、AID ツールの事例に対する考察や AID ツールに対する批判の検討を通して、SME が利用できる AID ツールに着目し、本研究において 2 種類の AID 教材シェルを開発することにした。本章の後半では、AID 教材シェルについての AID ツール上の位置づけと、2 種類の AID 教材シェルを選択した理由 (研究課題) について述べた。

2.2 AID 研究

2.2.1 AID の定義

AID は、1990 年代を中心に使用されてきた用語であるが、明確な定義づけは行われておらず、ID ツールもしくは ID ソフトウェアと呼ぶ場合も多い。

Kasowitz (1998) は、AID ツールが教育の開発に携わる人々を支援し、ID の物理的な作業を軽減するものとしている。また、AID ツールの強みは、効果的な教授開発のプロセスを通して、初心者や ID 以外の専門家をガイドできることにあり、ID 専門家が不足している場合や、SME (Subject Matter Expert) などが教授開発に責任を持つような状況で特に役に立つとしている。

Merrienboer and Martens (2002) は、コンピュータベースの ID ツール (AID と同義) を、デザイナーや教師が、ADDIE モデル (教授システムの分析、設計、開発、実施、評価のプロセス) によって構造化される 1 つ以上の様々な活動を助けるツールとしている。また、CBI (Computer Based Instruction) の歴史とも密接に関連していると述べている。一方で、コンピュータベースという言葉が付記していることから、この言葉を付けない場合は、コンピュータを用いない ID ツールも含むことになる。例えば紙面上のチェックリスト (根本・鈴木, 2005) 等が ID ツールとして提供されている。

2.2.2 AID ツールを整理している研究

AID ツールの開発研究は数多く存在するが、それらをまとめた研究のレビューもいくつか存在する。ここでは AID ツール研究を整理している論文に焦点をあてる。

Kasowitz (1998) は、エキスパートシステム、アドバイザリーシステム、情報管理システム、EPSS (Electronic Performance Support System)、オーサリングツールの 5 種類で AID ツール研究を整理している。例えば、アドバイスを提供するシステムは、ID の素養が無いユーザの開発作業を自動化するというより、インストラクショナルデザイナー (IDer) がアドバイスをするという ID 活動を自動化しているとみなす。

Spector and Ohrazda (2004) は、Kasowitz (1998) の分類を踏襲し、オーサリングツールを除く 4 種類について、AID ツールを 1 事例ずつ紹介している。また、AID ツールの歴史的展開、評価方法やオブジェクト指向のアプローチに関して考察を行っている。

Merrill (1997) は、ID ツール (この場合の ID は、Instructional Development として使用され、開発部分が中心である) を 5 つのレベルで整理している。適切な教授方略を組み込んだ学習志向のシステムが必要であるとの視点に立ち、既存の開発ツールを、プレゼンテーションツールなどの情報コンテナ (レベル 1)、教授方略の実装されていないオーサリングシステム (レベル 2)、テンプレートなどで教授のインタラクションを拡張したオーサリングシステム (レベル 3)、学習志向の ID ツール (レベル 4)、学習者に適応した学習志向の ID ツール (レベル 5) の 5 レベルに分類している。特に最初の 3 レベルを構造志向のシステムとして、教授方略の組み込まれた学習志向のシステムと意図的に区別している。

Murray (1999) は、25 件の ITS (Intelligent Tutoring System) のオーサリングツールについて、7つの分類に整理している。その分類は、カリキュラムの系列と計画、教授方略、シミュレーションや訓練、エキスパートシステム、複数の知識タイプ、特別な目的のシステム、知的な適応型ハイパーメディアである。さらに、各分類について、強み・制限・変更可能な部分という観点からまとめている。また、オーサリングツールの構成や機能、知識の獲得方法、設計上のトレードオフ、利用状況など、複数の視点から各ツールを詳細に分析している。

Tennyson and Spector (1995) は、ID プロセスに着目し、設計・開発・実施という3つの段階に分けて、1990年から1995年にかけてのAID研究を概観している。

Piskurich (2006) はIDの効率化(ショートカット)として、ラピッドプロトタイプングといったプロセス面だけでなく、ソフトウェアの利用を挙げ、分析ソフトやテスト開発ソフトを少数ではあるが紹介している。

Nieveen and Gustafson (1999) は、比較を目的としてツール自体の特徴を詳細に記述する枠組みを作成し、10事例に当てはめて分析を行っている。枠組みのカテゴリは、出力の種類、目的と証拠、開発プロセスと理論、タスク支援、対象となるユーザグループとし、さらに下位カテゴリや、その中の内容まで規定している。例えば、目的と証拠のカテゴリであれば、ツールの目的という下位カテゴリの中に、知識やスキルの転移・パフォーマンスの改善・構造的な学習の3項目を規定し、そこから対象のツールに該当する項目を選択するようにしている。

Merrienboer and Martens (2002) は、コンピュータベースのIDツールについて、Nieveen and Gustafson (1999) らの比較の枠組みを踏襲し(下位カテゴリは利用していない)、開発プロセスと理論、意図される出力、目的と証拠、対象ユーザ、タスク支援の5つの観点に分けて整理している。また、IDツールが開発プロセスに傾倒しているのに対して、今後は分析や設計プロセスおよび協働作業が重要になってくること、複雑な課題を扱う機会が増えてくると指摘し、開発に関連する以外の4事例のIDツールについて、比較の枠組みに基づいて整理している。

2.3 本研究における定義と事例整理の枠組み

本研究においてAIDという用語は、コンピュータの利用を明確にするという位置づけで用い、コンピュータベースのIDツールと同義と捉える。一方で、IDをより限定的に用いることとし、設計レベルのID理論が含まれているツールのみを対象とした。

先行研究にならい、本研究においても事例の整理を試みる。様々な分類・整理の枠組みが提案されてきたが、Merrienboer and Martens (2002) は、その他の分類枠の要素がある程度包括していると考えられる。Nieveen and Gustafson (1999) の枠組みを踏襲し

ており，Tennyson and Spector (1995) のプロセスや Merrill (1997) の教授方略の要素は「開発プロセスと理論」に，Murray (1999) や Kasowitz (1998) および Spector and Ohrazda (2004) の分類も「タスク支援」にある程度含まれるとみなすことができる．一方で，Murray (1999) は ITS に，Merrill (1997) は開発に特化しているため，ID ツールを分類する枠組みとして，そのまま活用するには検討の余地が残る．ただし，Murray (1999) や Kasowitz (1998) や Spector and Ohrazda (2004) のように，ツールの共通性によって分類（集約）を行い，分類ごとに特徴を検討することも，今後必要になってくると考えられる．

以上から，本研究における AID ツールを整理する枠組みは，Merrienboer and Martens (2002) を基にして，その他の先行研究も踏まえながら，次のようにした．

名称 システムのタイトルである．

文献 関連論文への参照である．

ID プロセス プロセスのどの段階を支援しているのかを提示することで，支援の適用範囲を明確にする．ツールは，複数のプロセスにまたがる場合が多い．ここでは，Nieveen and Gustafson (1999) や Gustafson (2002) にならない，標準的な ADDIE モデルを用いることとする．ADDIE のような直線的なモデルは複雑なタスクには向かない (Merrienboer and Martens, 2002) との指摘がある一方で，ラピッドプロトタイピングなども ADDIE の各工程が形を変えて網羅されている (鈴木, 2005a) との指摘もあり，ADDIE を採用することとした．

ID 理論 どの ID 理論（モデル）に基づいているのかを示す．Merrill (1997) は，教授方略（ID 理論）の実装が必要との立場をとり，その点で他の一般的なツールと区別している．ID においては，プロセスを踏むこと自体よりも，プロセスの各段階で何をどう行うのかの手法に，ID の研究知見が詰まっている (鈴木, 2005a)．設計において教材や学習支援システムの最終形のあるべき姿（青写真）の要件を記述した ID モデル・理論は，e ラーニングの質の向上にとって最も重要な要素と考えられる．よって，本研究においては設計レベルの ID 理論が組み込まれているツールのみを対象とした．なお，ID モデルの研究においては，モデルの構築や改善に関する理論的な研究と平行して，モデルを使うための技法も研究されてきた (鈴木, 1989b)．その 1 つとして，モデルの使い勝手を向上させるために，支援システム（AID ツール）の開発などが行われてきたという経緯もある．

目的と証拠 システムが何を目的としているのかと，システムが効果的であったことを示す評価結果があれば，あわせて示す．Spector and Ohrazda (2004) は，AID のこれまでの評価研究を踏まえながら，多くの研究ではシステム（ツール自体）の形成的評価と，ユーザがシステムで作成した学習環境の形成的評価が行われてきたとす

る。Gros and Spector (1994) は、主にコースウェア開発に関する AID ツールの評価の枠組みを提供している。評価については、開発プロセスだけでなく最終的な成果物も考慮に入れる必要があるとし、製品とユーザと文脈という3つのレベルから、プロセス（開発環境）と成果物（ここではコースウェア）の両方を評価するモデルを提案している。

入力 重要な操作の内容（設定など）や、データとして入力する内容を示す。入力が多岐にわたる場合は、すべてを記載できるとは限らないが、作業量の目安にもなり、出力と対応づけることで、システム像が明確になると考えて追加した。

出力 システムの出力として何が生成（表示）されるのかを示す。

タスク支援 ID 活動に対してどのような支援を行っているのかを示す。Merrienboer and Martens (2002) は、ID ツールを特殊な EPSS とみなせるとし、AID ツールのタスク支援について、①リソースやデータベースの提供による「ライブラリや情報支援」、②特定のタスクを実行するためのルールや指示を提供する「標準化支援」、③自動化されたツールやエキスパートシステムやウィザードを提供する全体あるいは部分的な「タスクの自動化」、④タスク実行を助けるジャストインタイムな学習教材を提供する「インストラクション」の4つを基本的な支援の種類として設計されるとしている。また、Nieveen and Gustafson (1999) は、ジョブエイドの種類を、工具箱、DIY キット、クックブック、全自動洗濯機というメタファーを用いて分類している。

自動化の程度 ID 活動がどれだけ自動化されているのかの目安を高、中、低で示す。ID 活動をツールが代行する（ツール中心の）場合は高く、ID 活動をツールが支援する（ユーザ中心の）場合は低いとし、それが混在しているツールを中程度とした。Spector and Ohrazda (2004) は、人間の活動をコンピュータによる実行に置き換えるような自動化を、強い (strong) 支援のシステムと捉えている。また、置き換えるのではなく、人間の能力を拡張するような自動化は、弱い (weak) 支援のシステムであり、一般にパフォーマンスサポートとみなせるとする。また、強いシステムは明確に定義された狭い領域でのみ機能し、開発プロセスに対応しているものが多く、分析や設計はほとんどないと述べている。

制限 システムにどの程度制限があるかを示す。例えば、適用領域や学習課題、柔軟性、拡張性、表現力に関してなど、特筆すべき制限がある場合には、記述することとした。Murray (1999) は、教授方略の制限や柔軟性の欠如など、各分類について制限の視点から分析しており、それを参考に項目を設定した。一方で、Jones et al. (1992) は、3種類のエキスパートシステム（AID ツール）を比較する際に、汎用性の観点（制限とは逆の観点）で分析している。特に自動化の程度が高い（強い）ツールは制限が多くなる。

対象ユーザ IDer・初心者 IDer, および SME など, 誰の利用を想定しているのかを示す. 明示されていなければ, システムの内容から判断した. これはツールの利用対象であり, 出力(成果物)の対象ではない. Kasowitz (1998) は, IDer の作業効率だけでなく, 初心者や SME にガイドすることが AID ツールの強みとしている. その他 上記の項目では示し切れない工夫等を記述する. 他にも, ツールの分類, ID 理論の学習, 標準化や再利用への対応, 開発や利用のコスト(作業時間の目安など), 実行環境, システムの複雑さ, 協同作業への対応, 利用状況などの視点も考えられる.

2.4 AID ツールの事例

これまでに数多くの AID 研究が行われてきたが, 今回は特徴の異なる 5 つの事例を, 作成した整理の枠組みにあてはめた(表 2.1). なお, ここに記載した内容は, ARCS 改善方略ガイドを除いて, すべて論文からの情報であり, 実際に利用してはいない.

1. GAIDA(Spector and Song, 1995) は, Guided Approach to Instructional Design Advising の略であり, 初心者や SME が教材開発を行う際に, ガニエの 9 教授事象 (Gagne and Medsker, 1996; ガニエ他, 2007) を適用できるようにアドバイスを提供するシステムである.
システムは, ガイダンスとレッスン(教材)事例の 2 つのモードから構成される. ユーザには, ガイダンスにおいて 9 教授事象の各事象の説明が提供される. また, 事例として 4 種類の教材(海軍の記章を同定する, 電気抵抗を分類するなど)が登録されており, ユーザが開発中の教材に最も近いものを選択し, その事例を参考にできる. ガイダンスでは, 各事象の説明に加えて, 事例教材が具体的にどのような工夫をしているのかの解説も表示され, そこから直接教材を参照できるようになっている. さらに, コースウェア開発のための留意点も提供されている. ガイダンスとレッスンファイルは独立になっており, レッソンの追加が可能となっている.
なお, ガニエの 9 教授事象は, 学習の情報処理モデルに基づいて学習支援のフェーズ(事象)を 9 つに分類した枠組みで, ID モデルの最古参であり最も広く用いられているものである(鈴木, 1989b). 9 つの事象は, ① 学習者の注意を喚起する, ② 授業の目標を知らせる, ③ 前提条件を思い出させる, ④ 新しい事項を提示する, ⑤ 学習の指針を与える, ⑥ 練習の機会を作る, ⑦ フィードバックを与える, ⑧ 学習の成果を評価する, ⑨ 保持と転移を高めるである(鈴木, 2002).
2. ARCS 改善方略ガイド (Suzuki et al., 2004) は, ARCS モデル (Keller, 1987; 鈴木, 1995) の観点から教材(授業)を評価し, 魅力(学習意欲)の側面について改善するための支援を行う Web システムである. 教材評価アンケートの収集および

集計機能を有し、その結果に応じて、あらかじめ登録されている ARCS モデルに基づく方略を提示する。アンケートは 9 段階のリッカート尺度で、4 要因の各下位分類 3 つずつの計 12 項目で構成される。

ARCS モデルは、学習意欲を注意 (Attention)、関連性 (Relevance)、自信 (Confidence)、満足感 (Satisfaction) の 4 側面にとらえ、授業の魅力を高める作戦を整理するための枠組みである。これまでの膨大な動機づけに関する心理学的研究や実践からの知見を統合して、実践者向けに使いやすい形にまとめたものである (鈴木, 2002)。

3. ADAPT-IT (Croock et al., 2002) は、Advanced Design Approach for Personalized Training Interactive Tools の略であり、4C/ID モデル (Merriënboer et al., 2002) の適用を促す 2 つのツールから構成される。

1 つ目のツールである Core は、複雑な認知スキルの分析と、コンピテンシーに基づく訓練プログラムの青写真を設計することを支援する。実際のプロセスは、複雑なスキルを分解し、タスクのクラスを並べ、学習タスクを設計する。その際に、理論を適用できるようにテンプレートへの入力や、視覚的なスキルの階層図の作成、理論に沿っているかをチェックする機能などを利用して作業を進めていく。また、プロセスはユーザの好みでどこからでも開始することができる。2 つ目のツールである Eval は、訓練プログラムの評価や訓練の基になっている青写真の改善を支援する。Web 上のアンケートフォームなどからデータを収集して分析を行い、その結果を表示する。評価はカークパトリック 4 段階評価モデルのうちのレベル 3 まで (反応、学習、行動) に対応する。

4C/ID モデルは、複雑な学習 (課題) のための環境が、学習タスク、支援情報、ジャストインタイムな情報、部分的なタスク練習の 4 つを、相互に関連する青写真の構成要素として記述できるとする ID 理論である。それらの青写真を作成していくための 10 段階の活動があわせて提案されている (Merriënboer et al., 2002)。

4. IDLE-Tool (Bell, 1998) は、ゴールベースシナリオ (Goal Based Scenario; GBS) (Schank et al., 1999; 根本・鈴木, 2005) に基づいた教材を作成するためのテンプレートベースのツールである。

IDLE (Investigate and Decide Learning Environments) とは、意思決定のために情報収集を行うという種類の GBS を示している。このツールは、SME がプログラミングや ID の専門的知識無しに IDLE の教材を作成することができる。そのサンプルとして鎌状赤血球症のカウンセラーの教材が搭載されており、学習者は遺伝カウンセラーの役割となって、患者の検査や、リスクの計算、専門家への相談を通して、最終的に患者へアドバイスを行う。学習者の活動は、問題、実行、意思決定、伝達、結論という 5 つのフェーズから構成されている。オーサリン

グは、ガイドされた事例の適用 (Guided Case Adaptation) と名付けられた手法に沿って進められ、ユーザはサンプル (テンプレート) を徐々に変更していきながら GBS 教材を作成する。

GBS は、シナリオ型教材を設計するための ID 理論であり、現実的な文脈の中で「失敗することにより学ぶ」経験を擬似的に与えるための学習環境として物語を構築するための理論である。GBS は、学習目標・使命・カバーストーリー・役割・シナリオ操作・情報源そしてフィードバックの7つの要素で構成される (根本・鈴木, 2005) 。

5. ID-Expert (Merrill, 1998) は、教授トランザクション理論 (Instructional Transaction Theory ; ITT) (Merrill, 1999; 鈴木, 2005b) に基づいて構築された学習環境とオーサリング環境を提供するシステムである。

ID 理論に基づく方略をアルゴリズムとして実装し、ナレッジオブジェクト (Knowledge Object) と呼ばれる知識ベースと分離して構成している。ユーザは知識を登録しさえすればよく、方略はシステム側で自動的に生成することになる。ID-Expert は IDENTIFY (同定) のトランザクションに対応しており、学習環境は提示、探索、練習、テストの4種類のインタラクションを学習者に提供する。オーサリング環境は、コースの作成、知識の登録、リソースの関連づけなどのトランザクションの設定を行う。また、学習者の状況などをパラメータとして選択でき、それに適した方略が提供されるようになっている。

ITT は、ID の自動化による効果的な教授と効率的な開発を目的とした ID 理論である。教授内容 (構成要素とその関係) を示した知識表現 (データ) を指定すれば、ID 原理に基づくあらかじめ内蔵された方略 (アルゴリズム) に基づいた学習環境が自動的に構築されるということを目指している。例えば ID の素養が無い SME でも、こういったツールがあれば効果的な学習環境を構築できるということになる。ITT では、ある種類の知識やスキルを獲得するアルゴリズム (方略) のセットをトランザクションと呼び、知識はナレッジオブジェクトとして表現する (詳細は第4章で説明する) 。

2.5 事例整理を通しての考察

本節では、AID ツール事例の整理を通しての考察として、AID ツール研究において留意しておくべき点について整理する。

表 2.1 AID ツールを整理する枠組みと事例

名称	GAIDA	ARCS 改善方 略ガイド	ADAPT-IT	IDLE-Tool	ID-Expert
文献	Spector and Song (1995)	Suzuki et al. (2004)	Croock et al. (2002)	Bell (1998)	Merrill (1998)
ID プロセス	設計	評価, 設計	分析, 設計, 評価	設計, 開発	設計, 開発
ID 理論	9 教授事象	ARCS	4C/ID	GBS	ITT
目的と証拠	理論の把握と教材の効果向上, 評価は形成的評価	教材の魅力を改善, 評価はユーザビリティ・小集団・ケーススタディ	効果的なトレーニングの設計と改善, 評価はプロトタイプユーザビリティ	GBS (IDLE) 教材の開発, 評価はフィールドテスト	学習効果と効率を向上した教材の開発
入力	作成する教材に近い事例の選択, 説明の欲しい事象の選択	Web 上のアンケート (項目固定) から収集される評価データ	複雑な認知スキルの内容 (Core), 評価データ (Eval)	作成する教材の内容	知識 (ナレッジオブジェクト, リソース) とトランザクション設定
出力	ガイド (理論の説明と事例)	集計データと方略リスト	訓練の青写真 (Core), 青写真の改善提案 (Eval)	ストーリー型教材	探索とガイダンスを組み合わせた学習環境
タスク支援	事例ベースのガイド (Guided Approach), 教材への適用はユーザが行う	アンケートは自動集計, 集計データ等の見方は説明有り, 方略の適用はユーザ次第	テンプレート, GUI の課題分析, 理論の適合性確認 (Core), アンケート生成と集計等 (Eval)	サンプル (テンプレート) を徐々に編集していくデザイン方略 (Guided Case Adaption)	自動的な教材生成. 知識登録は GUI 上で行う
自動化	小	中	中	中	高
制限	提供される事例の種類によって適用範囲が制限	登録している方略によって適用範囲が制限	開発部分はサポートしていない	GBS のうち IDLE (意思決定のための情報収集の活動) のみ対応	IDENTIFY (同定) のトランザクションのみ対応
対象ユーザ	初心者, SME	初心者, SME	IDer, 初心者, SME	初心者, SME	IDer, 初心者, SME
その他	事例の追加可能な構造		Core と Eval の 2 種類のツールから構成		

2.5.1 ID 理論の適用

ID 理論の教育への適用を支援する AID ツールは, 成果物が ID 理論の描く青写真の通りになることが目標となり, それをうまく支援することがツールで最も配慮すべき事項と

なる。

ID-Expert が基づく ID 理論である ITT は、AID を志向したシェル構築のための理論である。そのため、ITT に基づいたシステムを構築するということは、AID ツールを開発することとほぼ同義となる。ツール自体の青写真を描いている理論とも換言できる。一方でその他の理論をツールに組み込むためには、工夫が必要となる。GAIDA は 9 教授事象をただ解説するだけでなく、事例をリンクすることで、理論の適用を促進している。IDLE-Tool は、教材をテンプレート化し、それを編集していくことで、GBS に基づいた教材を開発できる。また、テンプレート自体が事例にもなっている。ADAPT-IT は階層図の作成のための GUI ツールを提供したり、4C/ID モデルに沿っているかをチェックできる機能を備えている。

ADAPT-IT や ARCS 改善方略ガイドは Web 上のアンケート機能を備え、簡単に評価を実施することが可能になっている。また、IDLE-Tool は、GBS 理論の扱う主要な課題を 8 種類に分類したうちの 1 つを実装することで、ツールの適用範囲をせばめ、それによってテンプレート化を実現している。GAIDA においても事例の種類がツールの適用範囲を制限しているとも言える。ID 理論の汎用的なガイドもできるが、一方で踏み込んだ支援を行うためには、適用範囲を狭める工夫が必要となっている。

2.5.2 ID 理論の学習

Spector and Ohrazda (2004) は、パフォーマンスサポートシステムにおいて、意識決定や問題解決の背後にある原理やプロセスを隠すような形態をブラックボックス、その逆をガラスボックス（透明なシステム）と呼んでいる。ユーザが専門性の獲得を望まないのであれば、システムはブラックボックスであるほうが望ましく効果的であり、ユーザが専門性の獲得やより高次の能力の獲得を期待している場合は、ガラスボックスが好ましいとする。本事例では、GAIDA のようなシステムは ID 理論（この場合 9 教授事象）を理解して適用してもらうことが目的であるためにシステムの透明性は高いが、特に ID-Expert のような自動化の程度が高いシステムは透明性が低いということになる。

Gustafson (2002) によると、初期の頃のツールは自然とブラックボックス的なものが多かったが、現在のツールの多くがガラスボックスとして設計されている。ブラックボックスは、ユーザがより有能になるかどうかに関係がなく、可能な限り効果的で効率的に仕事を行わせるためだけに設計される。それゆえに、ユーザは、タスクを成し遂げるためにツールに完全に頼り続ける。他方で、ガラスボックスのツールの設計は、ユーザがより有能になることがゴールの 1 つであり、場合によっては、ツールが必要なくなるか、初心者には簡単には扱えない進んだ機能を使うことが出来るようになる可能性がある。ブラックボックスよりも、高価で効率が劣るとしても、人的資本への投資を望む組織は、グラス

ボックスの方に興味を持つと述べている。

Uduma and Morrison (2007) は、AID ツールに対する上級者と初級者と未経験者の利用傾向を調べた。上級者はツールを教授方略データベースが付属するワープロのように扱うこと、初心者はツールが提供するアドバイスやガイダンスや補助に頼って作業し、ID の知識を強化するためにツールを利用していたこと、未経験者は設計手法を学ぶためにツールを利用していたことが挙げられた。これは、Designer's Edge と呼ばれる ID プロセス全体を支援するアドバイス型の AID ツールを対象としており、ID の自動化の程度がそれほど高くない場合の利用傾向である。ユーザが専門性の獲得を望まない場合や、生成型のツールのように自動化の程度が高い場合は、この限りではない。

ユーザが専門性の獲得を要する場合においては、初心者と未経験者には、特に AID ツールが基づいている理論等を学習させる工夫が必要であると考えられる。ユーザが専門性の獲得を望むか望まないかは状況によって異なると考えられ、どちらか一方をあらかじめ想定することはできない。よって、自動化の程度が高い場合においても、ID 理論を学習できるようなオプションを用意しておくことが望ましいと思われる。また、ユーザの ID における専門性のレベルや、ユーザ自身や組織的ニーズに応じて提供方法を変更するようなツールも考えられる。

2.5.3 標準化と再利用

近年は LMS (Learning Management System) 上で e ラーニングを提供する 경우가多く、SCORM (Sharable Content Object Reference Model) に対応しているかどうか教材やツールの選択基準の 1 つとなっている。事例では ADAPT-IT のみが SCORM に対応予定との記述があった。今後の AID ツールは標準化への対応も必要になってくるだろう。もう少し浸透してくれば、整理の枠組みの項目の 1 つとして追加すべき内容と言える。

また、教材の再利用や共有(検索)を目的として、メタデータを付与する LOM (Learning Object Metadata) の標準化が行われている。鈴木 (2005b) は、学習オブジェクトを単に並べるだけでは教授モデルを構築したとは言えず、学習を支援する観点からの学習オブジェクトの提示方法や組み合わせ方の提案が必要としている。このような工夫が今後の AID ツールに組み込まれてくると考えられる。

Wiley (2001) は、学習を促進することを意図した学習オブジェクトとして、ナレッジオブジェクトを紹介している。また、学習オブジェクトの応用について、学習支援を成功に導くためには、ID 理論が必要であるとしている。ナレッジオブジェクトは ID-Expert が採用しており、学習オブジェクトのあり方を検討する上で参考になるとと思われる。

2.5.4 AID ツールへの批判と課題

AID ツール開発は、まだ歴史が浅い (Gustafson, 2002) . 歴史が浅い分だけ、批判はそれほど多くなく、どちらかという手探りが続いている段階と言える。よって、AID 研究においては、批判よりは、AID ツールの紹介や今後の展望を示すレビューが多かった。本節では、AID ツールに対する批判を整理しておく。

Monshinskie (1998) は、自動化されたデザイン機能は、CBT (Computer Based Training) 開発を簡単にするために長い間探し求められた答えのように見えるが、教育技術者は、そのようなソフトウェアの実用性や受用性に疑問を呈しているとする。つまり、懐疑的に AID ツールを見ている利用者が多いことを示しており、AID ツールの効果が認知されていない現状が浮かび上がる。

一方で、Merrienboer and Martens (2002) は、これまでの AID ツールが、開発を重視し、分析・設計や実装・評価にほとんど関心が向けられなかったとしている。また、近年の ID の動向を踏まえ、設計プロセスの複雑性が増加しており、ID はもはや単独で 1 人の教師によるコースをデザインするものではなく、単純で直線的な ADDIE モデルは役に立たなくなってきたとする。さらに、IDer や SME やインストラクタなどのチームにおける共同作業の機会や、ステークホルダーのグループが大きくなり、組織的な成果などの必要性が増していると指摘し、AID ツールは開発プロセス以外にも目を向けるべきであると主張している。しかしながら、この議論は比較的複雑な ID を想定しているが、より単純なコースを開発する場合でさえ、AID ツールが生かされていない現状がある。

AID ツールはこれまでに多数開発されてきたが、普及しているとは言い難い。普及してこなかった原因について、直接言及している研究は見あたらなかった。Murray (2003) は、25 個の ITS オーサリングツールの活用度を示しているが、初期段階のプロトタイプでのコンセプトの検証が 6 件、プロトタイプを使っでの評価が 7 件、ほどほどに評価されるか使われているのが 8 件、よく使われているのが 4 件であった。この結果が示すように、よく使われていたツールはほとんどない状態である。プロトタイピングに留まっているのは、ツールを提供する目的よりも研究目的のためのツール開発が多いことを示していると思われる。これは ITS という比較的高度なシステムであることから理解に容易い。いくつかは商用化に至っているものの、販売からすでに撤退している事例も多いことから、ビジネスとしてはあまり成立しない可能性もある。また、現状で利用可能なものは、高度で複雑なツールが多く、利用するにはハードルが高い。

Monshinskie (1998) は、AID ツールの利用について、ASTD (American Society of Training and Development) のメンバーにアンケート調査を行った (有効回答は 481 件である)。その結果のうち、AID ツールに望まれる機能については、他の機能と比較し

て、テンプレートとの回答が圧倒的に多かった（テンプレートが第 1 位で 145 名、分岐とフィードバックが 2 位で 37 名であった）。これは利用者にとって、ID の知識は無くとも簡単に効果的にコンテンツを生成したいという要求があると推測される。ただし、Gayeski (1990) は、AID ツールが ID プロセスを単純作業化して創造性を失うと主張しており、このようなテンプレート利用などの自動化の程度を高くしていくこととは、トレードオフの関係にあると思われる。

Gustafson (2002) は、AID 研究が初期の頃のユーザは、ツールの使い方とそれらが支援するプロセスの専門性を持つことの両方が必要であったと述べている。その後、比較的短い期間で様々なツールが開発され、今後も増加傾向にあるとし、ID の知識を持たない初心者の SME から ID 専門家までのすべての人が、今後これらのツールに興味を持つようになると予測する。

Merrill は鈴木 (2006, p.128) の中で、「ID 専門職は教材ではなく作成ツール作りを」という文脈（コラム）において、インターネット上で誰もが発信者で誰もが教師とされている状況で効き目のない教材が大量にあるということを指摘し、化学者が手にすれば簡単にすばらしい化学教材が作れるような、ID 原理が内蔵されたツールが必要と指摘している。車にたとえると、車の運転には自動車修理工の知識は必要としないが、昔ならその必要があったとし、ID の現状はその昔の状態にあるとしている。ここで Merrill が目指しているのは、SME（化学者）が利用できる、教材生成型の AID ツールであり、そこに未来があると述べている。

Kasowitz (1998) が、ID 専門家が不足している場合や、SME などが教授開発に責任を持つような状況で AID ツールが特に役に立つとしており、SME は AID ツールの主要なユーザ層である。しかしながら、現状では SME が AID ツールを活用しているとは言い難い状況にある。よって、SME が簡単に利用可能なツールを開発し、誰もが利用可能な形で公開していくことが必要であると思われる。

2.6 SME 用の AID ツール

SME とは、特定の領域に関する専門家を指す。本研究では特に IDer と対比して用いることとし、SME は ID の素養を持たない、内容の専門家の意味で利用する。また逆に、IDer は内容の専門家でなく、教育の専門家であるため、質の高い教育を提供するためには、IDer と SME が協力する必要がある。内容の専門家がどこまでの専門性を要するのかは、特に明確には定められていないが、本研究では教材作成という状況において、その教材で教える内容を理解しているという程度も SME とみなしている。既存の AID 研究において、SME が利用できるための要件については、あまり議論されていない。

表 2.1 に整理したように、SME を想定している（あるいは利用できそうな）ツール

表 2.2 AID ツールのタスク支援の種類 (Nieveen and Gustafson, 1999)

種類 (メタファー)	特徴
工具箱	ただの道具の集まりである。ユーザは開発プロセスに道具を自由に適用するが、それらを利用した結果は、ユーザの専門性や経験に左右される。
DIY キット	製品を組み立てるためにあらかじめ用意された部品群が利用ガイドとともに提供される。ユーザの自由度は利用可能なコンポーネントによっていくらか制限される。
クックブック	良い結果を導くための処方を示すガイドライン集である。アドバイスのアプローチであり、コンピュータがユーザにアドバイスを行うが、実装するかどうかはユーザにまかせる。
全自動洗濯機	タスクのほとんどが支援システムによって実行 (制御) される。ユーザはいくつかのパラメータを設定するだけで、システムは良い結果を保証する。ユーザは ID プロセスの知識を事前に持つ必要がない。

は多いが、自動化の程度によって SME が ID の知識を身につけながら利用するのか、まったく必要としないのかによって、ツールの目的や利用の難易度が変わると考えられる。Spector and Ohrazda (2004) は、ツールを分類する視点として、支援の強さが強い (strong) か弱い (weak) かを示した。Halff (1993) は、AID ツールをアドバイス型のシステム (Advisory system) と生成型のシステム (Generative system) の 2 種類に分けて整理しており、生成型の方が強い (strong) であるとしている。強い支援の方が ID をブラックボックスとして扱うことになり、ID の専門性を高めることをあまり目的としない SME 向きであると考えられる。ID 理論を学習してもらう機会を提供することも重要であるが、それはオプションとすべきであろう。

Nieveen and Gustafson (1999) では、AID ツールのタスク支援 (ジョブエイド) の種類を、「工具箱 (toolbox)」、「DIY キット (do-it-yourself kit)」、「クックブック (cookbook)」、「全自動洗濯機 (automatic washing machine)」の 4 種類のメタファーに分類している。それぞれの特徴を表 2.2 に示す。最も自動化の程度が高いのは、全自動洗濯機であり、事前に ID プロセスに関する知識が必要ないとのことから、最も SME に適していると考えられる。

Merrill (1997) は、教材開発ツールを 5 つのレベルに分類しているが (表 2.3)、最初の 3 つのレベルはすべて構造志向であり、システムに含まれるインタラクションから生徒が何を学ぶかというよりむしろ、ツールがどのように動くのかに焦点が当てられているとする。レベル 4 や 5 の学習志向のシステムは、システムに組み込まれた教授方略が科学的な教授の原理に基づいている。これによってオーサリング作業が、かなり簡単になり、インストラクションがより効果的になるとし、適切な教授方略が組み込まれた学習志向のシステムの必要性を提案している。また、学習志向の ID ツールの長所として、プログラミン

表 2.3 教材開発ツールの段階 (Merrill, 1997, から抜粋して翻訳)

レベル	志向	項目	内容
1	構造	プレゼンテーションツールなどの情報コンテナ	プレゼンテーションツールやホームページ作成ツールといった、教材作成には全く配慮されていないが、情報を載せて手軽に提供できるツール。インストラクションよりはナビゲーションを提供する。
2	構造	オーサリングシステム（教授方略が実装されていない）	ToolBook に代表されるオーサリングシステムで、教材作成のプログラミングの負担を軽減するが、教材作成のためには分析や設計をユーザ側が行う必要がある。フレームから他のフレームへの分岐で教材を構成するものが主である。
3	構造	テンプレート等で教授インタラクションを拡張したオーサリングシステム	オーサリングシステムに学習者とのインタラクションを提供するテンプレートを用意し、ウィザード等の指示でテンプレートを埋めていくことで生成できるもの。ただし、ユーザは適切な系列化や、状況にあったテンプレートを用いて教授方略を計画しなければならない。
4	学習	学習志向の ID ツール (ID 原理に基づく方略を実装)	科学的に証明された教授の原理に基づく教授方略をあらかじめ実装したツールである。テンプレートは通常、1 種類のインタラクションを提供するが、これは学習者がある種類の知識やスキルを獲得するために必要となる完全なインタラクションのセットを提供する。
5	学習	学習者に適応した学習志向の ID ツール	学習者の様子をモニタしながら、学習活動の流れを動的に変更するなど、個々の学習者の状況に応じた制御を行う学習志向の ID ツールである。

グ時間の短縮（開発効率）はもちろんのこと、構造志向のツールよりも初心者の IDer や SME が、効果的な教材を開発できることを挙げており、このようなツールは SME に役立つと考えられる。なお、本章で取り挙げた ID-Expert は、レベル 5 に相当し。第 4 章で紹介する Instructional Simulator（こちらも ITT に基づいたツール）はレベル 4 に相当する。

また、Goodyear (1997) は、生成型の AID ツールは、ITS の研究や開発によるところが大きいのとする。Murray (2003) は既存の ITS オーサリングツールを整理しているが、ITS は、何を教えるかを規定する教授内容のモデルと、どのように教えるかを規定する教授方略を伴う、コンピュータベースの教授システムと定義づけている。一方で、日本教育工学会 (2000, p.13) によると、ITS は人工知能技術を応用した教授システムであるとし、対象領域知識を表現した教材知識、学習者の知識状態を表現した学習者モデル、教授方法を実現した教授知識、対話インタフェースから構成されるとする。Murray (2003) では ITS オーサリングツールの 1 つとして Instructional Simulator が紹介されているが、Merrill (1997) ではレベル 4（表 2.3）の学習者に適応していないツールとして紹介されており、日本教育工学会 (2000, p.13) の学習者モデルとの矛盾が生じる。Murray (2003) の定義であると、Merrill (1997) のレベル 4 以上が ITS に相当することになるが、本論文においては、日本教育工学会 (2000) の定義に従って人工知能技術を応用した高度なシス

テム，すなわち Merrill (1997) のレベル 5 として，ITS と捉えることとする．例外としては，Murray (2003) を参照する場合のみ，Murray の定義に従う．

第 1 章では，AID ツールにおける社会的ニーズとして，日本には IDer が少ない（ほとんどいない）こと，インターネットの普及によって誰もが発信者となる状況を踏まえ，SME が利用できるコンピュータ上の教材を生成するための AID ツールの開発が必要であることを示した．一方で，このようなツールが利用可能な状態になっているとは言い難いことが現状である．

Merrill (1997) における主張は，教材開発ツールにおいて，レベル 4 以上の学習志向のツールが必要であることを示しており，これは SME が利用する AID ツールとして適していると考えられる．Gustafson (2002) は，こういった SME の CBT 開発を支援する Merrill の成果は，特に注目すべきであるとしている．また，Merrill (1997) の示すレベル 5 のような高度なシステム (ITS) とまでいかなくとも，レベル 4 でも十分に効果的なツールは開発できると思われる．

以上の理由から，本研究においては，SME でも利用可能な AID ツールとして，生成型（あるいは全自動洗濯機のメタファー）で，Merrill (1997) のレベル 4 に相当する，ID 理論を内蔵した，コンピュータ上の教材開発（主に設計と開発プロセスを自動化する）ツールを研究対象とする．これを AID 教材シェルと呼ぶこととし，次節で定義する．

2.7 AID 教材シェル

2.7.1 定義と構成

シェルという用語は，一般に UNIX においてカーネルの外側にある殻の部分に相当するコマンドインタプリタの意味（山口ほか，1992）でよく使われている．また，Murray (2003) は，ITS の文脈で，シェルは ITS を構築するための汎用のフレームワークであるとする一方で，オーサリングツールはプログラミングができない人たちへのユーザインタフェースを伴うシェルであると定義している．Merrill et al. (1991) は，教授トランザクション（知識や技能の種類に応じた教授方略）をアルゴリズムとして実装しておき，知識をデータとして登録することで，教材を生成するシステムを教授トランザクションシェルと呼んでいる．知識は入れ替え可能であり，教材を配信する学習環境と教材を作成するオーサリング環境から構成されている．UNIX のシェルは複数存在し，ユーザは好みに応じて異なるシェルを利用するが，逆に教授トランザクションシェルは，1 つの枠組みを用意し，中身を入れ替えることで複数の異なる教材が生成される．これは Murray (2003) のオーサリングツールの用法とも同義である．本研究においては，Merrill et al. (1991) と同じ意味で「シェル」という言葉を用いる．

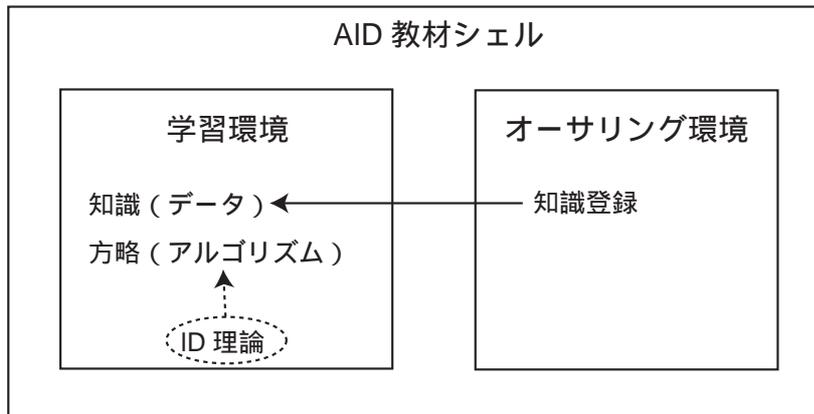


図 2.1 AID 教材シェルの最小構成

よって教材シェルは、コンピュータ上で動作し、知識（データ）とアルゴリズム（方略）から構成される学習環境と、知識を登録するためのオーサリング環境を備えるシステムとする。また、AID ツールの視点からは、前述したように本研究では設計レベルの ID 理論を踏まえていることを AID ツールの条件としているため、教材シェルにおいても、ID 理論に従った教授方略が実装されていることが必要である。また、自動化対象の ID プロセスとしては、設計と開発プロセスを支援することになる。Merrill (1997) の分類に従うと、学習志向の ID ツール（レベル 4）となる。

以上から、AID 教材シェルは、「知識（データ）とアルゴリズム（方略）から構成される、ID 理論を内蔵した学習環境と、知識を登録するためのオーサリング環境を備える、ID プロセスの設計と開発を自動化するコンピュータシステム」と定義する。AID 教材シェルの最小構成を図 2.1 に示す。AID 教材シェルは、「学習環境」「オーサリング環境」の 2 種類のサブシステムから構成される。最小としたのは、例えば、方略を制御するパラメータの設定や、知識や方略を最適に関連づけるルールなど、様々な制御が追加される可能性があるからである。

2.7.2 学習成果の分類

Gagne の学習の条件 (Gagne, 1985; ガニエ他, 2007) は、学習成果の 5 分類（言語情報、知的技能、認知的方略、運動技能、態度）のそれぞれに、その成果を獲得するのに必要な学習の内的条件（あらかじめ前提として保持しておくべき知識やスキル）と外的条件（学習の促進のために外側から行うべき働きかけ）を整理した ID 理論である。つまり、Gagne は異なる学習の成果には異なる学習の条件が必要であるとする。知識の種類（学習成果の分類）に応じて、教授方略も変わることになる。

Gagne の視点に立てば、AID 教材シェルはあらかじめ教授方略を内蔵するものであり、

表 2.4 認知領域における学習成果の分類 (ガニエ他, 2007, p.154 を抜粋)

能力	能力動詞	例
言語情報	述べる	1932 年の大統領選の争点について口頭で述べる
知的技能		
弁別	弁別する	フランス語の u と ou の発音を照合することで弁別する
具体的概念	同定する	それぞれの植物の根, 葉, 茎の名前を正しく言うことで同定する
定義された概念	分類する	家族の概念の定義を書くことで分類する
ルール	例示する	計算過程をすべて示しながら, 正の数と負の数の足し算を書いて例示する
高次のルール	生成する	ROI の見積もりを含めたビジネスプランを書くことで生成する
認知的方略	採用する	合衆国の地図を思い浮かべて州の名前を思い出す方略について, 利用する方略を説明することで採用する

生成する教材が枠組み（シェル）を通して提供されるため，その枠組みが対応する学習成果の分類によって，実装する方略が変わり，提供する教材の複雑さも変わってくることになる。

Gagne の学習成果の 5 分類のうち，認知領域に関わるのは言語情報，知的技能，認知的方略の 3 種類である。言語情報は，一度接した名前や記号，史実などの各種データを覚えて，それを思い出す課題，知的技能は，学んだルールなどを未知の例に適用する課題，認知的方略は自らの学習を効果的にするための作戦を習得する課題となる（鈴木, 2004）。ガニエの学習成果の 5 分類は，言語情報と知的技能を分けていることが特徴的であり，暗記した内容を後ほど述べられるようにすることと，身につけた内容を未知の課題に適用することを分けている。また，認知的方略は，知的技能の一種としながらも，自らの学習方法の選択に関わる点が，知的技能と異なっている。また，知的技能には，弁別，具体的概念，定義された概念，ルール，高次のルール（問題解決）という下位分類がある。参考までに言語情報，知的技能，認知的方略における学習目標の記述例を表 2.4 に示しておく。

以上のように，AID 教材シェルを開発する場合において，学習成果の分類を踏まえて構築する必要がある。

2.8 研究課題

本章の議論から，本研究においては 2 種類の AID 教材シェルを開発することを課題とした。

ITT は AID 教材シェル構築を志向した AID 研究の中核的理論であり，SME が利用可能な AID 教材シェルを目指す上でも欠かすことはできない。ITT の詳細は第 4 章で述べるが，ITT に基づく教材シェル（以降，ITT 教材シェル）は現在利用可能となっているものではなく，研究も途上にあり，開発して利用可能とすることは，AID 研究にとって意義

があると考えられる。そこで、本研究においても ITT を中心の 1 つとしてとりあげる。

また、ITT は、学習成果の 5 分類にあてはめると、知的技能の課題を対象としたものである。よって、より単純な学習課題である言語情報を扱った AID 教材シェルも必要であると考えられる。言語情報を学習するためには、例えばこれまでにドリル型教材が多数開発されてきたが、ID 理論を活用した教材シェルは提供されていないのが現状である（詳細は第 3 章で述べる）。ドリル型の教材シェル（以降、ドリルシェル）は、登録する知識構造が単純であるため、より SME が利用しやすい環境であると推測される。

そこで、本研究では、言語情報としてドリルシェルと、知的技能として ITT 教材シェルの 2 種類の AID 教材シェルを研究対象とし、開発を行うことにした。ただし、ITT においても言語情動的な要素が含まれ、ドリルシェルにおいても知的技能に対応は可能であるため、厳密に両者が重なる部分はある。

一方で、認知的方略は、学習の方法を学習するというメタ認知に関わる課題となるが、例えば、学習者がその方略に触れたり、直接教わることを通して身につけていくことになる。AID 研究で特に議論されてこなかった課題として、認知的方略への対応が挙げられる。これまでに、ID 理論をユーザが使えるようにするという視点では研究されてきたが、その方略を学習者に適用した際の、認知的方略への効果については研究されていない。教材開発を簡単にし、決められた方略をあらかじめ実装し、データを入れ替えることで複数の教材を提供できるという教材シェルの特性は、認知的方略に効果があると考えられる。

教材シェルのユーザは、教材作成者（オーサリング環境利用者）と学習者（学習環境利用者）の 2 種類が想定されるが、教材作成者と学習者が同様であること、つまり自分の学習のために教材を自ら作成することも考えられる。よって、オーサリング環境に ID 理論を学習する機能を付加しておけば、デザイナーの専門性を高めるだけでなく、学習者の認知的方略の能力を高めることにつながると思われる。これは SME が利用できることが前提となる。この場合はオーサリング環境上で ID 理論が開発者（＝学習者）にガラスボックスとなる。

また、教材シェルは、同様の方略を複数の教材に適用する形式であり、学習者は同じ方略を異なる事例に適用した教材に触れることになる。認知的方略は知的技能の一種であるため、学習環境で学習者に ID 理論に沿った方略を明確に提示することで、認知的方略を促進する環境になると考えられる。つまり、学習環境上で ID 理論の方略が、学習者にとってガラスボックスになるということである。

以上から、SME が利用できる AID 教材シェル開発を中心的課題としながら、認知的方略からの議論も本研究に含めることとし、前者をドリルシェルで、後者を ITT 教材シェルを通して検討することとした（表 2.5）。

表 2.5 研究課題（開発する 2 種類の AID 教材シェル）

AID 教材シェル	学習成果の分類	認知的方略の視点	章
ドリルシェル	言語情報	学習者自身が教材を作成する際にオーサリング環境で ID 理論を学習する	第 3 章
ITT 教材シェル	知的技能	学習者が複数の教材を適用した学習環境に触れることで ID 理論の方略を学習する	第 4・5 章

2.9 まとめ

本章では、AID ツールに関する研究の動向を概観した。また、先行研究を参考に AID ツールの分類枠を独自に作成し、ID 理論に基づく AID ツールの 5 事例をあてはめて整理し、ID 理論の適用・ID 理論の学習・標準化と再利用の観点から考察を試みた。さらに、AID 研究への批判や社会的ニーズを踏まえて、SME が利用できる AID ツールの必要性を示し、SME にとっては AID 教材シェルの形式が有効であることを述べた。これらの議論を通して、本研究では、SME が利用できるツールを目指して 2 種類の AID 教材シェルを開発することにし、Gagne の学習成果の 5 分類から、言語情報としてドリルシェルを、知的技能として ITT 教材シェルを対象とすることにした。また、SME が利用できる AID 教材シェル開発を第 1 の研究課題としながらも、AID 教材シェルは認知的方略の学習に適した環境であると考え、認知的方略の学習についても研究課題とした。

第 3 章

統合型ドリルシエル「ドリル工房」 の開発

3.1 はじめに

e ラーニングで提供される教材の 1 つにドリル型教材がある。これは主に学習内容の練習のために用いられ、よくクイズという形で提供されている。既存の LMS (Learning Management System) の多くはクイズを利用できるが、単に学習者の回答を要求し、フィードバックを行うものが多く、練習を支援するメカニズムには配慮されていない。

ドリルは、行動主義が全盛だった 1960 年から 70 年代にかけて、個別学習環境のドリル型 CAI (Computer Assisted Instruction) としてよく研究されてきた。その後、人工知能技術を応用して知的 CAI や ITS (Intelligent Tutoring System) といった知的システムに発展した。ITS は複雑なシステムであり、効果的な学習を可能とする反面、構築に手間がかかる。一方で、テスト理論である項目反応理論を利用したドリル型 CAI (許・繁樹, 1990) など、テスト研究を応用して効果的なアイテムの制御を行っている事例もある。ドリルの仕組みはテストと類似する部分もあるが、テストは診断や能力測定などの目的であるのに対し、ドリルは練習目的であり、アイテムに不正解した場合に理解するまで再出題の制御を行うという点などが異なる。

ITS やテスト理論を適用したシステムのような高度な制御を行わずとも、かなり単純な制御でも、相応の学習効果があると言われている (鈴木, 1998)。これまでにインストラクショナルデザイン (ID) の研究によって、練習を効果的に行うドリル制御構造が複数提案されてきた (例えば, Salisbury, 1988; Alessi and Trollip, 2001)。ドリル制御構造 (鈴木, 1998) とは、アイテムの選択および除去のメカニズム (鈴木, 1989a) である。主にフローチャートで表され、ドリルアルゴリズムと言い換えることもできる。Salisbury (1988) ではドリル構造と呼んでいる。

ID に基づくドリル制御構造を組み込んだ開発ツールは、日本だけでなく海外でもほとんど存在していない。単純な制御すらも利用されていないドリル型教材が多く存在している現状においては、こういった開発ツールが必要であると考えられる。そこで本研究は、ID で提案されているドリル制御構造の適用を支援する統合型ドリルシエル「ドリル工房」を開発した。

また、作成したドリルがどのように効果的に練習を支援するかについて、作成者が知る機会を用意することも重要である。例えば、ドリル制御構造 (ID 理論) を知ることは、ドリルに限らず、他の練習場面にも応用可能であり、教育の専門性を高めることにつながる。ドリルの学習者が自分のためにドリルを作成する場合は、認知的方略 (ガニエ他, 2007) の視点から、自らの学びを工夫するきっかけにもなると考えられる。そこで、本システムは、簡単にドリルを作成できることだけでなく、既存のドリル制御構造を整理し、作成者がドリル制御構造自体について学習できることを目指した。

ドリルシエルという用語は、鈴木 (1998) のドリル教材作成支援ツールと同義である。ドリル制御構造をあらかじめ組み込んでおき、内容 (アイテム) を入れ替えることのできるシステムを指す。また、ID で提案されている複数のドリル制御構造を1つのドリルシエルに統合したという意味で統合型という言葉を用いる。

研究で扱うドリルの役割は主にガニエの9教授事象 (ガニエ他, 2007) の練習 (事象6) とフィードバック (事象7) に相当し、学習する内容に関する導入や説明 (事象5以前の事象) を済ませてからの利用を想定している。また、ドリルは認知領域における言語情報に限定する。言語情報は、ガニエの学習成果の5分類の1つで (ガニエ他, 2007)、覚えたことを再生することが求められる。対照的な分類に知的技能があり、これは未知の課題に適用する能力の習得が求められ、言語情報とは必要とされる制御が大きく異なる。

3.2 先行研究

3.2.1 ドリル制御構造に関する研究

ID の分野では、Salisbury (1988) によって、単純なものから複雑なものまで複数のドリル制御構造が提案されてきた。関連して、Merrill and Salisbury (1984) は、単語帳のような単純なドリル制御構造に関する問題点を挙げ、その改善点を5項目に整理している。Salisbury (1990) では、ドリルに関する認知心理学の知見を踏まえ、ドリルの設計について、干渉、練習の間隔、復習の間隔、短期記憶の容量、記憶の情報表現という5つの視点から整理している。Alessi and Trollip (2001) はドリルの設計原理や2種類のドリル制御構造を提案している。

日本におけるドリル制御構造に関する研究としては、鈴木 (1989a) が言語情報と知的技

能のドリルの設計について，Salisbury (1988) や Alessi and Trollip (2001) の旧版の内容を参考に ID の視点でまとめている．また，水野 (2000) が分散効果の生起過程に関する認知心理学的知見に基づき，Low-First 方式を提案している．

3.2.2 ドリルシステム

ドリルシステムという言葉は，作成機能の有り無しに関わらずドリルを提供するためのシステムという意味で用いる．鈴木 (1998) は ID に基づく HyperCard 上のドリルシェルを開発し，正解消去型と空欄補充型のドリルを作成できるようにした．改善後には項目間隔変動型と状態前進型を組み合わせたドリル制御構造を実装している．Drill Designer 2.1(The University of IOWA, 1999) は，アイオワ大学で開発された，項目間隔変動型のドリルを作成するためのコマンドライブラリであり，既存の 2 種類のオーサリングツールに対応している．利用にはプログラミングの知識を必要とするため，一般向けとは言い難い．付属のデモ版は，アイテムの動きがわかるようにキューの状態（アイテムの並び）を可視化している．Hot Potatoes(Half Baked Software Inc., 2007) は，ヴィクトリア大学で開発されたドリルシェルで，多肢選択，並べ替え，穴埋め，内容一致，クロスワード型のドリルを作成できる．標準規格の SCORM (Sharable Content Object Reference Model) にも対応していて LMS に載せることが可能であるが，特に ID やドリル制御構造に関する記載は無い．

ドリル作成以外のシステムとしては，藤原ほか (2005) が情報処理入門科目における個別学習システムのドリル機能として，状態前進型のドリル制御構造を採用している．水野 (2000) や松浦 (2006) は，Low-First 方式による CAI 教材を開発している．また，携帯電話上のドリル (水野・松尾, 2006) や，PDA 上の手書き文字認識を利用した漢字ドリル (石塚ほか, 2004) など，提供形態も多彩になっている．標準化の視点では，QTI (Question & Test Interoperability) と呼ばれるクイズやテスト作成の標準規格に準拠したテストシステムが開発されており (仲林ほか, 2005)，ドリルについても今後対応を検討していく必要がある．

3.2.3 アイテムの形式

ドリルの先行研究では，アイテムの形式を 2 種類特定した．一方は，個々のアイテムに対して，問題と正答（刺激と反応）が 1 対ずつ登録され，各アイテムについて共通の問題文を設定し，選択肢には他のアイテムの正答を利用できる形式である．例えば，「次の地図記号は何を表すか」という共通問題文に対して，郵便局の地図記号を提示し，選択肢は，郵便局，学校，官公庁などのように他の地図記号の問題の正答も混ぜて構成する．Alessi

and Trollip (2001) は、対連合のアイテム (Paired-associate item) と呼び、鈴木 (1998) もこの形式を想定しているなど、過去のドリル研究に多い。言語情報の場合は、同じドリル群に含まれる他のアイテムが当該のアイテムと十分に紛らわしいという前提で、他のアイテムの正答を当該アイテムの正答と混ぜて選択肢を構成することが可能である (鈴木, 1989a)。この形式では対の関係 (問題と正答) を入れ替えることもできる。これを共通問題文形式と名付けた。

もう一方は、個々のアイテムに対して、問題文と選択肢 (正解を含む) の両方を登録する形式である。これは一般的なテストのアイテムと同様であり、例えば歴史 (鎌倉時代) のドリルで、鎌倉幕府設立の年代や、鎌倉幕府を開いた人物を答えさせる場合は、選択肢をそれぞれに用意しないと、明らかに正解がわかってしまうことになる。近年の多くのドリルツールや LMS および QTI は、こちらの形式を想定している。これを個別問題文形式と名付けた。

3.2.4 本研究との関連

ID 研究において、これまで複数のドリル制御構造が提案されており、それらを実装したドリルシステムはいくつか開発されている。一方で、ドリル制御構造を組み込んだドリルを簡単に作成して提供できるようなドリルシエルは、研究例が少なく、現状において利用可能な状態で提供されていない。既存のドリルシエルは、個々に1つのドリル制御構造を実装しているが、提案されているドリル制御構造は複数あり、例えば手厚く練習させる制御は、手軽さを失うことになるなど、トレードオフが存在する。よって、複数のドリル制御構造を統合し、ドリルの学習内容や利用場面に応じてドリル制御構造を選択できる環境が求められる。統合には、提案されてきたドリル制御構造を整理する枠組みが必要であり、Salisbury (1988) である程度進められているが、きちんと整理されてはいない。

また、ドリル制御構造自体を学べる環境を有するシステムは、Drill Designer 2.1 (The University of IOWA, 1999) のデモを除き、筆者の知る範囲では存在していない。既存のドリルシステムは、ドリル制御構造がブラックボックスである場合が多い。Drill Designer 2.1 のデモにおいても、1種類のドリル制御構造のキューの状態を可視化しているにすぎず、ドリル制御構造を学習する環境としては弱い。複数のドリル制御構造に対応するとともに、可視化だけでなくドリル制御構造の基礎を学習する教材なども追加する必要がある。

アイテムの形式 (共通問題文形式や個別問題文形式) については、既存のドリルシエルは、この2つの形式のどちらか一方にしか対応していない状況である。作成者が想定するドリルはどちらの形式も可能性があるため、両方に対応することが必要である。

利便性の面では、近年は教育の配信に LMS を導入する事例が増えているため、ドリル



図 3.1 ドリルを構成する 3 種類のプール (Merrill and Salisbury, 1984, を一部改変)

についても, Hot Potatoes(Half Baked Software Inc., 2007) のように SCORM に対応していくことも必要である。

以上から, 本研究の特徴的な点は, ID に基づくドリル制御構造を組み込んだドリルを簡単に作成できるドリルシェルを開発したこと, 複数のドリル制御構造を整理して統合したこと, ドリル制御構造を学習できる機能を設けたこと, 共通問題文形式と個別問題文形式の両方に対応したこと等が挙げられる。

3.3 ドリル制御構造の整理

3.3.1 ドリル制御構造の種類

ドリルの制御に関する基本的な流れ(サイクル)は, アイテムの選択, アイテムの提示, 応答の取得, 正誤判定, フィードバックを繰り返し, 終了後に評価を表示する (Salisbury, 1988)。また, ドリルは図 3.1 に示すように, 未学習のアイテムを入れるアイテムプール (IP), 学習中のアイテムを入れるワーキングプール (WP), 学習済のアイテムを入れるレビュープール (RP) という 3 種類のプール (アイテムの記憶領域) で構成される (Merrill and Salisbury, 1984)。

最も単純な制御のドリルは, WP のみで構成され, 出題するすべてのアイテムを WP に格納し, そこから 1 項目ずつ順番に出題される。正解であっても不正解であっても, アイテムは WP から取り出され, WP が空になった時点で終了, あるいは最初に戻って繰り返しとなる (原始的なドリル)。RP が追加されると, 正解したアイテムが RP に入り, 不正解したアイテムが WP に残ることで, 不正解したアイテムを集中的に学習できるようになる (2 プール型)。IP が追加されると, アイテムはまず IP に入り, 人間の記憶容量に配慮し, 7 ± 2 項目ずつ WP にアイテムを移動して出題する (3 プール型)。また, RP に日付の概念を加え, ある日付に達したらアイテムを WP に戻すことで復習機会を増加させたり (復習機会増加の 3 プール型), 事前テストやリハーサルを加え, RP (日付が 1 日, 2 日, 7 日) を 3 つ用意したものが最も複雑な状態前進型となる。リハーサルとは心理学で用いられる反復の意味ではなく, 問題文と正答を同時に提示することである。状態前進型は, 事前テスト状態 (既知のアイテムを排除する), リハーサル状態 (正答を覚える), ドリル状態 (アイテムを出題する), 復習状態 (既習アイテムを再出題する) のよう

に徐々に状態を移行していくドリル制御構造である。

また，Alessi and Trollip (2001) の項目間隔変動型は，不正解のアイテムを，待ち行列の最後尾に戻さずに何項目か後で，徐々に間隔を空けながら再出題していくドリル制御構造である。復習機会増加の3プールの型と似ているが，WPの構成法をはじめとして制御が異なる部分が多い。もう1つ提案されている単語帳型は，2プールの型とほぼ同様である。

ドリル制御構造のフローチャートを付録Aに載せる。

3.3.2 ドリル制御構造のスイッチ化

ドリル制御構造および制御要素の種類を表1に示す。制御要素は，Salisbury (1988) や Merrill and Salisbury (1984) を参考に，ドリル制御構造の構成要素として，学習を効果的にするための工夫（練習を支援するための制御）を，細かく分けたものと定義する。また，パラメータとは，制御に影響を与える変更可能な値（数字とは限らない）のことであり，例えば合格基準などが挙げられる。

表3.1は，制御要素をスイッチとして捉え，そのオンオフで既存のドリル制御構造が実現するように整理した。スイッチという概念を用いることで，各ドリル制御構造の制御要素を明確にし，特徴や違いをはっきりさせることを目的とした。スイッチはドリル制御構造を統合するために独自に考案したものであり，これによってドリル作成者が各ドリル制御構造を理解しやすくなることを狙っている。なお，各ドリルの制御構造および制御要素については，既存の研究成果を利用しているため，独自に考案したものではないことを補足しておく。特にSalisbury (1988) が，ドリルに必要な最低限の制御に対して，段階的に制御要素を追加していく視点でドリル制御構造を紹介しており，それを参考にして構造が明確となるように整理し直したことになる。

また，スイッチの切り替えによってドリル制御構造のカスタマイズも可能になる。鈴木 (1998) が行ったように，複数のドリル制御構造を組み合わせることが可能であり，制御要素にはそれぞれに利点や適用すべき状況があるため（例えば，アイテムを小分けにすることは，アイテム数が多い場合に適用する等），作成者が状況に応じて変更できることが望ましい。カスタマイズすることで，様々なドリルの制御を試行できるため，ドリル制御構造の学習を促進できる可能性もある。

スイッチ化によって，厳密にはいくつかのドリル制御構造を多少変更した。例えば，状態前進型は，WPに入っているアイテムが出題されるごとに，RPやIPを参照してWPへの補充処理を行うが，これは復習機会増加の3プールの型とはかなり異なるため，WP内の項目が無くなるまでは補充処理を行わないこととした。また，事前テストの処理については，Merrill and Salisbury (1984) を参考に，新しいアイテムがWPに入る時に事前テストを行い，正解のアイテムを除去し，WPが必要数になるまで，出題を繰り返す

表 3.1 ドリル制御構造と制御要素

制御要素	説明/パラメータ	ドリル制御構造					
		A	B	C	D	E	F
① アイテムを小さなまとまりに分けて出題する	短期記憶の容量制限への配慮 (Salisbury, 1988) とアイテム間の干渉を減らす (Merrill and Salisbury, 1984) ため、一度に出題するアイテム数は 7 ± 2 個とする。すべてのアイテムを一度にまとめて練習させようとせず、少しずつ無理なく練習させる。パラメータは WP の容量 (一度に出題するアイテム数) であり、デフォルトは 7 である (本システムは 5~9 の範囲で自動的に設定する。詳細は 3.6.2 を参照)。	×	×				×
② 出題順をシャッフルする	連続学習効果を取り除き、何度も繰り返して練習できるようにアイテムをランダムに並べ替える (Salisbury, 1988)。連続学習効果は、アイテムの順番を回答の手がかりとして利用することで、順番が変わると正答できなくなる場合を指す。シャッフルは、WP が空になってアイテムが補充された時か、IP がある場合は出題前の最初の時に行う。	×					
③ 不正解したアイテムを集中的に出題する	不正解したアイテムを WP に戻す (残す) ことで、学習済みのアイテムとは区別し、理解していないアイテムを集中的に練習できるようにする (Salisbury, 1988)。パラメータは、WP の最後尾に戻す (デフォルト) か、項目間隔変動かである。項目間隔変動は、学習者の短期記憶にアイテムの情報が残っているうちに、すぐに再出題することで、長期記憶への格納を促進する (Alessi and Trollip, 2001)。パラメータとして再出題間隔 (3, 7, 15... 等) も設定できる。	×					2
④ 合格したアイテムを一定期間後に復習として出題する	合格したアイテムを RP に入れ、一定期間後に復習のために再出題する。保持を高めるために、学習済みのアイテムを復習させることが推奨されている (Merrill and Salisbury, 1984)。RP は 3 つまで用意できる (出題して合格するごとに次の RP に移っていく)。各 RP にパラメータとして日付を設定し、その日付を越えた場合に、RP のアイテムが WP に移動して出題される。WP へ移動後に不正解した場合には、その時の復習状態を維持するか、ドリル状態に移行するかを設定できる。デフォルトは RP が 1 つで日付は 1 日である。	×	×	×		1	×
⑤ 事前テストを行う	最初に事前テストとしてアイテムを出題する。学習者がすでに知っているアイテムを排除し、練習の必要があるアイテムだけを選定する。事前テストに不正解の場合は、ドリル状態からリハーサル状態に進む。多くのアイテムを学習者がすでに知っている場合に、事前テストは特に有用とされる (Merrill and Salisbury, 1984)。事前テストは、WP に新しく未出題のアイテムが補充されるタイミングで行う。	×	×	×	×		×
⑥ リハーサルを行う	練習 (ドリル状態) の直前にリハーサル (問題文と正答を同時に表示) を行う。学習者が練習ではなく評価と意識してしまい、不正解を敬遠しないように、練習の初期に正解やヒントを求めに応じて与えることも考えられる (鈴木, 1989a)。学習者の知らないアイテムが多そうな場合は不正解ばかりになるので、まずリハーサルを行ってもよい。	×	×	×	×		×

A: 原始的なドリル B: 2プール型ドリル (Salisbury, 1988) C: 3プール型ドリル (Salisbury, 1988)
 D: 復習機会増加の3プール型ドリル (Salisbury, 1988) E: 状態前進型ドリル (Salisbury, 1988)
 F: 項目間隔変動型ドリル (Alessi and Trollip, 2001)
 IP: アイテムプール WP: ワーキングプール RP: レビュープール
 , ×: 列 (A~F) のドリル制御構造が, 行 (①~⑥) の制御要素を含む場合は, 含まない場合は×としている。
 1: パラメータは, 第1レビュープールが1日, 第2が3日, 第3が7日とし, 不正解時は復習状態維持となる。
 2: パラメータは, 項目間隔変動である。

こととした。

3.3.3 合格基準の取り扱い

合格基準は、ドリルの終了条件 (ドリルのサイクルを抜け出すための条件) と換言できる。ドリル制御構造には、合格基準の概念が入っているものと、学習者が停止するまで無限に出題が繰り返されるものがあり、この違いによりドリル制御構造も変わってくる。Alessi and Trollip (2001) によれば、合格基準は、個々のアイテムに対する基準と全体に対する基準という 2 種類がある。個々のアイテムの場合は、正解が連続で続いた回数、全

体の場合は何割が合格したかで、学習目標に到達したかを判断する。不正解時にはそれまでの正解回数をリセットし、不正解アイテムを集中的に出題すべきであるとしている。例えば項目間隔変動型は、不正解時に合格基準を1つ増やし、その数だけ不正解したアイテムを、間隔を増加させながらWPに配置する。また、項目間隔変動型はWPへアイテムを最初に読み込む際に、合格基準の数だけ複製して構成する。つまり、全5項目で合格基準が2回正解であれば10項目がWPに入ることになり、正解のアイテムを除去し、アイテムがWPから無くなった時点で合格基準を満たす。

Salisbury (1988) では、合格基準が状態前進型を除いて全く検討されておらず、ドリルを停止するのは学習者次第となっていた。状態前進型についてのみ、各状態のすべてに個別の合格基準が存在し、最終的にはアイテムがすべて除去されれば終了となる。本研究では合格基準を実装するために、状態前進型でなくとも状態という概念を利用することにし、各状態に合格基準を設定した。通常のWPのみで構成されるドリルは、ドリル状態とみなす。また、2プール型や3プール型のRPは、正解アイテムの復習ではなく不正解アイテムの再出題という意味合いが強いため、復習状態とはせずにWPとあわせてドリル状態とみなし、合格基準を設定することとした。また、日付のRPは、復習状態とみなし、あくまでもドリル状態で合格したことを前提に、復習アイテムとして出題するという位置づけにした。

また、合格基準を設定するということは、正解と合格という2つの概念ができることになり、合格基準が連続2回以上正解であれば、一度正解しても合格とはならない。よって、設定の内容に関わらず、合格基準の回数分だけは、WPに戻すように内部的に処理(保証)するようにした。一方で、ドリル状態や復習状態などのすべての状態に対して個別に作成者が合格基準を設定することは、わかりにくさを助長する可能性があるので、作成者が設定するドリル状態の合格基準と同じ基準を適用することとした。

3.3.4 その他のドリル制御構造の適用可能性

本研究では、ドリル制御構造の統合を目指して開発しているため、既存の他のドリル制御構造の適用可能性も展望しておく。鈴木(1998)では、項目間隔変動型と状態前進型を組み合わせた方式を提案していたが、リハーサル状態と項目間隔変動型のパラメータを設定することで、近い動きを実現できる。水野(2000)のLow-First方式は、項目間隔変動型と類似するドリル制御構造であるが、1項目ごとではなく、すべてのアイテムの正誤の結果と合格基準を踏まえて、次の出題順を決定するという点が異なっており、より複雑な制御を行っていると言える。不正解のアイテムを集中的に出題する制御要素(表3.1)にLow-First方式というパラメータを追加することが必要になる。

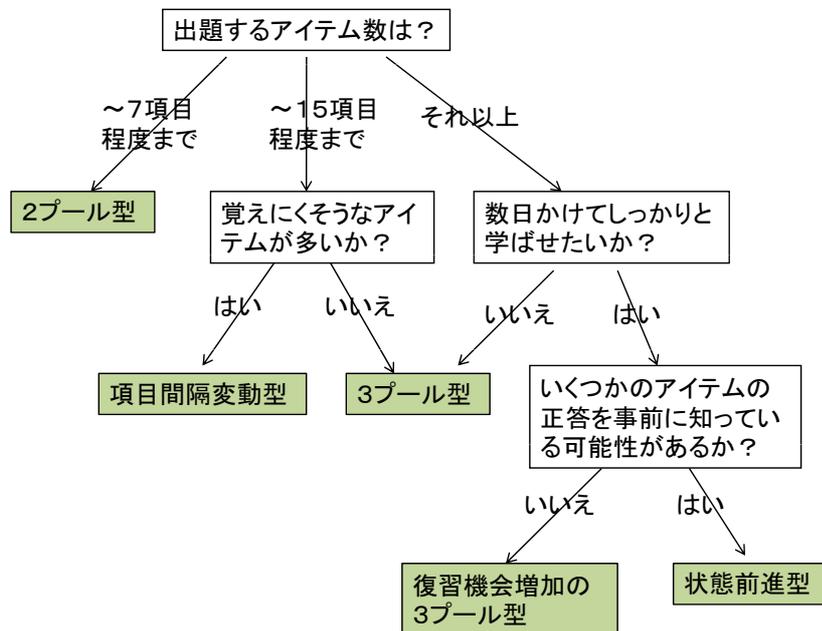


図 3.2 ドリル制御構造の選択チャート

3.3.5 ドリル制御構造の選択

既存のドリル制御構造に関して，作成者の選択を支援するために，選択チャートを作成した（図 3.2）．これまでにドリル制御構造は複数提案されてきたが，状況に応じてどれを選択したらよいかについては，ほとんど整理されてこなかった．Salisbury (1988) は，ドリル制御構造の選択について，学習課題の種類，学習スキル，プログラミングコストの 3 つを挙げている．しかしながら，学習課題の種類については本システムが言語情報のみであり，プログラミングコストについても，本システムにあらかじめドリル制御構造を実装しているため，特に留意する必要はない．学習スキル（復習等を自分なりに工夫してできるか）は，学習者次第の要素が強いため，今回は含めないこととした．

他には，例えばドリルの出題数に基づく選択が考えられる．システム側で把握できるため，自動的な選択が可能となる利点もある．短期記憶の容量の範囲内（7 項目以内）であるなら，アイテムを小分けに出題する制御要素は不要であり，2 プール型で十分となる．一方で，出題数が多い場合は，一度に覚えるアイテムが多すぎるために，3 プール型が必要となる．また，項目間隔変動型についても，2 プール型ほどではないが出題数が多い場合には不適當となる．厳密な規定はないが，一度に学習するには 15 項目程度が限界である（鈴木, 1989a）という指摘を参考にし，それ以上の場合は，3 プール型か復習機会増加の 3 プール型，状態前進型を候補とした．8～15 項目程度までは，項目間隔変動型と

3 プール型が考えられるが，項目間隔変動型は，不正解のアイテムを集中的に学習できるドリル制御構造であり，覚えにくそうな（紛らわしい）アイテムが多い場合に優れていると考え，その点で分岐させた．

学習者が学習を行う頻度によっても選択すべきドリル制御構造が変わってくる．1回しか学習されないようなドリルであれば，復習機会増加の3プール型や状態前進型は意味をなさない．むしろその場で完結した方が，学習者にとっても達成感が増す．よって一定期間利用する場合は復習機会増加の3プール型や状態前進型とし，そうでない場合は3プール型とした．また，既習状況によっても変わるとも考えられる．いくつかのアイテムの正答を事前に学習者が知っている可能性があるなら，「事前テストを行う」の制御要素を提供することが役に立つ (Merrill and Salisbury, 1984) ので，状態前進型が良いと考えられる．よって，学習者が既に知っているアイテムがある場合には状態前進型とし，そうでない場合は復習機会増加の3プール型とした．

3.4 システム概要

本システムはドリルシェルであるため，ドリルや ID についてよく知らない作成者であっても，アイテムを登録しさえすれば，ID に基づく効果的なドリルを作成できる．また，本システムはドリルを簡単に作成・提供できるだけでなく，ドリル制御構造自体を学ぶこともできる．本システムをの概念を図 3.3 に示す．本システムは，学習環境とオーサリング環境から構成される．システムを利用するにはアカウントを作成してログインが必要となる．システムのログイン後のメニュー画面を図 3.4 に示す．

提供している機能は，編集，実行，ダウンロード，学習の4種類である．作成者が本システムで作成したドリルを実際の学習に利用するためには，自分のローカル環境にダウンロードして設置する必要がある．よって，本システムの対象ユーザは，ドリルの作成者（ドリル制御構造の学習も含む）と，作成者がダウンロード後に設置したドリルを利用する学習者が想定される．

3.5 学習環境

ドリルの学習環境は，学習者が実際にドリルを用いて学習を進めるための環境である．あらかじめ設定されたドリル制御構造やパラメータに基づいて，1問ずつアイテムを出題し，学習者が回答を入力し，フィードバックとして正解か不正解，不正解の場合は正答，設定されていれば解説を表示していく．ドリルの学習環境の画面を図 3.5 に示す．ドリルは合格基準に達するまで繰り返される．

また，ドリルの学習環境は SCORM1.2 対応で生成することもでき，LMS 上に載せる

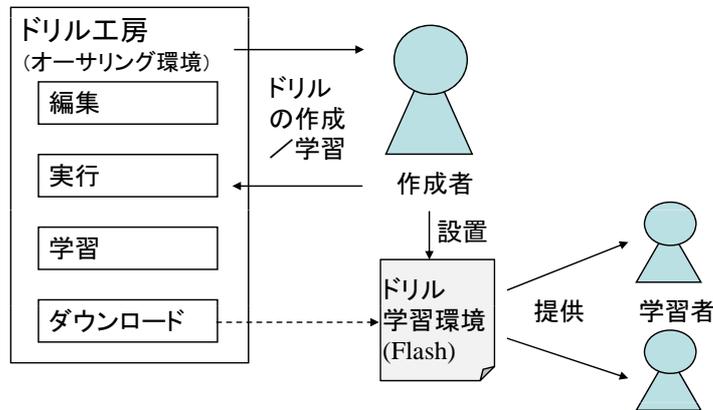


図 3.3 システム概念図



図 3.4 メニュー画面

ことで、利用状態の記録や、ドリルの中断と再開ができる。アイテム状態の保存には、`cmi.suspend_data` の領域を利用した。学習環境のファイル構成は、実行プログラムと設定ファイル (XML でドリル制御構造やパラメータ、各アイテムの情報を記述) が含まれる。



図 3.5 ドリル学習環境

3.6 オーサリング環境

3.6.1 編集機能

編集機能は、ドリルの新規作成や編集、アイテムの追加や削除を行うための機能である。ドリルは、必要最小限の情報を登録するだけで動くように配慮した。多くのパラメータはデフォルトで設定されており、作成者が要求しない限りは表示されない。編集機能は、ドリル作成とアイテムの編集という2つの下位機能から構成されている。

ドリル作成においては、ドリルのタイトルを入力し、ドリルの形式として「共通問題文形式」か「個別問題文形式」を選択し、選択肢数を入力する(図 3.6)。問題文形式を選択させるために、2種類の形式の違いに関する説明を表示するようにした。

アイテムの編集において、共通問題文形式の場合は、まず共通の問題文を入力し、その後で問題と正答のペアを入力していくことになる(図 3.7)。ここでは10項目分を一度に入力することが可能である。個別問題文形式の場合は、問題文と選択肢の入力が必要となる(図 3.8)。両者とも、回答へのフィードバック時に表示される解説を追加することもできる。利用可能なリソースの形式は、文字と画像と音声である。また、共通問題文形式は問題と正答を入れ替えることが可能なので、入れ替えた時に使う問題文を入力する欄も用意している。

ドリルのデザインについても、文字色や背景色を変更でき、作成後に公開する場所等に合った配色に設定できる。また、作成時のアドバイスとして、1つのドリルに含まれるアイテム群は、1つの学習目標に対応させることが原則(鈴木, 1989a)であること、ドリル



図 3.6 ドリル作成画面

を飽きさせないために 1 回 15 分程度に押さえる方がよい (Alessi and Trollip, 2001) といった説明を提示するようにした。

3.6.2 実行機能

実行機能は、作成したドリルを実行（確認）するための機能である（図 3.9）。実際にドリルで学ぶというよりはプレビュー（動作チェック）としての利用を想定している。

ドリルを実行すると、システムはアイテムを出題し、学習者が回答を入力し、フィードバックとして正解か不正解、不正解の場合は正答、設定されていれば解説を表示する。ドリルの実行前にドリル制御構造やパラメータを設定可能であるが、それを行わなくとも、デフォルトの設定ですぐに実行できる。

ドリル制御構造の選択においては、登録しているドリル制御構造の一覧から選ぶことになるが（図 3.10 左上）、出題するアイテム数はシステム側で判断可能なため、7 項目までは 2 プール型、8 から 15 項目までは項目間隔変動型、それ以上は 3 プール型をデフォルトとして自動設定することにした。これにより、作成者がデフォルト設定のまま利用したとしても、作成したドリルにある程度適したドリル制御構造が選択されることになる。また、3 プール型等でアイテムを小さなまとまりに分けて出題する際に、例えば 8 項目の時にまとまりを 7 項目に設定すると 7 と 1 でバランスが悪い。よって、バランスをとるため



図 3.7 アイテムの編集画面（共通問題文形式）

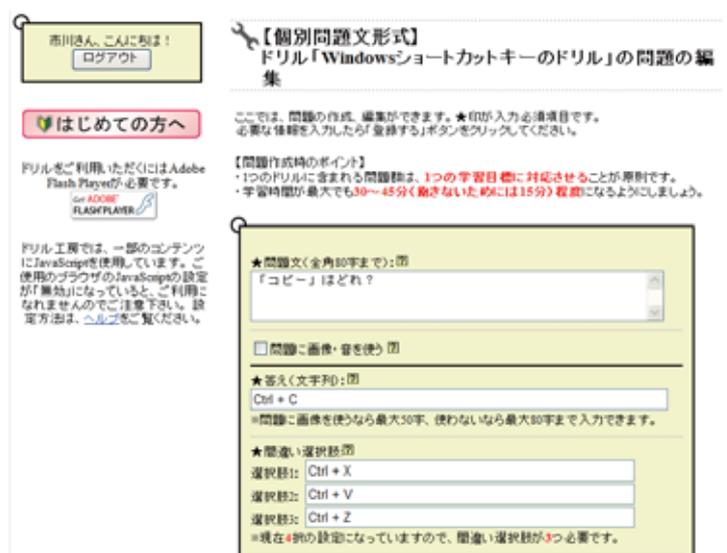


図 3.8 アイテムの編集画面（個別問題文形式）

に出題数を9から5の順で割り切れるものを探し、そうであればその数を、割り切れない場合は、余りの最も少ない数をデフォルトとした。

また、ドリル制御構造のカスタマイズ用に表1で示した制御要素のスイッチを用意した(図3.10右下)。各制御要素に固有なパラメータ設定の表示は、スイッチの切り替えに連動させている。それ以外のパラメータとして、合格基準、出題形式(再認の多肢選択式か、再生の記述式)、利用するアイテム数、評価結果の提示内容なども併せて設定できる(図3.10右上)。



図 3.9 ドリル実行画面



図 3.10 ドリル制御構造の選択とスイッチ

実装の段階においては、作成者のニーズを鑑み、ドリル制御構造に診断（ここでは正誤判定のみ）という制御構造を追加した。仲林ほか（2005）は、テストの実行形式の分類の中で、学習目的のテストとしてドリルを挙げ、1項目ごとに回答してフィードバックを行う形式ということで差別化している。診断を実装するために、回答直後に（即時）フィードバックを行うというスイッチを新設し、それをオフとすることで対応した。なお、即時



図 3.11 ダウンロード画面

フィードバックは本来ドリルに必須で備えるべき制御であるため、表 3.1 には含めていない。

3.6.3 ダウンロード機能

ダウンロード機能は、作成したドリルをローカル環境に持ち帰り、ネットワークの有り無しに関わらず、作成者が提供する e ラーニングや、もしくは自学自習に利用するための機能である。作成者はダウンロード前に、過去にドリルを実行した記録の一覧から設定を選択するか、ドリル名やドリル制御構造とパラメータを新しく設定する(図 3.11)。ダウンロードしたドリル(学習環境)は、その時に設定したドリル制御構造とパラメータのみで動き、実行機能のようにいつでも変更できるようなにはなっていない。また、ダウンロードするドリルは SCORM1.2 への対応(学習履歴の保存)を選択できる。

3.6.4 学習機能

学習機能は、ドリルの内部プロセスの可視化(シミュレーション)と、ドリル制御構造に関する教材や説明の提供を行うための機能である。可視化モードには、ドリル実行時(実行機能)からいつでも切り替えが可能であり、アイテムの順番や状態の一覧を表示し、回答するごとに、状態が更新される。可視化モードは4つの画面から構成され(図 3.12)、通常のドリル実行画面が左上に表示される。左下にはアイテムのリスト(表形式)が表示され、行はアイテムの順番を、列はアイテムの状態を示す。ドリルのパラメータは右上に表示され、いつでも変更が可能である。パラメータを変更すると、すぐにアイテムの状態



図 3.12 可視化モード (画面は 3 プール型の場合である)

も更新される。ドリルの制御ログは右下に表示され、制御のプロセスを戻ったり、進めたりするボタンを利用して、ドリルの動きを振り返ることができる。

教材や説明については、一通りのドリル制御構造は網羅しつつも、最低限 2 プール型と 3 プール型は理解してもらうという観点で作成した。Salisbury (1988) を参考にし、原始的なドリルから 3 プール型までを、シャッフルや各プールの追加という視点から段階的に解説するアニメーション教材とした (図 3.13)。また、各ドリル制御構造の説明時に、そのサンプルとして可視化モードを直接参照できるようにし、アイテムの動きを確認できるようにした。その他のより複雑なドリル制御構造は、それぞれの解説を用意し、さらにドリル制御構造と制御要素 (表 1) を、平易な言葉に書き換えて提示した。



図 3.13 ドリル制御構造の学習教材

3.6.5 開発環境

本システムは、Apache、PHP、JavaScript、HTML、MySQL、Adobe Flash によって構築した。Flash は、ダウンロード時の実行ファイルや教材に利用している。ドリル制御構造を拡張しやすいように、制御要素のスイッチや、それ以外の制御に必要となる処理を細かくクラス化した。なお、本システムの動作について、現状では Internet Explorer 6.0 でのみ確認を行っている。

3.7 評価

3.7.1 評価概要

評価は、本システムにおけるドリル作成のユーザビリティとドリル制御構造の学習について行った。評価方法は、システムを Web 上に公開し、予備実験の後に、ソフトウェア系の大学生 1~3 年生の計 12 名に対して数名ずつ利用してもらった。手順としては、作業課題の説明書を渡し、システムを利用してもらい、アンケートに記入してもらった。また、評価集計後に 5 名に対して結果をもとにしたインタビューを行った。評価は、単純なドリル作成、ドリル制御構造の学習、ドリル制御構造の適用という順番で実施した。

表 3.2 単純なドリル作成の平均時間（分，括弧内は SD）

N: A:5 B:5	共通問題文形式	個別問題文形式
課題 1	14.40 (2.94) [A 群]	13.00 (2.61) [B 群]
課題 2	11.20 (2.71) [B 群]	10.00 (1.41) [A 群]
計	12.80 (3.25)	11.50 (2.56)

3.7.2 単純なドリル作成に関して

本システムは，デフォルト設定のままでも ID に基づくドリル制御構造の中からある程度適したものを自動で組み込む．よって，まずは最も単純な手順で（ドリル制御構造やパラメータの設定を一切行わずにデフォルト設定のみを用いて），簡単にドリルを作成することができるかを確認する目的で評価を行った．また，アイテム登録の際に必要な 2 種類の問題文形式を区別できるかもあわせて確認した．

手順は，作業課題として作成するドリルの内容（共通問題文形式および個別問題文形式の 2 種類のアイテムのリスト）を提示し，2 種類のドリルを順番に作成してもらった．ドリル制御構造とパラメータはすべてデフォルトを利用し，実行画面で確認後，ドリルをダウンロードして実行するところまで（ドリル実行画面を見てもらう程度で学習はしない）を行った．アイテム数は両形式とも 7 項目で合計 200 字程度の文字数とし，手入力してもらった．作業課題の順番は，協力者を 2 群に分け，片方の群は作成する問題文形式を逆順とした．なお，機器等のトラブルで中断した 1 名と，作成をせずに他の機能を見ていたために共通問題文形式が 28 分と長時間になった 1 名については除外した．

結果を表 3.2 に示す．全員が適切に共通問題文形式と個別問題文形式を選択し，ドリルの作成を完了できた．平均作成時間は共通問題文形式が 12.80 分，個別問題文形式が 11.50 分であった．アンケートの結果からは，簡単にドリルを作成できたかの項目について 5 段階（5 が最も良い）で平均 4.58（SD=0.79）であったことや，本システムの良かった点（自由記述）として簡単に作成できたと回答した協力者が半数の 6 名にものぼったことから，簡単にドリルを作成できたと判断できる．

しかし，2 種類の問題文形式を理解できたかの項目について平均 4.58（SD=0.79）と高評価であったものの，インタビューでは 5 名全員が最初は迷ったと述べており，形式を選択せずにデフォルトで進み，おかしいということに気づいてから説明を読んだのが 2 名，最初から説明を読んで理解してから作成したのが 3 名であった．説明を見ればわかるようであったので，例えば，問題文形式はデフォルトを設定せずに，必ず選択させるという行為を行わせる必要がある．

3.7.3 ドリル制御構造の学習に関して

ドリル制御構造の学習については、システムの学習機能を利用して、ドリル制御構造が理解できたかどうかを確かめる目的で評価を行った。

手順は、学習機能の部分を教材として見てもらい、事前事後テストに答えてもらった。テストは合計 10 問の記述式とした。1 問 1 点としたが、記述式であるので、内容が多少あっていれば 0.5 点と計算した。問題の内訳は、シャッフルと IP と RP の利点の説明 (3 問)、2 および 3 プール型のアイテムの動きを示す (2 問)、各ドリル制御構造の説明 (5 問) であった。3 問分が 3 プール型よりも複雑なドリル制御構造の説明を課しており、さらに 1 問分は間違っても良いこととし、6 点以上を合格とした。事前と事後テストは、ほぼ同じ問題とし、アイテムの動きの 2 問のみ、アイテムの並びを変えた。

結果として、学習時間は平均 15.00 分 (SD=5.29) で、最低は 7 分、最高は 23 分であった。事前事後テストの結果を表 3.3 に示す。ドリルの内容をまったく知らない協力者であったので、事前テストは平均が 0 点に近かったのに対し (ただしシャッフルの利点だけは 4 割程度の正解率があった)、事後テストでは合格基準ぎりぎりの平均 6.12 点まで引き上げることができ、 t 検定を行った結果、有意差も確認された。よって、本システムを使うことで、ドリル制御構造を学習できたと言える。

一方で、合格基準は超えたが、あまり点数が高くなかった理由としては、3 プール型の説明や IP の利点の説明の合格率が 6 割前後であったことが挙げられる。アンケートからは、ドリル制御構造の多さや情報の過密性への指摘があり、協力者を混乱させてしまったようであった。2 プール型と 3 プール型を学ぶアニメーション教材は好評であったことから、最初からすべてを表示せずに、作成者が自分のレベルに応じて段階的に表示を制御できるような仕組みや、一カ所に学習教材をまとめるだけでなく、システムで必要そうな部分に随時説明を入れるなども考えられる。

また、アイテムの動きを示す 2 問が正解率 8 割以上と点数は良かったが、協力者がソフトウェア系ということが影響しているとも考えられる。インタビューでは教材のアニメーションが良かったという意見で全員一致しているが、可視化モードも併用して理解を深められたと答えたのは 1 名に留まった。可視化モードが理解できたかのアンケート項目では

表 3.3 事前事後テストの結果 (平均点, 10 点満点, N:12)

事前テスト	事後テスト
0.42 (0.45)	6.12 (1.85)

括弧内は SD, $t(11)=10.68$, $p<0.01$

平均 3.50 (SD=0.80) であり、比較的低い結果となった。インタビューからは、可視化モードはわかりにくかった、説明が無くてよく分からなかったという意見があったので、使い方の説明を充実させるなど、今後改善を行う必要がある。

3.7.4 ドリル制御構造の適用に関して

ドリル制御構造の学習に関する評価の延長線上で、ドリル制御構造の教材で学んだことを、実際に自分が作成するドリルに対して適用できるかを確かめる目的で評価を行った。手順は、他の機能も含めてシステム全体を眺めてもらい、自由な内容でドリルを1つ作成し、最適と思われるドリル制御構造を設定してもらった。この課題については、今回の協力者には難しいと考えたが、学習面の評価の補足的な意味で、今後の改善に向けての情報を得るために実施した。

結果として、登録アイテム数は平均 4.73 個 (SD=2.69) となり、作成できなかった協力者は1名で、ドリル制御構造の適用があまりわからなかったので作成を控えたということであった。既存のドリル制御構造を用いたのは9名、カスタマイズした者は2名であった。ドリル制御構造の選択理由としては、間違っただけのアイテムを何度も繰り返し学習してほしいという理由で2プール型を選択したり、アイテム数を多くこなせるという理由で3プール型を選択したり、間違えたら繰り返して覚えた方がよいという理由で項目間隔変動型とするなど、正当な理由をつけてドリル制御構造を選択していた。カスタマイズでは、3プール型に項目間隔変動型を追加したのが見られた。しかし、少数ではあるが、簡単なドリル制御構造で十分と判断して原始的なドリルを選択したり、アイテム数が少なく(7問であった)すぐに終わってしまうという理由で不正解時の制御を加えないなど、ドリル制御構造をあまり理解していないと思われる事例もあった。

インタビューからは、どんな問題にどのようにドリル制御構造を適用すればよいのかわからないといった指摘があり、ドリル制御構造とスイッチの関係をもう少しわかりやすく提示する必要がある。ただし、スイッチについては何度か使っていけば手を出すかもしれないという意見が示すように、システムを使い込むうちに使えるようになるとも考えられる。また、ドリル制御構造の選択チャートのボタンに気がつかなかったという協力者もいたので、選択チャートを目立つように表示することなどが必要である。

3.7.5 専門家への評価

ID やドリル制御構造を理解しており e ラーニングにも精通している専門家1名に、本システムを見せながら意見を聴取した。以下に頂いた意見を示す。意見聴取の結果からは、本システムは妥当であったことが伺えた。

- 学習者の自学自習用に活用できそうである。
- ドリル制御構造 (ID 理論) は役に立つはずであるが、なぜか既存のシステムに入っていない。
- e ラーニングで教材を作っている立場の人に便利そうである。
- カスタマイズ機能は、目先を変える (飽きさせない) 工夫としても使えるかもしれない。
- 可視化モードは ID の上級者向けである。
- ドリル数やアイテム数の増加に対して機能的な限界があるかもしれない。

3.8 知的技能への対応

本ドリルシエルは、言語情報への対応のみを検討してきたが、本章の最後に知的技能への対応も検討しておく。

3.8.1 知的技能とは

知的技能はガニエの学習成果の5分類の1つであり (ガニエ他, 2007), 何かを実行する能力 (手続き的知識) を意味する。対照的なものとして言語情報があり、覚えたことをそのまま再生することが要求される (宣言的知識)。言語情報は主に暗記となるが、知的技能は、未知の問題に学んだ内容を適用できることが要求される。知的技能は、その技能を身につけるための、より基礎的な下位目標の達成が前提条件となるような階層構造で整理できるとされ、学習課題分析には階層分析手法が用いられる。

知的技能には下位分類として、2つのものが異なるかどうかを判断できるようになる「弁別」、具体物の物理的な特性や属性を同定できるようになる「具体的概念」、物理的な属性ではなく定義で分類できるようになる「定義された概念」、規則を適用できるようになる「ルール」、複雑なルールの生成と使用ができるようになる「問題解決 (高次のルール)」がある。これらは、問題解決の前提条件がルール、ルールの前提条件が定義された概念というように、それぞれが次の技能習得の前提条件となっている。

また、言語情報と知的技能には、学習成果に本質的な差異があることから、学習を達成するための条件も異なり、教授方略 (ここでは特に練習方略) も異なる。さらに知的技能の下位分類それぞれでも学習の条件が異なることから、その効果的な練習方略も異なると考えられる。

3.8.2 既存の知的技能のドリル

知的技能のドリルについては、鈴木 (1989a) が知的技能の効果的なドリルのメカニズムについて、言語情報と対比しながら説明している。また、これまでの先行研究では、階層構造に着目した目標間の移動、知的技能の下位分類に関する方略、未知の問題の出題という3つの観点が主に取り上げられてきた。

階層構造への対応

知的技能は階層構造となるため、ある目標が達成できなければ、1つ下の階層に戻ることや、目標を達成後は上位の階層に進むなど、学習の系列が明確である。こういった階層構造に着目し、ドリルの結果によって、上位・下位の階層(目標)に移動するような仕組みが提案されてきた。Salisbury (1988) は知的技能の階層性に着目したドリルの制御方法として下位ドリル群を提案した。許・繁樹 (1990) は、中学2年生の式の計算を対象領域として、階層構造や項目反応理論を利用して、どの難易度の問題を出題するかを決定するドリル型 CAI を試作した。また、ドリルではないが、高橋ほか (2007) が開発している課題分析図に基づく e ラーニングシステムでは、階層分析図を学習項目の選択に利用しており、各項目についてドリルを配置していくことでも、近い内容を実現することができると考えられる。

下位分類に関する方略

知的技能の下位分類(主に概念)に関するドリルの制御方法やシステムが提案されてきた。Salisbury (1988) は、適応型概念学習ドリルとして、ある項目がどの概念の事例かを正しく答えられた場合、次の項目は他の概念の事例からランダムに選択されるが、間違った場合、次の項目は間違って答えた概念の例が選択されるという制御方法を紹介している。山本ほか (2004) は、次元分けとサブドリル構造(本稿の下位ドリル群と同義)を用いた、知的技能のための概念学習ドリルシエルの開発を行っている。次元分けの4つの値と階層構造をもとに、出題を制御している。また、藤原ほか (2005) は、情報処理入門科目におけるオンライン個別学習システムのドリル機能として、アルゴリズムに状態前進型ドリルを採用している。知的技能への適用とは特に明示されていないが、状態前進型ドリルも知的技能に応用できるとされている(鈴木, 1989a)。

未知の問題の出題

知的技能は、言語情報のような暗記ではないため、常に未知の問題を出題する必要がある。つまり、同じ練習問題を2回出題して2度目に正しく答えられたとしても、知的技能

が習得されたのか、丸暗記したために答えられたのかを見分けることが難しいからである(鈴木, 1989a)。このためには、十分な問題数を準備することが必要である。

また、複数の問題を自動生成するという方法もある。問題文の一部を変数として設定することで、ランダムな生成が可能とするようなシステムが提案されており、例えば数値計算に代表される知的技能のルールの適用などにおいては、この方法も有効であると考えられる。不破ほか(2003)のドリル型CAIシステムにおいては、問題文を乱数により毎回変えて提示するようになっている。さらにプログラム作成といった問題解決のレベルの出題や自動判定も行われており、知的技能のドリルにとって参考になる部分が多いと考えられる。

3.8.3 知的技能への対応

上記のように知的技能に対応したシステムは、開発者が意図したかどうかに関わらず、これまでもいくつか開発されてきたが、知的技能の特徴をきちんと捉え、練習の方略を検討している事例は非常に少ない。また、現状で一般に利用可能な、知的技能の方略を備えたドリルシエルは存在していない。そこで本研究は、知的技能への対応の足がかりとして、Salisbury(1988)で提案されている下位ドリル群の実装を試みた。

下位ドリル群

ドリル工房の知的技能への対応として、Salisbury(1988)で提案されている下位ドリル群(Subdrill Grouping)の実装を試みた。下位ドリル群は、関連する知的技能をドリル群として複数用意し、到達目標のドリルでつまづいた場合に、つまづきの原因となっている当該の下位目標のドリル群に自動的に移行する方法である。ある学習目標にむかって練習のでき具合に応じて徐々に難しいドリル群へと移動していく場合にも応用可能である(鈴木, 1989a)。

知的技能で提案されている制御方法の中で最初に下位ドリル群を取り上げた理由としては、知的技能にとって特徴的な階層構造を利用していたことに加え、現状のシステムとの親和性も考慮した。Salisbury(1988)は、1つのドリル内に様々な下位目標の内容を統合するよりも、下位ドリル群のように複数のドリルを組み合わせる手法の方が簡単であるとしており、現状のシステムの枠組みにドリル間の制御を追加することで、比較的容易に実装できると考えられたからである。

下位ドリル群の実装

下位ドリル群の手法を用いるときは、ドリル間の上位下位の関係を表示することが重要である(Salisbury, 1988)。そこで、階層分析図をGUIのエディタにより作成することに

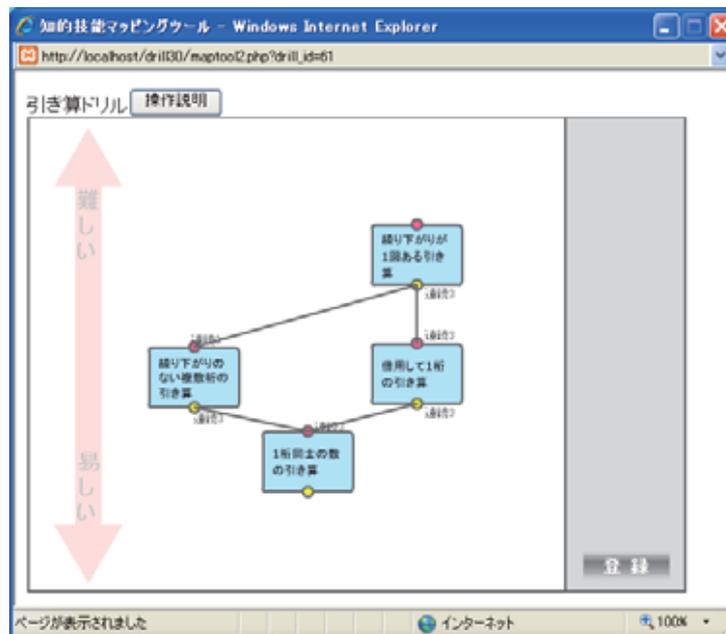


図 3.14 階層構造エディタ

した。ドリルの作成画面において、知的技能のドリル作成を選択すると、階層構造エディタが表示される（図 3.14）。エディタでは、各目標（1つの箱）を作成し、それに対応したドリルを関連づける。さらに各目標の上下を線でつなぐことにより、階層構造を表現する。1つのドリルを表す箱は自由に移動可能であり、箱の下部から下位へ、箱の上部から上位へ線を延ばしてリンクを貼ることになる。関連づけるドリルは既存のものを選択するか、新規に作成することになる。

ドリルを関連づけた後で、ドリル間を自動的に移動する条件として合格基準と不合格基準（便宜上このように呼ぶ）を設定する。合格基準には、異なる種類の問題に適用できた回数（例えば 5 回連続）や、正解と誤答の比率によって行う。不合格基準は逆に間違った回数や、正解と誤答の比率となる。ドリルの移動については、下位目標が 2 つ以上あった場合には、現状では特に履歴は参照せずに、任意の 1 つを選択するが、その練習が合格だった場合は、上位に戻るのではなく、同レベルにある他の目標に移動する。

下位ドリル群を実装する上でのその他の変更

ドリルの編集機能では、最初に言語情報が知的技能かの選択を行うようにした。各ドリル内の制御方法は、新たに知的技能専用のものは設けずに、言語情報のものをそのまま用いることにした。本来であれば、例えば概念用のドリルには専用の制御方法を用いた方が効果的と考えられるが、今回はそこまでは含めないことにした。

フィードバックについては、言語情報のようにただ正解を提示するだけでなく、概念の

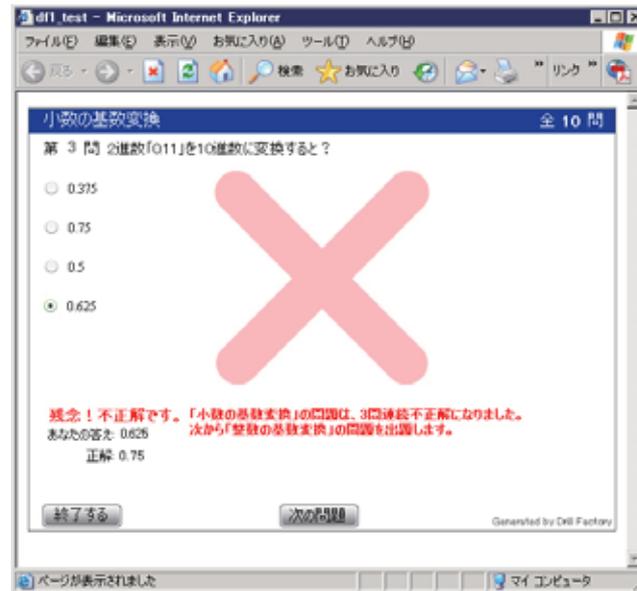


図 3.15 フィードバック

ドリルであれば定義や特徴を，ルールのドリルであれば，ルールやその正しい応用例のように，より意味のある形で行う必要がある (Salisbury, 1988)．よって，各問題の解説入力欄を広げるだけでなく，共通フィードバックの欄を設け，ルールなどを記述できるようにした．さらにドリルでの学習の結果として上位や下位に移動することになった場合は，その旨をフィードバック時に通知するようにした (図 3.15)．

また，ドリル実行時においても，階層アイコンを常に表示し，いつでも階層図を参照できるようにした．可視化モードでも同様に，現在どの下位ドリルにいるのかの位置を表示するようにした．

3.9 まとめ

本章では，eラーニングにおけるドリル作成の支援を目的として，IDに基づく複数のドリル制御構造を適用したドリルを作成できる統合型ドリルシェル「ドリル工房」を開発した．ドリルの編集，実行，ダウンロード，学習という4つの機能を実装し，特にドリルを簡単に作成できること，ドリル制御構造を学習できることを目指した．既存のドリル制御構造を制御要素のスイッチという概念を用いて整理し，ドリル制御構造のカスタマイズも可能にした．ドリル制御構造の学習機能として，教材だけでなくアイテムの動きを可視化できるようにした．評価の結果として，ドリルを簡単に作成できること，ドリル制御構造を学習できたことは確認されたが，可視化モードの難解さをはじめとして，いくつか課題が残った．

また，他の学習課題への適用の可能性を検討するために，知的技能への対応も検討し，下位ドリル群のプロトタイプ開発を試みて，動作することを確認した．今後は実際にドリルを設置する e ラーニング担当者などからの評価も必要である．また，それとは別に，QTI，携帯電話への対応などが今後の課題としてあげられる．

第 4 章

教授トランザクション理論に基づく 教材シェルの学習環境の開発

4.1 はじめに

AID 研究の中で、特に教材シェル構築のためのインストラクショナルデザイン (ID) 理論として、Merrill (1999) の教授トランザクション理論 (Instructional Transaction Theory ; ITT) が提案されている。この理論に従って開発されたシステムには、あらかじめ ID に基づいた教授トランザクションの方略がアルゴリズムとして組み込まれており、教授内容 (構成要素とその関係) を知識表現 (データ) として登録することで、効果的な学習環境が提供される。知識は、ナレッジオブジェクトと呼ばれる学習オブジェクトで表現している。これは、SME (Subject Matter Expert) が知識をシステムに登録しさえすれば、効果的な学習環境が提供できるということを目指している。コンピュータと親和性の高い既存の ID 理論は、例えば、STAR 遺産モデル (Schwartz et al., 1999) やゴールベースシナリオ (Schank et al., 1999) 等のようにいくつか存在するが、AID 教材シェル構築の枠組みとして提供されている ID 理論は ITT が唯一である。

ITT のシステムはこれまでに複数開発されてきているが、Merrill の提案した理論のすべてを実装しているとは言えない。また、現状において利用できるシステムは存在していない。Merrill 自身はインタビュー (鈴木, 2006) の中でも、この研究の重要性を指摘しながらも、すでに行っていないことを明言している。また、Merrill (1997) は、ツール自体にあらかじめ科学的な根拠に基づく ID 原理を内蔵した学習志向の教材開発ツールが必要であると指摘している。特に ID が普及していない日本の現状においては、SME が利用できる ITT のようなツールが必要であると考えられる。

そこで本研究では、ITT に基づく教材シェルの構築を行うことにした。これまでに教授トランザクションは 13 分類 (Merrill et al., 1992; Merrill and ID2 Research Team,

1996; Merrill, 1999) が提案されているが(表 4.1), 構成要素のトランザクションと, これまでにあまり実装されてこなかった抽象のトランザクションの中の 2 種類を研究対象とした。また, ITT については, 日本にはまったく紹介されておらず, 理論を紹介すること自体にも意義があると考えている。

本章では, ITT に基づく教材シエルのサブシステムの 1 つである学習環境の構築について述べる。次章でもう 1 つのサブシステムであるオーサリング環境をとりあげる。

4.2 教授トランザクション理論

4.2.1 理論の枠組み

ITT の目的は, ID の原則に基づく効果的な教授の提供と, ID プロセスの自動化による効率的な開発にある (Merrill, 1999)。ITT は, ガニエの学習の条件 (Gagne, 1985) を踏まえ, 異なる学習成果には異なる教授方略が必要であるという前提に立っている。

Merrill and ID2 Research Team (1996) によると, この前提に基づく ID 理論は, 図 4.1 に示すように, 学習される知識やスキルの記述的理論, この学習を促進するために必要とされる教授方略の記述的理論, 知識と方略を対応づける処方的理論で構成される。記述的理論は, 学習される知識やその獲得を促進する方略を記述することに利用される概念を同定する。処方的理論は, もし～であるなら～という処方的な形式で構成される。つまり, 学習者が特定の種類の知識やスキルを獲得するには, その種類に適切な教授方略を採用する必要がある。

これらの内容を ITT にあてはめると, 知識の記述的理論がナレッジオブジェクト, 方

表 4.1 教授トランザクションの 13 分類 (Merrill et al., 1992; Merrill, 1999; 鈴木, 2005b)

構成要素 (Component)	同定 (IDENTIFY)* 実行 (EXECUTE)* 解釈 (INTERPRET)*	実体の部品についての名前を言えて情報を覚える 活動の中のステップを覚えて実行する 過程の中の事象を覚えて起こることを予測する
抽象 (Abstraction)	判断 (JUDGE)* 分類 (CLASSIFY)* 一般化 (GENERALIZE) 決定 (DECIDE) 転移 (TRANSFER)	事例を配列する 事例を分類する 事例をまとめる 選択肢から選択する 新しい状況にステップや事象を応用する
連合 (Association)	伝播 (PROPAGATE) 類推 (ANALOGIZE) 代用 (SUBSTITUTE) 設計 (DESIGN) 発見 (DISCOVER)	他の文脈でスキルのひとつのセットを身につける 異なる活動や過程に例えることで, 活動のステップ や過程の事象を身につける ひとつの活動を他の活動を学習することに拡張する 新しい活動を創造する 新しい過程を発見する

表中の*印が本研究の対象である。

略の記述的理論は教授トランザクション，処方的理論はナレッジオブジェクト（知識）と教授トランザクション（方略）の選択および系列化などのルール（図 4.2）から構成される．ナレッジオブジェクトは学習オブジェクトの一種であり，知識（データ）をエンティティ，プロパティ，アクティビティ，プロセスの 4 種類のオブジェクトで記述する知識表現手法である．教授トランザクションは，人間とコンピュータのインタラクションにおいて，ある種類の知識を獲得するための提示・練習・学習者ガイダンスから構成される方略のセットである．これらをコンピュータ上で処理すると捉えると，知識はデータ，教授トランザクションはデータを処理するアルゴリズムとなる．

なお，ITT に関する Merrill の研究の展開については付録 B に記載する．

4.2.2 教授トランザクション

教授トランザクションとは，学習者が特定の知識やスキル（学習目標）を獲得するために必要な学習のインタラクション（相互作用）のすべてのことであると定義されている (Merrill, 1999) ．

また，教授トランザクションは，学習者とのインタラクションのセットであり，異なるナレッジオブジェクトに繰り返し使うことが可能なアルゴリズムである．Merrill (1987c) では，教授トランザクションをコンピュータと人間のインタラクションに限定しており，教授トランザクションという用語を利用するときは，コンピュータで学習を行う場合を想定している．教授トランザクションは，単にトランザクションと略される場合もある．



図 4.1 Gagne (1985) の前提に基づく ID 理論の 3 つの構成要素 (Merrill and ID2 Research Team, 1996, を翻訳)



図 4.2 ITT の 3 つの構成要素 (Merrill and ID2 Research Team, 1996, を要約および翻訳)

教授トランザクションには IDENTIFY や EXECUTE をはじめとする多くの分類が提案されている (Merrill et al., 1992) . これまでに 13 分類 (クラス) が特定され, これらのクラスがさらに構成要素 (Component), 抽象 (Abstraction), 連合 (Association) の 3 種類のスーパークラスにまとめられる (表 4.1) . なお, Merrill et al. (1992) と Merrill and ID2 Research Team (1996) におけるトランザクションの分類は若干異なっているが, Merrill (1999) が採用していた Merrill et al. (1992) に準拠する形とした . なお, Merrill (1999) で具体的に提案されているのは, IDENTIFY, EXECUTE, INTERPRET についてである . また, この分類は IDENTIFY が最も基礎的な内容として, 他のトランザクションの前提となっており, さらに EXECUTE, INTERPRET と複雑になっていくとされる .

教授方略は一般的には「のような学習課題を, のような学習者に教えるためには, のような教授方略を採用するのがよい」という形で記述される (日本教育工学会, 2000) . ITT においては, 教授トランザクション (方略あるいはアルゴリズム) を, ゴール (学習目標) と, それに必要な知識構造 (ナレッジオブジェクト), それらを獲得することを促進する情報提示・練習・学習者ガイダンスのセットで構成して整理している (Merrill, 1999) . よって, 本論文において, IDENTIFY などのように教授トランザクションを英語表記で用いる場合には, ゴール, 知識, 情報提示, 練習, 学習者ガイダンスをセットとした方略の意味として用いる .

Merrill et al. (1992) においては, すべてのクラスに対して, ゴールや知識や方略が整理されているが, あくまでも提案であり, 実装はされておらず, 実現のためには検討すべき事項が多く, 実現可能性もすべてにあるとは言い難い . Merrill (1999) では, IDENTIFY, EXECUTE, INTERPRET が詳細に提案されている . ただし, INTERPRET の練習については, 実装されておらず, あくまでも提案という形に留まっている . また, 学習者の状況 (パラメータ) に応じて, 方略 (アルゴリズム) の調整を行う提案も行われている (Merrill and ID2 Research Team, 1996; Merrill, 1999) .

4.2.3 ナレッジオブジェクト

ナレッジオブジェクトとは

ITT において知識はナレッジオブジェクトによって表現する . ナレッジオブジェクトは, 異なる関連した知識要素のコンパートメント (スロット) で構成されたコンテナと定義される知識表現である (Merrill, 1999) . エンティティ (物) ・ プロパティ (属性) ・ アクティビティ (活動) ・ プロセス (変化) の 4 種類がある (表 4.2) . 例えばスイッチというエンティティ (物) には, オンとオフのプロパティ (属性) があり, 電気を消すというアクティビティ (活動) によって, スwitchのプロパティがオンならオフにというプロセス

表 4.2 4 種類のナレッジオブジェクト

エンティティ (実体 ; entities)	世の中のオブジェクトを表す . 装置 , 人 , 動物 , 場所 , シンボル等を含む . (例えば電気のスイッチ)
プロパティ (属性 ; properties)	質的・量的なエンティティの属性 . 値 (value) のスロットや値に応じた描画 (portrayal) のスロットを持つ (例えば電気のスイッチの位置という属性 . 属性値は on/off)
アクティビティ (活動 ; activities)	世の中のオブジェクトに作用する学習者の行動 . 対応するプロセスへのトリガーを持つ . (例えば電気のスイッチを切り替えるという活動は , スwitchの位置を変えるというプロセスを呼び出す)
プロセス (変化 ; processes)	エンティティのプロパティ値を変化させる事象 . プロセスは , アクティビティか他のプロセスによって引き起こされる . 条件 (conditions) やそれに応じた結果 (consequence) のスロットを持つ . (例えばスイッチの位置を変えるというプロセスは , もし現在位置が on なら , スwitchの位置 (プロパティ値) を off にして , 電気を消すプロセスを呼び出す . もし現在位置が off なら , スwitchの位置 (プロパティ値) を on にして , 電気をつけるプロセスを呼び出す .)

(変化) が生じる . スロットは , 各エンティティにとって必要な情報を登録するための入れ物である .

また , これらのナレッジオブジェクト間の関係を PEAnet と呼んでいる (図 4.3) . 学習者はコントローラー (controller ; 通常はエンティティ) 上でアクティビティを実行する . それがプロセスをトリガーし , もしプロセスの条件 (condition) が真 (true) なら , トリガーされたプロセスは , あるエンティティのプロパティの値を変更する . プロパティの値が変化したとき , この値に関連づけられた描画 (portrayal) が変化するという仕組みになっている .

さらに , すべてのナレッジオブジェクトは , 名前 (Name) , 描画 (Portrayal) , 説明 (Description) という 3 種類の情報スロットを持つ (表 4.3) . この 3 種類だけという意味ではなく , オブジェクトによってはこれより多くのスロットを持つ . 逆に portrayal などには必要無い場合もある (スロットはあるが , 表示はされない場合など) . また , ナレッジオブジェクトは , 他の知識オブジェクトへのポインタのためのスロット持っている .

学習オブジェクトとしての位置づけ

Wiley (2001) は , ナレッジオブジェクトも学習オブジェクトの 1 つであるとみなしている . 学習オブジェクト (Learning Object) は , 技術的に支援される学習において , 利

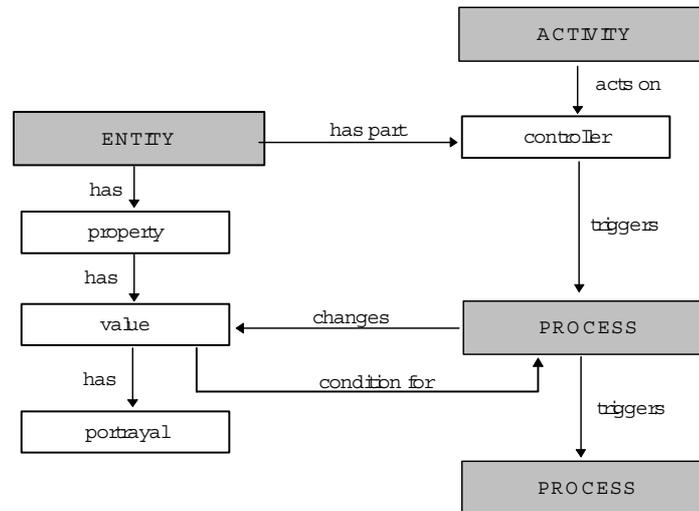


図 4.3 PEAnet(Merrill, 1999)

表 4.3 ナレッジオブジェクトが持つ基本的な情報スロット

名前 (Name)	1 つ以上の知識を参照するシンボルや用語を含む名前。(例えばエンティティ「スイッチ」という名前)
描画 (Portrayal)	質的・量的なエンティティの属性・ナレッジオブジェクトを表現するひとつ以上のマルチメディアオブジェクト(テキスト, 音声, 動画, 画像, アニメーション)を含む。(例えば電気のスイッチが on の状態の画像)
説明 (Description)	製作者がナレッジオブジェクトについての情報を配置することができる。いくつかのサブスロットに細分することが可能である。(例えば電気のスイッチの説明文)

用が再利用あるいは参照される, デジタルかデジタルでは無い, エンティティとして定義される (IEEE, 2007)。また, 学習オブジェクトの例として, 技術的な支援による学習において参照される, マルチメディアコンテンツ, インストラクショナルコンテンツ, 学習目標, インストラクショナルソフトウェアとソフトウェアツール, 人, 組織, 事象を挙げている。

学習オブジェクトについては, 再利用や共有(検索)を目的として, メタデータを付与する LOM (Learning Object Metadata) の標準化が行われている。例えば, 榎本・清水 (2006) は教育情報ナショナルセンターで提供する LOM 検索システムを開発し, 関ほか (2003) は学習者に対して学習オブジェクトを適応的に系列化するためのシステムを開発している。また, WBT (Web Based Training) コンテンツの標準規格として

SCORM (Sharable Content Object Reference Model) があり, 多くの LMS (Learning Management System) や教材作成ツールで採用されている。これにより, 異なるベンダの作成したコンテンツと LMS の自由な組み合わせが可能になっている。オープンソースの LMS (Learning Management System) においては, コースの内容やテストの作成などはできる (田中ほか, 2005) が, 学習教材のレベルでは, Web ページを編集するか, 資料や Flash などのコンテンツを登録するだけのものが多い (SCORM コンテンツを登録できるものもある)。LO の粒度の大きさの観点から見ると, LOM は比較的粒度の大きな WBT 教材などを対象としている。SCORM の SCO も LOM よりは粒度は小さいが, WBT 教材を 1 つの SCO とも見なせる。ITT のナレッジオブジェクトは, 教材を構成する 1 つの画像やテキスト, それらの処理などに対応し, 粒度が小さい。

Wiley (2008) は, 学習オブジェクトの仕様について, 次のように整理している。

- 学習オブジェクトの使用のための高度に規定されたアプローチ
- 学習オブジェクトの使用のための低度に規定されたアプローチ
- 学習オブジェクトの目録作成と検索のための高度に規定されたアプローチ
- 学習オブジェクトの目録作成と検索のための低度に規定されたアプローチ
- 学習オブジェクトの目録作成と検索のための中間的なアプローチ

ナレッジオブジェクトは, 「学習オブジェクトの使用のための高度に規定されたアプローチ」に分類される。LOM や SCORM などは, 「学習オブジェクトの目録作成と検索のための高度に規定されたアプローチ」に分類される。このように, ナレッジオブジェクトは, 他の学習オブジェクトと比較すると, 検索というよりは, 学習オブジェクト自体の学習への利用を志向しているモデルであると言える。

学習支援については, 例えば, SCORM2004 以前は, 学習者の状況に応じたコンテンツの系列化もできなかった (仲林ほか, 2006) など, まずは再利用や共有の観点が先行しており, 学習支援という観点からは脆弱な部分も多い。Wiley (2001) はまた, 学習オブジェクトの応用について, 学習支援を成功に導くためには, ID 理論が必要であるとしている。ITT は, 知識の記述としての学習オブジェクトと方略をセットで規定している ID 理論であり, 再利用や共有という考え方よりも, 学習支援が中心となっている。

4.3 先行研究

4.3.1 ITT システム

これまで研究されてきた Merrill による ITT のシステムは, ID-Expert (Merrill, 1998), IDVisualizer あるいは Instructional Simulator (Merrill, 1999, 2001a, 2003), IDXelera-

tor(Merrill and Thompson, 1999), XAIDA(Halff et al., 2003), Physics Lesson(Merrill, 2001b)である。表 4.4 に各システムのトランザクションへの対応状況やナレッジオブジェクトの利用の有無, それぞれで提示していた事例や方略を示す。オーサリング環境については次章で述べる。

これまで提案されたシステムは, 構成要素のトランザクション (IDENTIFY, EXECUTE, INTERPRET) および抽象のトランザクション (CLASSIFY) のみに対応しており, すべてを網羅したシステムを存在していない。また, ナレッジオブジェクトまで実装しないと ITT とは呼べないが, 知識構造は別に用意し, トランザクションの方略のみを利用しているものもある (Merrill and Thompson, 1999; Halff et al., 2003)。よって, 正確には, ナレッジオブジェクトを利用していない XAIDA や IDxelerator は対象外であるが, トランザクションの参考になるという意味で載せている。また, Physics Lesson は, いくつかのオーサリングツールで開発したとされるが, 物理実験の仮説検証用の専用のインタフェースであり, ナレッジオブジェクトは利用しているが, 方略は別物である。

よって, ITT の枠組みの中では, 基本的には IDVisualizer (Instructional Simulator) が最も参考になると考えられる。IDVisualizer は一方で, ITT(Merrill, 1999) で提案されているトランザクションおよびナレッジオブジェクトの利用という観点からは, IDENTIFY, EXECUTE は実装済みであるが, INTREPRET の練習は実装に至っていない。

また, システムにおいてはインタフェースもさまざまであるが, 複数のトランザクションを実装した際には, 学習者がガイダンス機能の選択を迷ってしまう恐れがあると考えられる。Merrill (1999) において, 明らかに学習者が制御する教授方法の種類が多すぎであることを編集者の Reiguluth が脚注 (p.408) で指摘している。例えば, ID-Expert は, 論文内では IDENTIFY のみに対応しており, 非常にシンプルであるが, 一方で, IDVisualizer は, メニューが豊富であり, 学習者が賢く選べるとは考えにくい。よって, トランザクションを実装する際には注意が必要である。

なお, ナレッジオブジェクトでシミュレーション環境の構築が可能となるが (Merrill, 1999, 2001a, 2003), IDVisualizer の別名の Instructional Simulator の言葉が示すように, 探索だけでなく直接的な教授を付加した, 簡易なシミュレーション (Merrill 2003) 環境である。よって, ITT のシミュレーションは, 複雑なシミュレーションを構築するというよりは, 単純なシミュレーションを構築することを対象にしていると言える。

4.3.2 ITT に基づく回転寿司の新人教育教材の試作

筆者らは, ITT の EXECUTE に沿った形で, 回転寿司の新人教育用教材を開発した (Ichikawa et al., 2007)。ある回転寿司の店舗において, 祝日などで人手が足りない場合

表 4.4 ITT 関連システムのトランザクションとナレッジオブジェクトへの対応状況

システム名	同定	実行	解釈	分類	KO	文献 / 備考
ID Expert					Yes	Merrill (1998)
ID Visualizer					Yes	Merrill (1999), シミュレーション .
IDXelerator					No	Merrill and Thompson (1999)
Physics Lesson					Yes	Merrill (2001b), 物理実験のシミュレーション .
XAIDA					No	Half et al. (2003), シミュレーション .

は対応しているもの、 は一部しか対応していないもの、 は多少異なるが対応していると捉えられるもの。
 ID Visualizer は Instructional Simulator と呼ばれる場合がある。
 同定～分類まではトランザクションを示す。KO はナレッジオブジェクトである。

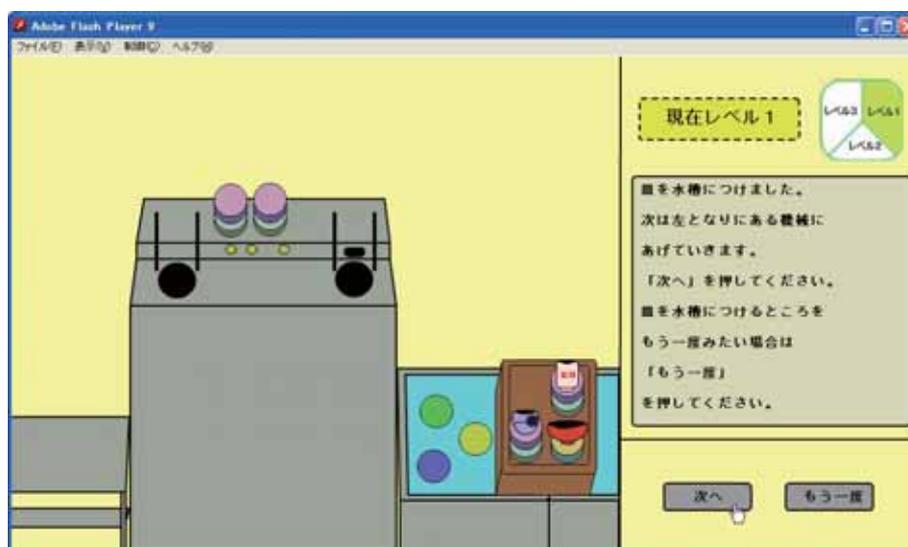


図 4.4 ITT に基づく回転寿司の新人教育教材 (Ichikawa et al., 2007)

に、臨時のアルバイトを雇う場合があるが、その一時的に雇うアルバイトのために必要とされるインストラクタが本来行うべき作業が滞ってしまうという問題が生じていた。その解決策として、ITT に基づく教材を開発して、仕事場の PC に入れておき、新人や臨時アルバイトは、その教材を利用してから業務を行うということにした。

教材は業務の中の洗い物のみを試験的に教材化し、その際に ITT の IDENTIFY (洗い物に必要な物品の名前や配置) は理解できているという前提で、EXECUTE (つまり洗い方の手順) を学習する教材を開発した (図 4.4)。EXECUTE には 4 段階 (レベル) の方略が提供されているが、短時間で学習させるために、レベル 3 とレベル 4 を統合して提供した。実際の現場で 3 名に評価を行ったところ、手順を失敗することなく業務を遂行できるようになり、教育時間を削減することに成功した。この教材はオーサリングは意識せず、方略のみを利用したものであり、ナレッジオブジェクトでは記述していなかったが、方略の有効性や簡単なシミュレーション教材の作成には有効であるとの見通しを立てることができた。

4.3.3 研究課題

以上から，本章の ITT に基づく教材シェルの学習環境の構築における研究課題を以下のように設定した．

Merrill (1999) で提案されているトランザクション方略のすべてを実装したシステムは無く，現状で誰もが利用できる状態にもなっていない．そこで，ITT で提案されている構成要素のトランザクションの学習環境を構築し，これまで提案止まりだった INTERPRET の練習の実装と，検討しか行われていない抽象のトランザクションの JUDGE および CLASSIFY のプロトタイプ開発を行うことにした．データにはナレッジオブジェクトを用いることにし，XML を用いて再定義した．また，トランザクションが増加すると，それだけ方略が多くなり，ユーザが選択できなくなることが考えられたため，学習目標（トランザクション）によって，メニューの表示を厳選するようにした．方略を整理して明示することにより，学習者が方略を理解する手助け（つまり認知的方略の学習）になることも期待できる．

4.4 構成要素のトランザクションの設計

4.4.1 システム概要

本システムは，学習環境とオーサリング環境から構成される．システム構成を図 4.5 に示す．本章では特に学習環境について述べる．システムは，ユーザにシミュレーションを提供する部分と，各トランザクションの処理を行う部分，起動時にデータを読み込む部分から構成される．

学習環境には，Merrill (1999) のシミュレーションエンジンおよび，IDENTIFY，EXECUTE，INTERPRET の方略を実装することとした．方略は基本的に Merrill(1999) を踏襲した．シミュレーションエンジンは，ナレッジオブジェクト間の関係性に基づいて構成される自由に探索できるシミュレーションを実現する部分である．ユーザは画面上のオブジェクトをクリックすることで，反応を得ることができる．

学習環境は，実際に学習を行う場として，自由に探索可能なシミュレーションを提供する（図 4.6）．画面上からは，学習目標の選択，学習ガイド（方略）の選択ができるようになっている．また，学習とテストという 2 種類のモードに切り替えることができる．

データは，ナレッジオブジェクトやトランザクション設定等の XML データと画像などのリソースファイルで構成される．なお，トランザクションの設定によっては，IDENTIFY のみの学習ガイドを提供する教材の作成も可能である．教材ページは，単に Flash を埋め込んだ HTML ファイルである．学習環境は，Web 上で（LMS に載せるな

どして) 提供できるように, Flash を実行環境として選択した.

なお, 本システムには, IDVisualizer(Merrill, 2001a, 2003) による運河の開門で船を移動させる仕組みを学習するコンテンツを, できる限り忠実に再現した (IDENTIFY, EXECUTE, INTERPRET のトランザクションを利用).

また, ITT に基づくことである程度の質は確保できると考えるが, より効果的に学習環境を提供するために, ガニエの9教授事象に基づいて全体の設計を行った. 表 4.5 にガニエの9教授事象の各事象と本学習環境で提供する内容を示す. 特に, 学習の展開(事象4~7)は基本的に ITT 従うが, 学習の導入(事象1~3)や学習のまとめ(事象8~9)については, 従来の ITT システムを参考にしながら, 独自に検討した. 学習目標を提示して強制的に進めていくというよりは, ITT 以前の研究から Merrill が重要視している学習者制御に配慮して, 学習目標は設定できるが, どのように利用するか判断は学習者にゆだねるようにした. また, アイコンを配置するなど, デザインもなるべくわかりやすくなるように配慮した. 学習が楽しくなるように, 効果音なども鳴るようにした.

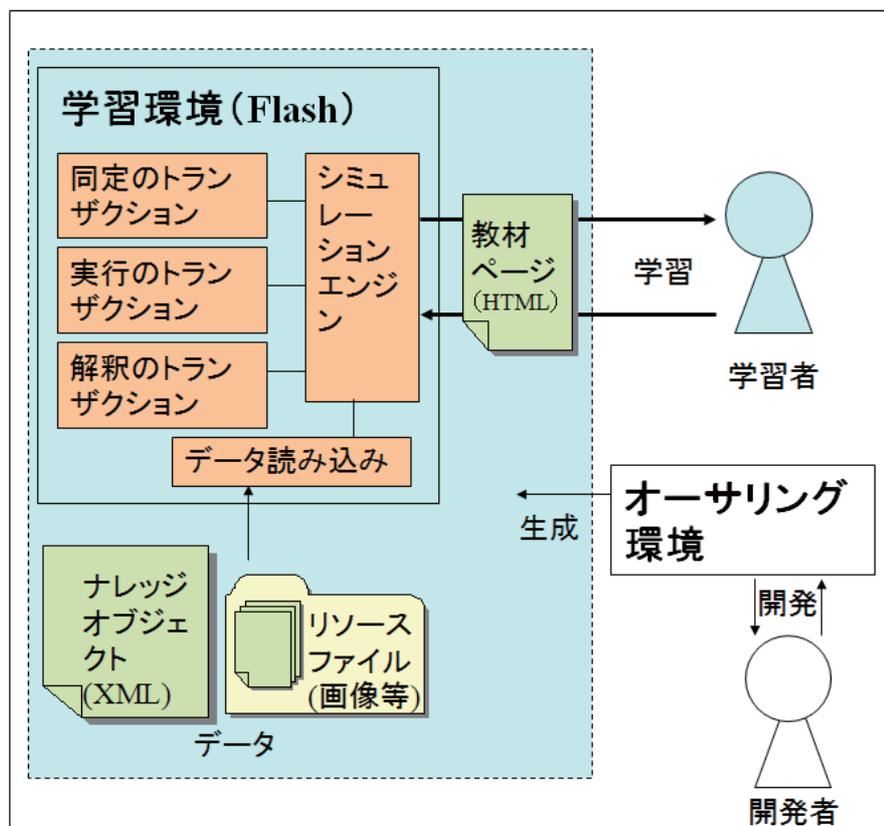


図 4.5 学習環境のシステム構成

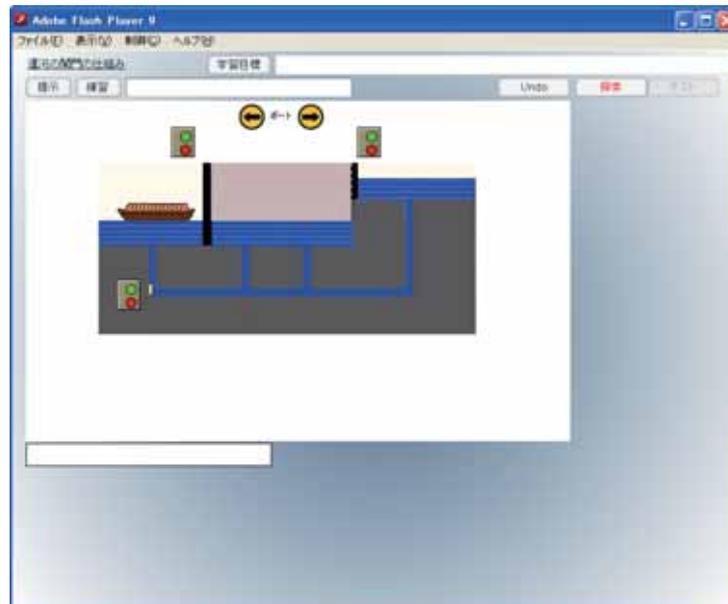


図 4.6 シミュレーション画面

4.4.2 データ仕様

ナレッジオブジェクトの実際の記述（具体的にどのようなスロットがあったのか）については、明確にその情報が提供されておらず、一部が (Merrill et al., 1992; Merrill and ID2 Research Team, 1996; Merrill, 1999) に載っているのみであった。今後のナレッジオブジェクトの共有化なども視野に入れ、XML で記述することにし、明確に仕様を定めた。データの一部を図 4.7 に示す。具体的な仕様は付録 C に記載する。しかしながら本研究のために（独自に）定めた仕様は、今後トランザクションを増やしたり機能を増やしたりすることで、修正の必要は出てくると考えられる。

また、ナレッジオブジェクトだけでは教材の構成上困難な部分があり、それは別の XML として追加することにした。例えば、教材名などが該当する。さらに、トランザクションごとの設定も記載する必要がある。ファイルは 1 つの XML ファイルにおさめることにした。具体的には、ルートが < ITT > となり、その中にナレッジオブジェクト (< KO >) や設定 (< System >) やトランザクション設定 (< Transactions >) を入れることにした。

4.5 学習目標（トランザクション）の選択による制御

Merrill (1999) において、明らかに学習者が制御する教授方法の種類が多すぎであることを編集者の Reigeluth が脚注 (p.408) で指摘している。ガイドのオプション（後述す

表 4.5 ガニエの 9 教授事象による学習環境の機能の整理

事象	本システムで提供する機能
1. 学習者の注意を喚起する	表紙を用意できるようにして、注意を引く工夫を行える場を提供する。Merrill (1998) の IDExpert は表紙作成機能を有し、ヨーロッパの地理を学ぶ教材の表紙において、写真付きで概要を説明している。
2. 学習者に目標を知らせる	学習目標を選択可能にし、画面上部に常に表示することで、目標を意識してもらおう。ただし、目標を設定しても自由な試行錯誤の探索の場は提供するようにする。
3. 前提条件を思い出させる	学習目標の選択のところで、現在選択中の学習目標の前提目標を明示する。また、各学習目標の階層的なつながりを表示する。
4. 新しい事項を提示する 5. 学習の指針を与える	ITT の情報提示の方略を実装する。
6. 練習の機会をつくる 7. フィードバックを与える	ITT の練習の方略を実装する。
8. 学習の成果を評価する	学習モードとテストモードを設け、探索（シミュレーション）で学習者が理解したと思った時点で、テストモードにうつり、テストを受けて学習の成果を確かめる。
9. 保持と転移を高める	今回は特に対応せず。すべての学習目標を達成した際のフィードバックを用意できるようにしてもよいかもしれない。

る「学習ガイド」に相当する）を考慮しても、かなりの数にのぼるため、これを学習者が賢く選択するのは困難であると考えられた。これには、学習者の状況を把握して（学習者のエンティティを作成する）、それに応じて方略を変更するという適応型の教授の可能性が提案されているが (Merrill and ID2 Research Team, 1996; Merrill, 1998)、その実装はかなり複雑になる。

一方で、ITT はトランザクションごとに、具体的な方略として数種類のガイダンスが提案されているため、トランザクションで厳選することが可能である。つまり、学習者にトランザクションを選択してもらえれば、その内容に応じて、ガイダンスを厳選することができると考えられる。そこで、学習目標としてトランザクションを選択できるようにし、それに応じてガイダンス（学習ガイド）を厳選することにした。トランザクションは、IDENTIFY を最も基礎的な内容とする階層的な構造になっている。Merrill (1999) と Merrill et al. (1992) を参考にすると、IDENTIFY, EXECUTE, INTERPRET の順に階層が上位になっていき、下位は上位のトランザクションの前提条件となる。Merrill et al. (1992) の提案においては、INTERPRET には必ずしも EXECUTE が必要でな

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<ko>
<entity id="1" visible="true" enable="true">
<name>船</name>
<description>
<text>運河の水路上で人々や物を運ぶために使われる平底の船である。</text>
<resource>./resources/ship_desc.jpg</resource>
</description>
<refactivity></refactivity>
<refproperty>1</refproperty>
<partof>3</partof>
<haspart></haspart>
<properties>
<property id="1" defaultval="下の運河"></property>
</properties>
<portrayal>
<resource id="1" type="image" src="resources/ship.png" w="100" h="100" x="115" y="170" />
<resource id="2" type="image" src="resources/ship.png" w="100" h="100" x="260" y="170" />
<resource id="3" type="image" src="resources/ship.png" w="100" h="100" x="260" y="123" />
<resource id="4" type="image" src="resources/ship.png" w="100" h="100" x="415" y="123" />
</portrayal>
</entity>

```

図 4.7 ナレッジオブジェクトの XML の一部

い場合があること，EXECUTE に INTERPRET が必要な場合もあることなどが示されていた。しかしながら，シミュレーション環境 (Merrill, 1999, 2003) の場合は，INTERPRET を理解するには，EXECUTE を知っておく必要があるため，EXECUTE の上位を INTERPRET とした。

学習者は，学習開始後に学習目標としてトランザクションを選択する (図 4.8)。学習目標は一般の人にもわかりやすいようにするため，簡潔に以下のように記述することにした。それぞれのトランザクションの日本語訳を当てるようにした。

IDENTIFY 構成要素を特定する

EXECUTE 正しい手順で実行する

INTERPRET 仕組みを説明できる

IDENTIFY は同定という用語がわかりにくいので特定とし，INTERPRET は解釈ではなく説明とした。選択画面には，次のおすすめに星マークを，すでに学習済の (テストに合格した) トランザクションにはチェックマークを表示するようにした。おすすめには，学習済のうち最も上位にある学習目標より 1 つ上位の学習目標に印がつくようにした。それは，下位のトランザクションは上位のトランザクションの前提条件であるという階層関係において，上位のトランザクションが学習済みであることは，下位は学習済みであるとみなせるからである。

トランザクション (学習目標) と学習ガイドを表 4.7 に示す。各学習ガイドの詳細は，次節以降で述べるが，各トランザクションとガイダンスの対応を表に示す。3 種類のトランザクションをあわせると 14 個もの学習ガイドがあり，これを各トランザクションごとに厳選して表示するようにした。画面の上部に学習目標と並べる形で学習ガイドを配置し

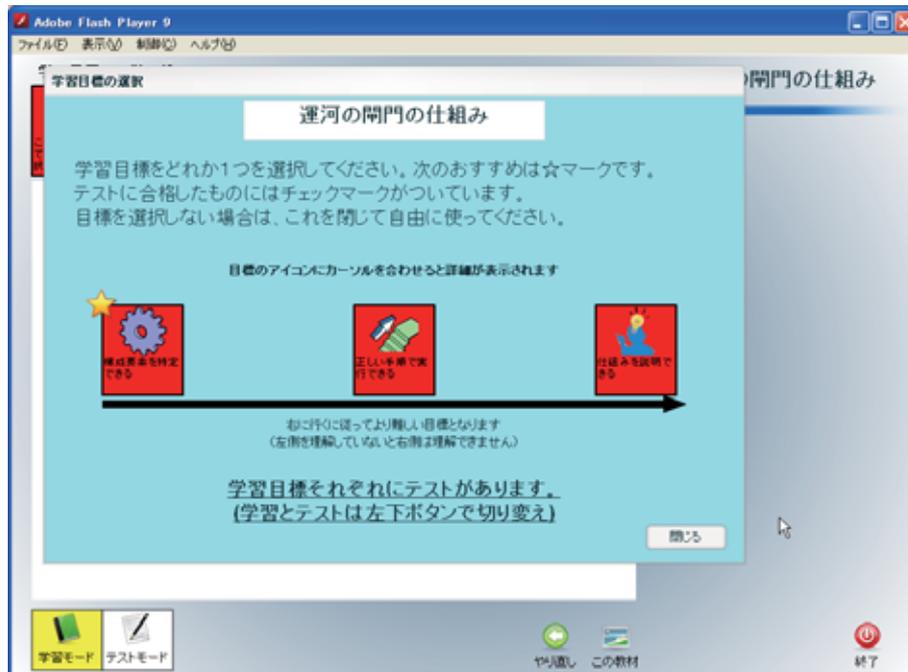


図 4.8 学習目標の選択画面

表 4.6 トランザクションに対応した学習ガイド

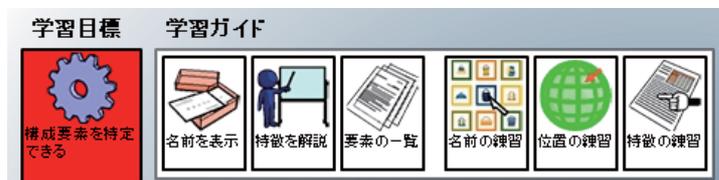
トランザクション	学習ガイド
IDENTIFY	名前を表示, 特徴を解説, 要素の一覧, 名前の練習, 位置の練習, 特徴の練習 (計 6 個).
EXECUTE	デモを見る, 指示通りに行く, 少しずつ自分で, すべてを自分で (計 4 個)
INTERPRET	理由を表示する, 状態を変更する, 結果を予測する, トラブルシュートする (計 4 個)

ている (図 4.9). EXECUTE については, 覚えるべき活動の手順が複数あると考えられたので, 学習目標とならべて下位目標を選択する欄を用意した.

4.6 IDENTIFY と EXECUTE の実装

IDENTIFY と EXECUTE については, Merrill (1999) と Merrill (2003) の内容をほぼ踏襲した.

学習目標として「構成要素を特定できる」(IDENTIFY)を選択時



学習目標として「正しい手順で実行できる」(EXECUTE)を選択時



学習目標として「仕組みを説明できる」(INTERPERT)を選択時



学習目標未選択時



図 4.9 学習目標を選択時（未選択時）のメニュー

4.6.1 IDENTIFY

IDENTIFY は、部品の名前・場所・機能の学習を目的としており、ITT の中では最も基礎的な学習となる。IDENTIFY に対応するナレッジオブジェクトは、エンティティである。

情報提示の学習ガイドとしては、カーソルの下にある部品名を表示したり、部品上でダブルクリックをすることで部品の解説を表示する（図 4.10）。また、各部品の解説を順番に見ていくこともできる。解説には、文章だけでなく、画像を用いることができる。部品名の表示については、学習ガイドにおいて「名前表示」アイコンにチェックしている時だけ機能するようにした（解説の表示も同様である）。



図 4.10 IDENTIFY の解説の表示



図 4.11 IDENTIFY の部品名練習

練習には、部品の名前を提示して学習者に部品を選択させるものや、部品を強調表示して部品の名前のリストからその部品を選択させるもの、機能を表示して部品名を選ばせるものがある（図 4.11）。選択後は正誤のフィードバックを行い、不正解だったときは、再度出題されるように制御している。

4.6.2 EXECUTE

EXECUTE は、手続きの学習を目的とし、4つのレベルのガイドを提供する。レベルが上がるごとに、難易度もあがる。どのレベルを行うかは、メニューの学習ガイドから



図 4.12 EXECUTE のレベル 1「デモを見る」

選択できる。なお、EXECUTE は、前提として IDENTIFY を習得している必要がある。これは、部品名や場所がわからないと、手続きを学習できないことを示している。

EXECUTE に対応するナレッジオブジェクトは、アクティビティであり、実行すべきステップは、すべてアクティビティで表現される。それに応じた処理は関連づけられたプロセスによって行われるが、学習者がここで覚えるのはアクティビティの順番ということになる。

レベル 1 (学習ガイドの「デモを見る」) は見るだけのデモンストレーションであり、順番に手続きのステップを実行してみせる (図 4.12)。学習者はただ眺めているだけであり、終了以外の操作はできない。何度でも繰り返し再生が可能である。

レベル 2 (「指示通りに行く」) は、各ステップで何を行うかを指示し、学習者が実行していくものであり、Simon Says と呼ばれる。学習者には、学習目標と次に行うべきステップのアクティビティ名を表示し、学習者の行動を待つ。学習者が誤ったステップを実行しようとしたときは、「それは<アクティビティ名>ではありません」と表示し、再度行うように促される。

レベル 3 (「少しずつ自分で」) は、ただ次のステップを行うように指示され、学習者に実行させる。具体的には、学習目標を提示し、「次のステップを実行してください」とだけ表示していく。誤ったステップを実行した場合には、レベル 2 と同様のフィードバックを提示する。

レベル 4 (「すべてを自分で」) は、何の助けも無しに学習者に自分で実行させることになり、学習目標を提示し、終了と思った時点で終了ボタンを押すように促される。終了ボタンを押すと、フィードバックとして、学習者の辿ったパスと正解のパスが比較表示され



図 4.13 EXECUTE のレベル 4「すべてを自分で」のフィードバック

る．不正解のパスがあった場合は，その部分が強調表示される．また，手順のステップ数の合計と，その中で手順がそれた数を表示する（図 4.13）．

4.7 INTERPRET の実装

INTERPRET については，なぜそれが起こったかの説明を表示するという方略以外は，これまでに実装されてこなかった．Merrill (1999, 2001a, 2003) で提案はされているが，説明の表示以外は，実装を踏まえたような具体的な提案には至っていない．とくに実装をしていないとの明示はないが，その後の論文の Merrill (2003) では INTERPRET は説明以外実装されていないと明言しており，Merrill (1999) では画面例も提示されていないことから，構想のみであったと考えられる．そこで本研究では，INTERPRET の提案に関して具体的な実装を試みた．

INTERPRET は，ある条件が与えられた時にどうなるのかを予測することや予期せぬ結果が起こった時に説明（トラブルシュート）ができるようになることを目的として，3つのレベルのガイドを提供する．実行のトランザクションが，手続きのステップを覚えるだけなのに対して，解釈のトランザクションは，その背後の処理（仕組み）を理解することになる．なお，レベル 1 は理由の表示と状態の変更，レベル 2 は予測，レベル 3 はトラブルシュートとなる．

INTERPRET に対応するナレッジオブジェクトは，プロセスである．EXECUTE がアクティビティであるのに対し，INTERPRET においては，そのプロセスに焦点をあてる．例えば，あるアクティビティを実行できるかどうかは，プロセスに記述された条件を



図 4.14 理由の表示画面

満たすかが問題となる。よって、そのアクティビティが実行できるかどうかの予測や、うまく実行できなかった場合のトラブルシューティングには、プロセスの条件を把握しておく必要がある。

4.7.1 理由の表示

学習ガイドの「理由を表示する」を選択しておくことで、学習者が行った操作（アクティビティ）毎に、どうしてそうなったのかの理由を表示する（図 4.14）。学習者の操作によって、何か変化があった場合は、「あなたが<アクティビティ名>を実行したとき、<エンティティ名>の<プロパティ名>が<プロパティ値>に変わった。なぜなら、<条件に対応したエンティティ名>の<プロパティ名>が<プロパティ値>であったからである」と提示する（条件の個数分だけ繰り返す）。逆に何も変化がなかった場合は、「あなたが<アクティビティ名>を実行したとき、何も起こらなかった。なぜなら、<エンティティ名>の<プロパティ名>が<プロパティ値>ではなかったからである」と提示する（必要なだけ繰り返す）。

プロセスの結果や条件から、学習者の意図した結果が特定できない場合もあり（これはナレッジオブジェクトの作成の仕方にもよる可能性がある）、その場合は無理に1つに絞るように推論することはせずに、「あなたの意図する行動が不明です」と表示し、可能性のある結果（やりたかったこと）について、「もし<可能性のある結果>にしたければ、<



図 4.15 理由の表示画面（学習者の行動意図が不明だった場合）

その条件 > にする必要がある」というような説明を複数の場合に分けて表示するようにした（図 4.15）。

4.7.2 状態の変更

Merrill (1999, p.421) は、「ある状況では学習者にシステムのいくつかのプロパティの値をセットできるコントロールパネルが提供される。生徒は異なる条件（プロパティ値）の結果を観察することや、これらの結果についての説明を受け取ることによって実験を行うことができる」と提案している。

学習ガイドから「状態を変更する」の項目を選択すると、図 4.16 のような状態変更用のコントロールパネルが表示される。これには、すべてのプロパティの状態が表示される。状態は、各行に「<エンティティ名>の<プロパティ名>が<プロパティ値>」と表示する。

状態を変更するときには、該当する行を選択すると、そのプロパティ値が下に表示されるので（「選択部分の状態」と書いている場所）、そこで変更したい値を選択し、変更するボタンをクリックすることで、状態が変わり、それによって画面も変更される。ただし、そのプロパティ変更がありえない状態になるときは、その理由が表示されるのみで、その変更は反映されない。

プロパティの変更がありえないかどうかは、その変更後のプロパティを結果（conse-



図 4.16 状態の変更（コントロールパネル）

quences)として持つプロセスを探し、それがあある場合にはその条件が現状で満たされるかを確認する。結果が見つからなかった場合には、それはできると判断し、そのまま値を変更する。また、説明画面も同時に表示するようにしておき、その結果をフィードバックすることで、何も起きなかった場合も、なぜそれが変更できなかったのかを表示する。

4.7.3 予測

予測の練習については、Merrill (1999, p.411) では、「このガイドは、システムを設定し、生徒に次に何が起こる？なぜ？というリストから、該当すると思われる項目（予測）を選択するように要求する。生徒はそれから手続きの次のステップを実行し何が起こるかを観察することによって、予測を確認する。生徒の予測の正確さは、システムに記録される」のように提案している。ただし IDVisualizer は Merrill (2001a, 2003) においても予測やトラブルシュートは実装されていないため、提案にとどまっている。そのため、画面例は紹介されていないが、Merrill (2001a) では、出題イメージ（図 4.17）が示されている。

本研究においては、これらの内容をほぼそのままにシステム上で実現することにしたが、プロトタイピングの段階で、最初の練習から理由までを回答することは難しいと指摘を受けたため、理由までは回答させずに、結果の予測のみとした。システムはまず、練習問題とする EXECUTE の学習目標（アクティビティをステップとする一連の活動の流れ

Prior to doing a step S is asked to predict what will happen.

When you open the outlet valve
 nothing will happen
 the lock water will lower

Because
 all conditions are met
 the lower gate is not open
 the upper gate is not closed
 the lock water is not high

図 4.17 予測の出題イメージ (Merrill, 2001a)

を示す目標)を構成するステップから、アクティビティ(ステップ)をシステムが選び、システムの状態を、初期状態からそのアクティビティの直前まで行った状況にする。そして学習者には、予測すべき状況が以下のように表示される。

「< EXECUTE の下位目標 > を行っている途中で、画面のような状態となっています。ここで、次のステップである<アクティビティ名>を行った場合に、結果としてどのような変化が起こるのかを予想してください。」

回答後は、実際に起こる状況を画面上でシステムがデモンストレーションし(カーソルが動き、そのアクティビティを実行する)、その後、フィードバックが表示される。フィードバックは正解、不正解と、どれだけ予測があっていたかを示し、不正解の部分は正答も提示する(図 4.18)。

また、練習問題の生成はランダムとし、出題した問題を記録しておき、すべてのプロセスが出題されるようにした。選択肢の生成については、4 択とした。選択肢は、次のように構成して、ランダム順で表示した。結果とは、次に行われるアクティビティがトリガーするプロセスの結果を指す。アクティビティを失敗させる確率を 25% とし、その場合の結果は何も起こらないということになる。

- 何も起こらない
- 結果(「エンティティ名」の「プロパティ名」が「プロパティ値」)
- 結果(トリガーされるプロセスがあれば、なければランダムに他のアクティビティ結果)
- 他のアクティビティの結果(不正解)

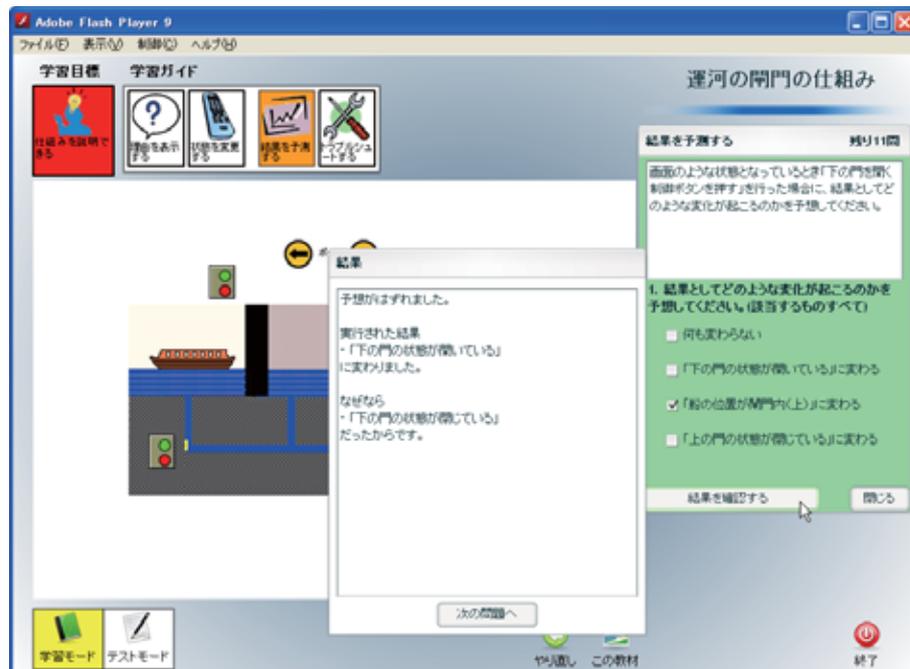


図 4.18 予測のフィードバック

4.7.4 トラブルシュート

トラブルシュートの練習については、Merrill (1999) によって、「このガイドは、手続きの誤りを時々盛り込むようにシステムを設定する。生徒は、手続きの次のステップを実行することと、何が起こったか？なぜか？を説明すること（結果と条件のリストから、何が起こったかなぜかを選択すること）を要求される (p.411)」と提案されている。本システムにおいては、この提案をできるだけ実現するように配慮した。練習問題は、誤りをまぜるように問題を生成する（誤りが入る確率は8割）が、誤りがあった場合には、実際にコントロールパネルを操作させてプロパティ値（誤りの状態）を変更させることにし、誤りが無い場合には、正解か不正解かと正答を表示するようにした（図 4.19）。誤りが無い場合でも、その理由を入力させる方法もあったが、冗長になると考えて入力させないことにした。また、たまには誤りをまぜることで、練習内にメリハリをつけることをねらった。問題文は次のように出題した。

「EXECUTE の下位目標」を行っている途中で、「<前のアクティビティ名>」のステップまで行っています。まず、次のステップである「<アクティビティ名>」を行って下さい。その結果を注意して見て下さい。その後、以下の問いに答えて下さい。



図 4.19 トラブルシューティングの画面

そして、学習者に実際にステップ（該当のアクティビティ）を実行させる。誤った操作を行った場合でも、特に何も告げずに次ぎに進むようにした。これは EXECUTE の復習も兼ねることになる。

その後、学習者は、問題があるのか無いのかを選択する。問題があると選択した場合には、コントロールパネルが表示されて、アクティビティが成功するように状態（プロパティ）を変更するように求められる。コントロールパネルの変更は即座に画面に反映される。学習者は、変更を終了したら、決定ボタンを押す。問題が無い場合は、単に決定ボタンが表示されるだけとなる。なお、コントロールパネルにおいては、ありえない状態への変更は不可能とし、そのような状態に変更しようとした場合は、変更できない旨が表示される。

決定ボタンを押した後は、問題があると選択した場合は、その変更がうまくいったかを実際にシステムがデモンストレーションを表示する（アクティビティを実行する）。その後、正解か不正解か、どの程度の一致であったか、不正解の場合は正答が表示される。問題が無いと選択していた場合も問題がある場合と同様に、システムがデモンストレーションを行い、正解か不正解か不正解の場合は正答が表示される。

4.8 テストモード

学習目標を明示するということは、その目標に達したかどうかを評価する必要がある。そこで、学習モード（通常のシミュレーション画面）とテストモードを用意し、スイッチを切り替えることで、現在選択している目標のテストが出題されるようにした。テストは、IDENTIFY が名前（5問）、位置（5問）、特徴（5問）の計15問、EXECUTE が下位目標の各手順を自分で間違わないように行うこと（レベル4）、INTERPRET がトラブルシュート（プロセスの数だけ、10個以上ある場合はランダムに10問）を出題することにした。テストの出題は基本的に練習と同様であるが、各トランザクションにおいて練習の中で最も難しいものが出題される。IDENTIFY は難しさがあまり変わらないが、部品の名前、位置、特徴を示す問題とした。テストは練習とは異なり、間違った問題が再出題されることは無く、1問ずつ正解か不正解かを表示していく。正答は表示しないようにした。ただし、出題のアルゴリズムは、間違ったものは再度表示されない以外は練習と同様である。

合格基準は、IDENTIFY が全問正解、EXECUTE が一回も間違わずにすべての手順を実行すること、INTERPRET は10問中9問正解とした。合格基準に満たなくなった時点で、システムは不合格と表示し、自動的に学習モードへと戻すようにしてる。また、学習モードからテストモードに移行すると、背景色がかわってテスト専用のインターフェースとなり、テストの出題状況（何問中何問が正解した等）が、画面上部に表示される（図4.20）。また、次の問題に行くときは、「第～問」と表示するようにした。なお、学習目標が選択されていない場合は、テストをクリックしてもなにも起きないようになっている（学習目標に対応したテストが出題されることは、ポップアップヘルプで説明している）。

4.9 評価

本システムは学習環境（自学自習教材）であるので、鈴木（2002）の形成的評価の枠組みを利用して行った。その前に、IDのことを理解している社会人1名と高校で非常勤講師をしている教職志望の大学院生1名に、プロトタイプの段階で適宜意見をもらいながら、改善を繰り返して開発をすすめた。また、学習環境が利用できる状態になったところで、1対1の形成的評価と小集団評価を実施した。



図 4.20 INTERPRET のテスト

4.9.1 1対1の形成的評価

本システムが自学自習用の学習環境としてきちんと学習できるものになっているのかについて、改善を目的として、ソフトウェア系の大学生1名ずつ計3名に対して、1対1の形成的評価を実施した。学習環境は運河の閉門とした。手続きとしては、本システムでの学習の事前事後でテストを実施した。テストについては、最初はテストモードでテストを行ってもらい、その後に学習モードで学習してもらい、学習できたと思った時点でテストモードを行ってもらった。また、実際に本システムを用いて学習している様子を筆者が観察し、学習後にアンケートとインタビューを実施した。アンケートは改善目的であったので、自由記述を重視し、本システムを操作して振り返ってもらいながら、気になった点を中心に回答してもらった。学習する内容には、運河の閉門を利用した。

結果として、3名のうちの1名は、事前テストに合格したため、実際に学習を行ったのは2名となった。2名は、事前テストに不合格であったが、事後テストには合格（合格基準は8割以上）した。また、最初の1名で不具合が生じた部分については修正してから、次の1名の評価に進めた。また、練習部分で残りの問題数を表示したほうがよいなど、出てきた問題点について、必要なものについては改善を行った。両者とも特に口だしすることなく、学習を進めることができていたので、学習環境は自学自習教材として独り立ちしていることが示唆された。

4.9.2 小集団評価

方法

1対1の形成的評価で出てきた問題について改善した後に、本システムの学習効果や使いやすさを確かめる目的で、ソフトウェア系の大学生5名に対して、小集団評価を実施した。小集団評価を実施する前に、他の1名に予備評価を行い、想定した手順通りに実施できることを確認し、不備があった点を修正した。

テストは1対1評価と同様に事前事後テストを行った。事前テスト、事後テストともにシステム上で行ったが、事後テスト合格後に、運河の閘門の仕組みの説明を自分の言葉で紙に記入してもらった。事前テストは、IDENTIFY、EXECUTE、INTERPRETを一度に行うような専用のプログラムを用意して、それを行ってから、通常の教材に進むようにした。事後テストは、各自が受けたいときに受けるように指示した。

また、アンケートは、システムの使いやすさに関する5段階尺度を6項目とシステムの良かった点や悪かった点、改善案の自由記述から構成した。アンケート後に個別にインタビューを実施した。

結果

結果としては、事前テストに全員不合格であったが、事後テストには全員合格した(表4.7)。5名分のデータと平均は以下のようになる。事後テストは合格するまで受けて良いことにしたために、事後テスト回数が多い協力者もいた。合格基準が厳しいこともあり、1回ですべてに合格したのは2名のみであった。特に、協力者Dは、学習ガイドは利用せずに、テストのみで学習していくという方略をとっていたために、回数が多くなっている。学習時間(事前テストを除いて、教材開始から事後テスト合格まで)は、平均が約23分であり、最大が30分、最少が19分であった。これはだいたいこちらの想定する程度であった。事後テストに合格するまでとしたが、合格するまでの学習時間としては、特に不自然に長い協力者はおらず、丁度よかったと考える。

評価後に確認したところ、運河の閘門を知らなかった者は3名(A, B, C)、閘門のイメージをなんとなく知っていた者が2名(D, E)であった。事前事後テストは、前節のテストモードで紹介した出題数や合格基準と同様である。また、ペーパーテストの結果は、単に手順を示した者が1名いたが、バルブの開閉によって水位を上下に調整することは述べていたので、理解はできていたものと推測された。

事前テストについては、平均が13点とかなり高い結果となった。これは、特にボタンなどの名称が推測できてしまうため、IDENTIFYで高得点であったことと、テストをする間に学習者が学習してしまい、INTERPRETなどは簡単な場合もあり、正解してしま

うことがあったようである。本来は事前テスト機能は用意していないが、運河のような簡単な教材は、テストだけでも解けてしまう場合があるようであった。

アンケートの結果は、表 4.8 のようになった。5 段階尺度で、5 が最も良い評価ということになる。最も高かったのは、学習の楽しさで平均 4.8 であり、かなりの高評価であり、ほとんどの協力者が最高の 5 をつけていた。その理由をインタビューしたところ、強制的にやらされないで、自分でいろいろと試したり動かしたりできることが良いという意見が多く、正解時に音が鳴ることなども挙げられた。デザインが見やすいという意見もあった。

最も低かったのは、とまどった点についてであり、全員なにかしらとまどった点があるということが示唆された。特に教材開始時に何をしたらよいのか一瞬とまどったという意見が多く、操作に慣れるのに時間がかかったという理由も挙げられていた。

学習目標を目指せたかについては、あまり意識しないで進めたと回答した協力者が 2 名もいた。学習ガイドが役に立ったというアンケート項目に対して、まったく役に立っていないと回答した 1 名は、その理由として、見ないで行ったため（テスト中心で進めた）とのことであった。その他の学習者は学習ガイドを使っていたが、全員 4 以上で好評であった。

自由記述では、自分のわからない部分や知りたい部分から学習できるので良かったことや、順番にすべてのガイドを行ったように観察された協力者は、レベルごとに段階を追って学習をできたことがよかったと回答していた。一方で、トラブルシューティングの問題の説明がわかりにくいという回答が多く、学習ガイドで複数のウインドウを表示すると（INTERPERT の状態表示）、どこを見てもいいかわからなくなることも挙げられた。

表 4.7 事前事後テストの結果と学習時間

協力者	事前テスト	事後テスト	事後テスト回数	学習時間
A	15 (10-0-5)	27 (15-2-10)	4 (2-1-1)	22
B	11 (6-0-5)	27 (15-2-10)	3 (1-1-1)	30
C	13 (6-2-5)	27 (15-2-10)	4 (1-2-1)	23
D	15 (11-0-4)	27 (15-2-10)	9 (2-2-5)	19
E	11 (6-0-5)	27 (15-2-10)	3 (1-1-1)	22
平均	13	27	4.6	23.2

括弧内は左から IDENTIFY, EXECUTE, INTERPERT の内訳
事前事後テストは点数、学習時間の単位は分である。

表 4.8 アンケート結果

質問項目 \ 協力者	A	B	C	D	E	平均
操作のわかりやすさ	4	4	5	3	3	3.8
学習の楽しさ	5	5	5	4	5	4.8
学習目標を目指せた	3	3	4	2	4	3.2
学習ガイドが役に立った	5	4	4	1	4	3.6
とまどった点はなかった	3	2	3	3	3	2.8
他の教材をやってみたいか	5	5	5	3	5	4.6

考察

まず、学習効果であるが、全員が事前テストで不合格だったのに対して、事後テストで合格していたことから、学習は成立していたと考える。また、時間的にもそんなにかかっておらず、効率も悪くないと思われる。観察していると、特に手順については、かなり短時間で学習が終了しており、学習ガイドが機能していたようであった。特に、学習が楽しかったとほとんど全員が回答しており、学習意欲の側面からも効果的な環境であったと示唆された。

学習目標については、学習目標を意識しなかったという協力者もいた。インタビューからは、あまり考えないで、単におすすめを選んでいったという協力者もいた。目標の説明の字が小さくて目立たなかったため、あまり注意しなかったという協力者もいたため、もう少し目立つように配慮する必要があるだろう。しかしながら、学習ガイドの多さからとまどったとする協力者はいなかったため、本研究の狙い通り、目標による学習ガイドの厳選はうまくいったと考えられる。

また、観察していると、学習者の学習の方法はかなり多様であった。準備された学習ガイドを順番にこなしていく学習者や、EXECUTE から開始する学習者、ほとんどテストだけを行う学習者がいた。これは別に悪いことではなく、全員が合格基準まで達したことを考えると、学習の多様性に配慮した環境になったと言える。一方で、学習ガイドを使わないで、かなり試行錯誤を繰り返して時間がかかっている学習者や、テストを何度も受け続ける学習者がいた。テストを何度も不合格している場合には、それに適した学習ガイドをおすすめするなどの提案（強制ではないが）を行う必要があるかもしれない。

事前テストの結果が示すように、今回のテーマであった運河の閘門の仕組みが、学習内容として比較的簡単であったと思われる。もう少し複雑な学習課題や、ボタンの名前などがまったく類推できないような課題のときに、どうなるかを確かめる必要があるだろう。また、今回は5名という比較的少人数で行ったが、もう少し多い人数で評価する必要もある。



図 4.21 IDENTIFY の教材（東北 6 県の地理）

テストや練習で出題される問題の選択肢数は、1 対 1 の形成的評価の時点で 10 択であったが、選択肢が多すぎるとの要望から 5 択としていた。しなしながら、ランダム抽出のために、例えばボタンに関する問題に対して、ボタンに関する選択肢が 1 問だけなど、確実にわかってしまうことがあった。まぎらわしい問題を出題するように、例えば同じプロパティをもっているものを選択肢にまぜるなど、選択肢生成をランダムだけでなく、多少制御をかけることも考えられる。一方でトラブルシューティングなどの問題では、例えば閘門の門が閉じる条件として開くが指定されていた場合、閉まっているものを閉めようとするという誤りを埋め込もうとする場合があり、非常にわかりにくいようであった。もう少し、自動生成の質を向上させる必要がある。

また、トラブルシューティングの日本語がわかりにくいという指摘が多くあった。この日本語のわかりにくさが、テストの点数を左右している可能性はある（例えばあらかじめ知っている人でも事前テストに答えられない場合も含む）。これも自動生成のために生じている課題であるが、むしろ言葉で行うのではなく、ステージ上で、画像を使いながら、出題や説明をするなど、視覚的に行った方がわかりやすい可能性があり、そのような方向も今後検討していく必要があるだろう。

4.9.3 他の教材への適用

開発した学習環境を運河の閘門以外に適用できるかを確かめるため、いくつかの教材の開発を行った。

まず、ID Expert(Merrill, 1998) によるヨーロッパの地理教材を参考にした東北 6 県の地理教材を図 4.21 に示す。これは IDENTIFY のみを利用した教材である。また、本章



図 4.22 EXECUTE の教材（回転寿司）

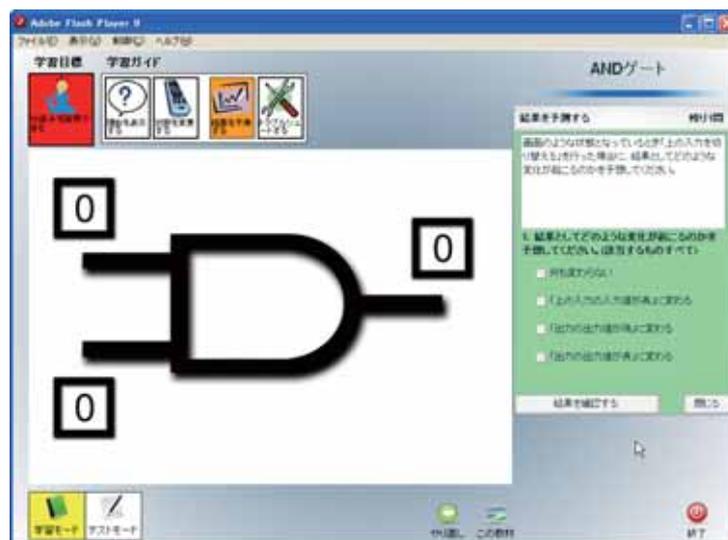


図 4.23 INTERPRET の教材（論理ゲート）

の前半で紹介した EXECUTE に対応した回転寿司新人教育教材を、今回開発した学習環境で再現したものを図 4.22 に示す。もともと本教材は IDENTIFY を既知とみなして EXECUTE のみを設計したが、本学習環境では、IDENTIFY をオンにすれば、若干情報を追加登録する必要はあるが、すぐに IDENTIFY と EXECUTE の両方に対応した教材にもできる。INTERPRET の例としては、論理ゲートの教材を図 4.23 に示す。

これらは、試験的に作成したものであるため、詳細には作り込んではいないが、IDENTIFY、EXECUTE、INTERPRET に対応した教材の試作例である。すべてに対応した例は、本章の船や、次章の電球のつけ方（簡単な教材ではあるが）がある。

なお，過去に回転寿司の教材を開発したことについて述べたが，100 時間程度は開発に費やしていた．本環境で開発した教材は，数時間で完成しており，開発の手間は 20 分の 1 以上になっていると思われる．

4.10 抽象のトランザクションの検討

4.10.1 抽象のトランザクションの特徴

抽象のトランザクションは，5 種類のトランザクションから構成され，学習者はナレッジオブジェクトに関するクラスとサブクラスとインスタンスの関係を獲得することができる (Merrill et al., 1992)．図 4.24 に，抽象の 3 種類のトランザクションとクラスの関係を示す．本研究では，特に判断と分類のトランザクションを中心に検討する．判断はクラス内のインスタンスを順序づけるトランザクション，分類はクラスによってインスタンスを識別するトランザクションである．なお，一般化は分類とは逆になる．例えば，パソコンというスーパークラスに，ノート型というサブクラスがあり，市販されているノート型 (インスタンス) があるとする．ノート型のクラスの特徴は，モニタとキーボード等が一体であり，持ち運びを意図した設計であるという「区別するプロパティ」によって表す (持ち運びを示すプロパティの値が Yes となる等)．それらの区別するプロパティに一致するインスタンスはノート型のクラスに属するとみなし，これを識別するのが分類である．判断は，例えば，提示された市販のノート型から，持ち運びが最も楽なものを選ぶことになる．順序づけの指標には，「次元プロパティ」が用いられ，重量が 1kg 以下，1.5kg 以下...，大きさは B5，A4... などとする．それぞれに重み付けを行い，重量と大きさを統合して判断することになる．抽象のトランザクションでは，クラスの関係を表すナレッジオブジェクトが新たに必要となる．その構想は先行研究である程度検討されているが (Merrill et al., 1992)，具体的な実装の視点では設計されていない．例外として，分類に関しては，実装までの議論がなされている (Merrill, 2001a)．

4.10.2 JUDGE と CLASSIFY の設計

教授トランザクションには，知識の記述と教授方略の 2 つの側面がある．今回はエンティティのみを対象として JUDGE と CLASSIFY の設計を行った．

知識構造

Merrill (2001a) を参考にし，クラスに関するナレッジオブジェクトを新たに設け，Kind と名付けた．Kind の構成を表 4.9 に示す．JUDGE には，1 つのクラスと 2 つのインスタンスを必要とし，その他の抽象のトランザクションには，1 つのスーパークラス，2 つ

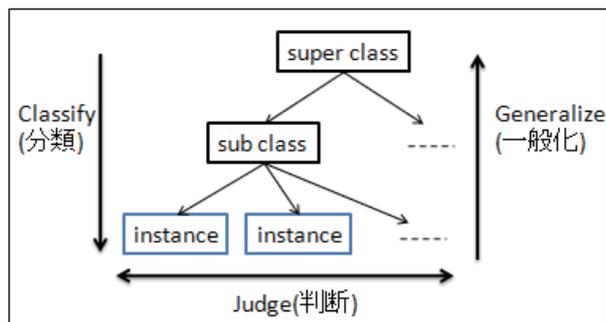


図 4.24 抽象トランザクションとクラスの関係

表 4.9 kind の主要なナレッジオブジェクト

Name	クラス名
Description	クラスに関する説明
Parent	スーパークラス名
Definitions	クラスを区別する（定義する）プロパティと値など
Dimensions	インスタンスを順序づける次元プロパティと値，各値の重み付けなど

のサブクラス，それぞれに2つのインスタンスを必要とする (Merrill et al., 1992)．これは，練習を行うには他の同位概念（不事例）を必要とするためである．スーパークラスの区別するプロパティと次元プロパティは継承される．

方略

Merrill (1999) を参考に，情報提示と練習の視点から，JUDGE と CLASSIFY の方略について検討した（表 4.10）．JUDGE と CLASSIFY の練習については，知的技能であるために，未知の問題を出題する必要がある．繰り返し練習を行うには，多くのインスタンスを登録しておく必要があるが，限界もある．プロパティが学習者の操作で動的に変化する場合はその時点での状態で出題し，さらに多少リアリティは失われるが，繰り返し練習を実現するために，例えばクラス名 + A, B …（例えばパソコンのクラスなら，パソコン A, パソコン B …）のようにインスタンス名を仮称で表現し，プロパティ値も自動で設定して出題することにした．

4.10.3 プロトタイプの開発

ここでは，実際に上記の内容で動くかどうかを確認するために，プロトタイプの開発を行った．技術的に実装できるかどうかの確認が中心であったので，インタフェースには全く配慮していないため，学習できるような状態とはなっていないが，上記の内容で技術的

表 4.10 JUDGE (判断) と CLASSIFY (分類) の方略

	情報提示	練習
判断	<p>順序づけとその理由を表示する：順序づけの名前とインスタンスの順序を提示する。次に、インスタンスの順序を決める事に使われる次元プロパティと値のリストを表示し、どのように順序づけを行うのかを説明する。同時に、次元プロパティの尺度について説明する。説明の内容は、「この順位は、<区別するプロパティ名>に従って行われます。例えば、<インスタンス名>はこの中で、<次元プロパティ名>が<値>であるので、最も順位が高いこととなります。これは、<次元プロパティ名>が<値>、<値>、<値>..という順で良いという尺度に基づいています。」となる。</p>	<p>最も良いものを選ぶ：学習者は、あるクラスのインスタンスを順序づける（あるいはベストを1つ選ぶ）ように指示される。すべての順位を示す（並べ替える）か、ベストを1つ選ぶかはランダムで変化する。学習者がインスタンスにカーソルを合わせると、そのインスタンスに関連づけられているプロパティが示され、そこから次元プロパティを見つけ、値を参照し、順序づけを行う。登録されているインスタンスを利用して出題するが、未知の問題を出題するために、例えばクラス名 + A, B ... (例えばノート型パソコンのクラスなら、ノート型パソコン A, ノート型パソコン B ...) のようにインスタンス名を仮称で表現し、プロパティ値も自動で設定して出題することにした。</p>
分類	<p>分類の見分け方を説明する：正事例（インスタンス）とその分類（クラス名）を表示する。その際にスーパークラスがあれば、その名前も表示する。次に、クラスメンバーシップを決定するのに使われる区別するプロパティやそれらのプロパティ値を伝える。次に、他の正事例（インスタンス）を提示して、同様に区別するプロパティとその値を説明する。最後に、不事例（同位クラスのインスタンス）を提示して、区別するプロパティの値が異なることを示す。</p>	<p>分類を練習する：未知の正事例と不事例をまぜてランダムにいくつか提示し、クラスに属するものとそうでないものを区別するように要求される（4つの選択肢に1~3個の正事例を入れて、正事例となるものを選択させる）。登録されているインスタンスを利用して出題するが、未知の事例を出題するために、スーパークラス名 + A, B ... (例えばパソコンのクラスなら、パソコン A, パソコン B ...) のようにインスタンス名を仮称で表現し（スーパークラスが無い場合には、ただの A, B とする）、プロパティ値も自動で設定して出題することにした。</p>

に動くことは確認された（図 4.25）。

なお、本プロトタイプは、エンティティを抽象化した場合の話であり、アクティビティやプロセスを抽象化することは本研究の範囲外としている。よって、この本検討は JUDGE および CLASSIFY に関する提案の一部分について行ったことになる。

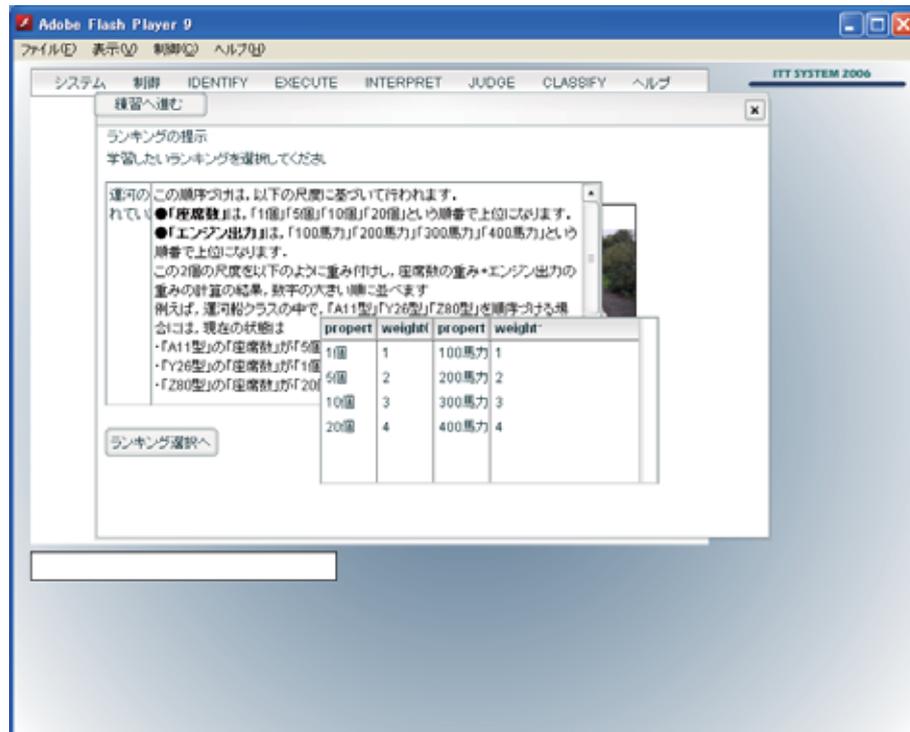


図 4.25 JUDGE の実装確認の様子

4.11 まとめ

本章では ITT 教材シェルの学習環境の開発について述べた。学習環境は学習目標（トランザクション）に応じてメニュー項目（方略）を厳選するインターフェースを採用し、アイコン等でわかりやすくなるように配慮した。INTERPRET はこれまでの先行研究のシステムにおいてほとんど実装されてこなかったが、学習ガイドとして、状態の変更や予測、トラブルシュートを実装した。少人数への評価ではあったが、学習効果が確認され、学習が楽しいという意見をもらうことができた。学習環境はデータを入れ替えることで他の教材に入れ替えることができるため、いくつかのコンテンツを試し、きちんと動作することも確認できた。今後は、より学習者にとってわかりやすくなるように改善するとともに、同じ枠組みで提供される複数の教材に触れることによって、学習者が認知的方略を学習できるかを検証していく必要がある。

第 5 章

教授トランザクション理論に基づく 教材シェルのオーサリング環境の 開発

5.1 はじめに

本章では、前章で検討した学習環境を生成するためのオーサリング環境の開発について述べる。オーサリング環境は教授トランザクション理論 (Instructional Transaction Theory ; ITT) に基づく教材シェルの 2 つのサブシステムのうちの 1 つである。

学習環境の中核論文に据えていた Merrill (1999) は、学習環境は詳細に記載されているが、オーサリング環境についてはほとんど記載されていない。唯一学習環境は、IDVisualizer を元に行っていることが示されており、IDVisualizer のオーサリングについては、Merrill (2003) と Drake (1997) に詳しい。Drake (1997) は IDVisualizer のアーキテクチャを最初に提案した研究である。

IDVisualizer のオーサリングは、基本的に、画面上にリソースを配置し、ナレッジオブジェクトの情報を入力していくようになっている。IDVisualizer は既存のオーサリングツールである ToolBook 上のアドオンとして提供されるため、画像などのリソースについては、ToolBook に備わっているリソースエディタなどを利用して作成する。WYSIWYG で教材を開発していくが、各ナレッジオブジェクトの情報を登録していく形式であるため、ナレッジオブジェクトの内容や各ナレッジオブジェクトの関係性をわかっていないと作成できないようなインタフェースとなっている。ナレッジオブジェクトをどのように入力させるかの工夫について、特に報告はされていない。

本章では、まず Merrill (2003) や Drake (1997) を参考にしながら、ITT 学習環境生成のために最低限必要な ITT オーサリング環境として上級者用の環境を開発した。これは、

ナレッジオブジェクトの内容を理解している人が利用できる程度のインタフェースを目指した。また、その上級者用インタフェースを足がかりとして、SME (Subject Matter Expert) の利用を想定して、ナレッジオブジェクトの構造を知らない人でも開発可能な環境とするための要件を洗い出し、そのプロトタイプを開発した結果について述べる。

5.2 先行研究

5.2.1 ITT におけるオーサリング

ITT のオーサリング環境は、これまでいくつか実装されてきている。主に ToolBook 上でアドインとして利用可能になっている。開発されてきたのは、ID-Expert(Merrill, 1998) ,IDXelerator(Merrill and Thompson, 1999) ,IDVisualizer あるいは Instructional Simulator(Merrill, 1999, 2003) の3種類である。また、ToolBook 以外の環境では、Electronic Trainer(Merrill, 1998) もある。IDExpert の製品版が Electronic Trainer であり、その後継が IDXelerator という位置づけになっている。また、IDVisualizer は、IDXelerator の第2世代という位置づけになっている (Merrill, 2003)。これまでのシステムの中で製品化されたのは、Electronic Trainer ,IDXelerator ,IDVisualizer の3つであるが、現在はどれも入手可能な状態にはない。ToolBook とは市販のオーサリングツールであり、現在は SumTotal Systems 社の製品である。

ID-Expert(Merrill, 1998) は直接教授 (direct instruction; 解説が流れたり、練習が提供されたりする) と探索型 (exploration; 自分で自由に動き回る) を含めた IDENTIFY のみの環境であったが、それを電子テキストのような形に特化させたのが IDXelerator である。一方で探索型を中心に拡張してシミュレーション環境にしたのが IDVisualizer となる。よって、これまでの ITT システムは、この2種類の流れにまとめることができる。

IDXelerator と IDVisualizer は、生成する教材のねらいが異なっており、IDXelerator は電子テキスト型の直接教授 (direct instruction) の学習環境であり、IDVisualizer は、直接教授に加えて、シミュレーションを含めたものであるとされる。IDVisualizer は、別名を Instructional Simulator と呼ぶが、教授 (Instruction) と付記しているのは、直接教授が含まれていることを示している。また、Instructional Simulator は精緻なシミュレーションを構築するためのものではないとし、シミュレーションという用語が不適切なので、Visulalizer という用語を用いたと述べられている (Merrill, 2003)。つまり、ITT に基づくシミュレーション型の学習環境は、複雑なシミュレーションよりは、単純なシミュレーションを推奨し、完全な探索型ではなく直接教授を受けながら、学習を進める環境である。

ITT のオーサリングの作業は、IDXelerator と IDVisualizer では大きく異なる。IDX-

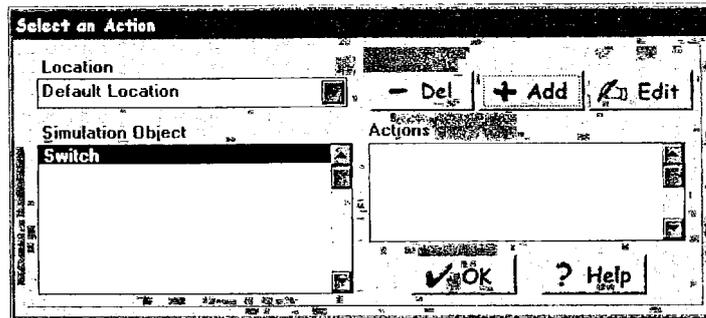


図 5.1 IDVisualizer のアクティビティ入力画面 (Merrill, 2003, Figure.10, p.199)

erator はナレッジオブジェクトを利用していないということが一番であるが，仮に利用していたとしても，作成方法は異なってくる．それは，シミュレーションを行う IDVisualizer は，PEAnet 構造を活用するために，エンティティとアクティビティとプロセスの関係が重要となる．もちろん，エンティティやアクティビティのみ学習する環境なども可能であるが，シミュレーションの場合はそれらの関連づけが必要である．

本研究はナレッジオブジェクトを採用しており，シミュレーション型教材の生成が可能なようにするため，IDVisualizer を参考にすることにした．もちろん，IDVisualizer でもシミュレーション以外の教材は作成できると考えられる．

IDVisualizer の典型的なオーサリング画面を図 5.1 に示す．ToolBook 上でリソースを配置した後で，メニューからオブジェクト（エンティティ）の作成を選び，名前を入力したり，プロパティを追加していくことになる．アクションやプロセスもメニューから選ぶことで，同様のウィンドウが表示されて作成していく．このように，1つ1つのナレッジオブジェクトの情報を入力するようなインターフェースとなっている．

5.2.2 本研究の課題

本研究の目的は，ITT 理論を知らない SME が簡単に学習環境を生成できるためのオーサリング環境を開発することにある．しかしながら，SME が簡単に作成できるということは，先行研究のシステムにおいてもほとんど工夫が実装されてきておらず，手探りで進める必要があった．このような場合は，要件を定義するためにソフトウェア開発のプロトタイプング手法によって開発を進めることが有効であると考えられる．また，学習環境を検証するにあたり，複数の教材を作成する必要もあった．

そこで，まずは ITT 理論を知っている（ナレッジオブジェクトを理解している）人が作成できる上級者用のオーサリング環境を開発した．上級者用の環境は，先行研究と同様に，ナレッジオブジェクトの情報をフォームから入力する形とした．これをプロトタイプとして実際にユーザに利用してもらうことで意見を聴取し，SME でも利用できる環境と

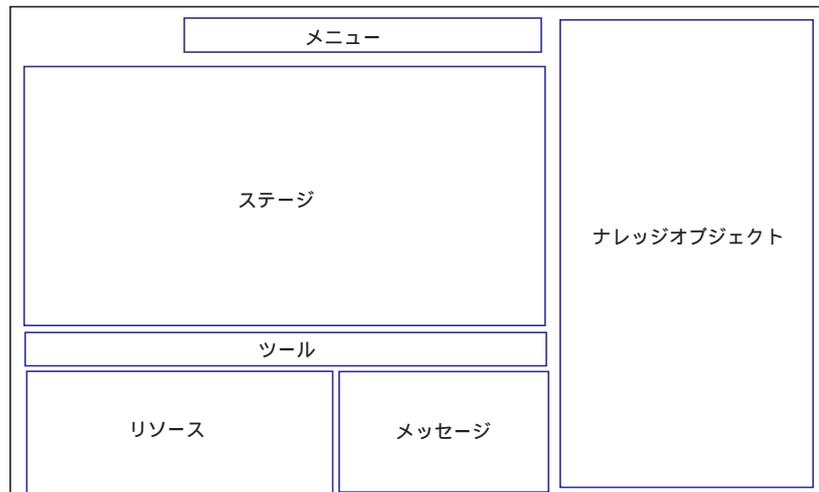


図 5.2 オーサリング画面の構成

するための要件定義を行い，SME 用のオーサリング環境の開発を進めることにした．

5.3 上級者用インタフェースの開発

5.3.1 概要

本研究では，まず従来の ITT オーサリング環境 (Merrill, 2003) を参考にしながら，ナレッジオブジェクトの情報を入力していく上級者用インタフェースを開発した．これにより，筆者が学習環境のサンプル教材開発を効率的に行う意図があった．また，これをユーザに利用してもらうことによって，SME が利用する上で必要とされるシステム要件を明らかにすることも目的とした．

オーサリング環境は，画像等のリソースを読み込みながら，ナレッジオブジェクトの登録と編集，トランザクションの設定を GUI 上で行う．作成後は，学習環境としてナレッジオブジェクトのデータや画像等のリソースと実行プログラムを 1 つのフォルダとして生成する．

システムは 6 つの領域から構成される (図 5.2)．メニュー領域には，システムが提供する機能を実行するためのメニューが配置される．ステージ領域は，リソースを配置して教材のインタフェースを構築していく部分となる．また，ナレッジオブジェクト領域は，4 種類のナレッジオブジェクトの情報を登録していく画面となる．リソース領域は，登録しているリソースが表示される．ツール領域は，画面上に文字を配置するラベル等を作成する際のツール群のボタンが配置される．メッセージ領域は，システムから何かメッセージを表示する必要がある場合に利用する．実際の画面を図 5.3 に示す．

基本的な作成の流れは，まず，リソースをステージ領域に配置することからはじめる．

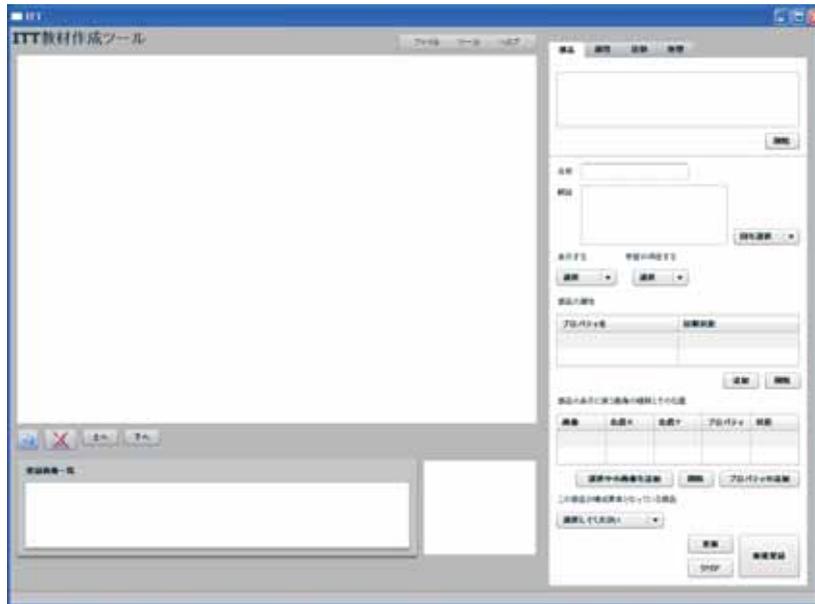


図 5.3 オーサリングの実際の画面

リソースを配置した後，そのリソースをエンティティとして登録する．それには，ナレッジオブジェクト領域にある入力フォームを利用してリソースの関連づけや名前等の入力を行う．その後，エンティティが持つプロパティや，プロセス，アクティビティをナレッジオブジェクト領域のフォームから入力する．作成者は，必要があればマニュアルに従って作業を進めることになる．ある程度完成したら，出来具合を確かめるために，プレビューを行い，実際に学習環境として動くかどうかを確認する．完成したら，書き出しを行うことで，学習環境が生成される．

5.3.2 機能

システムで提供する主な機能は次の 4 種類である．

(1) リソースの登録

リソースはデスクトップ上からステージ領域あるいはリソース領域にドラッグアンドドロップすることによって登録される．登録後は，リソースの領域に表示され(図 5.4)，次からはそこからドラッグアンドドロップをすることで，ステージ上に配置が可能となる．

また，リソースとして登録していても，ステージに無いものは，教材には利用されないことになる．ツール領域には，読み込んだリソースの重ね合わせを変更するボタンなどがある．

ステージ上に配置したリソースは，いつでも自由に移動可能であり，ステージ上のリ



図 5.4 リソース領域の画面

ソースにエンティティが関連づけられている場合は，そのリソースをクリックした際に，右側のナレッジオブジェクトの入力画面はそのエンティティが表示される．リソースは複製することもでき，例えば，操作するスイッチのように，同じ画像を多様する場合は，複製しながら配置することになる．

(2) ナレッジオブジェクトの登録と編集

エンティティ，プロパティ，プロセス，アクティビティの各ナレッジオブジェクトについて，必要な情報を登録する．例えば，エンティティは名前と解説の入力や，リソースとの関連づけ，部品となるエンティティや，関連するプロパティなどを選択する．

PEAnet 構造のナレッジオブジェクトにおいては相互に関連している部分がある（例えばプロセスの設定にはプロパティが必須など）ため，手順が行き来することになる．各ナレッジオブジェクトにアクセスしやすいように，画面右に入力欄を用意し，各ナレッジオブジェクト（エンティティ，プロパティ，アクティビティ，プロセス）をタブで整理することにした．

Merrill (2003) においては，作成の手順の例が示されており，エンティティ，プロパティ，プロセス，アクティビティの順番であった．アクティビティとプロセスは，開発の経験から，プロセスよりも先にアクティビティを考え（つまり，プロセスをトリガーするアクティビティから出発する），行き来があるのを承知しながらも，アクティビティとプロセスというタブの順番にした．

エンティティの登録

エンティティ登録用の入力欄を図 5.5 に示す．これは，オーサリング環境の右側に配置され，「部品」というタブをクリックすると表示される．画面上部はエンティティをリストする欄となる．エンティティを入力する際は，まずエンティティの「名前」を入力し，「解説」とそこで利用する画像を登録する．エンティティは，学習環境の IDENTIFY に利用されるため，例えば前章で紹介した IDENTIFY の学習ガイドの「名前を表示」ではここで入力した名前が，「解説を表示」にはここで入力した解説や画像が表示されることになる．

「表示する」は学習環境に表示するかどうか，「学習の項目」は学習の対象とするか（つまり，背景ではなく実際に覚えるべき項目とするか）を登録する．「部品の属性」の部分

図 5.5 エンティティ登録の画面

では、エンティティに関連づけられたプロパティの、そのエンティティにおける初期値を入力する。学習環境で最初に表示される時は、この初期値の状態となる。

「部品の表示使う画像の種類とその位置」では、エンティティに関連づける画像の登録（エンティティの表示のパターンすべて）と、その画像を表示する条件となるプロパティとその値（状態）を登録する。関連づける画像が1つの場合は、プロパティを登録する必要はない。

「この部品が構成要素となっている部品」では、登録中のエンティティを部品とするようなエンティティがあれば選択する（エンティティに関連づける画像の重ね合わせに影響する）。

入力が済んだら新規登録を押すことで登録される。また、過去に登録したエンティティ

図 5.6 プロパティ登録の画面

については、上部のリストに表示されているが、それを更新する場合は、リストからエンティティを選択して、その内容が各項目の欄に反映されるので、編集した後に、「更新」ボタンをクリックすることになる。

プロパティの登録

プロパティ登録用の入力欄を図 5.6 に示す。「属性」タブをクリックすると表示される。画面上部はプロパティをリストする欄となり、過去に登録したプロパティについては、そこを選択することで内容を更新できる。「名前」にはプロパティの名前を入力し、「状態」の項目にそのプロパティが取り得る値を追加していくことになる。

アクティビティの登録

アクティビティ登録用の入力欄を図 5.7 に示す。「活動」タブをクリックすると表示さ

図 5.7 アクティビティ登録の画面

れる。画面上部はアクティビティをリストする欄となり、過去に登録したアクティビティについては、そこを選択することで内容を更新できる。「名前」の項目にアクティビティの名前を入力し、「この活動から呼び出される処理」には、このアクティビティがトリガーするプロパティを選択する。「この活動を呼び出す部品」の項目からは、このアクティビティを関連づけて、クリックすることでアクティビティが実行されるエンティティを選択する。「手順の登録」という項目には、EXECUTE の目標を構成するステップ（アクティビティ）の順番を登録する。

プロセスの登録

プロセス登録用の入力欄を図 5.8 に示す。「処理」タブをクリックすると表示される。画面上部はプロセスをリストする欄となり、過去に登録したプロセスについては、そこを

図 5.8 プロセス登録の画面

選択することで内容を更新できる。「名前」の項目にプロセスの名前を入力し、「結果を示す属性の値」では、プロセスの結果を表すエンティティ名とプロパティ名とプロパティ値を選択して追加する。「その結果になるための条件」では、結果の部分を選択することで、その結果に対応した条件となるエンティティ名とプロパティ名とプロパティ値を選択して追加する。id は結果と条件の対応がわかるようにした通し番号である。

「呼び出すプロセス」では、このプロセスがトリガーするプロセスを選択する。

(3) プレビュー

作成した教材をプレビューで確認できる(図 5.9)。プレビュー画面では、学習環境に登録されているトランザクションのすべてをテストできる。ツールメニューからプレビュー



図 5.9 プレビュー画面

を選択すると、学習環境がブラウザ上に立ち上がるので、登録したナレッジオブジェクトがうまく機能するかを確認することになる。

(4) 読み込みと書き出し

以前作成した教材を読み込む機能と新たに作成した教材を書き出す機能である。書き出す内容は、1つのフォルダ内に、ナレッジオブジェクトのデータのXMLファイル、教材ページ（Flashを埋め込むHTMLファイル）、swf（Flash）ファイル、リソースファイル（フォルダ）となる。

5.3.3 実装環境

本環境は、Adobe Flex Builder3を用いてAdobe AIR（Adobe Integrated Runtime）として構築した。AIRは、Adobe Flashで構築したシステム（学習環境）との親和性が高い。また、作成者はデスクトップ上で開発を行うことを想定し、デスクトップアプリケーションを開発できるAIRを選択した。読み込み可能なリソースについては、Flashで読み込める形式（例えば画像であればJPEGかPNG）のみとなる。ナレッジオブジェクトのデータは、XMLファイルに書き出され、実行時にはそのファイルを読み込んで初期設定がなされる。

タイトル：「電球がつく仕組み」

(学習目標は、最終的に電球がつく仕組みを説明できることとなります。)

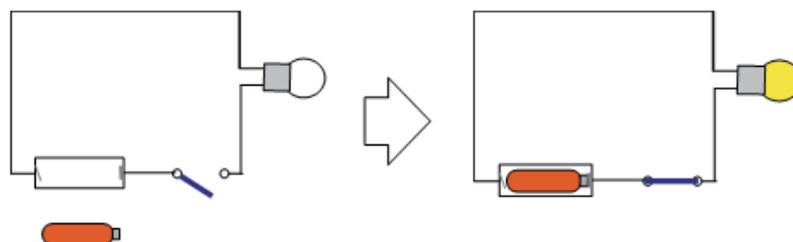


図 5.10 作業課題で提示した作成物のイメージ

5.4 上級者用インタフェースの評価

5.4.1 方法

上級者用のインタフェースを用いて、マニュアルを見ながら実際にシミュレーション教材を開発できるかどうかを確かめる目的でユーザビリティ評価を行った。評価はソフトウェア系の大学生4名にした。この4名は学習環境の評価にも参加しており、運河の閘門の学習環境を利用したことがあり、環境のイメージは持っている。オーサリング環境は、学習環境を一度は利用したことがある人を対象と考えているため、学習環境の利用経験者を対象とした。また、4名のうち1名はITT理論や学習環境の仕組みを知っている上級者（XMLソースレベルでの教材作成経験者）とした（以降、協力者Aとする）。

作成課題は、実験協力者が学習環境で学習したものと別の内容とした。これは、学習環境で学習すると、ナレッジオブジェクトの構造を理解することにつながるが、実際の作成者は作ろうとするシステムの学習環境を利用した経験がないと考えられるからである。想定作成者をSMEとするために、課題は電池を入れてスイッチをオンにすると電球がつく仕組みとし、誰もが知っている内容（簡単なシミュレーション）とした。作業課題は、混乱をさけるためにこちらからだいたいの作成物のイメージを提示し（図5.10）、さらに余計な開発時間がかからないように課題で利用する画像ファイルを配布した（図5.11）。

手続きとしては、1人1台にPCを用意し、最初に各自の作業するPCにAdobe AIRのランタイムと本システムをインストールし、画像ファイルをコピーして、利用できる環境を整えた。次ぎに課題を記載した手順書と、システムの操作方法を記載したマニュアルを配布して、課題の説明を行い、作成を開始してもらった。終了のタイミングは作成が終わったことを自己申告することにして作業時間に制限は設けなかったが、際限なく続く可能性があったため、最大2時間であるという目安を伝えた。作成終了後にアンケートに記入して若干のインタビューを行った。

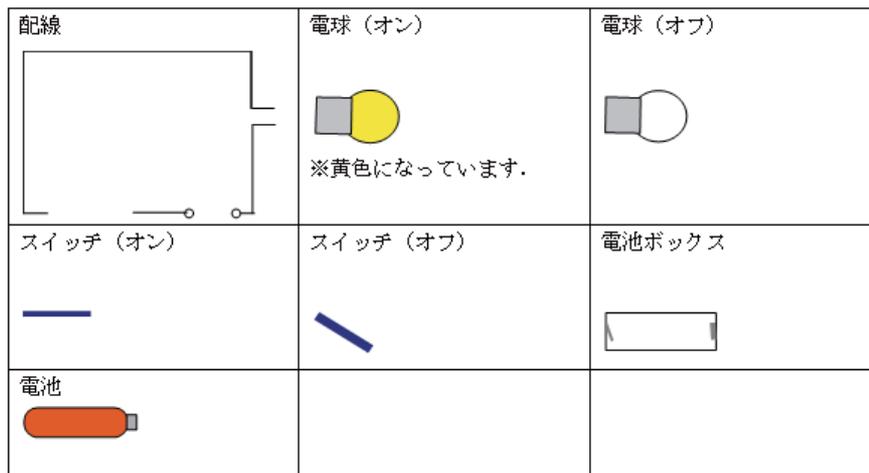


図 5.11 作業課題で用意した画像

5.4.2 結果

作業の結果，4名のうち3名が動く教材を作成することができ，1名は作成に至らなかった．よって，問題は多く存在するが，一応教材を作成できる環境であることは確認できた．作業時間とあわせて，表 5.1 に結果を示す．システムに予期せぬ不具合が生じてしまい，やり直した協力者もいたため，参考程度の作業時間である．協力者 C の作成した教材を図 5.12 に示す．

作業時間については，協力者 A（上級者）は，当然のことながら他人と比べて圧倒的に開発スピードがはやかった．XML ソースレベルでの教材作成経験があったため，ナレッジオブジェクトの XML ソースの入力を素直に行う本インタフェースは理解しやすい（XML の構造を知っていると使いやすい）と感じ，画像のドラッグが直感的でよかったとの意見であった．その他の協力者は，こちらが想定する程度の時間はかかっていたが，不具合のためのやり直しで，30 分程度余分にかかっている換算である．

協力者は，まず手順書とマニュアルを読むことから始めていた．その後，作成に着手したが，画像をステージ上に配置するものの，ナレッジオブジェクトの登録については最初はかなりわからない様子であった．実験中に各自の様子から，できない可能性も考慮し，課題内容を簡単にしてもよい（例えば電池は電池ボックスに入った状態でスイッチのみ切り替える）旨を伝えた．

ナレッジオブジェクトの記述については，名称のつけかたがさまざまであったが，エンティティ，プロパティ，アクティビティ，プロセスは理解していた．アクティビティとプロセスの関係を理解することに時間がかかったという意見が出た．

表 5.1 作業結果と作業時間

協力者	作業結果	作業時間	作成物の内容
A	完成	21 分	電池をクリックすることで電池ボックスに移動して、スイッチをクリックして切り替えることで電球が点灯する。スイッチのオンオフや電池の脱着も可能である。
B	完成	111 分	電池ボックスに電池は最初から入っていて、スイッチをクリックすることでオンになり、電球が点灯する。ただし、スイッチの切り替えはオンのみで点灯するまでの一方通行である。
C	完成	92 分	電池をクリックするとボックスに移動して、スイッチをクリックすることでオンになり、電球が点灯する。ただし、電池の移動とスイッチの切り替えは点灯するまでの一方通行である。電池がステージ上から多少はみ出していた。
D	未完成	120 分	オーサリング環境上では画像の配置はできていたが、ナレッジオブジェクトの登録はほとんどできてない状態であった。

協力者 D については、教材は未完成であったが、作業終了後に自分が作りたかったナレッジオブジェクトのイメージを紙面上に書き出してもらったところ、ナレッジオブジェクトの関係性は理解できていたようであった。協力者 D は、作成できなかった理由として、どのようにナレッジオブジェクトを登録すればよいのかがわからなかったと記載しており、インタフェースがわかりにくいことに起因していると考えられた。

作成方法としては、ステージ（画像の配置）部分をうまく活用しながら、アクティビティやプロセスを登録できた方がよいという意見や、入力項目を簡易化すべきであること、メニューでありえない項目は表示しないようにするなどの意見が挙げられた。また、終了後のインタビューからは、1つの状態を変化させること（例えば、スイッチのオンとオフで画像が変わること）に対して、複数の画像を登録することに違和感を感じた協力者が2名いた。スイッチの棒を回転して登録するなど、画像をステージ上で変形させて登録するものと思っていたようであった。

協力者の B や C からは、プレビューで動いたときは楽しいことや、わかりにくいエンティティの設定部分を改善すれば面白いシステムになること、画像の組み合わせから教材が生成できることが便利といった意見など、本システムの目指すところに好感をもっているようであった。

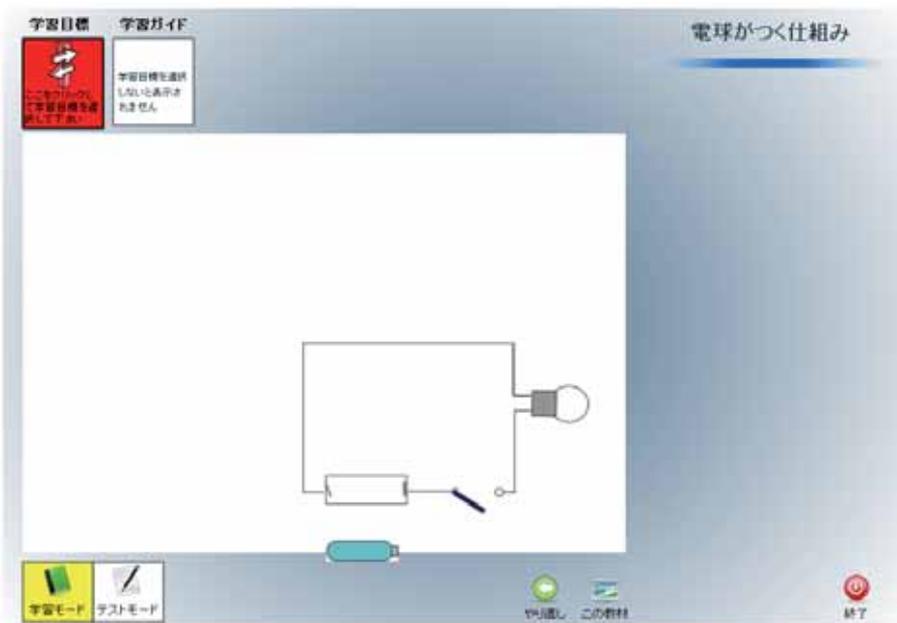


図 5.12 協力者 C が作成した教材

5.4.3 考察

従来の ITT のオーサリング環境より機能的には多少劣るが、上級者を想定した協力者 A は、短時間で教材を作成することができ、上級者用の環境としては十分機能していると考えられた。よって、上級者用のインタフェースとしては十分であると考えられる。

それ以外の 3 名のうちの 2 名については、マニュアルに頼りながらも教材が一応の形にはなった。ただしこれは、全員がソフトウェア開発の素養があることが少なからず影響しているとは考えられる。このままでは SME が利用することには、多くの改善を要するであろうが、協力者 B や C からはシステムに関して好意的な意見も得ることができ、今後改善していけば SME 用のインタフェースも提供可能であるとの見通しを得られた。

上級者以外が利用するためには問題点は山積みであったが、入力項目が複雑で多いことが困難さの 1 つの要因であった。学習環境と同様に学習目標から入力項目を厳選したり、その時点の作業に必要な無い項目は表示しないなどの工夫が必要である。また、教材作成の様子を観察していた際に、プレビューで動かないことを何度も試行していて悩んでいた協力者がいた。それはナレッジオブジェクトの整合性がとれていないことが原因であったため、プレビューの前にある程度自動検出するなどの作成支援機能が必要であると考えられた。

また、ナレッジオブジェクトの関係性は理解できていたようであったが、マニュアルの説明によるものであり、マニュアルを読まなくても作成を進められるようにすることが必

要である。例えば、ステージをうまく使いながらナレッジオブジェクトを登録できると良いという意見があったように、ナレッジオブジェクトの登録を、なるべく配置した画像を使いながらステージ上で視覚的に行っていけるようにすることや、ナレッジオブジェクトの関係性をステージ上に図示することで理解を深めるようにできるような支援が必要であると考えられる。作成プロセスに関しては、3名がまず電気の消えた状態（初期状態）から作成していく形で進めていたため、最初に初期状態を作らせるようなプロセスが自然に受け入れられる可能性があった。

完成した教材は、協力者 B と C については、電球が点灯するまでの一方通行であった。それに対して協力者 A はさまざまな可能性に対応していた。理論を知っていれば試行錯誤を行うシミュレーション教材を意識するが、そうでない協力者はその手順ができればよいと考えているようであった。これは、例えばシミュレーション環境で試行錯誤して学習を進めることや、各トランザクションを提供することが、教材利用経験者とは言え、理解していないと考えられた。課題の提示方法が悪かったとも考えられるが、学習環境での提供イメージを膨らませる（シミュレーションとして試行錯誤できる環境を構築できることを伝える）ことが必要である。

5.5 SME 用インタフェースの開発

5.5.1 システム概要

SME 用のオーサリング環境については、前節の上級者用インタフェースの読み込みと書き出し、プレビューなどの基本機能はそのまま踏襲した。上級者用インタフェースのプロトタイプ評価の結果や先行研究を踏まえ、SME が利用するために、入力の簡便化（作業の複雑性の解消）とナレッジオブジェクトの関係性の提示の 2 つの観点からシステムの工夫を試みた。また、SME が最初に画面を見た時点で挫折しないように、画面があまり複雑そうに見えないように配慮した。

5.5.2 入力の簡便化

上級者用インタフェースのプロトタイプ評価の結果からは、協力者 D は入力の難しさから完成に至らず、他の協力者からも入力項目の厳選の必要性が指摘された。そこで、入力の簡便化について検討し、ナレッジオブジェクトをペアにしての開発とトランザクションによる厳選を行うようにした。

ナレッジオブジェクトをペアで登録

上級者用インタフェースは4種類のナレッジオブジェクトについて、タブを切り替えながら、作成を進める形式であった。これは、ナレッジオブジェクトを知っている開発者にとっては、どうすれば効率よく開発できるか(ある種類のナレッジオブジェクトを登録する前に、登録すべき他のナレッジオブジェクトはあるのか)を意識しながらできるためにあまり問題とはならないが、ナレッジオブジェクトを知らない開発者にとっては、よく後戻りが発生することになる。また、ナレッジオブジェクトの4種類を別々に登録しようとする、入力欄が複雑となり、上級者用インタフェースの評価では、混乱の要因となっていた。これまでの先行研究においては、それぞれのナレッジオブジェクトの情報は個別のウィンドウから入力する形となっており、あまり配慮されてはこなかった。

そこで、入力や各ナレッジオブジェクトの行き来の手間を減らすため、各ナレッジオブジェクトの関連性の強いもの同士をペアで登録させることにした。ペアは、エンティティとプロパティ、アクティビティとプロセスとした。エンティティを登録してプロパティを登録することを繰り返し、それができた段階で、アクティビティとプロセスを同時に設定していくことになる。

プロセスには、プロパティを登録する必要がある、アクティビティは、トリガー元のエンティティを登録する必要がある。また、アクティビティはプロセスをトリガーするために、プロセスがあらかじめ登録されている必要がある。しかしながら、作成の考え方としては、アクティビティを先に考える方が自然であると思われた。そこで、まずボタンを登録し、アクティビティに関連づけたプロセスを登録するというようにした。シミュレーション環境の場合は、基本的に、ユーザの操作に対して何らかのアクションが起こることから、こちらの方が自然であると考えた。

システム上では、「部品の登録」としてエンティティとプロパティを登録するタブウィンドウと、「活動の登録」としてアクティビティとプロセスを登録するタブウィンドウを用意した。

部品の登録(図5.13)においては、「画像や特徴」の部分で、エンティティとリソースの関連づけや、プロパティとの関連づけを行う。プロパティ入力欄は普段は隠しておき、必要に応じて表示するようにした(図5.14)。

活動の登録(図5.15)においては、「部品の選択」のところで、まず起点となる部品(そのエンティティをクリックすることが活動の開始となる)を選択する。そして、「活動後に起こる変化のつながり」のところで、その活動(アクティビティ)によって呼び出されるプロセスを登録していく。プロセスは連鎖(トリガーする)分だけ登録していくことになる。未入力という部分をクリックすると、結果と条件という入力欄が表示される(図5.16)。そこで、例えば結果の欄をクリックすると、登録された部品(エンティティ)一覧



図 5.13 部品（エンティティとプロパティ）の登録



図 5.14 プロパティの登録（画面右下）

が直下に表示され，そこから部品を選択する．結果の欄に部品の絵が入り，プロパティ値が表示されるので，絵の下に表示されている三角ボタンによって，プロパティ値を切り替えて調整する．

以上のように，エンティティとプロパティ，アクティビティとプロセスをセットで登録していくようにした．



図 5.15 活動（アクティビティとプロセス）の登録



図 5.16 プロセスの結果と条件登録（画面右下）

トランザクションによる厳選

Merrill and ID2 Research Team (1996) は、知識獲得の手法として、「ガイドされた知識獲得 (Guided Knowledge Acquisition)」が可能であると提案している。これは、トランザクションによって、必要なナレッジオブジェクトの把握が可能であることを利用している。その提案の具体的な内容は、「一般的な用語で述べたゴールをユーザ（作成者、

SME) に提供し、ユーザが選択すると、システムが対象ナレッジオブジェクトを特定するために質問をする。例えば、異なる誤りの条件下で起こる事象を予測するというゴールを選択すると、システムは、エンティティの作成、プロセスの付加などをユーザに促すことができる」とする。これらは質問をして処方を提示するというエキスパートシステム的な支援であるが、本システムにおいても、例えばゴールの種類(教授トランザクション)によって、必要なナレッジオブジェクト(あるいはその中のスロット)を判断できるので、ナレッジオブジェクトの入力をガイドすることが可能であると考えられる。

本研究の学習環境はシミュレーション用であるため、PEAnet 構造が基本となる。トランザクションの階層的には、EXECUTE の上位が INTERPRET であるために、ナレッジオブジェクトのプロセス(INTERPRET に対応するナレッジオブジェクト)は EXECUTE には必要ないはずであるが、PEAnet 構造のシミュレーションの学習環境実現上、EXECUTE においてもプロセスは不可欠となる。よって、シミュレーションの場合は、トランザクションに関係なく、4 種類のナレッジオブジェクトが必要となる。つまり、シミュレーション環境上では、トランザクションによる入力項目の厳選機能はあまり意味をなさない。しかしながら、第4章で例を示したように、本環境においても、IDENTIFY だけは、シミュレーションではない(PEAnet 構造によらない)教材の開発も可能である。今後トランザクションを増やしたり、シミュレーションではない EXECUTE や INTERPRET への対応の可能性もある。

よって、本システムでは、今後も見据えた上で、トランザクションによって、必要なナレッジオブジェクトを表示するように、学習目標を選択させることにした。それによって、必要なナレッジオブジェクトの入力欄のみが表示される。例えば、IDENTIFY のみを選択したとすれば、エンティティの分しか入力欄が表示されず、エンティティと対で登録していくプロパティ入力欄も表示されないようになる。

開発した目標の設定画面(教材タイトルを含む)を図 5.17 に、IDENTIFY のみを設定した場合の部品登録画面を図 5.18 に示す。

5.5.3 ナレッジオブジェクトの関係性の表示

オーサリング環境は、基本的に PEAnet 構造による簡単なシミュレーションの構築が主目的である。よって、単に1つ1つのナレッジオブジェクトを入力していくのではなく、エンティティ・アクティビティ・プロセス・プロパティの関係性(お互いが影響しあうこと)の理解が必要となる。IDVisualizer の場合は、それぞれを関連づけながら、ナレッジオブジェクトを入力していくことになるが、視覚的にその関係を表示することには配慮されていなかった。

そこで、エンティティにアクティビティが関連づけられ、さらにアクティビティがプロ



図 5.17 目標の設定画面



図 5.18 IDENTIFY のみを選択した場合の部品登録画面

セスを呼び出していくというエンティティが起点となる関係性に着目し，その関係性（リンク）を提示することにした（図 5.19）．また，線を引くには，画面上に配置されているリソース同士に引かないと視覚的に表現ができないため，プロセスはエンティティに関連づけられたプロパティを変更するという視点から，リンク先は，プロセスによって影響を受ける（プロパティが変わる）エンティティとした．なお，本システムにおいては，基本的にプロパティはエンティティにしか関連付かないために，このようにしている．



図 5.19 ナレッジオブジェクトの関係性の表示

具体的には、作成者がステージ上にリソースを配置していき、ナレッジオブジェクトを登録・関連づけを行う際に、ステージ上に配置されたりリソース（エンティティに関連づけられたもの）を右クリックすると、関係するナレッジオブジェクトを線で結ぶようにした。これは、エンティティに関連づけられたアクティビティがトリガーするプロセスの結果として、変更されるプロパティを持つエンティティに線を引き、プロセスからトリガーがある場合には、さらにたどって線をつなげていく。この線はアクティビティやプロセスを表現している。

線を引くと同時にシステム側では、線に見出しをつけ（線のところに表示）、画面下部にエンティティ情報（名前）、アクティビティ情報（名前）、プロセス情報（結果のプロパティ名と値とエンティティ名）を表示する。それぞれの情報をクリックすると、それに該当するナレッジオブジェクトの詳細情報（入力欄）が、画面右のタブウインドウに表示される。線に付加するプロセスの見出しは、処理 1、処理 2.. とした。

このように視覚的に関係を示して構造の理解を促進することで、作成者の教材対象のモデル化の支援にもつながると考えられる。また、ステージ上でエンティティに対応した画像だけでなく、プロパティやアクティビティやプロセスのアイコンなども表示する（厳選して表示できるようにする）ことで、視覚的にわかりやすくしたり、そのアイコン上で入力させることで、ステージ上で作成させていくことも考えられる。

5.5.4 課題

今後は SME 用のインタフェースをプロトタイピングを繰り返しながら，ソフトウェア開発の経験が無い SME でも利用可能な環境を目指していく必要がある．また，ユーザビリティの専門家等に評価を行ってもらふことや，完成した段階でまとまった人数に対してユーザビリティ評価を行う必要もある．ITS (Intelligent Tutoring System) の分野では知識獲得に関する研究が行われているので，そのあたりの知見も参考になると考えられる．ナレッジオブジェクト間の整合性をチェックする機能も実装する必要がある．

本システムにはまだ ITT を学習する機能は搭載しておらず，理論の簡単な紹介とナレッジオブジェクトの簡単な紹介およびサンプル提示を行うヘルプの提供に留まっている．ID 理論自体を学習できるようには整備されていないため，情報を増やしていく必要がある．また，ID 理論を直接学習するというよりは，誰もが ITT を理解しながら，作業を進めていくような環境が適していると考えられる．ユーザの作業をモニタリングしながら，必要なときに必要な情報を提供して，オーサリング作業をサポートするような，EPSS (Electronic Performance Support System) のような環境としていくことも考えられる．設定されているトランザクションに応じたガイドということも考えられる．

また，第 4 章で議論した抽象のトランザクションへの対応としては，抽象のトランザクションのナレッジオブジェクトとして示した kind の入力方法は，他のナレッジオブジェクトと基本的に同じとした．エンティティ入力の際に同時にクラスを入力することも考えられたが，今回はクラス関連の登録をひとまとめにして登録させることにした．どちらの方法が良いのかは今後，検討の余地が残る．また，抽象のトランザクションにおいては，抽象の階層 (Merrill et al., 1992) を学習するということになるため，クラスの階層構造を表示することが効果的と考えられる．よって，クラスの専用エディタを用意して，クラスの階層と，そこに関連づけるエンティティやプロパティを設定できるようにする必要があるだろう．

5.6 まとめ

本章では，ITT 教材シェルのオーサリング環境について議論した．まず筆者らが学習環境のサンプル教材等を開発するための上級者用インタフェースを開発し，それをもとにして SME 用インタフェースのプロトタイプを開発した．

上級者用インタフェースについては簡単な評価を行い，上級者はきちんと作成できること，ソフトウェア開発の素養のある SME ならマニュアルがあればなんとか作成できるが，かなりの困難を要することが確認された．その評価をもとにしながら，SME 用イン

タフェースの要件について検討し，入力の簡便化としてトランザクションの種類による入力項目の厳選や，ナレッジオブジェクトをペアで入力していくようにし，ナレッジオブジェクトの構造の関係性を表示するような機能を追加した．今後は，改善を繰り返していく必要があるが，オーサリングとして最低限機能するような環境は整えることができたと言える．

第 6 章

考察

6.1 はじめに

これまでに、2 種類の AID (Automating / Automated Instructional Design) 教材シェルの開発について述べてきた。本研究における AID 教材シェルは、SME (Subject Matter Expert) が利用できることを第一とし、認知的方略の学習についても付加的な研究課題とした。本研究の成果は第 3 章から第 5 章の 2 種類の AID を志向した教材シェルの開発に集約されるが、本章では、AID 教材シェルを構築した経験に基づきながら、SME の利用や認知的方略、インストラクショナルデザイン (ID) 理論のツールへの実装、今後の展開に関して考察を試みる。

6.2 SME の利用について

本研究の主要な課題は、SME が利用できる 2 種類の AID 教材シェルの開発であり、1 つは言語情報のドリルシェル、もう 1 つは知的技能の教授トランザクション理論 (Instructional Transaction Theory ; ITT) に基づく教材シェルを対象とした。AID 教材シェルは、ID の素養のない SME が、ID 理論による教授方略を内蔵したシェルに知識を登録することによって、効果的な教材を提供できることを目指している。

ドリルシェルにおいては、SME が利用することを想定し、できるだけ簡単にドリルが作成できるように配慮した。具体的には、デフォルト設定のまま、アイテムを登録するだけで、ID 理論 (ドリル制御構造) を内蔵したドリルを生成するようにした。評価の結果からも、大学生が初見で指定されたドリルを短時間で作成できたことから、SME が利用できる AID 教材シェルを構築できたと考えている。また、これまで ID 理論が提案されながらも、活用されてこなかった資産を有効活用できたと考えられる。これは、ドリルシェルが対象とする学習課題は言語情報で、提供形態がドリルという単純な構造であるこ

とや、登録する知識が作成者にとってある程度自明であること（問題文や選択肢で構成されること）も大きな理由であったと考えられる。

ITT 教材シェルはドリルシェルよりは、かなり複雑な構造であった。特に知識はナレッジオブジェクトという一般に馴染みのない（インストラクショナルデザイナーにも馴染みがあるわけではないが）特別な表記法で表現するため、4種類のナレッジオブジェクトの理解と、実際に学習対象をナレッジオブジェクトへとモデル化することが必要となる。特に、SME が利用するために、入力作業の簡便化や、ナレッジオブジェクトの関係性の提示などの工夫を行った。先行研究と比較して、多少の制限はあるが、大幅に使いやすさは向上したと思われる。SME が作成する環境ができたとは言い切れないが、SME が利用するための土台となるシステムは開発できたと考える。本研究では言及していないが、知識獲得は ITS (Intelligent Tutoring System) の分野で研究されており (Murray, 2003)、今後 ITS の成果も参照していく必要があるだろう。

さらに、ITT 教材シェルの場合には、学習環境の複雑さの問題もあった。ITT で提供される方略が多いために、どのように提供すべきかが課題となった。教材シェルの構成上、そもそも学習環境が活用されなかったら本末転倒である。これは、ドリルシェルにも言えることであるが、ドリルは単純な構造でどのような内容かを理解している人が多いため、問題とならなかった。ITT の学習環境については、トランザクションごとに必要な方略を厳選して提示することや、何度も評価を通して作り込んでいったことによって、利用可能な状態に近づけることができた。

また、実際の開発を通して、SME が利用できる AID 教材シェルの開発はかなりの労力を要することが明らかとなった。それは、AID 教材シェルは学習環境とオーサリング環境の双方をデザインする必要があることと、SME が利用するツールを作るためには、かなりの作り込みが必要となることに起因する。複雑さを表面に出すと SME には敬遠されるツールになってしまうため、応用的な部分は隠すなど、専門性の向上の段階を踏む形で、徐々に複雑なことを可能とするような形態が有効と考えられる。ドリルシェルにおいては、アイテム登録の方法を共通問題文形式と個別問題文形式の2種類を用意したために、評価でも何人かは混乱していた。アイテムの形式が2種類あることに着目し、より汎用的にと考えて実装したものの、逆に複雑性を増してしまう要因となった。SME はシンプルなインタフェースを求めると考えられるため、思い切った機能削減（割り切り）も必要であろう。

6.3 認知的方略について

本研究においては言語情報と知的技能に対応した AID 教材シェルを開発したが、AID 教材シェルは、学習者が効果的な学習方法自体を学ぶという認知的方略の視点からも効果

が期待できると考えられ、研究課題の1つとしていた。

教材開発者にとって、AID 教材シェルのオーサリング環境は、ID 理論の学習の場としても有用である。ドリルシェルにおいては、AID 教材シェルが学習環境と一体となっていることを生かし、ID 理論自体を学習するための教材として機能することを示した。教材シェルは、学習者自らが自分の学習のために教材を作成するという考えられ、特にドリルシェルはそのような状況も視野に入れて、ドリル作成者を対象にオーサリング環境上に ID 理論自体を学習できる機会（つまり認知的方略を学習できる機会）を提供した。評価の結果、利用者は、自分でドリルを作成でき、かつ ID 理論（ドリル制御構造）を学習できることが確認され、効果的に練習を進めるための認知的方略の学習を促進する環境を構築（少なくとも方略に触れる機会を提供）できたと考えている。これはツールのグラスボックス化につながる。

特にドリルシェルでは、ID 理論（ドリル制御構造）の流れを知ってもらうために、アイテムの動きを可視化するなどの工夫を行った。可視化は、オーサリング環境と学習環境の両方をもつ教材シェルの強みである。評価としてはあまりよい結果とはならなかったが、もう少しわかりやすく作り込むことで、よりよい ID 理論の学習環境が提供できると思われる。さらにドリルシェルにおいては、アイテムの動きがわかるアニメーション教材を作成したところ好評であった。単にマニュアルを提供して説明をするだけでなく、理論を理解するための教材をセットで提供することは効果的であったと考えられる。

一方で、学習者にとっては、AID 教材シェルの学習環境の作り方によって、認知的方略を学習する機会を提供できることが考えられた。つまり、AID 教材シェルは同じ教授方略（枠組み）を持ちながら、様々なテーマを対象とした教材を生成できるため、同じ枠組みの複数の教材に触れることで、共通に提供されている教授方略に気づき、自らの学習方略とする機会となる可能性がある。これには、方略自体を整理して学習者に明示する必要がある。ITT 教材シェルは、学習環境は学習目標と方略を明示するインタフェースを学習者に提供しており、学習環境自体が認知的方略の学習につながると考えられる。今後は複数の教材を作成し、複数の教材に触れることによる認知的方略への効果を検証していく必要があるが、そのための体系的な基盤は準備できたと考えている。

6.4 ID 理論の実装について

本研究の対象とした AID ツールは、すでに提唱されている既存の ID 理論を組み込んだもの限定した。よって、AID 教材シェル構築の際には、教材シェルへ ID 理論の組み込みを行ったが、第3章で検討したドリルシェルと、第4,5章で検討した ITT に基づく教材シェルとでは多少異なった。

ITT は、そもそもが AID ツール（教材シェル）構築のための理論であるため、ID 理

論を適用することが、すなわち教材シェル開発とほぼ同義であった。つまり、この場合の ID 理論のツールへの組み込みは自然に行うことができた。これは、データ構造としてナレッジオブジェクトが決まっていたこと、アルゴリズムとして教授方略が提供されていたこと、それらがコンピュータで処理可能な内容であったことが挙げられる。Merrill and ID2 Research Team (1996) では、ITT の構成を、知識の記述的理論、方略の記述的理論、知識と方略をつなぐ処方的理論の 3 つを挙げていたが、これらの理論がすべてコンピュータでの利用を想定していたことになる。

ドリルシェルは、ITT のように AID ツール構築のための理論ではないが、ドリル制御構造（方略あるいはアルゴリズム）の理論であるため、コンピュータには組み込みやすい状態であった。しかしながら、ドリル制御構造は、これまでに複数の制御構造が提案されており、一長一短があったため、1 つの制御構造のみに統一することが不可能であった。よって、アイテム数などの、ユーザの状況に応じて、異なるドリル制御構造が適用されるようにした。また、ユーザが適用できるように選択チャートを用意したり、上級者用としてカスタマイズ機能も用意した。ただし、言語情報のみの対応であるので、知的技能については、また別のドリル制御構造や、知識の構造を検討する必要性が生じてくる（知的技能の場合は階層構造が必要である）。

両方の事例とも、学習課題の性質（ITT はトランザクション、ドリルシェルは学習成果の 5 分類）によって、適用する方略が変わってくるため、その制御を行うようにした。また、両方ともコンピュータに親和性の高い理論であるため、コンピュータに特化していない多くの ID 理論とは性質が異なると考えられる。Spector and Ohrazda (2004) は、自動化の程度が高い（人間の活動をコンピュータに置き換える）場合は、明確に定義された狭い領域でのみ機能すると述べており、ID 理論の適用範囲は広くても、学習課題の性質等で範囲を狭めることによって、AID 教材シェル化が可能となる場合もあると思われる。

6.5 AID 教材シェルの展開

本節では、本研究で取り上げた AID 教材シェル（あるいはその ID 理論）の今後の展開について考察する。

6.5.1 ITT の批判的検討

ITT は学習課題の分類がかなり細かく分類されているが、ドリルシェルと同様に、各トランザクションがガニエの学習成果の 5 分類にあてはめるときに、どうなるかを検討していく必要がある。例えば、CLASSIFY や JUDGE などは、エンティティを対象とした場合、どちらも定義された概念（知的技能）と捉えることができる。ITT のトランザクショ

ンは、ほとんどすべてのインタラクションに対応していると述べているが、どれだけの範囲に適用できるのかは、学習課題の分類を行っている他の ID 理論や、多くの事例にあてはめてみる必要がある。

また、ナレッジオブジェクトは教材の対象を記述するが、ソフトウェア開発においては一般的に UML (Unified Modeling Language) と呼ばれるモデリング技法が活用されている。ナレッジオブジェクトと UML は、アクティビティがアクティビティ図に、プロセスがステートマシン図、第 4 章の最後で議論した kind はクラス図などへの対応づけが考えられ、さらに UML は他にも多くの種類のモデル図が用意されている。こういったモデリング技法からナレッジオブジェクトを捉え直すことができる可能性もある。また、UML からナレッジオブジェクトへの変換などを行うことで、システムのことを学ぶ教材を簡単に生成できる可能性もある。

6.5.2 異なる学習理論からのアプローチ

また、学習理論の観点から、Merrienboer and Martens (2002) は、AID ツールの比較の要素として、ID 理論だけでなく、あわせて学習理論 (パラダイム) を挙げていた。鈴木 (2004) は、行動主義、認知主義、構成主義の持ち場について、学習課題の複雑さと学習者の熟達度の 2 つの軸で整理することを提案している。さらに問題解決を志向して、使える研究成果は何でも使おうとする折衷主義の重要性を指摘している。よって、こういった主義の視点からも異なったツールが多く開発されることが必要である。今回開発したツールは、比較的知識の構造化が容易である行動主義のドリルシェルと、認知主義の ITT 教材シェルとしたが、今後は、より複雑な課題を扱ったり、協調的な学習環境を生成するような構成主義的な教材シェルについても開発していく必要があるだろう。

6.5.3 AID 教材シェルの構成要素間の共有可能性

本研究では、ドリルシェルと ITT 教材シェルを開発したが、ドリル部分は API (Application Programming Interface) 化することで、他の AID 教材シェル開発に役立つと考えられる。例えば、ITT の IDENTIFY などに練習としてドリルの要素が組み込まれていたが、提案されている練習の方略は、ドリル制御構造の観点からは、まったく配慮されていないものであった。例えば、IDENTIFY の部品の名前を覚えるなどのアルゴリズムとして、ドリル制御構造 (項目間隔変動型等) の採用が考えられ、今回は実際に ITT 教材シェルの学習環境に一部を実装した。よって、ID 理論においても、規模の小さい基本的な方略を API 化することで、他のシステムへ流用出来る可能性があると考えられる。これらは、AID 教材シェル構築の負担軽減となると思われる。

6.5.4 マクロ的な方略の AID ツールとの連携の可能性

Reigeluth (1993) は, AID の機能のアプローチとして, マクロ, ミドル, マイクロという 3 種類で整理を行っていた. 本研究の AID 教材シェルは両方とも教材開発のみで, 特にコース管理機能などは無いため, マイクロと言える (ITT でも Merrill (1998) はコース管理機能を備えており, Reigeluth はミドルの位置づけにしていた). 例えば, マクロ的なツールは主にコースの作成に利用されるため, コース管理は違うツールで行い, AID 教材シェルはよりマイクロ的な利用を行うことで AID ツール間のすみ分けが可能である. 例えば, 高橋ほか (2007) はミドル的なツールで ID の課題分析図 (学習目標に到達するまでに必要な要素を明らかにした図) を用いたインタフェースを提供するが, そこにドリルの学習環境や ITT の学習環境などを載せていく可能性はある.

6.5.5 シェル上でのテンプレート提供

本研究では, 教材シェルの中身を入れ替えることで, 異なる教材を提供できるような学習環境を開発してきた. しかしながら, 例えばナレッジオブジェクトで記述したとしても, 異なるシェルを通せば, 違った環境を提供できるとも考えられる. IDXelerator はナレッジオブジェクトを用いていないが, 教授トランザクションの方略を用いており, ナレッジオブジェクトで提供可能のはずである. また, Physics Lesson も物理の仮説検証の環境であり, それも他の仮説検証の教材に利用できる可能性がある.

これは, ナレッジオブジェクトでなくても, ドリルにおいても, シェル (学習環境) を, 携帯電話へ対応させたり, ホームページ上にバナーとしてドリルを設置する (井本, 2006) など, シェルを組み替えることで, さまざまな環境でドリルを行ってもらうことが可能になる.

教材シェルは, 中身を入れ替えることで, 教材を生成するものであったが, UNIX シェルのようにユーザの好みに応じて複数のシェルを用意することも考えられる. ユーザの好みよりは, 学習するものに応じてシェルを変更すること, あるいはそういった数多くの種類に特化できるように, 画面構成を変更するなども考えられる. そういったシェルの構成をうまくコンポーネント化できれば, 適用できる教材の幅が広がると考えられる. IDLE-Tool が GBS の情報収集の学習に特化してテンプレートを開発していたことを踏まえると, ITT 教材シェル上で特定の領域に特化したテンプレートを提供するという事も考えられる.

また, Merrienboer and Martens (2002) が ID プロセスにおける協同作業の重要性を指摘していたように, 教材シェルを Web 上などに公開し, テンプレート作成環境を提供

することで、様々なテンプレートが共有できれば、より有効な AID 教材シェルになると考えられる。

6.6 まとめ

本章では、SME でも利用可能にすることを旨とした 2 種類の AID 教材シェルの開発を通して出てきた問題点等を考察した。SME が利用できるようにデザインするために、ドリルシェルと ITT 教材について出てきた課題を整理した。また、認知的方略の学習にとって AID 教材シェルは有効な形態である可能性を示した。また、AID 教材シェルの今後の展開として、ITT の批判的検討、異なる学習理論（パラダイム）からのアプローチ、モジュール化、マクロ化、テンプレート化について議論した。

第7章

結論

7.1 本研究の成果

本研究では、日本のインストラクショナルデザイナー（IDer）が少ない現状とインターネットの進展により誰もが発信者（教材作成者）となり得る状況や、AID（Automating / Automated Instructional Design）ツール研究のレビューから、SME（Subject Matter Expert）が利用できる AID ツールとして、あらかじめインストラクショナルデザイン（ID）に基づく教授方略を内蔵して知識を登録することで効果的な学習環境が生成される教材シェル（AID 教材シェル）の必要性を示した。Gagne の学習成果の分類をもとに、言語情報のドリル型の教材シェル、知的技能の教授トランザクション理論（Instructional Design Theory ; ITT）に基づいた教材シェルの 2 種類を開発することとし、SME が利用できる AID 教材シェルを開発し、その一部を公表することができた。また、AID 教材シェルは認知的方略の学習を促進する環境としても有効であるという可能性を示すことができた。本研究の成果を 3 点にまとめて以下に述べる。

成果の 1 点目は、e ラーニングの普及によって必要性は増加したが、日本には紹介されてこなかった AID ツール研究を整理して動向を明らかにしたことにある。本研究では、AID ツールのレビュー研究に着目し、どのような分類で AID ツールが整理されているかを示すことで、AID ツールの全体像を明らかにした。また、先行研究を参考に AID ツールを整理していくための、計 12 項目から成る独自の枠組みを作成した。特に AID ツールは、ID 理論に基づいていることが必須であるとの視点に立ち、枠組みには ID 理論という項目を設けた。また、著名な ID 理論に基づく異なる種類の AID ツールの 5 事例をこの枠組みにあてはめ、多様な AID ツールの特徴を明らかにした。また、AID ツール研究の整理を通して、SME が利用可能な AID 教材シェルの必要性と AID 教材シェルが認知的方略の学習を促進する可能性を示した。

成果の 2 点目は、SME でも利用可能な言語情報に対応した AID 教材シェルとして、ド

ドリルシェルを開発したことにある。これまでに ID の分野では複数のドリル制御構造が提案されてきたが、既存のツールにはドリル制御構造が組み込まれておらず、効果的な練習を支援しているとは言い難い状況であった。そこで、ドリル制御構造をシステム上に統合し、SME でも簡単にドリルを開発できる統合型ドリルシェル「ドリル工房」を開発した。ドリル制御構造は複数提案されていたため、状況に応じて最適なドリル制御構造を適用できるようにした。また、ドリル制御構造を作成者が学習できるようにし、可視化モードとしてドリル実行中にアイテムの状態遷移がわかるようにした。さらにドリル制御構造を制御要素の組み合わせとして整理することで、各ドリル制御構造の違いを明確にしてわかりやすくするだけでなく、制御構造自体のカスタマイズも可能にした。評価の結果、ドリルは簡単に作成できたこと、ドリル制御構造を学習できたことは確認された。また、他の学習課題への適用の可能性を検討するために、知的技能への対応も検討し、下位ドリル群のプロトタイプ開発を試みて、動作することを確認した。この開発を通して、AID 教材シェルが、ID 理論を知らない人でも効果的な教材を生成できること、逆に ID 理論を知りたい人には効果的な学習環境となることを示すことができた。学習者自らがドリルを作成し、ID 理論に触れることによる認知的方略の学習効果も期待される。ドリルシェルは Web 上での公開も開始した。<http://ichi.et.soft.iwate-pu.ac.jp/df/>

成果の3点目は、SME でも利用可能な知的技能に対応した AID 教材シェルとして、ITT に基づいた教材シェルの開発したことにある。ITT は AID 研究の中核的理論であるが研究途上にあり、ITT の研究を進めることは、AID ツール研究にとっても意義のあることと考えられた。ITT のシステムへの実装は、構成要素のトランザクションを中心に、一部しか行われてこなかった。学習環境は、シミュレーション環境とし、構成要素のトランザクションのである IDENTIFY (部品の名前や位置を同定できる)、EXECUTE (手続きを実行できる)、INTERPRET (なぜそうなるのかを説明できる) のすべてを実装した。特にこれまで検討段階に留まっていた INTERPRET の練習については、予測とトラブルシューティングの設計と実装を行った。また、提案されている方略をすべて実装するとメニュー項目が多く学習者が混乱するとの判断から、トランザクションの種類によってメニュー表示を厳選するよう工夫した。これは、方略をわかりやすく学習者に提示することにより、学習者が同じ枠組み(方略)で違う教材に触れることによって、認知的方略の学習を行うことも意図した。形成的評価の結果、学習環境を利用して学習効果があることは示唆された。また、さまざまなテーマに適用できることも確認した。さらに抽象のトランザクションの JUDGE (順位付けができる) と CLASSIFY (分類できる) について、設計と簡易なインタフェースによるプロトタイプ実装を行い、動作することを確認した。

また、オーサリング環境については、上級者用のインタフェースを開発後に、そのプロトタイプ評価の結果を踏まえながら、SME 用のインタフェースを開発した。入力の煩雑さを少なくするために、特に入力の行き来が多いエンティティとプロパティ、アクティビ

ティとプロセスの各ナレッジオブジェクトのペアについて、同時に入力する方法を提案した。またトランザクションによって、特に IDENTIFY のみの教材において、入力項目を厳選する機能を設けた。さらに、本研究で実装した学習環境はシミュレーションの構築が中心であるため、PEAnet 構造と呼ばれるナレッジオブジェクト間の関係性がわかるように、リンクを表示する機能を設けた。これによって SME が利用できるようにしていくための基盤となる環境を整えることができた。

7.2 今後の課題

本研究においては、SME でも利用できるような AID 教材シェルの開発を目的としてきた。ドリルシェルにおいては、すでに活用の段階にあるため、実際の e ラーニングのコンテンツ開発場面で活用してもらうことを通して、実用に耐えうるかを確認しながら、さらに作り込みを行っていく必要がある。ITT 教材シェルについては、学習環境は複数の教材を作成し、それらを利用していくことによる認知的方略への効果を確認するための評価を実施していく必要がある。オーサリング環境については、SME が利用できるようにプロトタイピングを繰り返しながら、完成度を高めていく必要がある。また、ITT は Gagne の学習成果の分類と比べるとかなり細かく学習課題を分類しており、両者を比較しながら、ITT 自体の批判的検討を行っていく必要もあるだろう。また、ナレッジオブジェクトはソフトウェア開発に利用されるモデリング手法である UML (Unified Modeling Language) と類似している部分もあるため、UML の視点からナレッジオブジェクトを再検討していくことも考えられる。

付録 A

ドリル制御構造

Salisbury (1988) と Alessi and Trollip (2001) によるドリル制御構造の日本語訳 (筆者による) を以下に載せておく。

- 2 プール型 (図 A.1)
- 3 プール型 (図 A.2)
- 復習機会増加の 3 プール型 (図 A.3)
- 状態前進型 (図 A.4)
- 項目間隔変動型 (図 A.5)

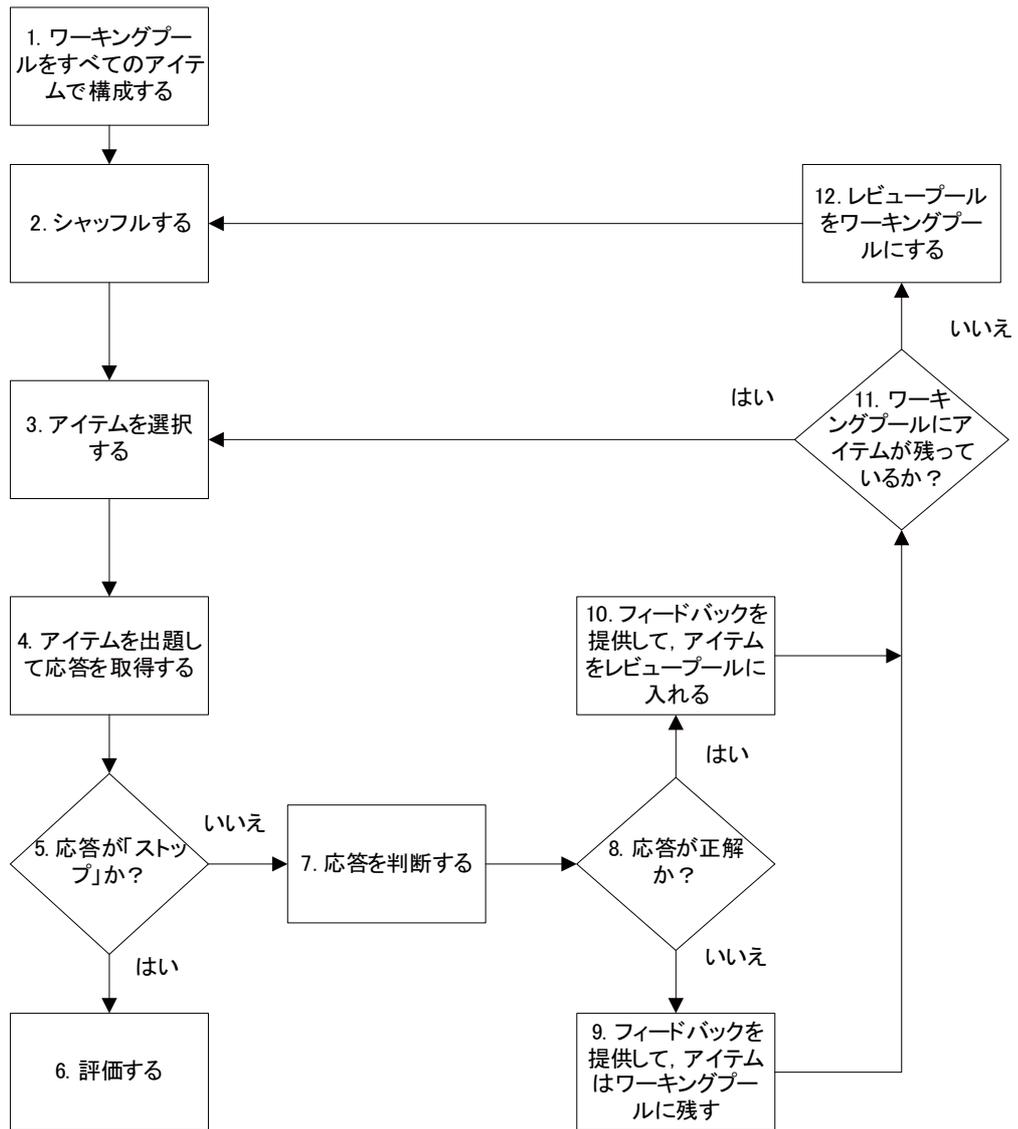


図 A.1 2 プール型 (Salisbury, 1988, fig.4.4, p.109 を日本語訳したもの)

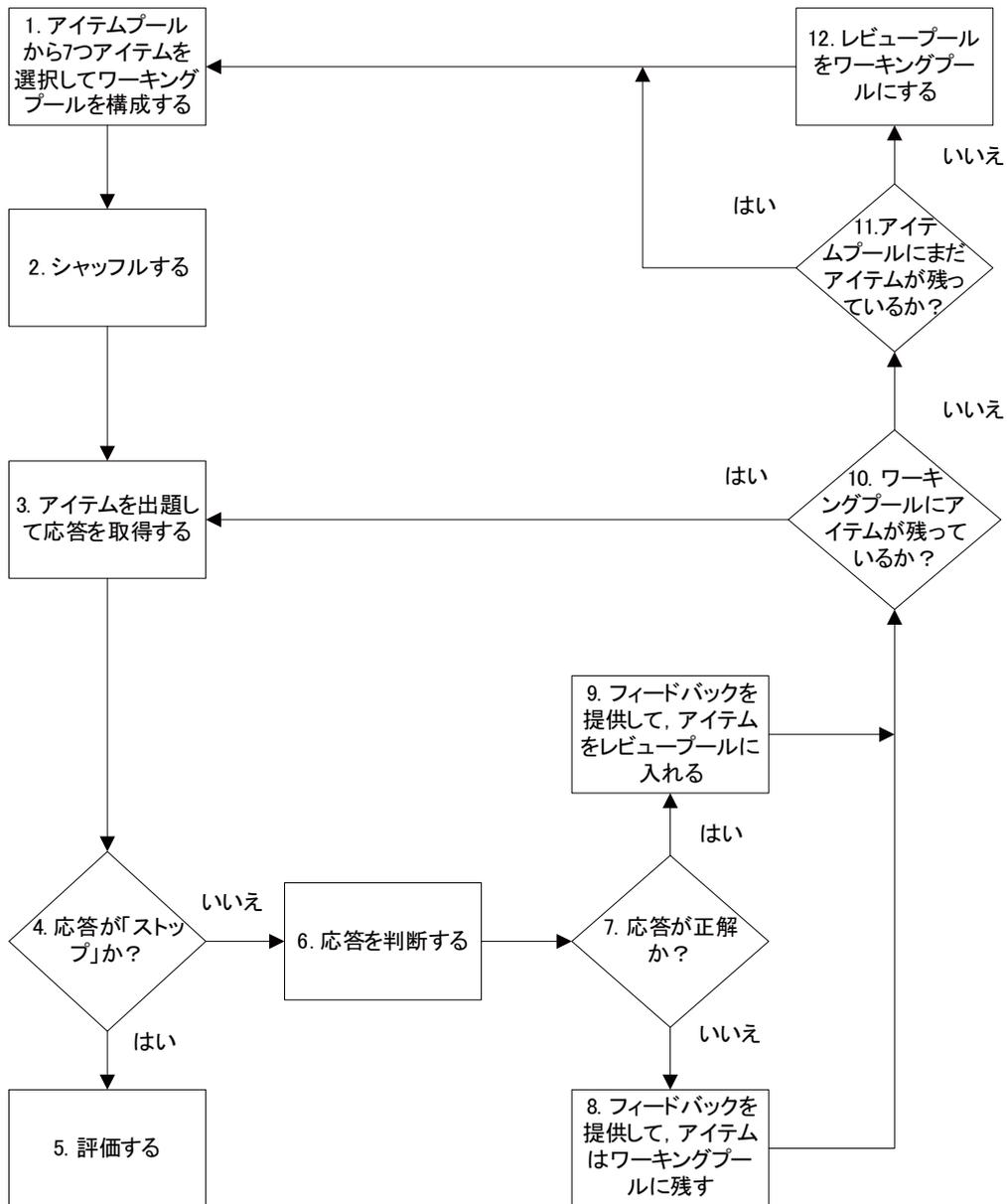


図 A.2 3 プール型 (Salisbury, 1988, fig.4.5, p.111 を日本語訳したもの)

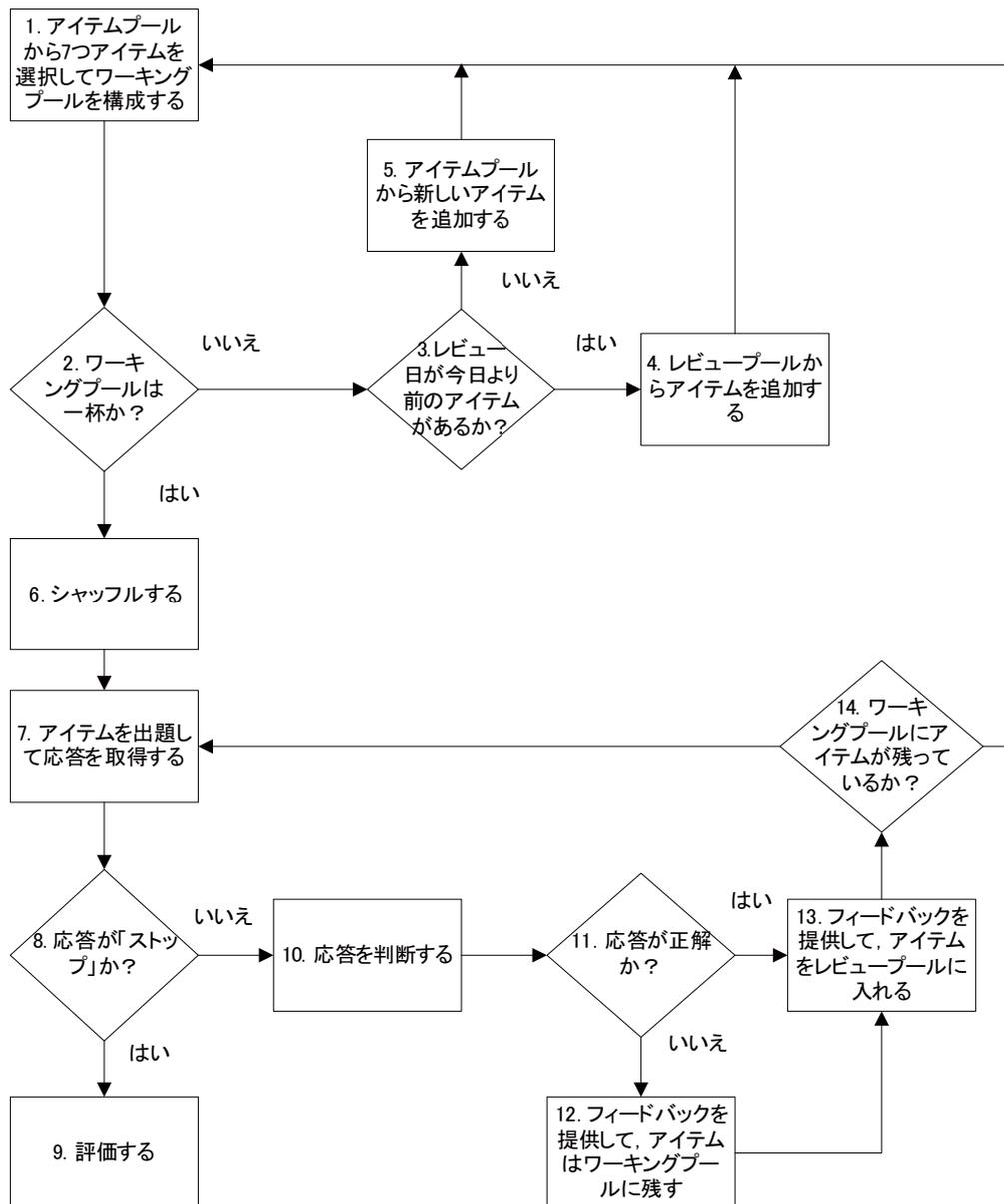


図 A.3 復習機会増加の3プール型 (Salisbury, 1988, fig.4.6, p.113 を日本語訳したもの)

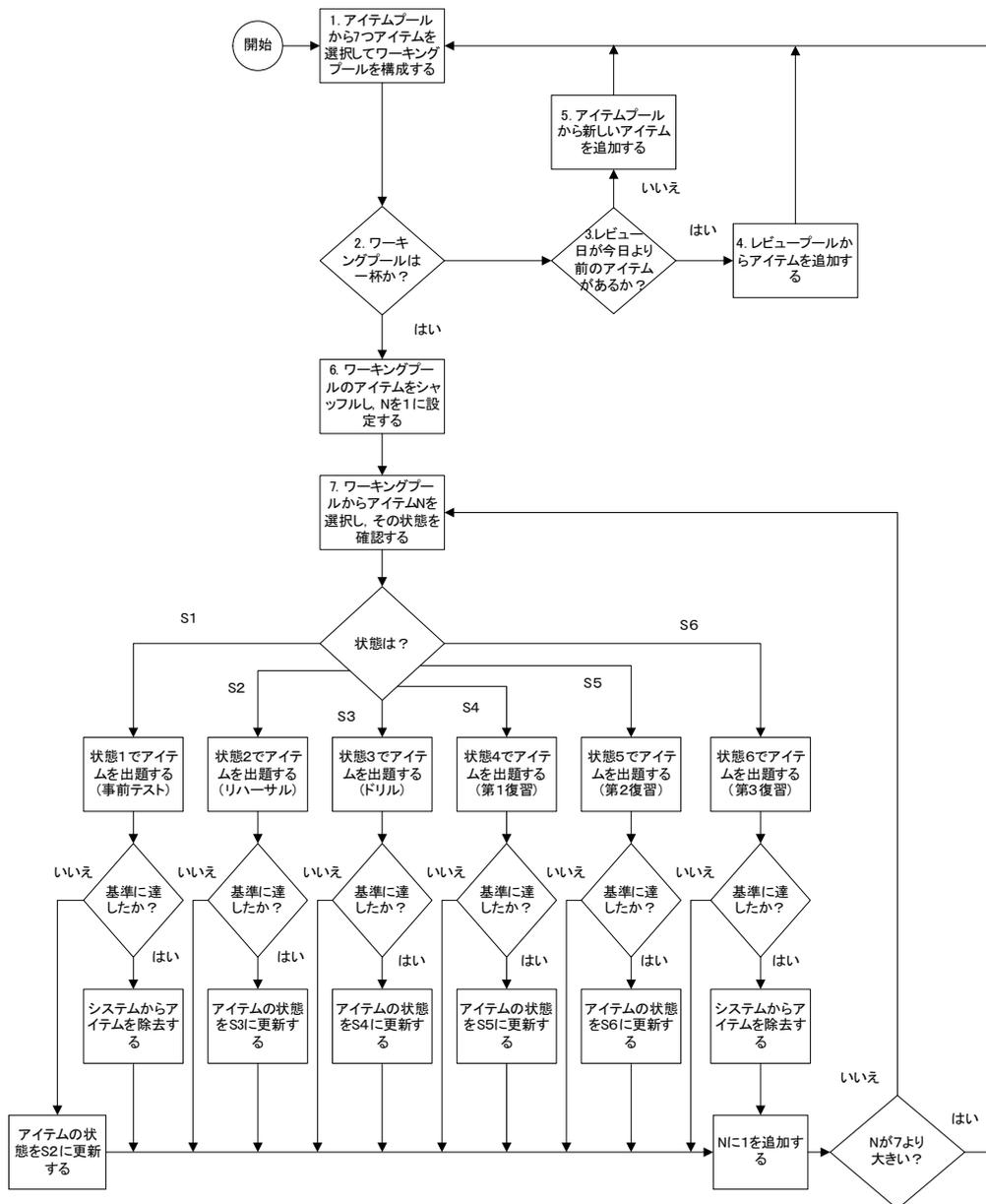


図 A.4 状態前進型 (Salisbury, 1988, fig.4.7, p.115 を日本語訳したもの)

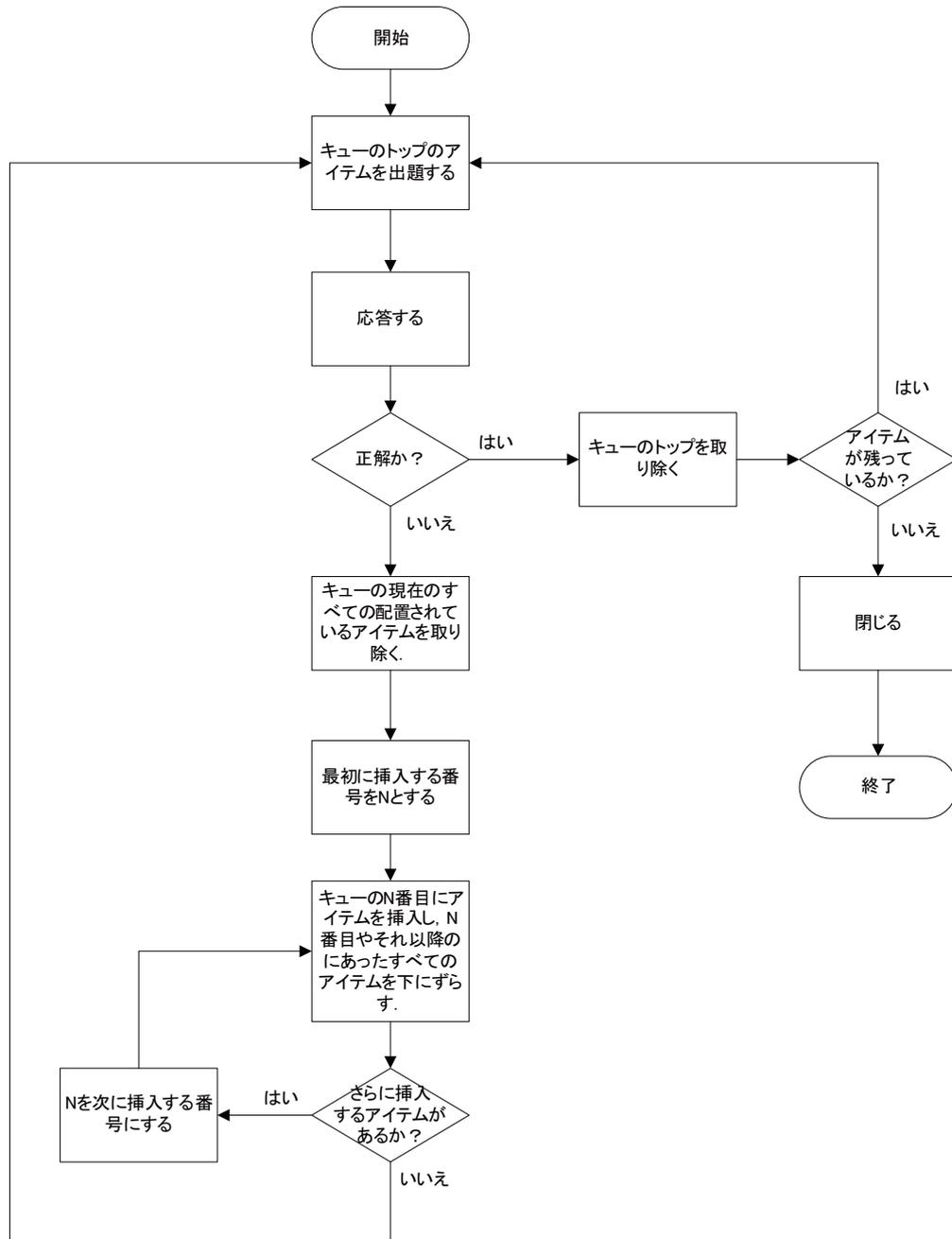


図 A.5 項目間隔変動型 (Alessi and Trollip, 2001, fig.6.4, p.195 を日本語訳したもの)

付録 B

教授トランザクション理論の展開

教授トランザクション理論 (Instructional Transaction Theory; ITT) に関わる展開はかなり複雑であるが、ITT に関わるこれまでの研究の整理を試みる。Merrill のこれまでの貢献を概観した論文 (Young and Reeves, 2007) においては、Merrill の主要な貢献は、画面構成理論 (Component Display Theory; CDT)、精緻化理論 (Elaboration Theory)、ITT、ID 第一原理 (First Principles of Instruction)、Pebble-in-the-Pond モデルの 5 つが挙げられており、ITT は Merrill の代表的な研究の 1 つとして位置づけられている。

ITT という名称が最初に論文として紹介されたのは、Merrill et al. (1991) である。ITT は、Merrill (1983) による画面構成理論 (Component Display Theory; CDT) と Gagne (1985) の学習の条件を発展した ID 理論である (Merrill and ID2 Research Team, 1996)。CDT は、行動 (performance) の種類と内容 (content) の種類のマトリックス (図 B.1) に基づき、このセルのどこに学習目標が当てはまるかを確認できれば、それに沿った方略が提供されたとする ID 理論である。

CDT から ITT への教授モデルの発展は、学習課題の性質を見極めて、それぞれの課題習得に効果的な条件を詳細に分析することで、教授環境を自動的に構築するモデルを模索

LEVEL OF PERFORMANCE (行動のレベル)	FIND (見つける)				
	USE (使う)				
	REMEMBER (覚える)				
		FACT (事実)	CONCEPT (概念)	PROCEDURE (手続き)	PRINCIPLE (原理)
		TYPE OF CONTENT (内容の種類)			

図 B.1 Component Display Theory の Performance-Content マトリックス (Merrill, 1983, に翻訳を付記)

するものである(鈴木, 2005b)。Merrill (1999) は ITT を CDT の精度 (precision) を高める試みであるとし, ID の自動化としてコンピュータに実装するためには CDT では不十分であり, 知識表現, 教授方略, ID の処方をより正確に記述する必要があったとしている。また, CDT は分析志向であり, 異なる種類のゴールに対する教授方略のコンポーネントを重視したが, ITT は統合志向であり, 教授トランザクションの中にこれらのコンポーネントを統合することを重視している。CDT と ITT に関して, 完全に一致するとはまではいかないが, IDENTIFY が事実, EXECUTE が手続き, INTERPRET が原理に対応づけられるとしている。

また, CDT に基づいて開発された TICCIT を開発した後の Merrill (1985) や Merrill (1987c) の研究あたりから, オーサリングシステム関連の研究が進められ, さらに 1987 年あたりから ID のエキスパートシステムの開発 (Merrill, 1987a) など, AID 関連研究を中心に行うようになった。この頃の内容を CDT を拡張した Component Design Theory (Merrill, 1987b) と呼ぶ場合もある (あるいは CDT2 や, 画面構成理論を小文字の cdt として大文字の CDT を用いる場合もある)。これらの延長線上に ITT が提案されている。Merrill (1994, p.327 脚注) においては, この Merrill (1985) や Merrill (1987c) の論文が ITT の土台になっていると述べている。

ITT が提案された 1990 年にはまだナレッジオブジェクトという用語は存在せず, Jones et al. (1990) によって提案された知識フレームの 3 分類 (エンティティ, アクティビティ, プロセス) に対して属性 (attributes), 構成要素 (components), 抽象 (abstractions), 連合 (associations) という 4 種類の精緻化を行う EFN (Elaborated Frame Network) と呼ばれる知識ベースの利用を想定していた。Merrill et al. (1992) になると PEAnet と呼ばれるエンティティとアクティビティとプロセスおよびプロパティを関係づけて, シミュレーションで利用するという取り組みに発展し, プロパティがはじめて追加された (それまでは attributes であった)。この Merrill et al. (1992) が現在のナレッジオブジェクトの原型と言える。ナレッジオブジェクトという用語がはじめて登場したのは, Merrill and ID2 Research Team (1996) である。ナレッジオブジェクトは Merrill の近年の研究で継続的に利用されている。

教授トランザクションという用語が登場したのは, 1987 年の 3 つの論文からである (Merrill, 1987a,b,c)。教授トランザクションは, 学習者とコンピュータ間のインタラクションに使われていた言葉である。

近年の Merrill の論文には教授トランザクションという用語はほとんど登場しておらず, Merrill (2001a) から, Components (Components of Knowledge Object, Components of Instructional Strategy あるいは Knowledge Components, Strategy Components) という用語を使うようになっている。Merrill (2001a) では, 「教授トランザクションが, この論文で同定された知識と方略のコンポーネントでより正確に記述される。教授トランザ

クションは、方略コンポーネントと知識コンポーネントの適切な組合せを必要とする。教授トランザクションは、与えられた教授ゴールに適切な方略コンポーネントと知識コンポーネントの組合せによって定義される」と述べている。一方で、それ以降の論文では、これらのコンポーネントを用いる場合の文脈は、コンピュータ利用に限ってはいない場合が多い。Merrill (1999) では、ITT はコンピュータだけでなく他の伝達モデルのための教授をデザインするためにも利用できると述べているが、トランザクションはもともと人間とコンピュータの間のインタラクションに用いてる言葉であり、これらは、特にコンピュータ利用と限ってはいないため、トランザクションという言葉を利用しなくなってきたと考えられる。もちろん ITT の考え方はその後の Merrill の研究にも継承されている。トランザクションクラスの観点からは、Kind あるいは Kinds-of と呼ばれる知識および方略のコンポーネントが提案されており (Merrill, 2001a, 2002), これは Merrill et al. (1992); Merrill (1999) の CLASSIFY や GENERALIZE に相当する。

ITT は Merrill and ID2 Research Team (1996) の冒頭において、進行中の研究であるため過去の研究と矛盾している内容もあると自ら指摘しているように、常に進化系の研究である。Merrill (1999) が ITT の内容を示す文献としては最後である。これまでの展開が示すように、ITT の理論や関連研究は常に変化している状態にあり、用語や手法も混在しているため、どこかの時点の研究を見定めて、着手する必要があった。ITT 研究は関連研究も含めると数多くあるが、Merrill (1999) は、ID 理論としてそれまでの研究の集大成として整理されている論文であった。また、その論文が載っている文献はグリーンブックと呼ばれる Reigeluth が編集している ID 理論の著名な文献の第 2 作目でもあり、信頼性が高く、かつ ITT という名称では最も新しい内容であるとして、基本的にこの内容に従って開発することにした。また、ITT では提案が不足している部分については、他の研究を参照にすることにした。特にナレッジオブジェクトの具体的な記述方法については、Merrill (1999) ではかなり不足していた。なお、ITT の構想や研究を捉える意味で整理している論文は Merrill and ID2 Research Team (1996) であり、Merrill (1999) と相互補完の関係にある。本研究の基幹論文もこの 2 つである。

付録 C

ナレッジオブジェクトの XML 仕様

ナレッジオブジェクトの XML 仕様を以下に載せる。

- ナレッジオブジェクトの XML 仕様 (表 C.1)
- エンティティの XML 仕様 (表 C.2)
- プロパティの XML 仕様 (表 C.3)
- アクティビティの XML 仕様 (表 C.5)
- プロセスの XML 仕様 (表 C.4)

XML 仕様 (表) を参照する上での注意事項：

- 名前の@は属性を表す。
- 回数の*は必須項目を表す。

表 C.1 ナレッジオブジェクトの XML 仕様

項番	名前	説明	回数	備考
0	ko	ナレッジオブジェクトを記述する (ルート)。	1*	
1	entity	エンティティを記述する。	0 以上	表 C.2 に詳細を記載
2	property	プロパティを記述する。	0 以上	表 C.3 に詳細を記載
3	activity	アクティビティを記述する。	0 以上	表 C.5 に詳細を記載
4	process	プロセスを記述する。	0 以上	表 C.4 に詳細を記載

表 C.2 エンティティの XML 仕様

項番	名前	説明	回数	データ形式
1	entity	エンティティの情報を記述する .	1*	
1-a	@id	通し番号	1*	整数
1-b	@visible	表示するかどうか	1*	ブール値
1-c	@enable	利用するかどうか	1*	ブール値
1-1	name	名前	1*	文字列
1-2	description	解説を記述する .	0,1	
1-2-1	text	解説文	1	文字列
1-2-2	resource	解説に用いる画像等のファイル名	0,1	文字列
1-2-2-a	@type	リソースの形式 (image / audio / video)	1	固定文字列
1-3	partof	所持するエンティティへのポインタ (id)	0,1	整数
1-4	haspart	部品となるエンティティを記述する .	0,1	
1-4-1	part	部品となるエンティティへのポインタ (id)	1 以上	整数
1-5	properties	保持するプロパティを記述する .	0,1	
1-5-1	property	プロパティの値	1 以上	文字列か整数
1-5-1-a	@id	プロパティへのポインタ (id)	1	整数
1-5-1-b	@default	プロパティのデフォルト値	1	文字列か整数
1-6	portrayals	画面上へ表示する際のリソースを記述する .	0 以上	
1-6-1	resource	関連づけるリソースへのポインタ (id)	1 以上*	整数
1-6-1-a	@x	リソースの X 座標	0,1	整数
1-6-1-b	@y	リソースの Y 座標	0,1	整数
1-6-1-c	@w	リソースの幅	0,1	整数
1-6-1-d	@h	リソースの高さ	0,1	整数
1-6-1-f	@property	プロパティへのポインタ	0,1	整数
1-6-1-g	@value	プロパティの値 (この時に表示)	0,1	整数

表 C.3 プロパティの XML 仕様

項番	名前	説明	回数	データ形式
2	property	プロパティの情報を記述する .	1*	
2-a	@id	通し番号	1*	整数
2-1	name	名前	1*	文字列
2-2	description	解説を記述する .	0,1	
2-2-1	text	解説文	1	文字列
2-2-2	resource	解説に用いる画像等のファイル名	0,1	文字列
2-2-2-a	@type	リソースの形式 (image / audio / video)	1	固定文字列
2-3	valuelist	値が文字列の場合に利用	0,1	
2-3-1	value	取り得る値	1 以上	文字列
2-4	valuenum	値が数値の場合に利用	0,1	
2-4-1	@max	最大値	0,1	実数
2-4-2	@min	最小値	0,1	実数
2-4-3	@interval	間隔	0,1	実数

表 C.4 プロセスの XML 仕様

項番	名前	説明	回数	データ形式
3	process	プロセスの情報を記述する .	1*	
3-a	@id	通し番号	1*	整数
3-1	name	名前	1*	文字列
3-2	description	解説を記述する .	0,1	
3-2-1	text	解説文	1	文字列
3-2-2	resource	解説に用いる画像等のファイル名	0,1	文字列
3-2-2-a	@type	リソースの形式 (image / audio / video)	1	固定文字列
3-3	consequences	結果を記述する	1	
3-3-1	consequence	取り得る値	1 以上	文字列
3-3-1-a	@target	エンティティへのポインタ	1	整数
3-3-1-b	@property	プロパティへのポインタ	1	整数
3-3-1-c	@value	プロパティ値	1	文字列か整数
3-3-1-1	condition	プロパティ値	0 以上	文字列か実数
3-3-1-1-a	@target	エンティティへのポインタ	1	整数
3-3-1-1-b	@property	プロパティへのポインタ	1	整数
3-3-1-1-c	@value	プロパティ値	1	文字列か実数
3-4	trigger	プロセスへのポインタ	0,1	整数

表 C.5 アクティビティの XML 仕様

項番	名前	説明	回数	データ形式
4	activity	アクティビティの情報を記述する .	1*	
4-a	@id	通し番号	1*	整数
4-1	name	名前	1*	文字列
4-2	description	解説を記述する .	0,1	
4-2-1	text	解説文	1	文字列
4-2-2	resource	解説に用いる画像等のファイル名	0,1	文字列
4-2-2-a	@type	リソースの形式 (image / audio / video)	1	固定文字列
4-3	trigger	プロセスへのポインタ	0,1	整数

謝辞

本論文の作成にあたり，お世話になった方々にここで感謝を述べます。

まず本研究を遂行する上で，多大なるご指導を頂いた熊本大学大学院社会文化科学研究科の鈴木克明先生に深く感謝申し上げます。主指導教員をお引き受け頂き，社会人学生として遠隔地からの在籍ではございましたが，ご自宅やリエゾンオフィスでのゼミなど，場所を問わずご指導を頂いたことに，重ねて深謝いたします。インストラクショナルデザインという分野との出会いを作って頂いたのも先生ですし，先生無しに本研究の実現はありませんでした。

同大学院の中野裕司先生には，1年次から副指導教員としてご指導頂き，短期修了に関する申請時なども含めて，さまざまな面でご協力頂きました。深く感謝申し上げます。

同大学院の喜多敏博先生，大森不二雄先生，山下徹先生には審査委員会に加わって頂き，本研究のご指導，ご助言を頂きました。皆様のおかげで論文としてまとめることができました。深く感謝申し上げます。また，根本淳子先生や松葉龍一先生をはじめとする教授システム学専攻の教員の皆様および大学院生の皆様には，本研究関連で学会発表した際や熊本に足を運んだ際に，暖かく迎え入れて頂き，かつ皆様の普段の研究教育活動を目にして，頑張ろうという意識が高まりました。熊本大学大学院社会文化科学研究科の井ノ上憲司君は同じ専攻に在籍するという立場から，普段熊本大学に行けない私のサポートをして頂いたり，遠隔授業の準備などを協力頂きました。ありがとうございました。

株式会社シエンの高橋暁子さんには，システムの開発や評価の面でいろいろとお世話になりました。特にドリルシェルの開発に関しましては，たびたびの仕様変更で大変だったと存じますが，多大なる協力を頂きましたこと，感謝申し上げます。

岩手県立大学ソフトウェア情報学部の阿部昭博教授には，本研究を推進する上で，研究がしやすいように職場の環境を整えて頂いたり，気にかけて頂いたり，支えとなって頂きました。阿部教授のご協力がなければ，短期修了はできなかったと思います。深く感謝申し上げます。また，同学部の藤原康宏先生には，評価にご協力頂きました。ありがとうございました。

同学部の宮澤芳光君，高橋恭平君には，ITTに基づく教材シェルの開発を親身になって

手伝って頂きました。お二人の尽力があって、ITT 教材シェルが形となりました。ありがとうございました。同学部阿部昭博研究室に所属する大学院生と学部生の皆様にも、評価の協力者として参加して頂きました。また、本研究の成果のいくつかは、卒業生の佐藤晶一君、鎌田奉訓君、菊池陵君への指導の過程で得られたことも含まれています。感謝申し上げます。

最後に、本研究を生活面で支えてくれた家族、みゆき、維十と両親に感謝し、本論文の謝辞と致します。

発表論文

論文

- 市川尚・高橋暁子・鈴木克明 (2008.10) 複数の制御構造の適用と学習のための統合型ドリルシェル「ドリル工房」の開発．日本教育工学会論文誌，32(2)：157-168
- 市川尚・鈴木克明 (2008.3) インストラクショナルデザイン自動化ツールの研究動向．教育メディア研究，14 (2) : 33 - 44

国際会議

- Ichikawa, H., Kikuchi, R., & Suzuki, K. (2007.9) Development of an instructional material and deliver system based on Instructional Transaction Theory. A paper presented at the Korea-Japan Joint International Conference (5th): Learning media and technology for future education and training, Pusang, Korea, .(Prceedings, 29-35).
- Ichikawa, H., Takahashi, A., Sato, S., Kamata, T., & Suzuki, K. (2007.7). Development of an Integrated Drill Shell “ Drill-Factory ”. A paper presented at 8th International Conference on Information Technology Based Higher Education and Training (ITHET 2007), Kumamoto, JAPAN 13A3-2 (Paper No.134)

国内発表

- 市川尚，鈴木克明 (2008.10) 教授トランザクション理論に基づく教材シェル：抽象トランザクションの検討～判断と分類を中心に～．教育システム情報学会第 33 回全国大会講演論文集：98-99
- 市川尚，高橋暁子，鈴木克明 (2007.12) 統合型ドリルシェル「ドリル工房」の知的技能への対応．日本教育工学会研究報告：195-200

- 市川尚，鈴木克明（2007.12）AID ツール関連研究のレビュー（2）. 日本教育メディア学会研究報告：27-32
- 市川尚，鈴木克明（2007.9）教授トランザクションに基づく教材シエルのオーサリング環境の開発．教育システム情報学会第32回全国大会講演論文集：196-197
- 市川尚，鈴木克明（2007.9）AID ツールに関する研究のレビュー（1）. 日本教育工学会第23回全国大会講演論文集：719-720
- 市川尚，鈴木克明（2007.5）教授トランザクション理論に基づく教材シエルの実行環境の開発．教育システム情報学会研究報告，22(1)：31-36

参考文献

- Alessi, S. M. and Trollip, S. R. (2001) *Multimedia for learning*, Massachusetts: Allyn and Bacon.
- Bell, B (1998) Investigate and decide learning environments: Specializing task models for authoring tools design. *Journal of the Learning Sciences*, Vol. 7, No. 1, pp. 65-105.
- Clark, R. C. and Mayer, R. E. (2007) *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*: Pfeiffer & Co, 2nd edition.
- de Croock, M. B. M., Paas, F. G. W. C., Schlanbusch, H. and van Merriënboer, J. J. G. (2002) ADAPT-IT: Instructional Design (ID) tools for training design and evaluation. *Educational. Technology Research and Development*, Vol. 50, No. 4, pp. 47-58.
- Drake, D. L. (1997) Design and Development of a Computer-based Simulation Authoring System for Problem-Solving Instruction. Ph.D. dissertation, Utah State University.
- Gagne, R. M. (1985) *The Conditions of learning and theory of instruction (4th ed.)*, New York: Holt, Rinehart and Winston.
- Gagne, R. and Medsker, K. (1996) *The conditions of learning: Training applications*: Harcourt Brace/ASTD.
- Gayeski, D. M. (1990) Are you ready for automated design?" *Training and Development Journal*, Vol. 27, No. 10, pp. 61-62.
- Goodyear (1997) Instructional Design Environments: Methods and Tools for the Design of Complex Instructional Systems. in Dijkstra, S., N. M. Seel, F. Schott, and R. D. Tennyson eds. *Instructional Design Volume 2*: Lawrence Erlbaum Associates, Inc.
- Gros, B. and Spector, M. (1994) Evaluating automated instructional design systems:

- A complex problem. *Educational Technology*, Vol. 34, No. 5, pp. 37-46.
- Gustafson, K. (2002) Instructional Design Tools : A Critique and Projections for the Future. *Educational Technology Research and Development*, Vol. 50, No. 4, pp. 59-66.
- Half Baked Software Inc. (2007) Hot Potatoes Version 6.2.4.0. <http://hotpot.uvic.ca/>.
- Half, H. M. (1993) Prospects for Automating Instructional Design. in Spector, J. M., C. M. Polson, and J. D. Muraida eds. *Automating Instructional Design: Concepts and Issues*, pp. 67-131.
- Half, H. M., Hsieh, P. Y., Wenzel, B. M., Chudanov, T. J., Dirnberger, M. T., Gibson, E. G. and Redfield, C. L. (2003) Requiem for a development system: Reflections on knowledge-based generative instruction. in Murray, T., S. Blessing, and S. Ainsworth eds. *Authoring tools for advanced technology learning systems*, pp. 33-60, Amsterdam, The Netherlands: Kluwer Academic Publishers.
- Ichikawa, H., Kikuchi, R. and Suzuki, K. (2007) Development of an instructional material and deliver system based on Instructional Transaction Theory. in *A paper presented at the Korea-Japan Joint International Conference (5th): Learning media and technology for future education and training*, pp. 29-35, Pusang, Korea.
- IEEE (2007) The Learning Object Metadata standard. <http://www.ieeeeltsc.org/working-groups/wg12LOM/lomDescription/>.
- Jones, M. K., Li, Z. and Merrill, M. D. (1990) Domain knowledge representation for instructional analysis. *Educational Technology*, Vol. 30, No. 10, pp. 7-32.
- Jones, K., Li, Z. and Merrill, M. D. (1992) Rapid prototyping in automated instructional design. *Educational Technology Research and Development*, Vol. 40, No. 4, pp. 95-100.
- Kasowitz, A. (1998) Tools for Automating Instructional Design. in *ERIC Digest(ED420304)*.
- Keller, J. M. (1987) Development and use of the ARCS model of motivational design. *Journal of Instructional Development*, Vol. 10, No. 3, pp. 2-10.
- van Merriënboer, J. J. G. and Martens, R. (2002) Computer-based tools for instructional design: An introduction to special issue. *Educational Technology Research and Development*, Vol. 50, No. 4, pp. 5-9.
- van Merriënboer, J. J. G., Clark, R. E. and De Croock, M. B. M. (2002) Blueprints for complex learning: The 4C/ID-model. *Educational Technology Research and Development*, Vol. 50, No. 2, pp. 39-64.

-
- Merrill, M. D. (1983) Component Display Theory. in Reigeluth, M. C. ed. *Instructional design Theories and Models: An Overview of Their Current Status*, pp. 279-333, Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Merrill, M. D. (1985) Where Is the Authoring in Authoring System?" *Journal of Computer-Based Instruction*, Vol. 12, No. 4, pp. 90-96.
- Merrill, M. D. (1987a) An Expert System for Instructional Design. *IEEE Expert*, Vol. 2, No. 2, pp. 25-37.
- Merrill, M. D. (1987b) The New Component Design Theory: Instructional Design for Courseware Authoring. *Instructional Science*, Vol. 16, No. 1, pp. 19-34.
- Merrill, M. D. (1987c) Prescriptions for an Authoring System. *Journal of Computer-Based Instruction*, Vol. 14, No. 1, pp. 1-8.
- Merrill, M. D. (1994) *Instructional Design Theory*: Educational Technology Publications, Inc.
- Merrill, M. D. (1997) Learning-oriented instructional development tools. *Performance Improvement*, Vol. 36, No. 5, pp. 51-55.
- Merrill, M. D. (1998) ID Expert: A Second Generation Instructional Development System. *Instructional Science*, Vol. 26, No. 3-4, pp. 243-262.
- Merrill, M. D. (1999) Instructional Transaction Theory (ITT): Instructional Design based on Knowledge Objects. in Reigeluth, M. C. ed. *Instructional Design Theory and Models Vol. II: A New Paradigm of Instructional Theory.*, pp. 397-424, Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Merrill, M. D. (2001a) Components of Instruction toward a Theoretical Tool for Instructional Design. *Instructional Science*, Vol. 29, No. 4-5, pp. 291-310.
- Merrill, M. D. (2001b) A knowledge object and mental model approach to a physics lesson. *Educational Technology*, Vol. 41, No. 1, pp. 36-47.
- Merrill, M. D. (2002) Knowledge objects and mental models. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects*, pp. 261-280.
- Merrill, M. D. (2003) Using Knowledge Object to Design Instructional Learning Environments. in Murray, T., S. Blessing, and S. Ainsworth eds. *Authoring Tools for Advanced Technology Learning Environments*, pp. 181-204, Amsterdam, The Netherlands: Kluwer Academic Publishers.
- Merrill, M. D. and ID2 Research Team (1996) Instructional Transaction Theory: Instructional Design based on Knowledge Objects. *Educational Technology*, Vol. 36, No. 3, pp. 30-37.
- Merrill, P. F. and Salisbury, D. F. (1984) Research on Drill and Practice Strategies.

- Journal of Computer-Based Instruction*, Vol. 11, No. 1, pp. 19-21.
- Merrill, M. D. and Thompson, M. B. (1999) The IDXelerator: Learning-Centered Instructional Design. In Jan van den Akker, Robert Maribe Branch, Kent Gustafson, Nienke Nieveen and Tjeerd Plomp (Ed.) *Design Approaches and Tools in Education and Training*, pp. 265-277. Kluwer Academic Publishers.
- Merrill, M. D., Li, Z. and Jones, M. K. (1991) Instructional Transaction Theory: An Introduction. *Educational Technology*, Vol. 31, No. 6, pp. 7-12.
- Merrill, M. D., Jones, M. K. and Li, Z. (1992) Instructional Transaction Theory: Classes of Transactions. *Educational Technology*, Vol. 32, No. 6, pp. 12-26.
- Monshinskie, J. F. (1998) A Survey of Multimedia Developers Concerning the Use of Automated Instructional Design Software. *Journal of Instructional Delivery Systems*, Vol. Spring, pp. 26-32.
- Murray, T. (1999) Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, Vol. 10, pp. 98-129.
- Murray, T. (2003) An Overview of Intelligent Tutoring System Authoring Tools: Updated Analysis of the State of the Art. in Murray, T., S. Blessing, and S. Ainsworth eds. *Authoring Tools for Advanced Technology Learning Environments*, pp. 491-544, Amsterdam, The Netherlands: Kluwer Academic Publishers.
- Nieveen, N. and Gustafson, K. (1999) Characteristics of computer-based tools for education and training development: An introduction. in van den Akker, J., R. Branch, K. Gustafson, N. Nieveen, and T. Plomp eds. *Design approaches and tools in education and training*: Kluwer Academic Publishers.
- Piskurich, G. (2006) *Rapid Instructional Design: Learning ID Fast and Right (2nd ed.)*, San Francisco: Pfeiffer.
- Reigeluth, C. M. (1993) Functions of an Automated Instructional Design System. in Spector, J. M., C. M. Polson, and J. D. Muraida eds. *Automating Instructional Design: Concept and Issues*, pp. 43-66: Educational Technology Publications, Inc.
- Salisbury, D. F. (1988) Effective drill and practice strategies. in Jonassen, D.H. ed. *Instructional designs for micro- computer courseware*: LEA.
- Salisbury, D. F. (1990) Cognitive Psychology and Its Implications for Designing Drill and Practice Programs for Computers. *Journal of Computer-Based Instruction*, Vol. 17, No. 1, pp. 23-30.
- Schank, R., Berman, T. and Macpherson, K. (1999) Learning by Doing. in In C.M.Reigeluth (Ed.), *Instructional Design Theory and Models Vol. II: A New*

-
- Paradigm of Instructional Theory.*, Hillsdale, NJ: Lawrence Erlbaum.
- Schwartz, D. L., Lin, X., Brophy, S. and Bransford, J. D. (1999) Toward the Development of Flexibly Adaptive Instructional. in *In C.M.Reigeluth (Ed.), Instructional Design Theory and Models Vol. II: A New Paradigm of Instructional Theory.*, Hillsdale, NJ: Lawrence Erlbaum.
- Spector, M. and Ohrazda, C.(2004) *Handbook of Research on Educational Communications and Technology*, Chap. Automating Instructional Design: Approaches and Limitaions, Hillsdale, NJ: Lawrence Erlbaum Associates.
- Spector, M. and Song, D. (1995) Automated instructional design advising. in Tennyson, R. and B Ann eds. *Automating instructional design: Computer-based development and delivery tools*, New York: Springer-Verlag.
- Suzuki, K., Nishibuchi, A., Yamamoto, M. and Keller, J. M. (2004) Development and evaluation of Website to check instructional design based on the ARCS Motivation Model. *Information and Systems in Education*, Vol. 2, No. 1, pp. 63-69.
- Tennyson, R. and Spector, M. (1995) Automated Instructional Design: An Introduction. in Tennyson, R. and B Ann eds. *Automating instructional design: Computer-based development and delivery tools*, New York: Springer-Verlag.
- The University of IOWA (1999) Drill Designer 2.1. <http://www.alessiandtrollip.com/>.
- Uduma, L. and Morrison, G. (2007) How do instructional designers use automated instructional design tool? *Computers in human behavior*, Vol. 23, No. 11, pp. 536-553.
- Wiley, D. A. (2001) Learning objects Need Instructional Design Theory. in Rossett, A. ed. *The ASTD E-Learning Handbook*, pp. 115-126, New York: McGraw-Hill Companies.
- Wiley, D. A. (2008) *The Learning Object Literature*, Chap. 29, pp. 345-353: Lawrence Erlbaum Associates.
- Young, L. A. and Reeves, C. T. (2007) M. David Merrill - A Significant Contributor to the Field of Educational Thechnology. *Educational Technology*, Vol. July-August, pp. 51-53.
- 青山学院大学総合研究所 e ラーニング人材育成研究センター (2008) , <http://elpc.o2en.aoyama.ac.jp/> .
- 石塚丈晴・堀田龍也・小川雅弘・山田智之 (2004) 小学生を対象とした PDA を用いた漢字ドリル学習システムの開発 . 日本教育工学雑誌 , Vol.27 , Suppl. , pp.225-228 .
- 井本玲雄 (2006) パナー型学習支援システムの開発と学習行動の調査 . 岩手県立大学ソフ

- トウェア情報学部 2005 年度卒業論文集 .
- 榎本聡・清水康敬 (2006) 教育情報のメタデータ化と検索システムの構築 . 情報知識学会誌 , Vol.16 , No.4 , pp.63-72 .
- 大森不二雄 (編) (2008) IT 時代の教育プロ養成戦略 . 東信堂 .
- 梶田将司・角所考・中澤馬志・竹村治雄・美濃導彦・間瀬健二 (2007) 高等教育機関における次世代コース管理システムの構築に向けて . 日本教育工学会論文誌 , Vol.31 , No.3 , pp.297-305 .
- ガニエ・ウェイジャー・ゴラス・ケラー (著) 鈴木克明・岩崎信 (監訳) (2007) インストラクショナルデザインの原理 . 北大路書房 .
- 許紅・繁樹算男 (1990) 項目反応理論と教授内容の階層的構造表現による問題項目の提示順序の最適化 . 日本教育工学雑誌 , Vol.14 , No.2 , pp.73-80 .
- 経済産業省商務情報政策局情報処理振興課 (編) (2007) e ラーニング白書 2007/2008 年版 . 東京電機大学出版局 .
- 廣濟堂 (2008) 学習天国ぼすと . <http://kenten.jp/> .
- 清水康敬 (2007) 社会基盤としての学習オブジェクトの現状と展望 . 日本教育工学会論文誌 , Vol.31 , No.3 , pp.259-269 .
- 鈴木克明 (1989a) テレビ番組による外国語教育を補うドリル型 CAI の構築について . 放送教育研究 , Vol.17 , pp.21-37 .
- 鈴木克明 (1989b) 米国における授業設計モデル研究の動向 . 日本教育工学雑誌 , Vol.13 , No.1 , pp.1-14 .
- 鈴木克明 (1995) 魅力ある教材設計・開発の枠組みについて-ARCS 動機づけモデルを中心に- . 教育メディア研究 , Vol.1 , No.1 , pp.50-61 .
- 鈴木克明 (1998) HyperCard 上のドリル教材作成支援ツールの開発研究-教材設計モデルを用いた評価と使い易さの評価をもとに- . 日本教育工学雑誌 , 第 22 , No.1 , 43-55 .
- 鈴木克明 (2002) 教材設計マニュアル . 北大路書房 .
- 鈴木克明 (2004) 詳説インストラクショナルデザイン : e ラーニングファンダメンタル . 日本イーラーニングコンソーシアム .
- 鈴木克明 (2005a) e-Learning 実践のためのインストラクショナル・デザイン . 日本教育工学会論文誌 , Vol.29 , No.3 , pp.197-205 .
- 鈴木克明 (2005b) 教育・学習のモデルと ICT 利用の展望 : 教授設計理論の視座から . 教育システム情報学会誌 , Vol.22 , No.1 , pp.42-53 .
- 鈴木克明 (2006) 人間情報科学と e ラーニング (第 5-8 章) . 放送大学教育振興会 .
- 関一也・松居辰則・岡本敏雄 (2003) e ラーニング環境での学習オブジェクトの適応的系列化手法に関する研究 . 電子情報通信学会論文誌 , Vol.J86-D-I , No.5 , pp.330-344 .
- 高橋暁子・市川尚・阿部昭博・鈴木克明 (2007) 課題分析図に基づく自己管理学習支援型

- e ラーニングシステムの開発．日本教育工学会論文誌，Vol.30，Suppl.，pp.25-29．
- 田中裕也・井ノ上憲司・根本淳子・鈴木克明 (2005) オープンソース CMS の実証的比較分析と選択支援サイトの構築．日本教育工学会誌，Vol.29，No.3，pp.405-413．
- 玉木欽也 (監修) 齋藤裕・松田岳士・橋本諭・権藤俊彦・堀内淑子・高橋徹 (2006) e ラーニング専門家のためのインストラクショナルデザイン．東京電機大学出版局．
- 常磐祐司 (2007) e ラーニングを支える最新技術動向～ハイビジョン，Web2.0 およびユビキタスデバイス．教育システム情報学会誌，Vol.24，No.2，pp.137-147．
- 藤原康宏・大西仁・永岡慶三 (2005) 情報処理入門科目におけるオンライン個別学習システムを利用した授業実践とその効果．日本教育工学会論文誌，Vol.29，Suppl.，pp.109-112．
- 不破泰・中村八束・山崎浩・大下眞二郎 (2003) Web を用いた CAI システムによる大学講義の高度化とその評価．教育システム情報学会誌，Vol.20，No.1，pp.27-38．
- 仲林清 (2007) e ラーニングにおける技術標準化とオープン化．日本教育工学会論文誌，Vol.31，No.3，pp.285-295．
- 仲林清・中村明仁・吉岡俊正・相良貴子・加賀田俊・永岡慶三 (2005) 標準規格に準拠したオンラインテストシステム．日本教育工学会論文誌，Vol.29，No.3，pp.299-307．
- 仲林清・清水康敬・山田恒夫 (2006) e-Learning 標準化技術の開発と実践の新しい展開：SCORM と LOM を中心に．人工知能学会誌，Vol.21，No.1，pp.92-98．
- 日本 e ラーニングコンソーシアム (2008) ，<http://www.elc.or.jp/>．
- 日本教育工学会 (編) (2000) 教育工学事典，実教出版．
- 根本淳子・鈴木克明 (2005) ゴールベースシナリオ (GBS) 理論の適応度チェックリストの開発．日本教育工学会誌，Vol.29，No.3，pp.309-318．
- 水野りか (2000) Low-First 分散学習方式の効果の CAI での実験的検討．日本教育工学雑誌，Vol.24，No.2，pp.111-120．
- 水野りか・松尾崇史 (2006) 認知心理学的知見に基づいた Low-First 方式の携帯電話用 e-Learning システムの効果と学生の評価．教育システム情報学会誌，Vol.23，No.4，pp.185-196．
- 山口和紀・于旭・中村敦司・新城靖・西山博泰・古瀬一隆・石川佳治・佐々木重雄・林謙一・萩原一隆・金谷英信・鈴木孝幸・黒石和宏 (1992) The Unix Super Text [上]．技術評論社．
- 山本雅之・藤原康宏・鈴木克明・赤堀侃司 (2004) 次元分けとサブドリル構造を用いた概念学習シェルの開発．教育システム情報学会第 29 回全国大会講演論文集，pp.307-308．
- 松浦執 (2006) 初等物理学 e-Learning でのドリル反復学習支援．日本教育工学会論文誌，Vol.29，Suppl.，pp.193-196．