

宇都宮大学大学院

博士論文

プログラミング教育を支援する問題解決環境
に関する研究

平成23年3月

宇都宮大学大学院工学研究科

エネルギー環境科学専攻

寺元 貴幸

[目次]

1. 序論	1
2. プログラミング教育と支援環境	5
2.1. 教育用 PSE システム	5
2.2. プログラミング教育	5
2.3. 教員が感じるプログラミング教育の問題点	6
2.4. 学習者が感じるプログラミング教育の問題点	8
2.5. プログラミング教育支援環境への要求	12
3. 教育支援環境の構築	13
3.1. 教育支援用フレームワーク	14
3.2. 授業支援システムの概要	16
3.3. TSUNA-TASTE システム	21
3.3.1. TSUNA-TASTE システムの概要	22
3.3.2. 学生機エージェント	22
3.3.3. 学生機用モジュール	24
3.4. ネットワークによる授業支援	26
3.4.1. ネットワーク構造	26
3.4.2. WWW サーバ	27
3.4.3. データベースシステム	27
3.4.4. ブローカシステム	29
3.4.5. 通信データの構造	30
3.5. TSUNA-TASTE の適用例	31
3.6. プログラミング試験の支援	34
3.7. コンパイル情報を利用したプログラミング支援システム	37
3.7.1. コンパイラ情報の取得	37
3.7.2. コンパイラのラッピング	38
3.7.3. コンパイル・エラーの分類	39
3.7.4. 個人ごとのエラー情報	41
3.7.5. 成績とコンパイル・エラー	42
3.7.6. コンパイル・エラーに対するアドバイス	45
3.8. アルゴリズム可視化ツール	46
4. 実践的競技システムへの適用	50
4.1. ネットオークション型システム	51
4.1.1. 競技ルール	51
4.1.2. 競技用の石垣・石垣枠	52
4.1.3. 競技用システム	53

4.1.4.	サブシステム	54
4.1.5.	システムの障害対策	55
4.1.6.	順位決定処理	56
4.1.7.	まとめ	58
4.2.	高セキュリティ型ネットワークシステム	58
4.2.1.	競技ルールとシステムの不具合	58
4.2.2.	再試合の検討	60
4.2.3.	再試合システムの開発	61
4.2.4.	再試合の運営	62
4.2.5.	まとめ	63
4.3.	アルゴリズム対戦型システム	64
4.3.1.	競技ルール	64
4.3.2.	競技システム	66
4.3.3.	競技シミュレータ	67
4.3.4.	まとめ	68
4.4.	インターネット対戦可能な準リアルタイム対戦システム	68
4.4.1.	競技ルール	68
4.4.2.	競技システム	70
4.4.3.	インターネット対戦システム	71
4.4.4.	まとめ	73
5.	結論	75
5.1.	プログラミング試験の評価	75
5.2.	コンパイル・エラーに対するアドバイス	76
5.3.	アルゴリズム可視化に対するアンケート	78
5.4.	教育支援フレームワークの問題解決環境としての評価	79
5.5.	まとめ	81
	参考文献	82
	研究業績	86
	謝辞	87

1. 序論

日本の全国の全公立学校（小学校，中学校，高等学校，中等教育学校及び特別支援学校）にパーソナルコンピュータ(Personal computer:パソコン)が導入され始めたのは1985年頃であるが，その後Windows 95に代表されるグラフィカルユーザインターフェース(Graphical User Interface: GUI)を中心としたOSの普及，パソコンの低価格化などを追い風に教育現場にも広くパソコンとインターネットが普及をはじめた．その後，1990年から当時の文部省（現在の文部科学省）がパソコン環境整備のために予算を割り当て，2011年3月には生徒3.6人あたりに1台のパソコンを導入する方針をかかげてIT新改革戦略として予算を計上してきた．しかし現実的には7.2人に1台程度(H21.3現在)の普及率にとどまっている．またネットワークに関しても同じく2011年3月までに学校内のLAN普及率を100%にするとの目標を定めているが，LANの普及率についても64%程度(H21.3現在)[1]と目標を大きく下回っている．

パソコンの普及率の低さに関しては本論文の主旨とは直接関係ないので議論しないが，このようにパソコンの普及率が低いことから，現在日本の教育現場においてパーソナルコンピュータは1人が1台を所有するという真の意味でのパーソナルコンピュータではなく，パソコン演習室において1台のパソコンを多数で共有し，学生・生徒が集団で演習を行うために使用されている事を示している．学生にパソコンを貸与する私立大学や私立高校もあるが，非常に限られた例であり全国的に広く普及している制度ではない．このため，演習室を対象とした支援システムは現在でも強い要望がある．そして今後しばらくはパソコン演習室にクラスが移動し，そこで各種の演習を行うタイプの学習方法が継続することを示している．

次に教員側のパソコンを使った学習指導に関しても，平成21年3月1日に文部科学省の行った「教員のICT活用指導力の基準の具体化・明確化に関する検討会」でのアンケート調査[1]によると「活用できる」，または「割とできる」と自信のある教員の割合は以下のようにになっている．（複数回答）

- 教材作成のためにICTを活用（79.9%）
- ICTを活用して校務分掌等に必要な情報を収集し文書等を作成（75.1%）
- 児童生徒がICTを活用してわかりやすく発表・表現できるよう指導（52.0%）
- 教員間における必要な情報の交換・共有化（58.8%）

このアンケート結果から，プレゼンテーションソフトを使って全体に資料を

掲示するような指導に関しては、ある程度自信を持って対応できているが、個別指導や情報の交換・情報の共有化に関してはまだ半数程度しか自信が無いことを示している。この点に関しても教員をできるだけ丁寧に支援するシステムが求められていることが認識できる。

高等専門学校（以後高専）や大学など高等教育機関では導入されるパソコンの台数や能力も高く、パソコンを利用したプログラミング教育やシミュレーション演習も数多く行われるようになった。しかしパソコンやネットワークが普及しても、教員が学生の様子を十分に知ることができない状況は以前から続いておりこの部分を支援するための仕組みが強く求められてきた。

パソコンを使った演習時に教員が知りたい情報として、学生のパソコン操作の履歴がある。学生がパソコンに対してどのような操作を行っているのかその情報を詳細に収集することができれば、教員は学生についてより多くの情報を得ることができる。学生の様子が分析できれば、演習の内容や指導の方法を工夫することが可能になる。また、特に指導が必要となる学生の特定も容易となり、高い教育効果が期待できる。

さらに学生の指導を行うことができるシステムがあれば、より高度な授業支援が可能となる。1クラスの人数は小中学校や高専では40名程度、大学では100人規模になることもある。このように多くの学生を教員1名で面倒見ることは不可能であり、先ほどのアンケートのように個別指導ができないという教員の意見に繋がっている。大学の場合はこのような状況を改善するためにティーチングアシスタント制度がある。このようにティーチングアシスタントの制度が存在することは、演習時の個別指導が重要であることを認識していることを表している。本論文では、このような個別指導の問題を解決するシステムに関して研究を行ってきたのでその検討の過程から構築したシステムおよびその評価について述べている。

本論文では以上のように学生側と教員側の双方を支援し、円滑な教育が実現できる支援環境の構築に関して研究を行ったのでその背景や概念、そして構築したシステムについて述べる。このような特定の問題を解決するシステムを、コンピュータ分野では問題解決環境（Problem Solving Environment: PSE）と呼んでいる[2]–[5]。狭義の定義としてPSEは計算科学などの複雑で大規模な計算を行うために必要となるハードウェア・ソフトウェアを、効率よく動作させるシステムを示す用語として利用されることが多い。しかし本来はもう少し広義の意味を含み、今回開発を行っている授業支援に特化したシステムも問題解決環境と呼ぶことができる。

パソコンを使った教育は小中学校のリテラシー教育に始まり、高校では情報処理教育として、プライバシーや著作権など情報の取り扱い、ワープロ演習、

表計算ソフトの演習など少しずつ高度になっていく。高専・大学ではさらにプログラミング教育や各種専門教育用のソフトウェア（画像解析ソフト，回路シミュレータ，三次元 CAD など）の利用へと進んでいく。いずれも効率的な学習を行うには多くの工夫が必要であり，そのための学習支援ソフトウェアも数多く存在している。しかし残念ながら今のところ全ての領域をカバーする教育支援環境は存在しない。一般的なパソコン操作やファイル共有を行う授業支援システムは，もちろん製品として数多くあるが，逆にこれらの授業支援システムは，特定領域で学生・教員が望むような動作にカスタマイズできるような柔軟性に乏しい。

本論文ではプログラミング教育という分野に焦点をあて，この領域を支援する事を対象として考えた。プログラミング言語教育（以下プログラミング教育）については，パソコンの操作方法だけでなくファイルの保存場所などのパソコン環境やコンパイラ環境，そしてアルゴリズムの学習など学習者に要求する項目が多く，初学者にとってはハードルの高い科目になっている。そこで最初にプログラミング教育でどのようなことが問題となっているのか現状を調査した。そして教員および学習者がどのようなことを問題として感じているのかを調べた。その上で，適切な指導を行っていくにはどのような点を支援し改善すればよいのか検討している。このように 2 章ではプログラミング教育にどのような支援が必要かについて詳しく述べる。

3 章ではその検討を元にまず教育支援用のフレームワークを設計し，その概要について述べる。そしてそのフレームワークを利用したネットワーク分散型の授業支援システム **TSUNA-TASTE** の構築を行ったので詳しく述べる。また **TSUNA-TASTE** を実際の教育現場に適用し授業支援を行ったのでその利用状況を報告する。特にコンパイラ情報の取得とアドバイス表示に関して詳しく述べる。また **TSUNA-TASTE** の適用から，学生がプログラミング教育にアルゴリズムの可視化を求めている事が判明したので，アルゴリズムの可視化を可能にするシステムを開発し，それを実際に利用してきた。その様子についても述べる。

4 章では教育支援用フレームワークを授業支援システム以外に適用した事例について述べる。プログラミング教育は，その実践力の高さがもつとも重要である。この実践的な場として高専では約 20 年前からプログラミングコンテストを行ってきた。このコンテストのうち競技部門について，平成 19 年から平成 22 年の 4 年間本教育支援用フレームワークの仕組みを活用してシステムの設計・構築，そして実際の競技を行ってきた。システムは 4 例あり 4.1 節でネットオークション型システムのシステムを，4.2 節で高セキュリティ型ネットワークシステムを，4.3 節でアルゴリズム対戦型システムを，そして 4.4 節でインターネ

ット対戦可能な準リアルタイム対戦システムについてそれぞれ述べる。これらのシステムは競技のルールや開催条件により、フレームワークをベースにその都度モジュールやシステム環境を変更し、それぞれの問題に適用していったものである。

最後に 5 章でプログラミング授業支援の評価に関して、プログラミング試験での運用、コンパイル・エラーに対するアドバイスシステム、そしてアルゴリズム可視化について報告し、さらにシステム全体に関して検討しまとめる。

2. プログラミング教育と支援環境

2.1. 教育用 PSE システム

教育分野において問題解決環境 (Problem Solving Environment: PSE) の果たす役割は非常に大きく、現実的に高い教育効果が得られる適用例も多いなど PSE の得意な分野の 1 つである。教育分野の PSE システムは、次のように大きく 3 つに分類することができる[6]。

第 1 は教員側をサポートし講義や演習を支援する PSE システムである。近年教育現場へのコンピュータ普及が進むにつれ、情報関連の教育時間が増加している。また e-Learning の普及が始まり、多くの教育機関や企業においても Web ベースの教育が実践されるようになってきている。このように教育現場では多くのパソコンが利用されているが、指導教員が学習者すべての学習状況を正確に把握することは困難である。このような問題を解決するシステムは学習状況把握型 PSE と呼ばれる。

第 2 は学習する仕組みの作成をサポートする PSE システムである。学習者はいつでもどこでも学習できる学習環境を切望しており、e-Learning はその 1 つの有効な解決方法である。しかし、e-Learning システムを作成する事自体が大きな問題であり、この問題を解決する仕組みが必要となってきた。このように e-Learning などの教材作成を支援するシステムを教材作成支援 PSE と呼ぶ。

第 3 は学習者が抱える特定の問題を解決する PSE システムである。学習者は非常に多くの問題を持っている。これらをデータベースから検索して学習者に適切なアドバイスや実例を示すシステムを発想支援型 PSE と呼ぶ。

残念ながら現在の段階ではこれら全ての問題を 1 つの PSE システムで解決する教育用万能 PSE システムは存在しない。しかし、対象者や対象分野をある程度限定すれば、有効な PSE システムの作成は十分可能である。今回本論文では、学習者の状況を把握する第 1 の学習状況把握型 PSE と、第 3 の学習者にアドバイスを行うことで学習者を支援する発想支援型 PSE の両方の機能をもつ支援環境の構築を目指している。また e-Learning 教材ではないが、プログラミングのアルゴリズムを可視化することにより、学習者を支援するシステムについても開発を進めている。このように教育分野における PSE の可能性を探り、実際に教育現場で実践してその結果を検討した。

2.2. プログラミング教育

情報専門学科の標準カリキュラムとして J07 が 2006 年度から 2007 年度にかけて情報処理学会の情報処理教育委員会の下で策定され、2008 年度に修正と

補正が行われた[7]. これは世界標準である米国 IEEE/ACM の CC2001-CC2005 を土台として、日本の情報専門教育の状況に対応した見直しを行い、コンピュータ科学 (J07-CS), 情報システム (J07-IS), ソフトウェアエンジニアリング (J07-SE), コンピュータエンジニアリング (J07-CE), インフォメーションテクノロジー (J07-IT) の5つの領域と、広く情報について学ぶ内容を定めた一般情報処理教育 (GE) についてまとめたカリキュラム標準である. 非常に幅広い情報教育分野のなかで、各知識領域について体系的に学んでいくべき指針となるカリキュラム構成や最低限習得させるべき項目をコアカリキュラムとして列挙しており、今後の情報処理教育の大きな指標となっている.

高専におけるプログラミング教育はこのなかのコンピュータエンジニアリング (J07-CE) に分類され、「PF プログラミングの基礎」「AL アルゴリズム」, 「PL プログラミング言語」などにその詳細が記述され、ユニット数やコア時間が制定されている. 特に「PF プログラミングの基礎」には 38 時間ものコア時間が設定されるなど、非常に多くの時間が必要であることが明記されている.

このように、プログラミング教育では学習すべき内容はすでに決められているが、これをどのように教育し、さらにどのように評価するかに関しては、難しい問題点がある. プログラミング教育の難しさとしては以下の点が考えられる.

- 表現が抽象的であり、修得には十分な学習時間を必要としている.
- プログラミング言語・およびプログラミング環境は多種多様で教育標準とよべる環境を規定できていない.
- 多くのアルゴリズムは抽象度が高く、そのままでは理解しづらい.
- プログラミングの学習には実際にプログラミングを行う演習が不可欠であり、座学だけでは理解が難しい.

これらの状況は 2.4 節に示す学生へのアンケート結果からも伺うことができる.

2.3. 教員が感じるプログラミング教育の問題点

プログラミング教育のようにパソコンを使った演習が増えることは、学生にとって新たな知的好奇心を喚起し、創造的な学習によってさまざまな成果を上げる優れた面がある. しかし、反面演習にパソコンを使用することにより学習上の問題も発生している. その学習上の問題の一つが教員側の指導に関する問題である. 教員自身のスキルが仮にどんなに高くても、その教員が十分に観察し指導できる学生は一度に一人に限られるため、指導していない学生やクラス全体の状況が十分に把握できないという問題である. 通常の授業や講義の場合、学生とのコミュニケーションを大切にしながら進めていったとしても、進行の

主導権は教員にあり，基本的には全体が一定の流れに沿っている．しかし，パソコンを使った演習時は学生自身のスキルのばらつきから（このばらつきは年々大きくなるようである）少し時間が経過すると，進捗状況がバラバラになり教員は個別対応を余儀なくされる．この問題により，教員がすべての学生のようにすを十分に把握することは困難な状況が発生する．また教員が学生個々の対応にまじめに取り組めば取り組むほど，指導を受けられない学生の待ち時間が長くなり，さらにこの問題は大きくなる．

そこで教員に代わって学生のようにすを観察するシステムが必要となる．プログラミング教育やアプリケーション教育などコンピュータを利用した学習では，学生の操作状況を把握することが重要であり，このことから学生の操作状況を収集する何らかの仕組みが必要とされてきた．学生がパソコンに対してどのような操作を行っているのかという情報を詳細に収集し分析することができれば，教員は学生についてより多くの情報を得ることができ，さらに学生の指導を行うことができれば教育効果も高まると考えられる．また，これらの情報を別の側面から分析すれば，学生自身のさまざまな特徴のほかに演習環境や問題解決環境の利用状況，さらにはシステムの問題点などを調査することも可能となる[8]－[14]．

本論文ではこの問題を解決する環境として，PSE システムの第 1 の分類である教員を支援するタイプの教育用 PSE である授業支援システムを構築した．この授業支援システムは学生のパソコン操作のようすをパソコン上に常駐させたソフトウェア（以後学生機エージェントと呼ぶ）で観察し，このデータをネットワーク経由でサーバに収集・分析することで学生の状態を把握するシステムである．どのようなデータを収集するのかといった設定は教員が操作するパソコン（以降教員機と呼ぶ）の上のウィンドウ画面や Web インタフェースにより行い，その設定が学生機エージェントに転送される．また，学生機エージェントは学生が演習に不必要なアプリケーションを起動した場合に警告を発生するといった比較的単純な処理に関しては学生機エージェント自身が判断し，自動的に警告を出す仕組みを持っている．これらの仕組みを利用することができれば多忙な教員を支援し，学習効果の高い授業が可能になると考えた．

また授業支援システムは各教員が行う講義や演習によりその支援内容を変更できる機能を持つことが必要である．プログラミングの演習と TOEIC の演習とではパソコンで使用するアプリケーションも，演習の方法も大きく異なる．こういった各教員のニーズを柔軟に反映できることが教育用 PSE システムには求められると考える．

ネットワークを利用した起動アプリケーションの調査やパソコンの遠隔操作，システムの利用制限を行うソフトウェアは既に市場に複数存在しているが，授

業支援用 PSE システムとして能力を発揮するためには教員が求める情報の収集と蓄積，データの分析，そして学生の指導をすべて統合的に行うことが必要である．この点で学生の監視を行う既存アプリケーションとは目的やアプローチが異なっている．

2.4. 学習者が感じるプログラミング教育の問題点

学習者自身がプログラミングについてどのように感じているかを知るためにアンケート調査を行った．このアンケートは津山高専情報工学科 2 年生を対象として 2003 年度末に行った．この学生は 1 年生から約 2 年間プログラミングを学習した学生であり，多くの学生が学習を始める段階ではプログラミングの初心者であった．アンケートは 2 年生の終了直前に実施した．学習内容は 1 年目に JavaScript でプログラミングの概念と初歩的な構文の基礎を学習し，2 年目に C 言語により変数やデータ型，そして条件分岐や繰り返しを組み合わせたアルゴリズムの基礎を学習する．週 1 回 2 時間，年間で約 30 週（60 時間）程度の演習を行っており，J07「PF プログラミングの基礎」の要件を概ね満たした内容となっている．

最初に学習意欲に関して以下のような質問を行った．これは学習意欲そのものが低いと，今後のアンケートの信頼性が疑われるからである．

Q1. 「自分にとってプログラミングを習得する事は将来役にたつと思いますか？」

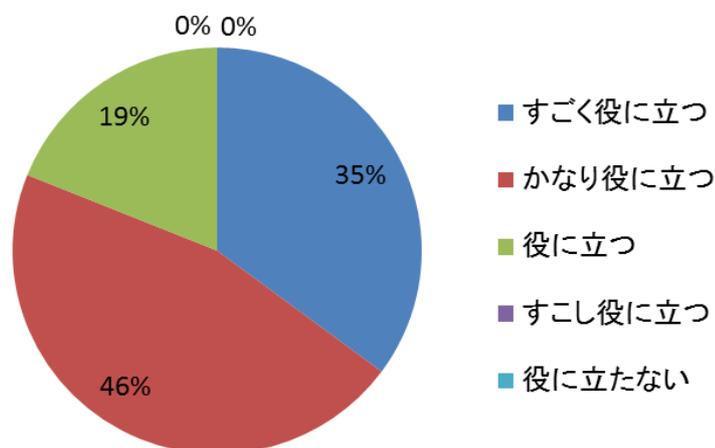


図 2-1 プログラミングの必要性

図 2-1 が示すように，学習者はプログラミング能力を将来役に立つスキルと考えており，プログラミング教育の必要性を強く認識している．このように約 8 割の学生がプログラミング技術の習得を強く望んでおり，学習に前向きに取り

組んでいる。またプログラミング技術を役に立たないのものと判断している学生は0%であり、プログラミング教育が重要であるかことを学生自身も理解していることが分かる。

ではこれらの学生に次の質問でプログラミングを好きか嫌いか尋ねてみた。

Q2. 「プログラミングが好きですか？」

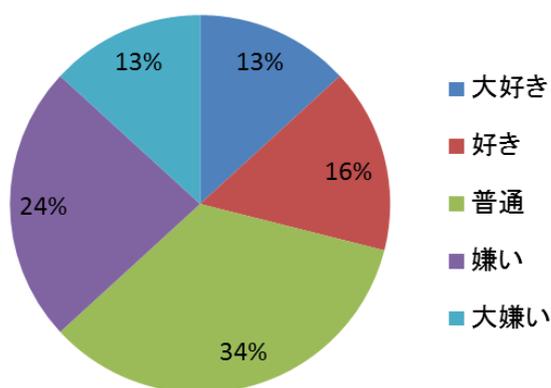


図 2-2 プログラミングの好き嫌い

図 2-2 が示すように、「普通」が 34%に対して「好き」が 29%、「嫌い」37%となり、学生の嗜好がはっきりと二分化している様子がうかがえる。このようにクラス全体が二分化していることは、教員にとっては非常に教えにくい状況である。好きな学生に合わせて講義や演習を進めると嫌いな学生には難解になりますます嫌いになる。嫌いな学生に合わせて話をすると好きな学生にとっては物足りない気分となる。そして一度「嫌い」になってしまった学生を「好き」に変えることは経験上とても難しい。つまりプログラミングが得意な学生はあまり手をかけなくてもどんどん好きになり、一度壁にぶつかって嫌いになった学生は、短期間では好きにはならない傾向がある。このような傾向からも、学生をつまずきをいかに早期に発見し、適切な指導によりすくい上げる事ができるかが重要となると思われる。

では好きな学生も含めてプログラミングが簡単か、それとも難しいかを尋ねてみた。

Q3. 「プログラミングは難しいですか？」

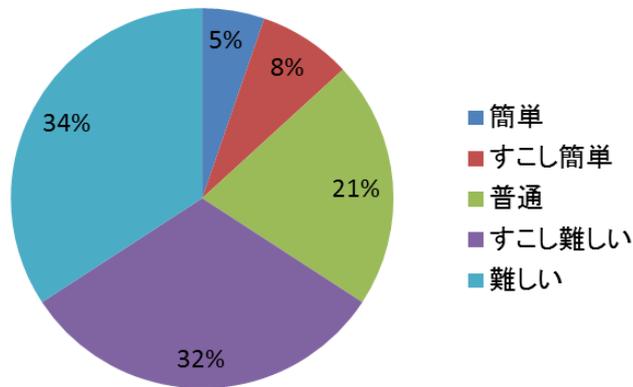


図 2-3 プログラミングの難易度

この問いに対しては、1割程度が「簡単」「少し簡単」と答え、7割程度が「すこし難しい」「難しい」と答えており、プログラミングはかなり難しいと感じている学生が多いことが分かった。この難しいと感じている部分を明確にして、その部分を集中的にわかりやすく教育することで学習効果を高める事ができると考えられる。

プログラミングの演習時に学生が戸惑い、よく質問するのがコンパイル時のエラーメッセージに関する質問である。そこでコンパイル・エラーに関して、以下のような質問を行った。

Q4. 「コンパイル・エラーをわかりやすい言葉にしてくれると助かりますか？」

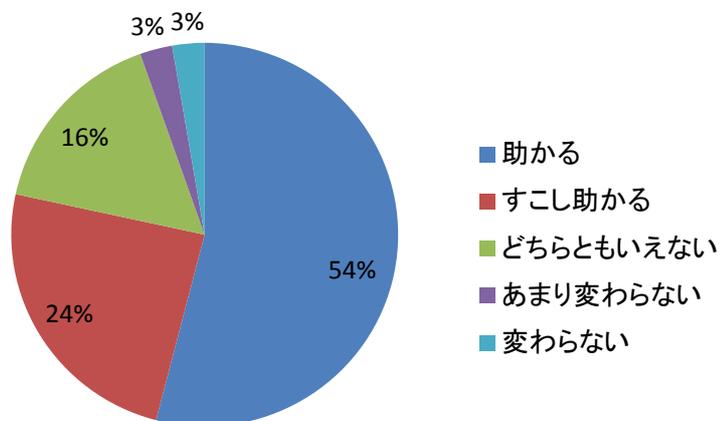


図 2-4 コンパイル・エラーへのアドバイス

この問いに対して、8割近い学生が「助かる」と答えており、コンパイル・エラーに関して多くの学生が難解と感じておりアドバイスを求めていることが判明した。つまりこの問題を支援することにより多くの学生の学習を支援できることが判明した。これよりプログラミングの教育支援環境にはコンパイル・エラーに関する何らかの支援が必要であることが示された。この結果を受けて、3.7節に示すコンパイル情報を利用したプログラミング支援システムに関する研究を行った。

プログラミングの演習において学生の多くがコンパイル・エラーに苦手意識を持っていることが分かったが、その他にはどのような項目が苦手なのか自由記述で回答を求めた。

Q5. 「プログラミングで分かりにくいと感じる項目はなんですか？」

表 2-1 プログラミングの苦手領域

項目	割合
アルゴリズム全般	31%
繰り返し処理	19%
ポインタ	14%
If文の条件分岐	11%
関数	9%
変数	5%
構造体	4%
配列	4%
全部	14%

表 2-1 が示すようにアルゴリズム全般や制御構造が把握できていない学生の割合が多く、アルゴリズムに関しても何らかの支援が必要であることが判明した。

なお一般的には難しいとされるポインタや構造体に対する回答の割合があまり高くないのは、2年生ではポインタや構造体は導入部分しか学習していないため、まだ本当の難しさを体験していないためと考えられる。より複雑な内容は上級生（高専の場合は3年生以上）になってからであり、このアンケートではそれほど高い値を示さなかったと考えられる。

これらの結果から授業支援システムには何らかの形でアルゴリズムを可視化してわかりやすくするための仕組みが必要と考えた。この結果を受けて、3.8節に示すアルゴリズム可視化に関する研究を行った。

2.5. プログラミング教育支援環境への要求

本章においてプログラミング教育の問題点を教員側から検討した場合と、学習者側から検討した結果を以下にまとめる。

- 教員に代わって学生の操作状況を収集し蓄積する。
- プログラミングに関する各種情報を収集する。
- 教員に収集した情報を提示する。
- 教員が授業・学生別の設定が行える。
- 教員が学生機にメッセージを送ることができる。
- 教員の手によらず教育支援システムから自動的に学生機にアドバイスを表示，警告の表示，システムへの介入ができる。
- プログラミングに関してコンパイル・エラーを分かりやすくアドバイスできる。
- アルゴリズムの分かりやすく可視化して提示できる。

上記の要求を満たすために 3 章で教育支援環境の構築に関して述べる。

3. 教育支援環境の構築

2章の分析により，プログラミングの教育支援を行うには2.5節に示した機能が必要となることが判明した。例えばパソコンを使った演習時には教員が学生のパソコン操作の履歴を知る事ができる機能が必要となる。学生がパソコンに対してどのような操作を行っているのかという情報を詳細に収集し分析することができれば，教員は学生についてより多くの情報を得ることができ，さらに学生にアドバイスを提示することで効率的な教育が可能となる。また，これらの情報を別の側面から分析すれば学生自身の特徴のほかに演習環境や問題解決環境の問題点等を調査することも可能となる。さらにプログラミング教育ではコンパイル・エラーに関する支援と，アルゴリズムに関する支援が重要であることも判明した。

そこで，本章ではプログラミング教育のための教育支援環境を提案したい。教育支援環境を構築するために，まず環境に求められる機能を教育支援フレームワークとしてまとめる必要がある。そしてこの教育支援フレームワークの実践としてネットワーク分散型の授業支援システム（TSUNA-TASTE）を構築した。教育支援フレームワークは，システムが持つべき特徴および機能の基盤となる設計思想を示したものであり，これを3.1節に示す。このフレームワークを実際にインプリメントしたシステムが授業支援システムであり，その詳細を3.2節以降に述べる。

システムの構築のために，次に示す開発段階を経て目的となるフレームワークの実現を行った。まず学生のパソコン操作の情報を収集するシステムを作成した。このシステムは学生のパソコン操作をパソコン上に常駐させたソフトウェア（学生機エージェント）で観察し，このデータをネットワーク経由でサーバに回収・分析することで学生の状態を把握するシステムである[15]–[21]。収集するデータの設定は教員機上のウィンドウ画面により行い，その設定が学生機エージェントに転送される。また，学生機エージェントは学生が演習に必要なアプリケーションを起動した場合に警告を発生するといった比較的単純な処理に関しては学生機エージェント自身が判断し，自動的に警告を出す仕組みを持っている。

この第1段階として専用アプリケーションを用いて学生機と教員機の間で通信を行い，学生機の様子を把握して操作が可能なシステムの構築を行った。本システムは教員が演習室で学生を直接指導する場合には有効であるが，専用の教員機だけがシステムにアクセスすることから汎用性に乏しく，また集めたデータを長期的に保存し広く活用する事ができなかった。そこで第2段階として，

システム全体を汎用性の高いアーキテクチャで設計し直し、データベースや Web サーバを導入することで、より利便性の高いシステムへと移行を行った。このとき、それぞれの機能を別々のコンピュータ（サーバ）で実行できるように設計し、分散処理が可能な構造とした。これは将来データ量が増え各種処理の負荷が高まった時に、必要時応じてサーバを追加してフレキシブルにシステムの拡張が行えることを目指した。このネットワーク分散型授業支援システムを 3.3 節から 3.8 節に示す。

3.1. 教育支援用フレームワーク

2 章のプログラミング教育に関する問題点の分析により、プログラミング教育を支援するシステムに必要な機能は明らかになってきた。そこで 2.5 節に示した機能を実現するためにまず教育支援用フレームワークを提案する。このフレームワークはプログラミング教育の支援に必要な機能が列挙されており、またシステムをどのように設計すべきかその指針がまとめられている。フレームワークの概要を情報の流れを中心として図 3-1 に示す。

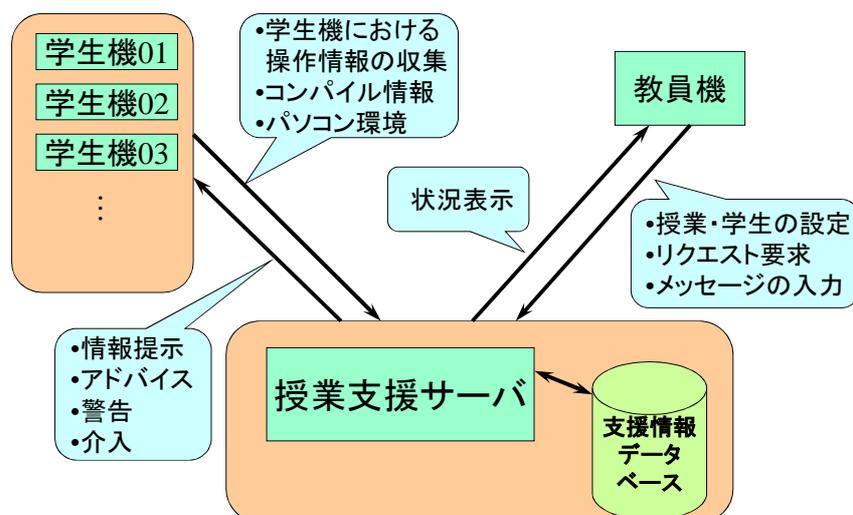


図 3-1 教育支援用フレームワークの概要

図 3-1 に示すように、システムを流れる情報は授業支援サーバを中心に教員機から学生機へリクエストの要求に基づき、学生機から情報を取り出して、授業支援サーバへ転送する。その情報を必要に応じて教員機が閲覧するというのがシステムの概要である。それぞれの機能における要項を以下に述べる。

[学生の情報収集・情報提示に関して]

- 学生パソコン操作に関して、キーボード、マウス、アプリケーション、プロセス、ウィンドウ情報等を収集する機能を持つ。

- 学生パソコンへの介入・情報提示などサーバ（もしくは教員側）から操作を行う機能を持つ。
- 情報収集・操作に関してシステムが自立的に動作する機能を持つ。
- 複数のタイプの実行（逐次実行，定期実行，連続実行）が可能である。
- 各機能を別々にコントロールするために，機能はモジュール化されており，さらに各モジュールの制御をカスタマイズ可能である。
- モジュールの追加により容易にシステムへ機能が追加できること。
- 授業支援用ではない一般的なソフトウェアに関する分析可能とするため，既存のアプリケーションやプログラムを授業支援用のモジュールにラッピングして機能に組み込むことが可能であること。
- システムは自動的に起動し，学生が故意に機能を停止することはできない。

[教員の指示に関して]

- 教員はシステムへの指示やデータの閲覧をネットワーク経由で行うことが可能である。
- カリキュラム（学年，クラス，教員，科目，演習室等）に応じてシステムの振る舞いを変更可能である。
- 学生の様子を分かりやすく表示可能である。
- 学生の様子はデータベースに保存し，教員は自由にデータにアクセス可能である。

[ネットワークに関して]

- 教室の単位でネットワークのトラフィックを抑制する機能がある。
- ネットワークを転送するデータはセキュリティに十分配慮する。
- 学生が演習場所を変更した場合に対応可能である。
- 通信プロトコルは **TCP/IP** ソケットを使用する。

[システムに関して]

- 学生が使用するパソコンの **OS** はもともと普及率の高い **Windows** システムとする。
- システムの設定変更等のインターフェースは **Web** ブラウザを基本とする。
- サーバ **OS** およびデータベースソフトは基本的にフリーウェアや **GNU General Public License** ならびに **BSD** ライセンスを使用する。
- 学生機以外のネットワーク部分はデータ転送効率の高い **Java** 言語を使用する。
- 学生自身も自分の操作記録を閲覧できるなど，情報公開の仕組みがある。

上記のフレームワークを実現するためには分散エージェントとネットワーク・データベースを基盤技術とした問題解決環境が必要になる。以降実際に構

築した授業支援システムについて 3.2 節から 3.8 節に示す。

3.2. 授業支援システムの概要

教育支援用フレームワークの概念をうけて、実際に稼働するシステムを実装する。第 1 段階として専用アプリケーションを用いて学生機と教員機の間で通信を行い、学生機の様子を把握して操作が可能なシステムの構築を行った。第 1 段階の開発では演習室でのパソコン演習を対象としたシステムとなっており、この段階ではまだプログラミング教育に特化されていない。以下にどのような操作が可能か詳しく述べる。

本システムの機能を、教員の操作からまとめると次のようになる。

- 1) 教員が専用アプリケーションにより授業支援を行う演習室を決定する。
- 2) 各学生機上に常駐する学生機エージェントは、学生機の OS から必要な情報を収集し教員機のエージェントに転送する。
- 3) 教員機のエージェント上に学生機の状態が表示される。
- 4) 禁止アプリケーションを発見した場合は「注意」「警告」「強制終了」を選択して学生機のエージェントに直ちにメッセージを送り学生機の画面に指示を出す。

このように本システムの操作はそのほとんどを教員機上で行う。教員機では各種機能の設定や情報の提示、指示内容の入力を行う必要があり、システムを使い易くするという観点からも教員機のエージェントは非常に重要である。もちろんプログラミング教育以外への適用も視野に入れ、コンピュータを専門としない教員でも利用可能なようにメニューやボタンにより選択する GUI 形式を採用している。

授業中は、各種の情報を教員に提示することで円滑な授業の実現を行う。構築したシステムでは、学生の起動アプリケーションの情報、マウス・キーボードの操作履歴の取得が可能である。マウスとキーボードの操作状況から学生の様子（どの程度まじめに演習を行っているか、どの程度 Windows の操作に熟達したか等）が推測できる。これらの設定は可能な限り手動と自動の切り替えが可能とし、教員の負担を軽減すると同時に処理の柔軟性の向上も目指している。

図 3-2 は演習時に学生が起動しているウィンドウ情報を表示するためのインタフェースである。これはウィンドウの左側に各パソコン（6 台分）が起動しているアプリケーションの一覧を表示している。巡回を「自動」に設定すると巡回時間に従って、表示するパソコンを切り替えることで演習室すべてのパソコンの様子を時分割で表示する。なお津山高専で使用する演習室は 48 台を基本構成としているので、10 秒間隔で巡回した場合 80 秒で演習室全体をモニタリングできる。

ウィンドウのタイトルを利用して「使用許可アプリケーション」と「使用禁止アプリケーション」を設定し、使用禁止のアプリケーションを発見した場合は強調表示する。右上の管理状態を自動的に設定すれば、使用禁止のウィンドウを発見した場合、最初に学生へ警告を出し、それでも終了しない場合は、エージェントに対してアプリケーションの強制終了メッセージを送信する。管理状態が半自動の場合には、その都度教員の指示を仰ぐことになる。

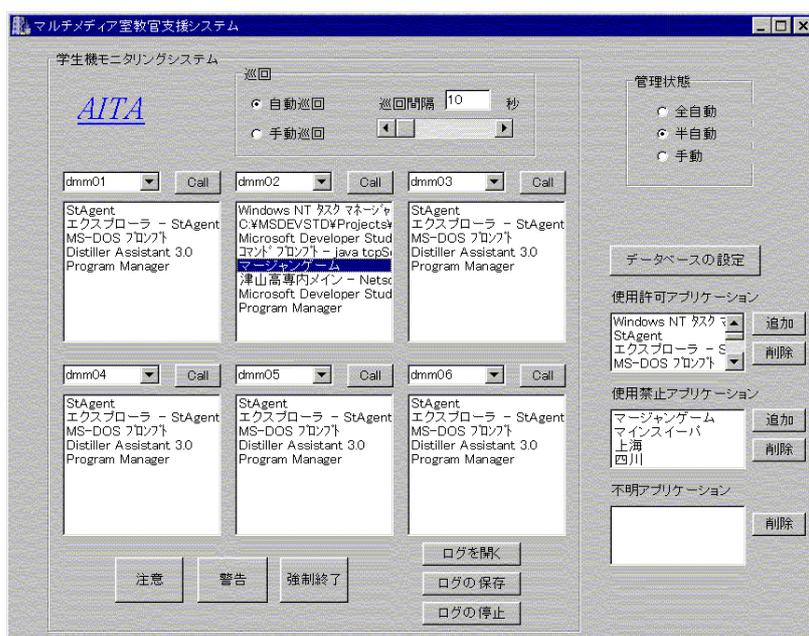


図 3-2 ウィンドウ名を表示する教員機インタフェース

しかし、学生機で起動しているウィンドウ名は表示に統一性がなく、アプリケーションによってはファイル名のみが表示されるなど、正確にアプリケーションを特定できない事例が見つかった。そこでシステムからプロセス名を取得しアプリケーションの分類を行った。各学生機の起動しているプロセスの分類とアプリケーション強制終了の様子を図 3-3 から図 3-5 に示す。図 3-3 にアプリケーションの分類ダイアログを示す。これはシステムが過去に収集した既知のプロセス名を読み込み、「許可アプリケーション」と「禁止アプリケーション」への分類を行う。このダイアログにより、数多く存在する Windows のプロセスの分類が可能となった。

また各パソコンを巡回して表示する形式は、試用した教員より使いにくいという意見が出されたので、表示方式を変更して演習室全体が見えるように変更した。その画面を図 3-4 に示す。これは、1 番の学生を選択すると、その学生が起動しているアプリケーションの一覧がウィンドウの右上に表示され、使用

を許可しているアプリケーションには○印のアイコンを、使用が禁止されているアプリケーションには×印のアイコンがつけられる。禁止アプリケーションへの警告を行うように設定すると、学生機上で禁止アプリケーションを発見した時は図 3-5 のように学生機の画面上に警告パネルを表示して警告する。アプリケーションの起動を完全に禁止するのではなく、一定時間内の起動についてのみ許可するといった設定も可能である。

次に学生のキーボードの操作状況を表示した例を図 3-6 に示す。キーボード操作の情報を収集するエージェントからは学生の平均入力速度、キーボードの押し下げ時間など以下のような情報が送られてくる。

- 入力速度（キー入力／分）
- アルファベットならびに数字の入力数
- バックスペースならびにデリートキーの入力数
- その他のキーの入力数
- キー押し下げ時間の平均値・分散値
- キー入力間隔の平均値・分散値

これらのうち各学生のキー入力速度（毎分の入力数）を教員機上に表示しているのが図 3-6 である。左側には全学生の入力速度を棒グラフとして表示している。この表示データは学生が操作してから約 30 秒程度のタイムラグ後に更新される。このうち一人の学生を選択（この場合は 2 番の学生）すると、右側にその履歴を折れ線グラフとして表示する。

更にシステムが収集した学生のキーボード操作のデータを分析した結果を図 3-7 に示す。図 3-7 は二人の学生のキーボード操作において、タイピングの速度に対するキーの押し下げ時間の平均値を一定期間継続して測定し、その分布を示した例である。各プロットは 10 秒ごとのタイピング速度とその時の押し下げ時間の関係を示している。

この分析結果よりタイピングの最高速度は(b)の学生がごくまれに 300 [strokes/minute] 程度まで速くなることがあるが、いずれも約 250 [strokes/minute] を上限としている点では共通しており大きな差異はない。これに比べキーボードの押し下げ時間の分布に関しては表 2-1 に示すように明らかな違いが見られた。

表 3-1 キー押し下げ時間の分散値

学生	分散
Student A	20.34
Student B	12.26

図 3-7 (a)の学生はパソコン経験の短い（1 年以内）学生であり，図 3-7 (b)の学生は(a)の学生に比べ長い経験（3 年以上）を持つ。

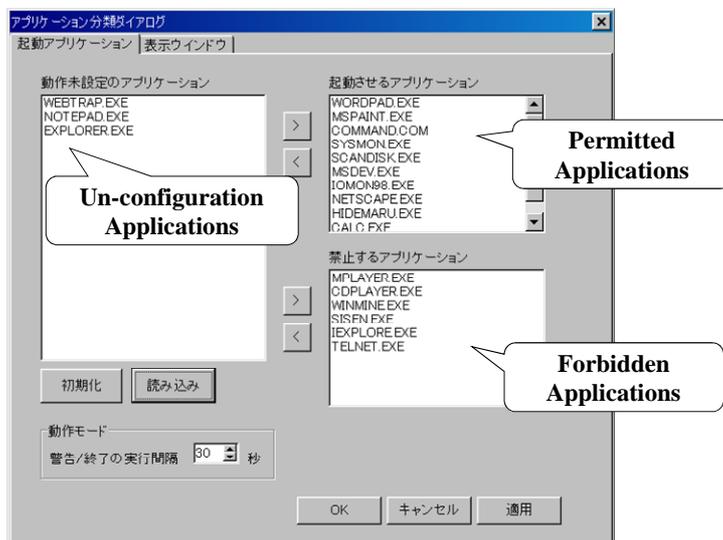


図 3-3 アプリケーション分類の設定画面

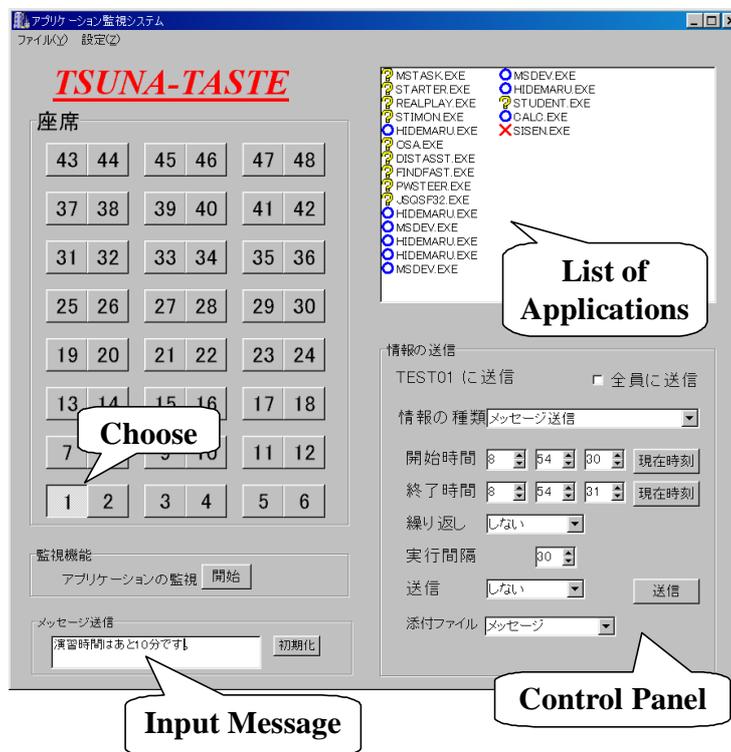


図 3-4 各学生機上のアプリケーション管理



図 3-5 禁止アプリケーションの強制終了

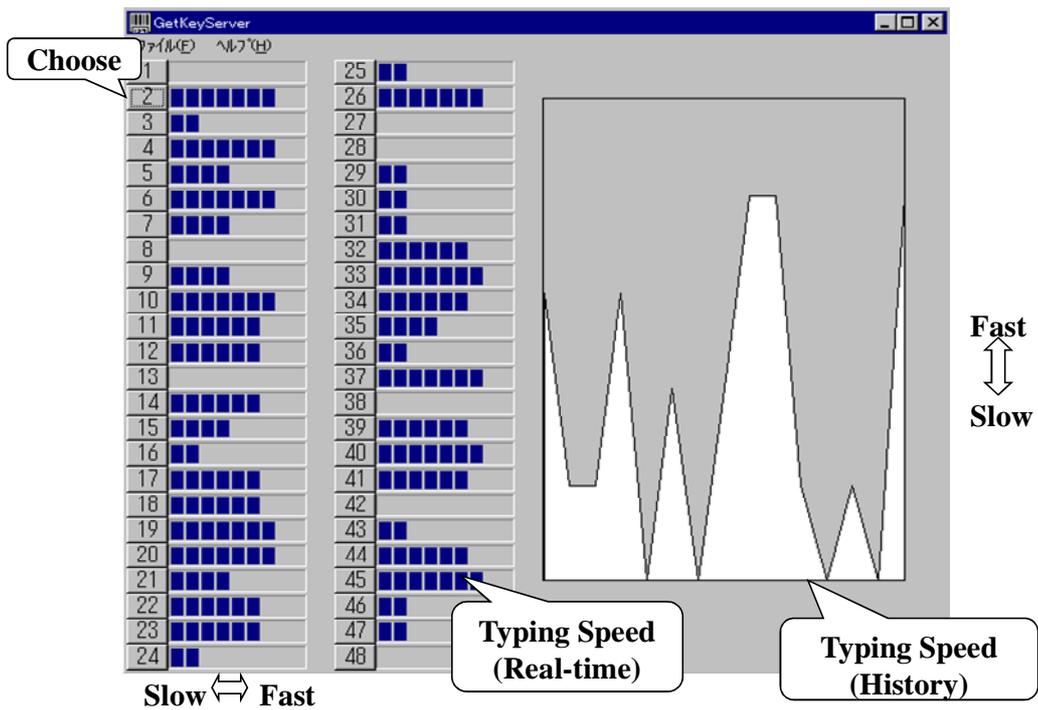


図 3-6 キーボード操作状況の表示

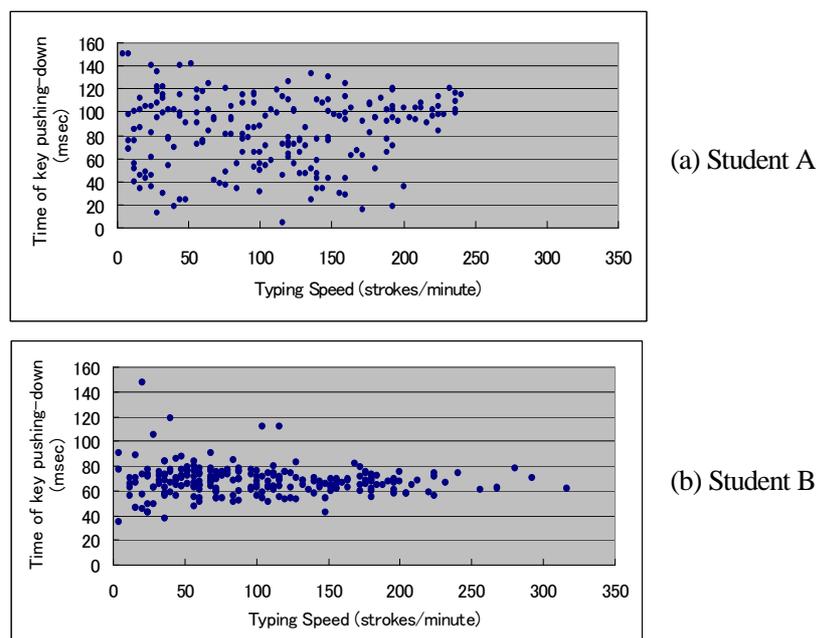


図 3-7 キーボード押し下げ時間の分布

3.3. TSUNA-TASTE システム

3.2 節で示した授業支援システムはプロトタイプ的なシステムであり，汎用性，拡張性をあまり考慮していなかった．そのため以下のような問題点が明らかとなった．

- 教員が使用するソフトウェアが専用アプリケーションであるため，機能追加を行うためには再インストール作業が必要である．
- 学生機と教員機の接続関係が固定されているため，複数の演習室を横断的に管理することが不可能であり，複数の演習室を使用する講義・演習に対応できない．
- 学生機で同時に複数の機能を動作させることが困難である．

以上の問題点を解決するために第 2 段階として以下のシステム変更を行った．

- 教員が操作するアプリケーションを専用アプリケーションからブラウザ経由で閲覧および操作可能なシステムに変更する．これによりインストール作業が不要となり，LAN 内であればどこでも利用可能となる．
- データベースシステムを導入し複数の演習室を統一的に管理できるようにデータの一元管理を行う．
- 学生機で動作させる機能をそれぞれモジュール化し，各モジュールを統合的に管理するソフトウェアを導入することで複数の機能を同時に実行可能とする．
- 将来サーバの負荷が増加することが予想されるため，複数のサーバで機

能を分割できるよう分散処理が可能なシステム構造とする。

この段階でシステム名を授業支援システム **TSUNA-TASTE** とした。これは **Tsuyama National College of Technology Teaching Assistant System** の略であり、**Teaching Assistant** をネットワーク上に構築したいという希望を示している。ネットワークを利用した起動アプリケーションの調査やパソコンの遠隔操作、システムの利用制限を行うソフトウェアは既に市場に複数存在しているが、本システムは教員が求める情報の収集と蓄積、データの分析、そして学生の指導をすべて統合的に行うことを目的としている。またデータベースを利用することにより設定情報の変更や新しい機能の追加といった柔軟性が増し、ブラウザを利用したデータの利用範囲の拡大が可能となった。

3.3.1. TSUNA-TASTE システムの概要

授業支援システム **TSUNA-TASTE** は、ネットワーク分散技術とデータベース並びに **Web** サーバを基盤技術として設計を行った。これにより、まず複数教室の管理が可能となった。さらに教員が複数同時に接続し演習を行うことが可能になった。さらに学生自身が自分のデータを確認できるようになり、システムとしてオープンな構造となった。図 3-8 に **TSUNA-TASTE** の概略図を示す。

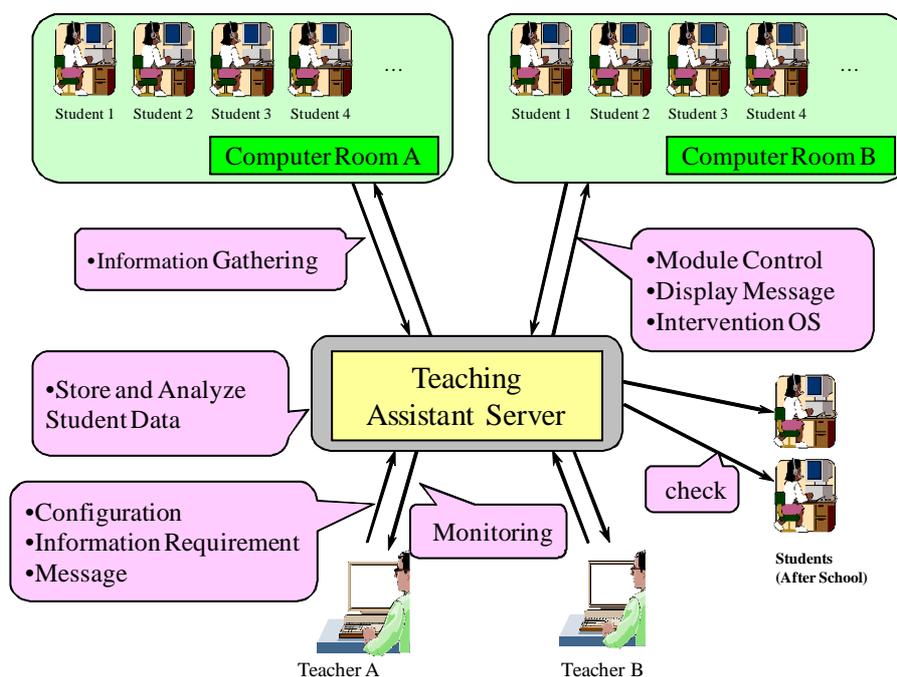


図 3-8 TSUNA-TASTE の概略図

3.3.2. 学生機エージェント

授業支援システムのうち学生機上で動作するソフトウェアを特に学生機エー

ジェントと呼んでいる。ここでは、この学生機エージェントについて詳細に述べる。学生機エージェントは、学生機パソコン上で情報の収集やサーバへのデータの送受信を行うソフトウェアの総称である。パソコン操作の情報収集やメッセージ表示など表 3-27 に示す 7 種類の異なった動作を行う。このように各機能はそれぞれモジュールという処理単位に分割している。それぞれのモジュールは比較的単純な動作のみを行うように設計しており、学生機エージェントがこれらのモジュールの動作を一元管理する。このように処理機能をモジュールに分割することにより、学生機エージェントへの機能追加が容易となり、また不要なモジュールは実行しないため、メモリ消費と CPU 負荷を少なくできる。モジュールについては 3.3.3 項で詳細に述べる。

学生機エージェントは学生機上のメモリに常駐し、授業支援サーバからのメッセージに従い学生機上でモジュールを起動し、必要があればサーバへデータを転送する。学生機エージェントが起動するモジュールにはシステム情報の収集や画面へのメッセージの表示、設定ファイルの書き込みなど幾つかのタイプがあり、動作も一度だけの実行で終了するタイプ、繰り返し処理の必要なタイプなど複数ある。このように異なったタイプのモジュールを円滑に動作させるために次の 4 つの仕組みが必要となる。

- サーバからメッセージを受け取る部分 (Receiver)
- 受け取ったメッセージを解析して、作業 (タスク) の管理を行う部分 (Kernel)
- 情報収集など実際の作業を行う部分 (Module)
- 収集したデータをサーバに送り返す部分 (Sender)

この学生エージェントの内部構造を図 3-9 に示す。Receiver は常にネットワークのポートを監視し、サーバからのメッセージを待つ。サーバからメッセージが送信するとメッセージに含まれるヘッダ部分を解析して (図 3-9①) その妥当性を検証する。メッセージが妥当であれば Kernel にデータを渡し、モジュールの実行時間、繰り返しの有無、繰り返し回数など実行のスケジュール情報を学生機エージェント内部に持つタスクテーブルに順次格納する (図 3-9②)。Kernel は定期的にタスクテーブルを調べそのスケジュールに基づきモジュールの起動・終了を管理する (図 3-9③)。起動されたモジュールは独立して動作し、情報を収集するタイプのモジュールは収集したデータを一時ファイルに記録し、学生機エージェントにデータの送信を依頼する (図 3-9④⑤)。データ送信の依頼を受けた Sender はデータに必要なヘッダ情報を付加して送信を行う (図 3-9⑥)。

また、学生機エージェントの設計において次の三つの事が重要となる。i) 画面上でも速度的にも学生にその存在を意識させない。ii) パスワードなどセキュリティ

ティに関係する情報をそのままネットワークに流したりネットワークのポートがウィルスの進入口にならないようにする. iii)小さなデータベースを持ち, 比較的簡単な判断は自立的に行えるよう設計する.

これら学生機エージェントの機能は, 使用するパソコンの OS に大きく依存する. 最初に作成した学生機エージェントは Microsoft 社の Windows 95 (一部 Windows NT でも可能) を基準に作成を始めた. 現在は Windows XP, Windows Vista, Windows 7 をベースとしている. Windows Vista, Windows 7 でも動作可能であるが, OS のセキュリティが厳格になり, インストールと運用には工夫が必要になっている. とくにネットワークの通信においてはパーソナルファイアウォールの設定を正しく行わないとサーバとの通信ができない場合がある.

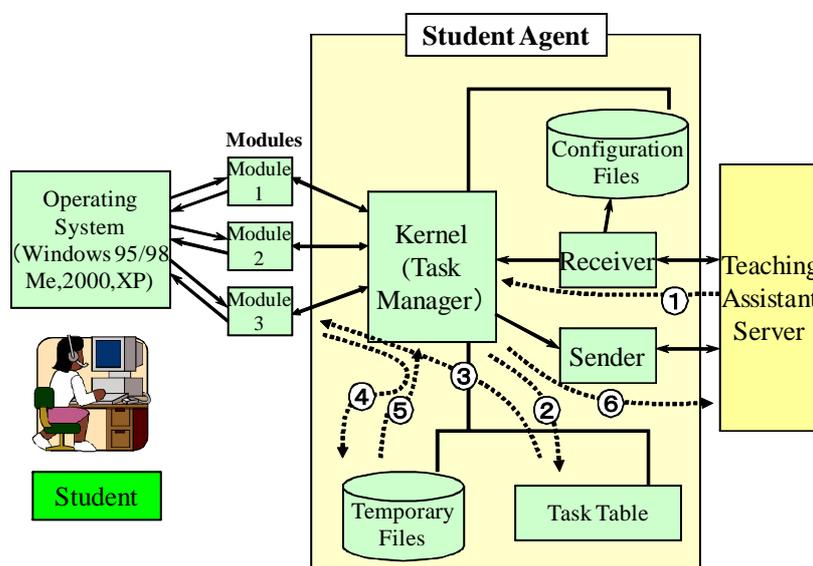


図 3-9 学生機エージェントの内部構造

3.3.3. 学生機用モジュール

学生機エージェントにより起動されるモジュールは, 学生機エージェントとは独立して動作する. これは学生機エージェントをまったく変更することなく新規のモジュールを追加するだけで機能の追加が可能であることを意味している. また学生機の動作に影響が出ないように, 個々のモジュールをできるだけ小さな機能に分割し, 実行サイズを小さく (50KB~200KB 程度) している.

モジュールの動作には次の 3 タイプある.

- 一度だけの実行で処理が終了するタイプ (ファイルの転送, メッセージ表示等)
- 定期的に行う必要があるタイプ (実行アプリケーション名の取得等)

- 継続的に実行されるタイプ（キーボードやマウスの操作状況の記録等）

3.3.2 項で述べた **Kernel** がこれらのモジュールの管理をタスクテーブル上で行う。各モジュールはプロセス ID で管理されているため、異なったモジュールを同時に複数起動することが可能である。ただし同種のモジュールを複数同時に起動することは禁止している。これは繰り返し実行のようにサーバから定期的に起動命令が来た場合、同じモジュールが複数起動して混乱しないように、排他的な処理を行っている。もし、同時に同種のモジュールが必要になった場合は、モジュールをコピーして名称を変更することで対応する。

教室内に配置される学生機エージェントと授業支援サーバの間をネットワークで接続し相互に通信する必要がある。この通信には **TCP/IP** ソケットを利用したメッセージ交換方式を採用した。メッセージは 3.4.5 項に述べる宛先や通信の正当性を確認するヘッダ部分と送信データのボディ部分から構成される。**Receiver** はデータを受信する場合、最初にヘッダ部分の妥当性の確認を行う。本システムはその性格上ネットワークからの不正な侵入には十分注意しなければならない。このため **Receiver** はパスワードによる認証を行い不正なパケットの侵入を防ぐようになっている。また、ポート番号についてもチェックを行っている。認証後に、データの宛先や生成時間、ボディ部分のサイズのチェックを行う。機能的にはキーロガーと同様にユーザの入力を記録する事も可能であるが、万一に事を考えて、そのようなデータをそのままネットワークへ流すことはしていない。また安全面から **SSL (Secure Sockets Layer)** など暗号化された通信を行う事が望ましく、パソコンの能力が許せば **SSL** による通信も検討している。ただし、**SSL** を導入するにはサーバ証明書を購入する費用や定期的な発生する証明書の更新などの問題があり運用が複雑になる。また各クライアントからの通信が集中するサーバで復号化処理により **CPU** が高負荷になることが懸念されたため、学内の **LAN** で使用する場合は簡単な共通鍵暗号方式による暗号化のみを行っている。今回実装しているモジュールの一覧を表 3-2 に示す。

表 3-2 モジュール一覧

ID	Name	Function
1	CollectProcess95	Collection of process name from Windows 95/98.
2	CollectProcessNT	Collection of process name from Windows NT, XP, Vista, 7.
3	CollectWindow	Collection of a Windows name.
4	AppDelete	Termination of application process.
5	GetMouse	Get mouse operation information.
6	GetKey	Get keyboard operation information.
7	DisplayMessage	Indication message on Student window.

3.4. ネットワークによる授業支援

授業支援システム TSUNA-TASTE は 3.1 節で述べたようにネットワークを経由してデータを収集する事を基本に開発を行ってきた。これは開発当初から一貫したものである。ただ、情報の集め方や転送方法に関しては、転送される情報量の増大に対応してサーバの性能やネットワーク構成を変更してきた。ここではネットワーク全体の構造と、そこでやりとりされるデータの形式について述べる。

3.4.1. ネットワーク構造

従来の TSUNA-TASTE は教員機上に独自のインタフェースを作成し、そのインタフェース画面を使ってシステムと対話を行う構造となっていた。演習中はこの構造で問題ないが、学生の操作履歴等は各研究室などからも自由に閲覧できると便利である。そこで新たなインタフェースとして Web ブラウザを使用する方法を検討した。この構造にすれば学生自身も自分のデータを閲覧することが可能となる。また、接続するユーザの情報や講義のデータなど扱うデータの種類が増加したため、システムが扱うべきデータ量やデータの種類が増加した。これらデータの管理に本格的なデータベースシステムを導入した。

これらのシステム構成は現在多くのシステムで用いられている 3 層構造モデルをベースにしている。Web システムのサーバ群は、一般的に役割を 3 層に分割し、それらの独自性を高めることで、システムの拡張性やメンテナンス性の向上を目指すシステムが 2005 年頃から急速に拡大してきた。3 層は次の 3 つのシステムからなる

- 1) Web サーバ : Web ブラウザからのリクエスト受け付け、データの表示を行う。
- 2) アプリケーション(AP)サーバ : 論理的な演算処理を担当し、各種の判断を行う。今回は授業支援サーバがアプリケーションサーバとなる。
- 3) データベース(DB)サーバ : データの保存・参照・更新を行う。

AP サーバには、任意の処理を実行するプログラムの稼働環境となるミドルウェア「AP コンテナ」を配置する。具体例として、例えば「JBoss Application Server」(米 Red Hat) [22]やオープンソースの「Apache」[23]「Tomcat」[24]などがある。今回は Apache と親和性の高い Tomcat を使用した。また開発言語は主に Java を使用した[25]。図 3-10 に授業支援システム TSUNA-TASTE のネットワーク構造を示す。

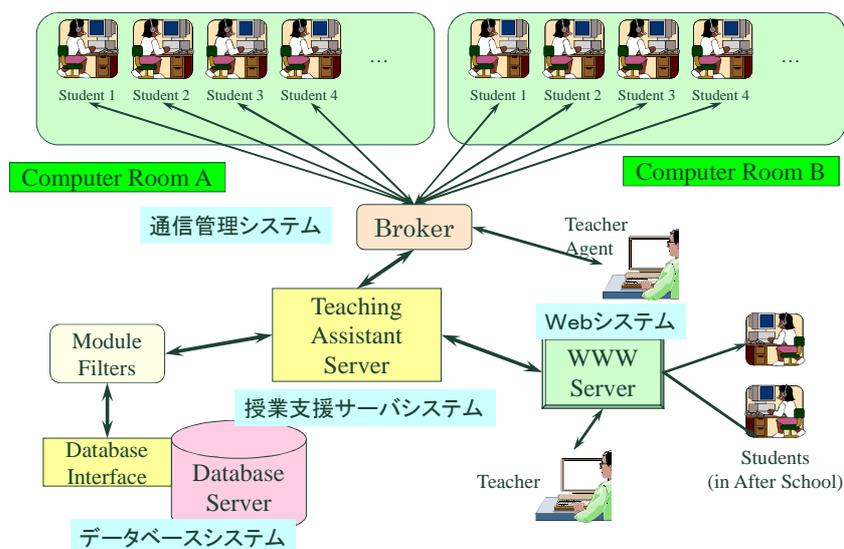


図 3-10 TSUNA-TASTE のネットワーク構造

3.4.2. WWW サーバ

授業支援に関するデータはすべてデータベースサーバに格納されるが、そのデータベースに対して直接コマンド (SQL 文) を発行して検索や設定を行うのは現実的でない。そこでデータの検索等は WWW サーバ上のスクリプトとして実行し、各ユーザはブラウザから必要なメニューを選択するだけでデータにアクセスする構造が必要である。必要なデータは授業支援サーバを経由してデータベースサーバから得る。WWW サーバを使用することにより、以下の利点がある。

- 同一ネットワークであればどこからでもデータへアクセスできる。
- データの設定・表示のインターフェースが作成しやすい。
- インターフェースとして Web ブラウザを使用すれば、特別なアプリケーションをインストールする必要がない。
- データベースサーバと連携することで安全性を確保しやすい。

一般的に Web サーバには、Web ブラウザとの通信機能を持つ常駐サービス「httpd (http デーモン)」を配置する。代表例は、オープンソースの「Apache HTTP Server」である。本システムでも Apache を使用した。

3.4.3. データベースシステム

データベースサーバは講義に関連する全てのデータを管理する。データはそれぞれ a)担当教員名や各クラスの学生個人データ (氏名や学籍番号等) など通常 1 年間を通して変化しないデータ, b) 学生の座席位置に関する情報など定期的・突発的 (パソコンの故障等) に変更される可能性があるデータ, c) 学生機から収集したデータ, 更に d) その他講義に必要な付随的なデータに分類され

る。これら複数のデータを有機的に結合して授業支援をおこなう必要がある。データベース内のデータ構造を図 3-11 に示す。

学生機エージェントの各モジュールが収集したデータは、使用するモジュールごとに固有のデータ形式を持っている。そのため、これらのデータをデータベースに格納する場合にはそのモジュール専用のフィルタが必要となる。このモジュール用のフィルタでデータの整形を行った後に、データベースインタフェースからデータベースへデータを格納する。データを読み出す場合も同様である。

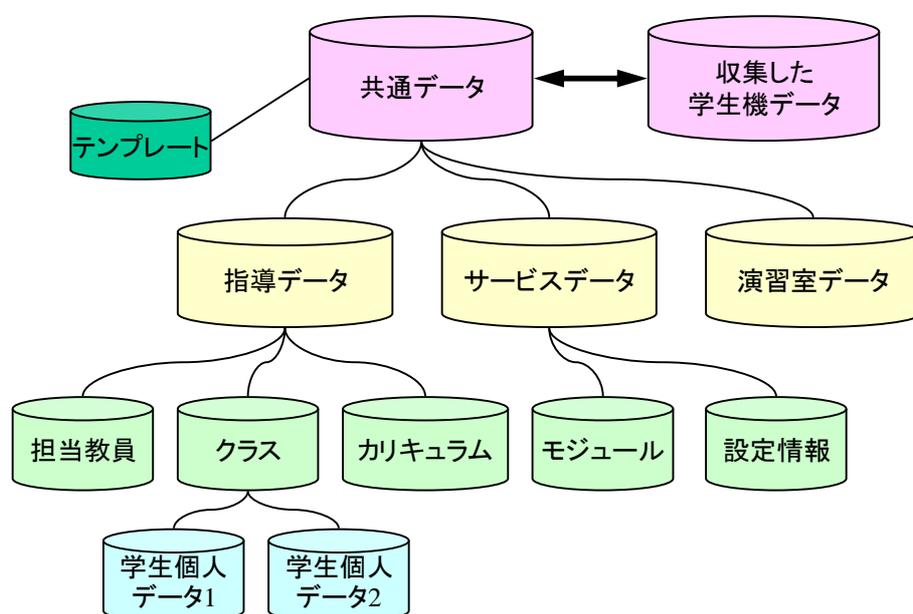


図 3-11 データベースの内部構造

データベースを導入してデータを一元管理できるようになると、データへのアクセス権とセキュリティの確保が非常に重要となる。また、学生の授業に関するデータの取り扱いは、漏洩による問題が発生しないよう慎重に行う必要がある。そこで本データベースでは、システム管理者、授業担当教員、その他教員、受講学生、学生本人、その他学生のようにユーザと講義との関係を明確にし、そのアクセス権を管理する。

DB サーバは、データの管理機構を備えたミドルウェアである「DBMS (Database Management System)」を配置したサーバである。具体例として、例えば「Oracle Database」(米 Oracle) [26]やオープンソースの「PostgreSQL」

[27]「MySQL」[28]などがある。今回はシステム全体をフリーウェアで構築しており、また、設定関係の資料が豊富であることと JDBC ドライバ[29]が安定していたことからデータベースには「PostgreSQL」を選択した。「PostgreSQL」は BSD ライセンスに類似するライセンスにより配布されているオープンソースのオブジェクト関係データベース管理システムである。

3.4.4. ブローカシステム

Web システムの基本は上記の 3 層構造となる。しかし、授業支援システムに関しては通信の主体が演習室のパソコン群であり、通常の Web サービスとは情報の流れ（トラフィック）が少し異なる。そのため通信の中心部分にデータの仲介を行うシステムとして「ブローカ」を挿入することとした。現在ではサーバの負荷を分散するためにロードバランサーや Layer 7 Switch など、多くの技術や製品が存在するが、開発当初は非常に高価であった。そこで、これらの装置と動作原理は異なるが、独自にネットワークのトラフィックをとりまとめるシステムを構築した。

授業支援システムでは通信の多くが学生機と支援サーバの間で行われるため、授業中は授業支援サーバに通信が集中する。通信負荷を考慮すると、授業支援サーバから通信機能の一部を分離させた方が効率がよいと考えた。そこで各学生機エージェントと授業支援サーバの間の通信を扱うブローカを導入し、学生機の台数増加やブロードキャスト通信の問題を緩和した。ブローカを介して通信を行う利点をまとめると次のようになる。

- 授業支援サーバの通信負荷を軽減。
- 複数の教室・科目への対応。
- 授業中の座席変更への対処。
- ネットワーク設定の自由度の増加。

支援システムの運用時には全学生に同じメッセージを送信する状況（ブロードキャスト配信）が頻繁に発生する。この場合ブローカまでは一つのメッセージで送り、ブローカで各学生に配分する方式を採用することで通信が効率的になる。そのためブローカは通信の最も多く発生する学生機の近くに置くことが望ましい。学内に複数の演習室があり授業支援サーバならびにデータベースサーバを（ネットワーク的に）近い場所に置くことができない状況でも、ブローカを演習室の近くに導入することで通信負荷を軽減できる。具体的には、演習用パソコンを接続するスイッチング HUB もしくは、その上位にある HUB に接続するのが理想的である。また演習室ごとにブローカを設置できれば更に望ましい。

別の問題として演習室ではパソコンのトラブルにより学生が急に席を変更しなければならぬといった状況が時々発生する。この場合、不完全なデータの

送信やソケットのコネクションが残り次回の接続に失敗する可能性がある。サーバ側でこの対応を行うことも可能であるが、ブローカで一度通信を全て受信し、データを整形してサーバに転送することでこの問題を解決した。これにより、学生が席を移動するなどの急なトラブルにも対応可能となった。

ブローカにはネットワーク機能の高いLinux系またはBSD系のサーバマシンが利用できるようJava言語を用いて開発を行った。なお参考のためにC言語でも開発を行ってみたが、ソケットが接続完了するまでの時間が増加する現象が発生したため、実用上問題となる可能性があったので通信部分に関してはJava言語を中心に開発を行った。

3.4.5. 通信データの構造

システムにおいてブローカシステムを効率よく動作させるには、通信データのデータ構造に関しても工夫が必要となる。これは教室内に配置される学生機エージェントと授業支援サーバの間をネットワークで接続し相互に通信する場合、その間にブローカが入り通信を効率よく振り分ける必要があるためである。この通信にはTCP/IPソケットを利用したメッセージ交換方式を基本として採用した。

通信するソケット上にヘッダ部分とボディ部分とから構成される通信データの構造を作成した。独自データ構造の作成は、汎用性を損なうイメージがあるが、通信そのものはTCP/IPソケット（ISOモデルの第5層）であり、この部分に関しては完全に互換を持っている。この上位である第7層のアプリケーション層での実装であり、通信の汎用性は損なわれない。また仮にブローカが動作していない場合でも、学生機エージェントからの通信の送り先として直接サーバを指定することで通信が可能である。

サーバから学生機エージェントへの送信、および学生機エージェントからサーバへの送信、さらにブローカ間との通信すべてにこのプロトコルを使用する。表 3-3 にそのヘッダ部分を示す。

3.3.2 で述べたReceiverがデータを受信する場合、最初にヘッダ部分の妥当性の確認を行う。本システムはその性格上ネットワークからの不正な侵入には十分注意しなければならない。そのためReceiverはトークンによる認証を行い不正なパケットの侵入を防ぐようになっている。表 3-3 の「Token for authentication」がこのトークンとなっている。トークンは学生（パソコンの利用ユーザ）がログインに使用するパスワード等とは無関係であり本システムのソケット通信のみで使用する。

このまた、ポート番号についてもチェックを行っている。認証後に、データの宛先や生成時間、ボディ部分のサイズのチェックを行う。

表 3-3 通信データの構造

Function	Data Size [byte]
Token for authentication	32
Message type	4
Sender group ID	4
Receiver group ID	4
Sender Name	4
Receive Name	4
Message ID	4
Message create time	4
Body data size	4
Option	64

通信部分に関しては SSL による通信の暗号化にも対応できるようにした。ただし、すべての通信を SSL で行った場合サーバの負荷が急激に増加するため、通信の安全性と利便性のバランスを考慮して導入する必要がある。また、サーバ証明書の問題から LAN 環境において SSL 通信を行うメリットはほとんどない。ただし、通信がインターネットを経由する場合は SSL による暗号化が必要になると考えられる。

3.5. TSUNA-TASTE の適用例

プログラミング演習やその他パソコンを使った演習に使用しているマルチメディア演習室のシステム上に TSUNA-TASTE システム、データベースサーバ更に Web を追加した場合の適用例を以下に示す。

図 3-12 に講義に関するデータを表示した例を示す。従来システムでは教員や講義によりサービスの内容を変更するには、システム自体の設定を変更する必要があり煩雑であった。本システムではデータベースサーバを利用して、カリキュラム表・教員・学生更に教室といった講義に必要なデータを一元管理することで、支援の内容を具体的なサービスとして提供することが容易となった。設定内容についても Web ブラウザ上で確認することが可能である。

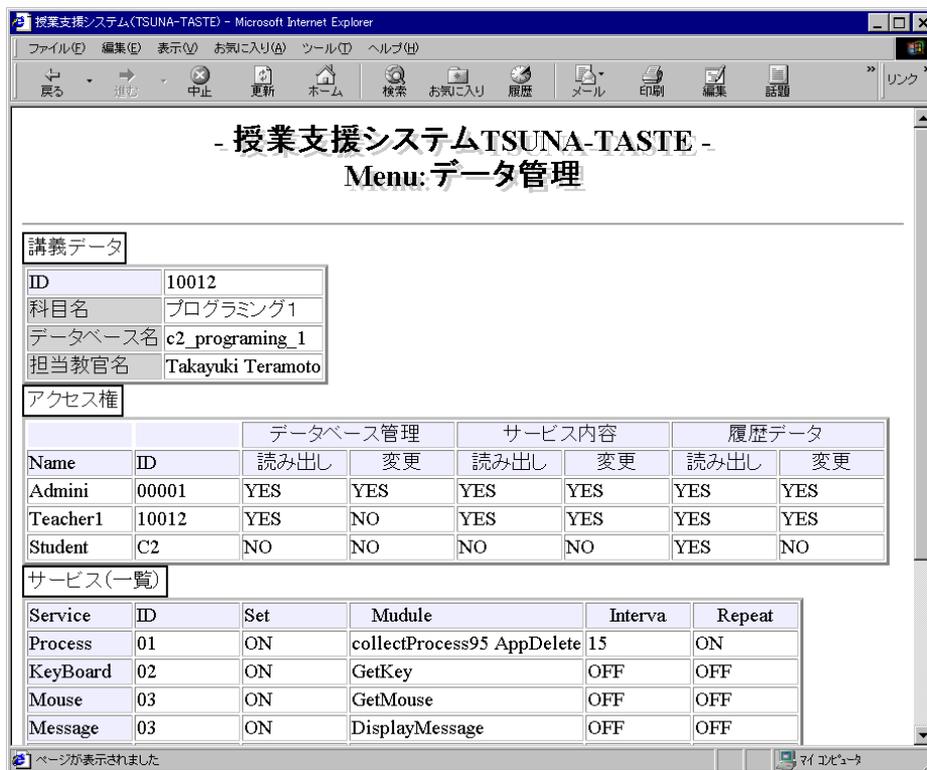


図 3-12 Web による講義データ設定画面

本システムは一部のデータに関して学生もデータベースへ接続し、データを調べることが可能となる。たとえば学生が自分自身のアプリケーション履歴やキーボードの入力速度等のデータを閲覧できれば、学習意欲や態度に効果があると考える。この場合、データへのアクセス権に関しては、データベースとWWWサーバを連携させて管理を行う。これにより非常に細かいアクセス制御を設定が可能になる。

図 3-13 に Web ブラウザによる教員用の管理画面を示す。Microsoft 社の Internet Explorer を標準のブラウザとしているが、Firefox や Opera など他のブラウザでもできるだけ同じレイアウトで表示されるよう、特別なタグやアドビシステムズ社の Adobe Flash などの使用は避け、テキスト情報主体で表示を行うようにしている。このため見た目はシンプルとなっている。

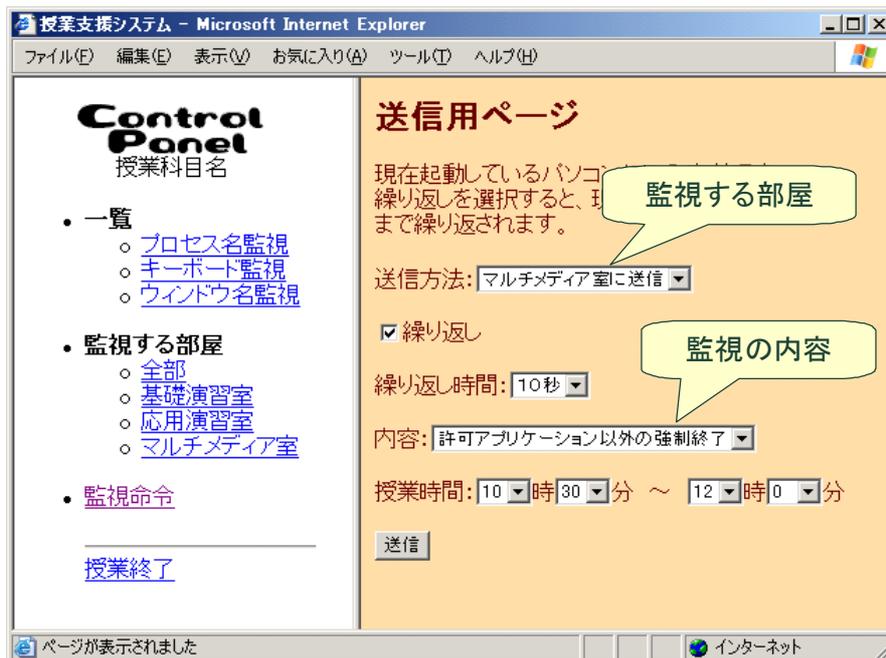


図 3-13 TSUNA-TASTE の管理画面（教員用）

図 3-14 に学生が起動しているアプリケーションを Web ブラウザで表示している例を示す。前述の図 3-4 と同様に、各学生が起動しているアプリケーションの一覧がウィンドウの右上に表示され、使用を許可しているアプリケーションには「○」印のアイコンを、使用が禁止されているアプリケーションには「×」印のアイコンがつけられる。起動しているアプリケーションがシステム使用最小限のもの（起動直後の状態）の場合は「OK」と表示される。また未分類の場合は「？」印がつけられる。図 3-4 が教員専用のウィンドウであり学生へのメッセージの送信など多くの機能を持つのに対し、図 3-14 の機能はアプリケーションを分類して一覧表示することに限定されるが教員だけでなく学生もそのデータを閲覧できる。教員と学生の双方が Web ブラウザにより閲覧できる環境を整えることで以下のような効果が期待できる。

- 教員は学生の操作履歴を有効に利用できる。
- 学生自身も自己評価を行うことができる。
- 講義に関するデータを一元管理することで、広範囲な支援を行うことができる。

また図 3-15 にはキーボード・マウスの操作状況を示す。

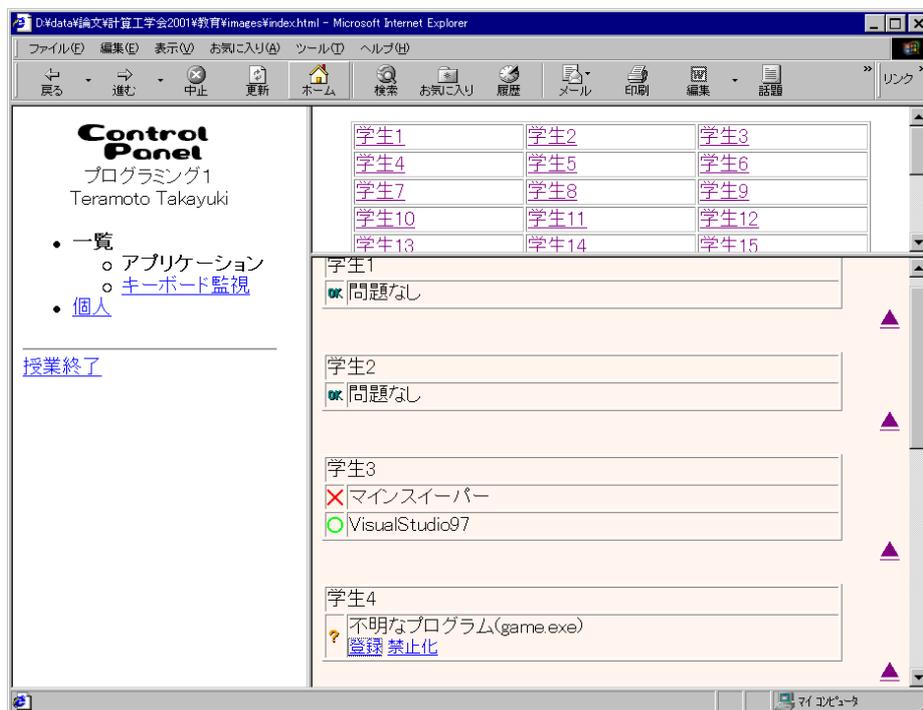


図 3-14 起動アプリケーション一覧 (公開情報)

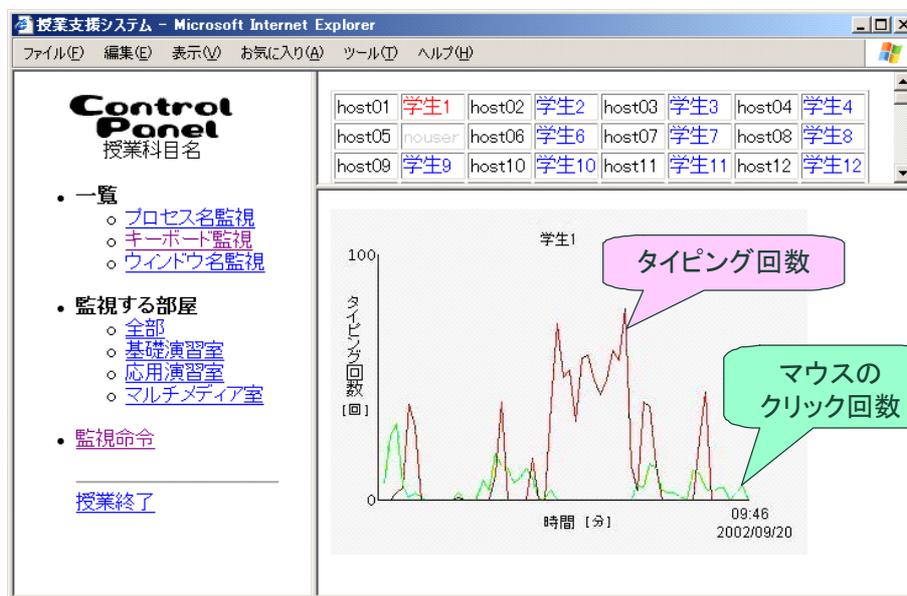


図 3-15 キーボード・マウスの操作状況 (公開情報)

3.6. プログラミング試験の支援

コンピュータを利用した授業が増加することにより、コンピュータを使った試験を行う必要性や要望が増加してきている。そこで本システムをプログラミング試験の支援に適用することで、ネットワークに接続されたパソコンを利用

した場合でも公正に試験を行うことができるようなシステムの構築を行った。もしこの機能が実現できれば Web ベースの e-Learning システムでも公正な試験が可能になり適用範囲は非常に広いと考えられる。高度情報通信ネットワーク社会推進戦略本部 (IT 戦略本部) が掲げる「e-Japan 重点計画」では、「教育及び学習の振興並びに人材の育成」の中で日本での e-Learning 普及を大きなテーマに掲げており、今後はますます e-Learning の利用が増えることが予想される。[30]

試験で使用するコンピュータがスタンドアローンの場合、各パソコンのアクセス権を適当に設定することで比較的容易に公平な試験に適用させる事ができる。しかし現在のパソコン演習室はユーザ管理をサーバで行い、ファイルもローカルディスクではなくサーバ上のストレージに保存される。また Web ベースの e-Learning システムは当然ネットワーク接続が必須条件となる。この場合すこしコンピュータに詳しいユーザなら容易に他人と連絡を取ることが可能で、公正な試験の実施が困難な状況であった。

試験を行う上で問題となるのは不正なアプリケーションの使用による不公平が生じることである。不正なアプリケーションを利用してどのような事が可能か検討した結果、以下のような事が考えられた。

- メールソフトによる情報交換
- チャットによる情報交換
- ファイル共有による解答のやりとり
- ブラウザによる不必要な検索
- 不要アプリケーションの起動 (電卓を使用してはならない試験で電卓のアプリケーションを起動するなど)

アプリケーションの起動を制御する市販ツールもあるが、教員が必要な時だけ、必要なアプリケーションを自由に設定できるツールは少なく、各種の試験に適用するには不十分と感じられた。

また、これらの行為を Web ベースの e-Learning システムで防ぐ事は原理的に不可能である。Web ベースのシステムは、ブラウザ内で行われる動作に関しては制御可能であるが、それ以外のアプリケーションに影響を与える事は不可能である。

授業支援システム TSUNA-TASTE は学生機上のアプリケーションを制御することが可能であり、試験を公平に実施することが可能な環境が提供できると考えた。TSUNA-TASTE は禁止アプリケーションを起動できないように設定する機能を有している。これをさらに厳しく設定するために許可されたプロセス以外はすべて起動不可能とするモジュールを作成した。このモジュールにより、パソコンを使ってもプログラミングの試験が公平に行える環境が構築できた。

今回のプログラミングの試験では以下のものだけを使用許可とした。

- エディタ (bcpad.exe, TeraPad.exe, AzEdit.exe)
- メモ帳
- コマンドプロンプト
- コンパイラ (bcc32.exe)
- テスト用ファイル (test01.exe, ……test04.exe)
- システム用プロセス

これらの設定ファイルは、試験科目や状況により変化する可能性があるので、あらかじめ各学生機に配布するのではなく、ネットワーク上にあるサーバからモジュールの起動毎にコピーする設計とした。この機能により教員であれば試験中でも自由に変更することが可能となった。

また本校の場合 3つのコンピュータ演習室があり、これら全てを 1つの Microsoft Windows Server のアクティブディレクトリの単位として管理している。しかし全ての教室で試験を行う事はほとんどなく、演習室ごとに異なった運営を行う必要がある。そのため、教員が試験開始と同時に設定を有効にできるよう、設定の変更は全て Web 上で行うことができるようになっている。試験開始時に教員は管理用の Web サーバにアクセスし、図 3-13 のような制御画面から設定を変更したり、学生機エージェントに指令を出したりすることができる。演習室名や、制御の方法、監視の内容などを自由に設定することができる。

これ以外に Web 上では表示していないが、図 3-16 に示すようなコンパイル実行回数などについても情報を収集している。

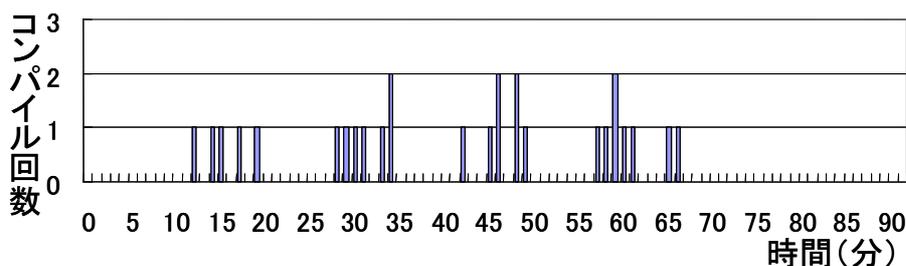


図 3-16 コンパイルの実行回数 (学生 1 名分)

なお TSUNA-TASTE システムそのものプロセスを停止させられると、情報を収集することができなくなってしまう可能性がある。これを防止するためサーバ側で電源が入っている (ping に応答する) 状態であるにも関わらず、TSUNA-TASTE の学生機エージェントから応答がない場合は教員側に警告を出すよう設定している。

今回実際に本校情報工学科 2 年生 (46 名) の「プログラミング 2」学年末試

験（90分）に適用し、その効果を調査した。実施後に各学生の操作記録を詳細にチェックし、不正が行われていないことを確認し、公正な試験が実施できたことが確認できた。

今回の試験は Web ブラウザを使用していないが、授業支援システムはウィンドウタイトルによる制御が可能のため、Web ブラウザを利用した e-Learning に適用することも可能である。この機能により目的としている Web サーバ以外の URL には接続できないようにする事が可能である。ただし、厳密な運用に関しては、本システム以外にも proxy サーバの設定など、いくつかネットワークに関するサポートが必要と思われる。

3.7. コンパイル情報を利用したプログラミング支援システム

本授業支援システムの適用目標はプログラミング教育を支援することを主な目的としている。プログラミング学習は演習が非常に重要であり、しかも演習時に学習者への適切な指導・アドバイスを必要とする分野である。また、学習者のレベルや状況により柔軟に対応することも求められる。プログラミングのソースコードに対して自動的に評価したり、アドバイスを与える研究も多くある [31]–[44]。3.6 節で示したように、TSUNA-TASTE はパソコンを使用したプログラミングや e-Learning に応用可能なことが実証できた。しかし、プログラミングの試験に応用できたといっても、学生のプロセスを制御したにとどまっておき、学生がどのような問題でプログラミングができていないのか、またその場合に有益なアドバイスを与えることはできていない。またそのために必要な情報も不足している。

そこで学習者がどのようなコンパイルを行っているのか知るためのコンパイラ用モジュールを新たに作成し、学習者の学習状況をコンパイルのエラーメッセージから取得することを考えた。コンパイラのエラー情報から、どのような間違いを犯しているのか、また同じ間違いをすぐに解決できずに何度も何度もコンパイルを行う学生はいないか。さらにエラーの種類から学習者のプログラミングレベルにあった応答メッセージの返答を行うなどの方法が考えられる。学習者の理解度などの傾向を調べる為に、収集したコンパイル・エラー情報をもとに分析するためのシステムに関して述べる。

3.7.1. コンパイラ情報の取得

授業支援システム TSUNA-TASTE に学習者のコンパイル情報をリアルタイムで収集し、データベースに蓄積するモジュールの追加を検討した。収集する情報はコンパイル実行の日時、ユーザ名、送信元ホスト名、コンパイルしたファイル名、コンパイル結果文字列の 5 項目（図 3-17）となっている。

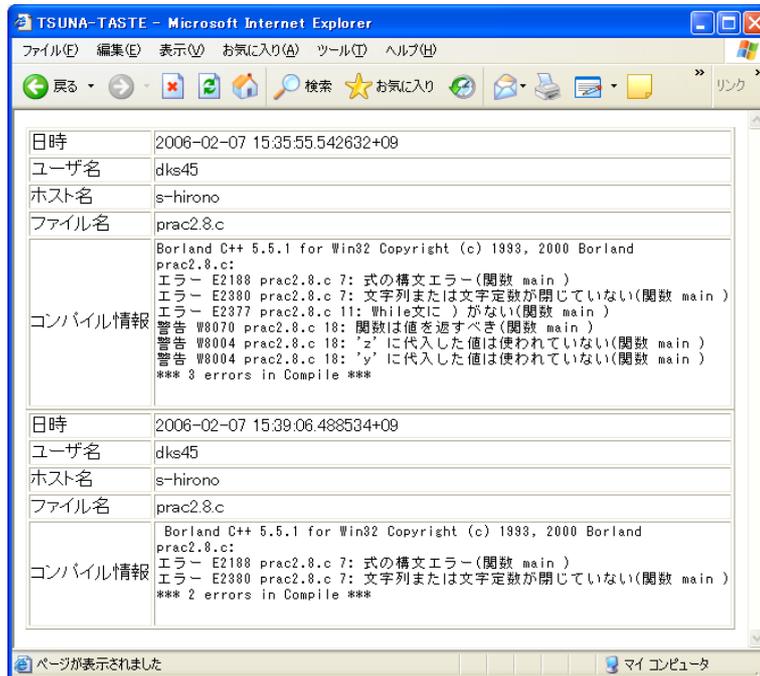


図 3-17 収集するコンパイル・エラー情報

3.7.2. コンパイラのラッピング

コンパイラがどのようなエラーコードを出しているのかをコンパイラから直接取り出すことは基本的に困難である。市販のコンパイラのようにそのソースコードが入手できないアプリケーションの場合、逆アセンブルを行う必要があるが、商用ソフトウェアでは使用契約によって逆アセンブルなどによるリバースエンジニアリングを禁止していることが一般的である。そこで、コンパイラそのものには手を加えず、これをラッピングしてコンパイラの出力だけを取り出し、それをサーバに転送する方式で情報を取得した。このメリットは、利用者に対してラッピングによる環境の変化を全く意識させることなく情報が取得できる点にある。

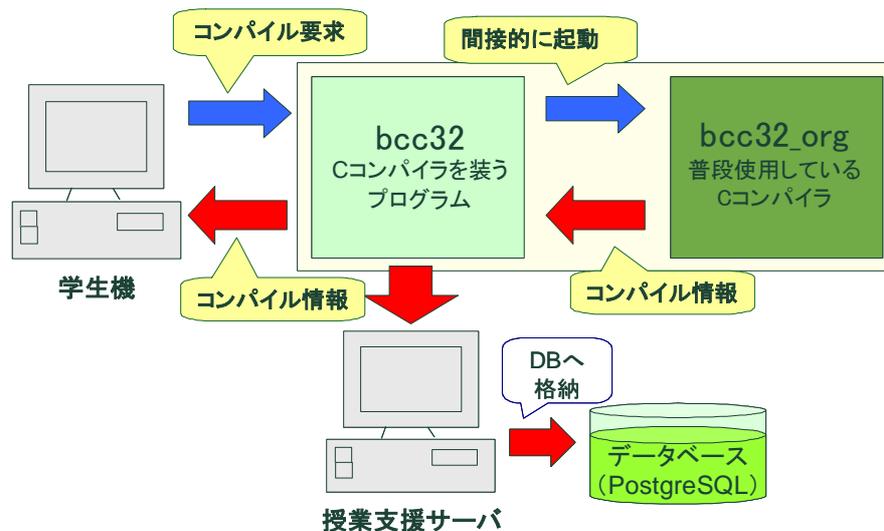


図 3-18 ラッピングテクノロジーによるコンパイル結果の取得

3.7.3. コンパイル・エラーの分類

コンパイラからの情報が収集可能となったので、収集したコンパイル結果のデータ分析を行った。データベースに登録されたデータから特定のデータを取得する為にはSQL文によるクエリをデータベースに送信して結果を得る必要がある。あるクラスにおける1ヶ月分の、コンパイル回数とエラー回数を表3-4に示す。

表 3-4 コンパイル・エラー回数 (月別集計)

日付	エラー回数	コンパイル回数
Apr-06	21945	36867
May-06	0	21979
Jun-06	776	2139
Jul-06	289	1120
Oct-06	415	1295
Nov-06	95	288
Dec-06	225	818
Jan-07	108	521
合計	23853	65027

4月、5月の回数が他の月に比べて突出しているが、これは教育システムのリプレース作業が完了していなかったため、送信が正しく行われずパッケージが輻

轉したために、正しくカウントできない学習者があったためである。この正しくカウントできなかったユーザに関しては、今回の評価からは除外し、システムが安定動作したときの情報を元に、特定のグループの 6 月（学習初期段階）と 12 月（学習中期段階）のコンパイル・エラー状況を比較したのが表 3-5 である。

表 3-5 学習時期によるコンパイル・エラーの比較

6 月		12 月	
エラー番号	回数	エラー番号	回数
E2451	73	E2451	36
E2379	54	E2379	32
E2040	35	E2194	28
E2225	1	E2380	1
E2027	1	E2048	1
E2277	1	E2323	1

ここでのコンパイル・エラーのエラー番号とエラーの内容の対応を表 3-6 に示す。

表 3-6 コンパイル・エラー番号の内容

エラー番号	エラーの内容
E2040	宣言が正しく終了していない
E2048	認識できないプリプロセッサ指令
E2054	else の位置が誤っている
E2134	複合文に } がない
E2141	宣言の構文エラー
E2188	式の構文エラー
E2190	不要な }
E2194	ファイルが見つからない
E2206	不正な文字
E2379	ステートメントにセミコロン (;) がない
E2451	未定義のシンボル

この対比から、ある程度の集団になると学習の初期段階と中期段階において、際立った違いが見つけにくいだが、特定のコンパイル・エラーに着目してみると、

初期に多く見られた E2040（宣言が正しく終了していない）が中期ではほとんど見られなくなり、代わりに E2194（ファイルが見つからない）が増えている、これは12月ごろの学習内容が「ファイル操作」に関するものであったことを反映している。

3.7.4. 個人ごとのエラー情報

グループにおけるコンパイル・エラーの傾向はクラス全体の傾向を調べる場合に有益である。また各個人の情報を分析し、それぞれのユーザにあったメッセージや指導が必要となる。表 3-7 にユーザ A とユーザ B のコンパイル・エラーの分布を示す。

ユーザ A は比較的プログラミングが得意な学習者であり、数多くのエラーを出しているが特定のエラーを続発させることはない。逆にユーザ B は初歩的なエラーを多数発生させており、コンパイル・エラーの意味を正しく理解できていないと思われる。またファイル構造を理解していないと思われるエラーも出しており、この点においても早急に指導が必要なユーザであることが予想される。

表 3-7 ユーザによるコンパイル・エラー情報

ユーザ A		ユーザ B	
エラー番号	回数	エラー番号	回数
E2040	9	E2194	22
E2188	7	E2379	15
E2134	6	E2054	8
E2377	4	E2188	8
E2379	4	E2451	5
E2376	4	E2141	5
E2054	3	E2376	4
E2194	3	E2377	3
E2380	2	E2193	3
E2206	2	E2134	2
E2378	2	E2128	2
E2127	1	E2087	1
E2451	1	E2206	1
E2209	1	E2127	1
E2087	1	E2380	1
E2141	1	E2378	1

TSUNA-TASTE では学生のモチベーションを向上させるために、収集したデータの一部を学生が閲覧できるようにしている。図 3-19 にユーザ名（この場合「s-taka22」）を指定して、エラー発生状況を確認したい月とそこから何か月分のデータを閲覧したいかを指定する。すると、月ごとにエラーコード別の発生数を知ることができる。このユーザの場合は 11 月をピークにして、その後コンパイル・エラーが激減していることがわかる。これ以外にも、ユーザを二人選んでそれぞれの月ごとのエラー数とエラー番号に対応した回数が表示される。このような情報を公開することで、学生からは非常に張り合いがあるというアンケート結果が出された。

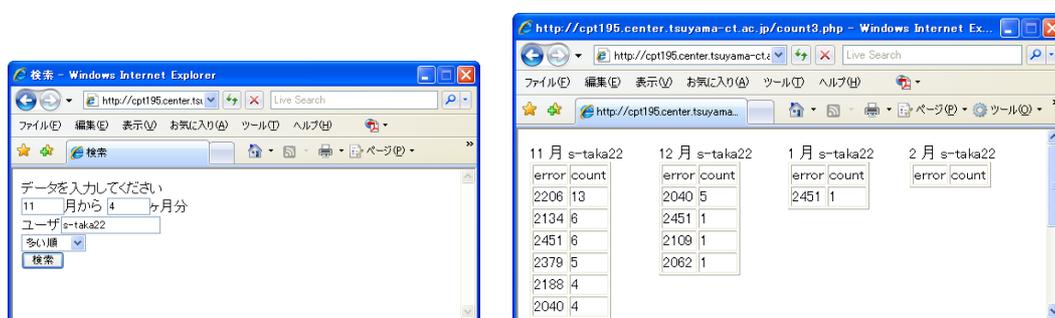


図 3-19 学生によるコンパイル・エラーの閲覧

3.7.5. 成績とコンパイル・エラー

コンパイル・エラーの発生状況から学習者の熟達度が推測できるかどうかを検討する必要がある。そのため、学習者のコンパイル回数、コンパイル・エラーの発生回数、そして学習者の熟達度の間に相関関係があるかどうか検証した。ここで熟達度は実技試験も含めたプログラミングの成績としている。

最初に成績に対するコンパイル回数について分析を行った。図 3-20 に成績とコンパイル回数の関係を示す。コンパイル回数は左側が前期分のデータを右側が後期分のデータを集計したものである。また表 3-8 に成績に対するコンパイル回数並びにエラー回数の相関係数を示す。これによると、成績に対するコンパイル回数は正の相関をもっていることが分かる。また後期になり学習が進むほど相関関係はより強くなり、学習者の熟達度とコンパイル回数にはある程度関係があることが証明された。

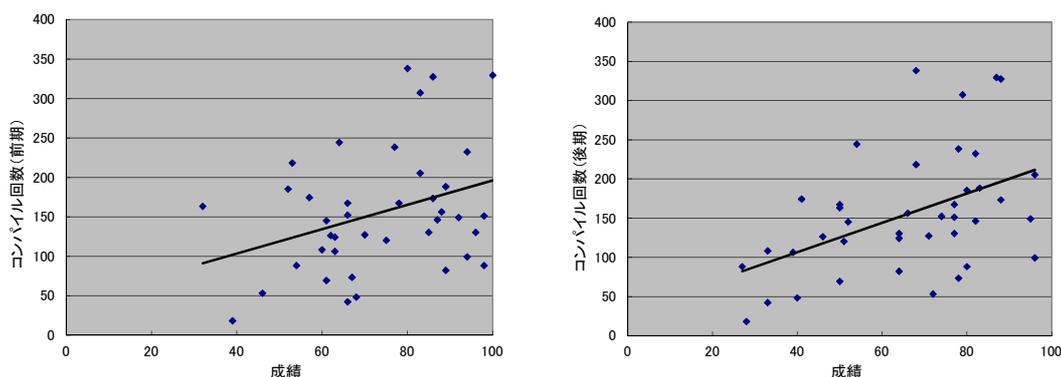


図 3-20 成績とコンパイル回数 (左：前期 右：後期)

表 3-8 成績とコンパイル・エラー回数の相関係数

	コンパイル回数	エラー回数
成績 (前期末)	0.33	0.25
成績 (後期末)	0.47	0.45

次にコンパイル時に発生したエラーの回数が成績とどのような関係にあるか分析したのが図 3-21 である。エラー発生回数は左側が前期分のデータを右側が後期分のデータを集計したものである。なお、コンパイル時のエラーは1回のコンパイルに対して複数個発生することがあるが、今回はこれらをまとめて1回とカウントした。成績に対するエラー回数には弱いながら正の相関があり、前期から後期へと学習が進むにつれて、相関が強くなることが分かる。ただし、当初は熟達するほどエラーが減少することが予想されていたが、実際の結果は、熟達してもエラーの回数は減らないことがわかった。

コンパイル回数に対するエラー発生割合を調べたのが図 3-22 である。この結果から、エラー発生割合は成績によらず、ほぼ 50%で一定であることが判明した。表 3-9 にそれぞれの標準偏差を示す。これにはいくつかの要因が考えられるが、当初は成績が悪い学生ほどエラー発生率が多いことが予想されたが、本システムのアドバイスシステムが有効に働き、初心者でも短期間でエラー箇所が特定できたことが要因の一つとも考えられる。

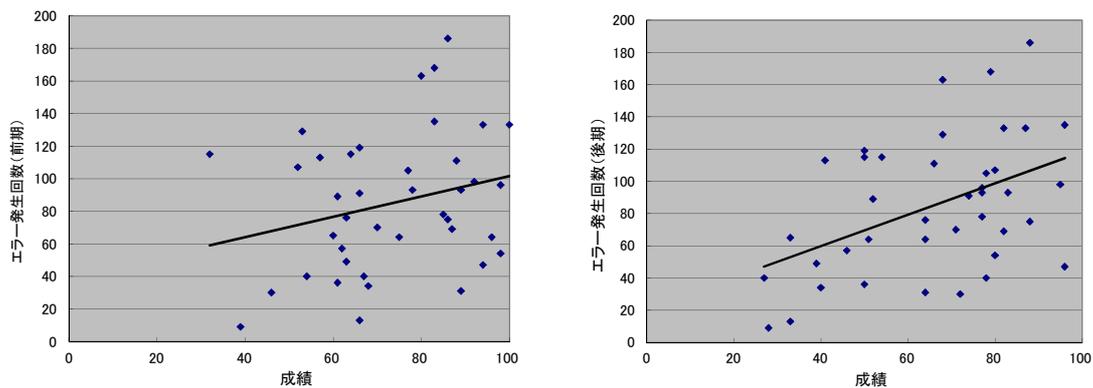


図 3-21 成績とエラー発生回数 (左：前期 右：後期)

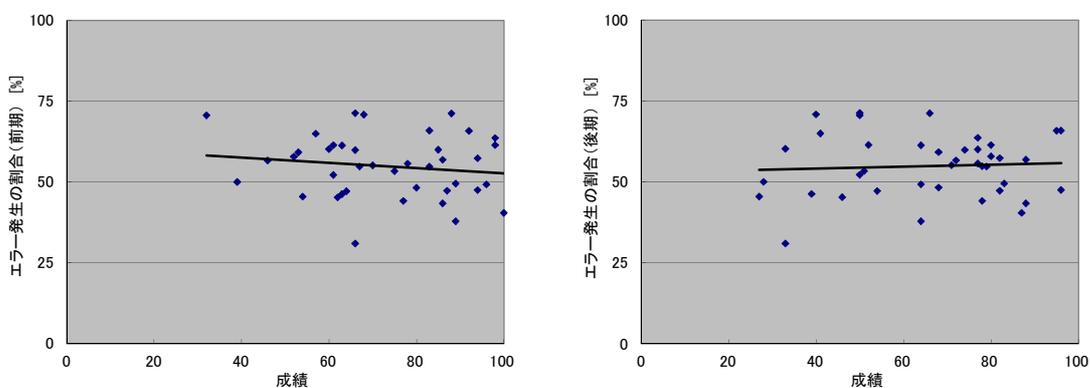


図 3-22 成績とエラー発生の割合 (左：前期 右：後期)

表 3-9 成績とエラーの標準偏差

	コンパイル回数
成績 (前期末)	0.095
成績 (後期末)	0.096

特定のエラーと成績に強い相関が得られたのが E2451 (未定義のシンボル)のエラーである。エラーの発生回数と学生の成績に関して関係を調べてみると、サンプル数は少ないが相関係数は-0.95となり強い負の相関があることが判明した。その様子を図 3-23 に示す。この結果より、コンパイル・エラーの種類や個数から、学習者の熟達度が推測可能であることがある程度証明された。しかし、特定のエラーコードに着目した場合、そのサンプル数がかなり限られるため、短期間で推測するにはまだ研究の余地があると思われる。

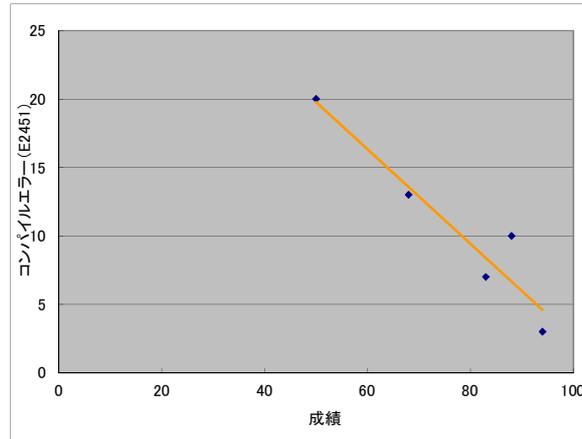


図 3-23 コンパイル・エラー (E2451) と成績

3.7.6. コンパイル・エラーに対するアドバイス

授業支援システム TSUNA-TASTE では学習者のコンパイル情報をリアルタイムで収集し、データベースに蓄積している。収集しているデータは送信されたデータの日付、送信元ホスト名、コンパイルしたユーザ名、コンパイルしたファイル名、コンパイル結果の 5 項目となっており、収集したデータをそれぞれデータベースに格納している。

データが格納されると、簡単な分析を行うことができる。支援サーバの分析データ内にあらかじめ、コンパイル・エラーのパターンに対する、初心者向けメッセージが一覧表として格納してある。初心者が理解困難なエラーメッセージの場合は、理解しやすいと思われるアドバイスを検索することができ、その一覧表に基づいて図 3-24 の様に学習者のパソコン画面に表示を行う。メッセージにはエラー番号やエラーの発生した行番号などの情報も引き継がれており、より適切にアドバイスできるよう工夫されている。

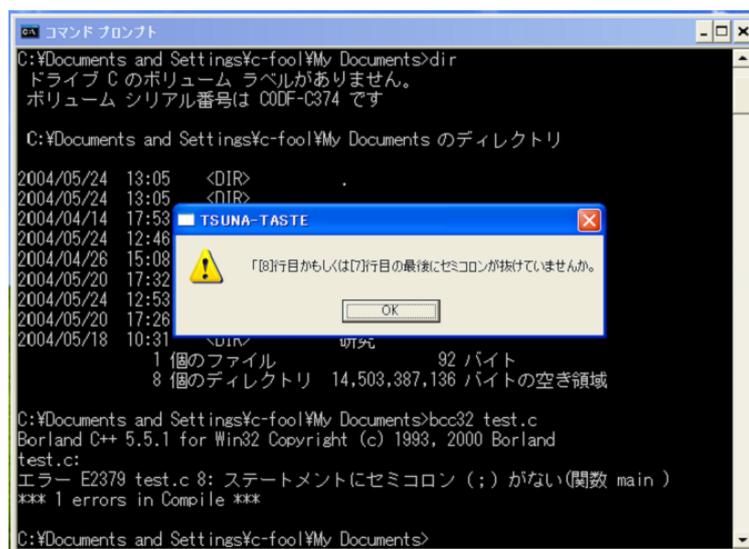


図 3-24 コンパイル・エラーに対するアドバイス

このエラーメッセージ表示は初学者にとっては非常に有効である。しかし、プログラミングの上級者にとっては、あまり有益な情報は含まれない。むしろ、ポップアップを冗長に感じることもある。本来は学習者のプログラミングレベルを自動的に判別し、上級者にはこのようなアドバイスは表示しない、もしくは別の形でアドバイスすることができると良い。残念ながら、現在のところ自動的に学習者を分類することができていないので、データベースを直接操作して、メッセージが不要と思われる学生にはメッセージを表示しないよう調整している。3.7.5 項に示すように、コンパイル・エラーの発生状況からある程度熟達度を推測する方向性が見えたため、今後学習者の自動分類が可能になればこのような作業も自動化できると考えている。

3.8. アルゴリズム可視化ツール

前節まで述べてきたように、学習者のパソコン操作の状況や学習状況をリアルタイムで収集し、その状況に応じて適切な指示を出すことにより教員および学習者双方を支援するシステムとして、授業支援システム **TSUNA-TASTE** を構築してきた。

このシステムにより学習者の学習（コンパイル）履歴をできる多機能に分析し、学習者の苦手する領域や、熟達度別にどのような問題で学習が止まっているのか検討を行ってきた。2.4 節の学生アンケートより学生がプログラミングを苦手とする要因について調査したところ、アルゴリズム全般の理解が十分でないことが判明した。これはソースコードを読んでも、それがどのように実行されていくのかイメージをつかめない学生が多数存在することを示していると思

われる。プログラムが実行されていくイメージをつかめない学生にとって、プログラミングはあまりにも抽象度が高い作業になる。学習の初期段階でこのようにある程度イメージが描けない状況が続くと徐々に苦手意識に繋がると同時に、コンパイル・エラーを積極的に理解する努力を放棄する傾向へと繋がる。

この問題を解決するためソースコードの実行過程を可視化して提示することにより、学習者がプログラムの制御構造のイメージを描きやすくするためのアルゴリズム可視化ツールの作成が行われてきた[45]–[47]。これらのツールはプログラミング関係の授業で採用されることの多い C 言語用のインタプリタの一種として実現している。そこで、今回新たにアルゴリズムを可視化するために、C インタプリタの開発と TSUNA-TASTE 用のモジュール開発を行った。

このアルゴリズム可視化ツールは C 言語の文法に従って記述されたソースコードを逐次解読し実行しながら、実行した内容やその様子を記号やアニメーションを使って表現する機能を持つ。またプログラム実行途中におけるメモリの状態を表示する機能や関数の説明を提示する機能なども持っている。ソースコードを解読・実行する速度は調節可能であり、また 1 行ずつ解読・実行して停止させることや連続実行中に一時停止させることも可能である。

このアルゴリズム可視化ツールが扱えるデータ構造、実行可能な命令、そしてその他の機能をまとめると以下のようなになる。

- 使用可能なデータ型／データ構造
int 型, char 型, float 型, double 型, 二次元までの配列, ポインタ, 定数
- 使用可能な命令
if 文, elseif 文, else 文, while 文, for 文, printf 関数, scanf 関数, 代入文, ++ (インクリメント) 演算子, -- (デクリメント) 演算子
- テキストエディタ機能
元に戻す, やり直し機能, カット／コピー／貼り付け機能
- 最近使ったファイル (過去 10 ファイルまで) の表示
- 印刷機能
- 実行方式にコマ送り実行を行う機能
- フローチャートの全体を確認できる機能
- 動作速度の変更機能
- 関数ビューアの実装(関数の書式や使用例等を調べることができるサブプログラム)

アルゴリズム可視化ツール実際の操作画面を図 3-25 に示す。

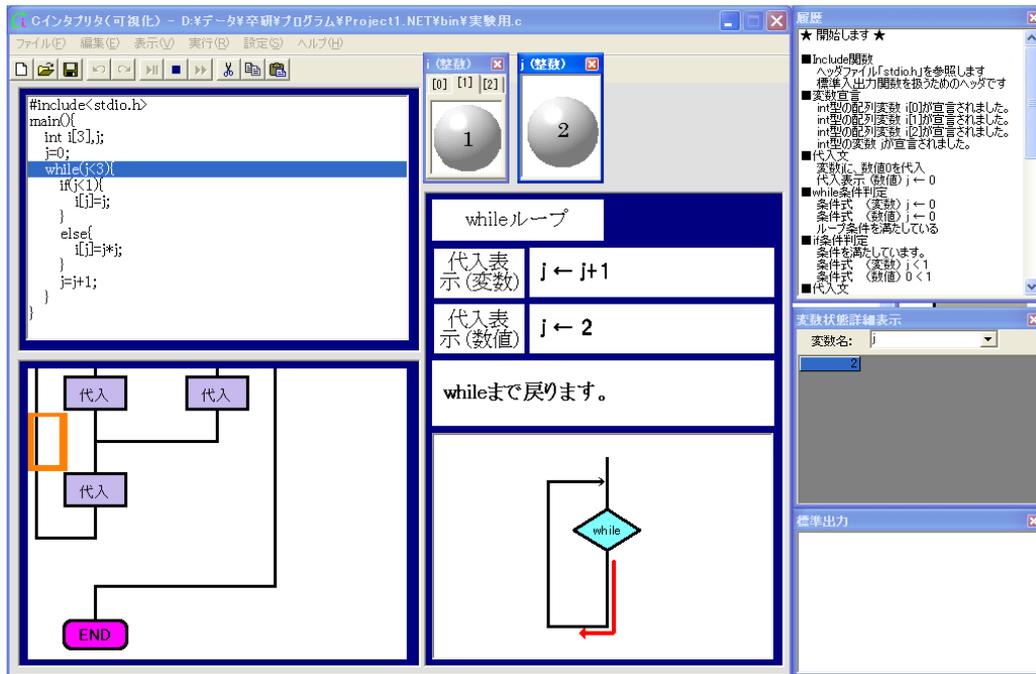


図 3-25 アルゴリズム可視化ツール

このユーザインタフェースは次のような操作が可能であり、これらの操作を通じて学生のプログラミング作成を支援する機能を備える。

- ① ソースコードのエディタ (図 3-25 左上)
 - C 言語のソースを表示する
 - 簡単なエディタ機能を持つ
 - ファイルからの読みこみと書きだしも可能
- ② フローチャートの表示 (図 3-25 左下)
 - ソースを解析してフローチャートを描画する
 - 実行中は対応する場所 (実行している行) をマークアップして表示する
- ③ 実行内容の説明 (図 3-25 中央中)
 - 現在実行中の命令の説明や処理を分かりやすく表示する
- ④ アニメーションによる処理内容の表示 (図 3-25 中央下)
 - 代入操作や比較処理などはアニメーションによって可視化する
- ⑤ 変数状態の表示 (図 3-25 中央上)
 - 変数名や型、値などを表示
 - 配列はタブで切り替えて表示
- ⑥ 変数状態詳細の表示 (図 3-25 右中)
 - 変数を表形式で表示 (2次元配列に有効となる)
 - 変更された場所をマークアップして表示

- ⑦ 履歴の表示 (図 3-25 右上)
 - 実行過程の履歴をテキストとして表示
- ⑧ メモリ内の状態を表示
 - 主記憶を仮想的 (0 番地からアドレスを割り当てる) に表現
- ⑨ 標準出力の表示 (図 3-25 右下)
 - printf 関数などの出力先として標準出力を表示する

このアルゴリズム可視化ツールはコンパイラではなくインタプリタタイプの実行環境となっている。そのため 3.7 節で利用したコンパイラは利用しない。またコンパイラのように標準入出力を使用してテキストデータの入出力を行わないため、3.7.2 項で述べたラッピング技術を使うことができない。この理由によりアルゴリズム可視化ツールを TSUNA-TASTE に組み込み情報を収集するには独自のモジュールとして組み込む必要がある。アルゴリズム可視化ツール自身に操作履歴をリアルタイムで書き出す機能を追加し、その情報をモジュールを使って TSUNA-TASTE に読み込むこととした。その時の動作を図 3-26 に示す。

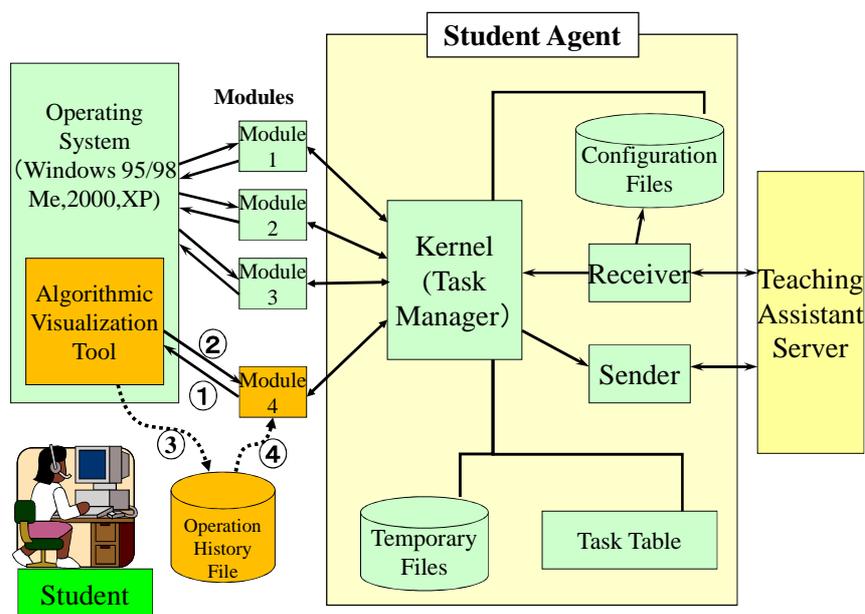


図 3-26 アルゴリズム可視化ツールの組み込み方法

このアルゴリズム可視化ツールを実際の演習で使用した時の学生のアンケート結果を 5.3 節に示す。

4. 実践的競技システムへの適用

教育支援用フレームワークを元に授業支援システム TSUNA-TASTE の構築を行ってきた。教育支援用フレームは非常に柔軟な設計が可能となっており、TSUNA-TASTE においてもモジュールの変更や追加，データベースの変更や追加，さらにネットワークの分散化などその柔軟性を確認できた。まだあらゆる教育分野をカバーする万能 PSE の開発には及ばないが，プログラミング教育をターゲットとして，試験支援やコンパイル情報の取得による学習者支援，またアルゴリズム可視化ツールによる学習者の理解を補助するシステムなどを構築してきた。

教育支援用フレームワークをさらにプログラミング教育の実践的な教育に応用できないだろうか検討した結果，全国高専の学生を対象として毎年開催されている全国高等専門学校プログラミングコンテスト[48]（以下高専プロコン）の競技部門に応用する事を考えた。このような分野へ適用可能であることが検証できれば，その柔軟性と有効性が実証できると考えた。

高専プロコンは自由部門，課題部門そして競技部門の 3 部門から構成されている。自由部門と課題部門は毎年基本的な運用方法に変更がないため，各会場に合わせて配置を調整することで実施可能である。これにくらべ競技部門は毎年テーマが変更され，その都度競技運用のためのシステムを構築しなければならない。過去の多くの大会では全国プログラミングコンテスト委員会の競技部門担当がシステムを作成し，主管校が運用を行うスタイルで開催を行ってきた。第 18 回大会は津山高専を主管校として開催されることになり，津山高専でシステムを開発することとなった。競技内容を検討するうちに，教育支援用フレームが適用可能であることが判明したため，実際の競技システムの開発を行ったのでその様子を 4.1 節に示す。

授業支援システムは競技システムを前提として開発を行っていた訳ではないが，そのフレームワークの柔軟性から，システムの一部を活用する事ができた。ただし，グラフィカルな面に関して，TSUNA-TASTE では Java を使ってきたが，Java だけでは思ったような画面が構成できないため，マイクロソフトの .net フレームワークも活用して開発を行った。

また，第 19 回大会（4.2 節），第 20 回大会（4.3 節），第 21 回大会（4.4 節）においても競技部門の運用システムの作成を担当することになり，フレームワークの一部を使用することとした。毎年競技ルールを含めて，競技環境全体が変更され，開発環境や実行環境も異なるため，全く同じものを使うことが難しいが，システムの根幹部分に使用することができた。以下のその様子を述べる。

4.1. ネットオークション型システム

4.1.1. 競技ルール

第18回高専プロコンは津山高専を主管校として、2007年10月に岡山県津山市で開催された。第18回大会の競技テーマは、津山城の有名な石垣を題材とした「石垣工務店」に決定された。この競技は、図4-1に示す石垣枠に見立てた木製のボード上に石にみたてた様々な形のピース（実際には木製）をはめ込んで、できるだけ隙間無く埋めることで頑丈な石垣を作ることを目的としている。ただし石垣の作成に使用するピースは自由に使えるのではなく、ネットワーク上のオークションサイトで入札し、他のチームよりも高い価格で落札しなければ使用することができない。当然定められた所持金以内で入札価格や購入の優先順位を決める必要がある。また、オークションに出されているピースには数に限りがあり、最低落札価格も定められている。他チームの入札価格を公表しない非公開方式のオークションで入札を受け付け、申し込み締め切りと同時に他のチームとの値段が比較され、どのチームが落札できたのか決定される。その情報が公開され、落札したチームに実体であるピースが配布される。

このような入札が1回戦は4回、準決勝は5回、決勝では7回行われる。もし同じピースを複数のチームが同じ値段・同じ希望順位で入札した場合、関係するチームだけによる再入札が行われる。この部分はルールが少し複雑になるためここでは割愛する。

ネットワーク上で落札された石は実際のピースとして、各チームに配布されステージ上で組み上げることになる。この配布作業も短時間でしかも間違いなく行う必要がある。最終的にピースを受け取ったチームが枠のなかにはめ込んでその完成度を競う。

これらの競技を支えるために、入札システムや入札結果を全てのチームに公開する情報公開システム、石垣に見立てた木製のピース、組み立てる石垣枠等が必要となる。これらをすべて津山高専が設計し製作した。また、入札システム、情報公開システム、そしてピースを配布するシステムの開発に教育支援フレームワークを一部変更して適用した。

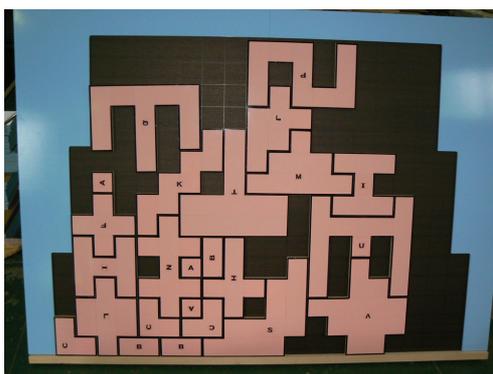


図 4-1 石垣をモチーフとしたパズル競技

4.1.2. 競技用の石垣・石垣枠

競技はコンピュータとネットワークを利用して入札を管理するバーチャルな世界と、実際に木製のピースを配布し枠上に組み立てるリアルな世界の両方で構成される。競技は図 4-2 に示すようなピース ([A]-[W]の 23 種類) を、図 4-3 に示す枠に並べることで行われる。この競技で使用するピースおよび枠は、この競技独自のものであり、もちろん販売などされていない。そこでそれぞれ素材の選択から設計、加工、組み立て方法の検討を行う必要があった。

当初ピースは木製がふさわしいと考えた。しかし実際に作成するには天然の木材は値段が高くしかも加工が難しいため、中密度繊維板(MDF: medium density fiberboard)と呼ばれる木質繊維を原料とする成型板(ファイバーボード)を使用することとした。

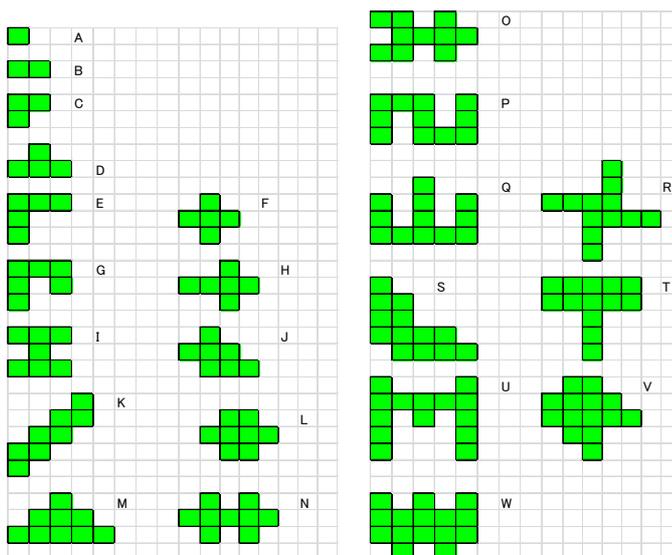


図 4-2 ピース形状の一例

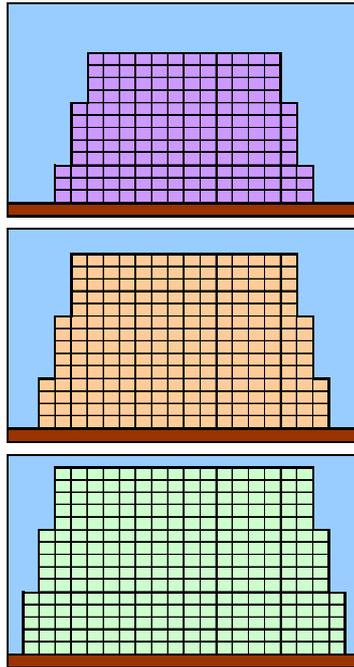


図 4-3 石垣枠の形状（上：1回戦，中：準決勝，下：決勝）

石垣枠は、ピースを並べる部分の形状と面積が試合の進行により異なる。図 4-3 の一番上が 1 回戦・敗者復活戦用（セル数：166），真ん中が準決勝戦用（セル数：222），一番下が決勝戦用（セル数：270）の枠である。このように枠の形状と面積を素早く変更する必要があるため、石垣枠はメインの茶色いベース部分と面積を変更するためのパネル部分（青色）の組み合わせで実現することとした。また、青色のパネルは真ん中で左右に分割されており、持ち運びが容易に行えるようになっている。これら石垣枠を全部で 9 台作成し 9 コート同時に試合ができるようにした。

4.1.3. 競技用システム

競技では、各チームが必要とするピースを非公開式入札により決定する入札システムが必要である。この入札を管理するためのネットワークおよびコンピュータシステムを開発する必要がある。システムのネットワーク構成を図 4-4 に示す。これらのシステムが競技の進行に合わせてどのような動作をするのか順に説明する。

競技の進行はすべて「運用管理サーバ」で制御する。また競技や参加チームに関するデータはすべて「入札管理用サーバ」内のデータベースにより管理し、同時にこのサーバ上で動作する Web サーバにより、各サブシステムに情報が提供される。フレームワークでは運用管理サーバ、データベースサーバそして Web サーバを別々のサーバで動作させることを前提に開発しているが、サーバ設置の容易性や移動のしやすさから、サーバを 1 台に集約する事も可能であり、本

競技システムでは1台に集約した。

このデータを元に競技の進行を管理するのが「運用管理サーバ」である。競技は、競技開始から第1回入札開始ー第1回入札終了ー落札ピースの計算ー第1回開札ー第2回入札開始と進行する。オークションに出されるピースの形状や個数、最低落札価格などは試合ごとに異なっている。

第1回目の入札が始まると、競技者は各コートに割り当てられた「入札用パソコン(9台)」から希望のピースに優先順位と値段をつけて入札する。一定時間経過後第1回目の入札が締め切られ、開札となる。落札結果は各「入札用パソコン」に表示されると同時に、「情報公開用サーバ」にデータが転送され、Web APIによりxmlデータと、Webベースによるhtmlデータとして公開される。このサーバから競技者は各自が持ち込んだ「競技者用パソコン(9台)」を使って落札に関する全ての情報を閲覧できる。この情報は試合終了、選手待合室のパソコンにも転送し、参加者が自由に閲覧できるようにした。これにより、入札の傾向を分析するなどの戦略を立てるチームも現れた。

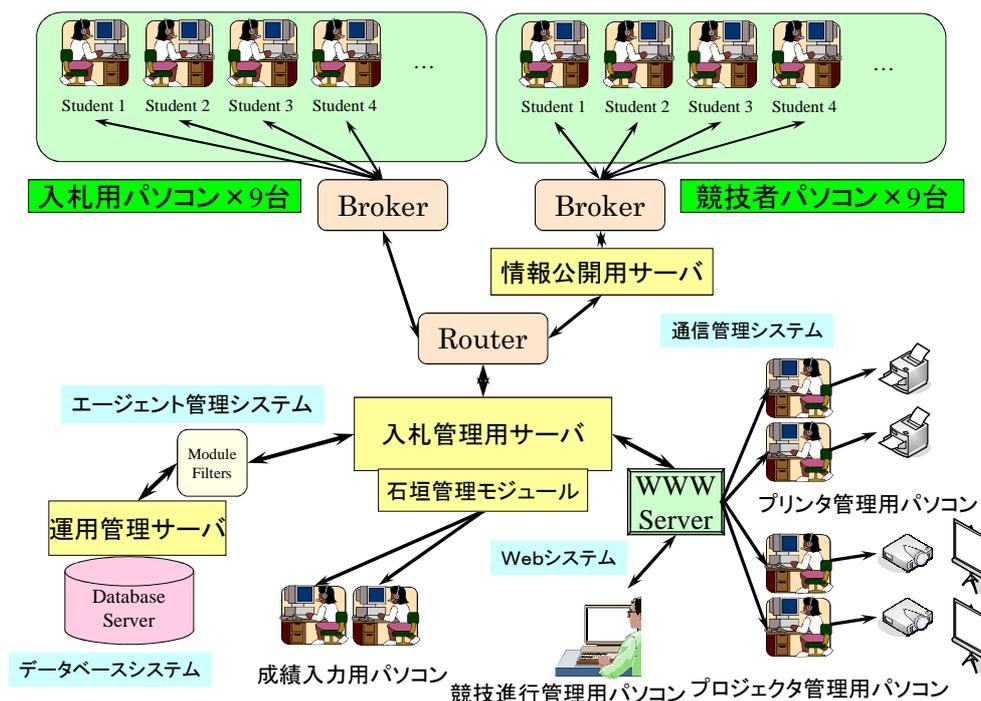


図 4-4 18回大会の競技システムのネットワーク構成

4.1.4. サブシステム

入札を管理するメインシステムのほかに、競技運営を支援するためにいくつかのサブシステムが必要となる。サブシステムは「落札結果印刷」「タイマ」「競技情報表示」「成績入力」の四つがある。これらに関しては、教育支援用フレー

ムワークでは対応するモジュールが存在しなかったため、別途開発を行う事となった。

「落札結果印刷」は、リアル世界の石を配送するピース配布係の学生のための出力システムである。「プリンタ管理用パソコン」に接続されたプリンタに開札と同時にその情報が A4 で 1 枚印刷される。この印刷結果をピース配布係の学生が持ってピースが置いてある倉庫へ行き、自分の担当するチームのピースを正確に集めて各コートに届ける。このときに、同じ用紙をもう 1 セット印刷し、各コートの審判に配布する。これは間違ったピースが配布されないようするための対策である。

「タイマ用パソコン」は競技の進行状況を示すテロップを表示し「時計」「入札中」「再入札中」などの情報をプロジェクタへ映し出すサブシステムである。

「競技情報表示用パソコン」はオークションに出されているピースの種類や個数、落札価格や各チームの持つピースの数など、競技に関するすべての情報を表示するサブシステムである。この画像のビデオ出力をプロジェクタへ接続し、大型スクリーンに映し出すことで競技の進行を補助する。オペレータが司会の指示により、適切な情報に切り替えて競技進行を盛り上げた。

「成績入力用パソコン」は、4.1.6 項の図 4-7 示す競技終了後審判から提出される審判用紙のデータを入力し、最終的な順位を決定する。

4.1.5. システムの障害対策

普段使用する授業支援システムでは、仮にシステムが止まっても講義や演習そのものに直接的な影響はない。そのため通常時の運用ではあまり障害対策について考慮せずに開発してきた。しかし高専プロコンの競技は一度始まってしまうと 1 試合 20 分程度必要であり、コンテスト全体の日程がかなり厳しいため、試合の停滞・やり直しは全体の進行に大きな影響を与えてしまう。このためシステムには高い信頼性が要求される。そこでいくつかの障害対策を施した。まず、競技で使用するパソコン (27 台) はすべて地元企業から借り受けて準備した。パソコンはすべて OS やドライバの仕様を統一し、故障した場合もすぐに予備機と交換できるようにした。

サーバおよびネットワーク機器に関してもすべて 2 セット用意し、トラブルがあった場合は機器ごと入れ替えることで対応できる準備をした。またプリンタに関しても通常 2 台で運用するが、どちらか 1 台が故障しても、残りの 1 台で競技が継続できるよう工夫した。

競技者が持ち込むパソコンによる DOS 攻撃や不正なアクセス、ウィルス等による競技の中断を防止するため、競技者の持ち込むパソコンを接続するネットワークと入札系のネットワークを完全に分離しルータによって接続することで影響が出ないようにした。また、競技者パソコンからの不正アクセスを防ぐた

めに、情報公開用サーバへの接続回数に上限を設定し、それを超えてアクセスするパソコンには情報を提供しないなどの対策を行った。入札パソコンにおいても、不正な入力でシステムが停止しないよう、入力デバイスをマウスとテンキーのみとし、キーボードへの入力ができないようにして運用を行った。

実際に運用している競技システムの様子を図 4-5 に、また、競技の様子を図 4-6 に示す。



図 4-5 運用中の競技システム



図 4-6 競技風景

4.1.6. 順位決定処理

競技が無事終了すると、各チームの点数をカウントしなければならない。リアルの世界での石垣において、できるだけ隙間なく石を埋めたチームが上位となる。いくらオークションで落札しても、石垣枠にちゃんと組み上がっていな

いピースは点数とならない。そのため、サーバだけではこの点数をカウントできない。なお隙間の数が同じ場合はさらにいくつかのルールにより、順位が決定される。その判断に必要なデータを審判が計測し、図 4-7 に示す審判用紙に記入する。この部分もピースの配布同様にヒューマンエラーの発生する可能性が十分に考えられる。そのため次に示す 3 重のチェック体制でミスを防いだ。

第 1 にサーバに保存されている各チームが落札したピース情報を競技終了と同時に印刷しコート審判に渡す。この情報と枠上で使用していないピースから、枠上の有効ピース数や面積が計算できる。第 2 に盤面上のピース数を数え先ほどのデータと照合を行う。また盤面上の空き面積を数えその面積についてもチェックを行う。第 3 に競技者に測定結果に間違いがないことを確認し競技参加者にサインをもらうことで、チェックが完了となる。

この審判用紙を本部の成績入力用パソコンで入力し、結果表示待ちとなる。この入力も 2 回チェックを行い、間違いがないことを確認する。最後に司会者の合図で、最終結果をプロジェクトに表示して競技が終了する。なお、万一のトラブルに備えて、各コートの試合結果は全てデジカメで撮影し保存した。

審判採点用紙

予行演習	1回戦	敗者復活戦	2回戦	決勝
108	166	166	222	270

試合番号 _____ 試合 _____
 コート番号 _____ 高専名 _____
 審判氏名(サイン) _____

競技者確認 (サイン) _____ 以下の結果で間違いありません

未使用・無効ピース

ピース番号	セル数/1個	ストーン数	セル数
A	1	X	=
B	2	X	=
C	3	X	=
D	4	X	=
E	5	X	=
F	5	X	=
G	6	X	=
H	6	X	=
I	7	X	=
J	7	X	=
K	8	X	=
L	8	X	=
M	9	X	=
N	9	X	=
O	10	X	=
P	11	X	=
Q	12	X	=
R	12	X	=
S	13	X	=
T	13	X	=
U	14	X	=
V	14	X	=
W	15	X	=
合計			

競技者確認 (サイン) _____
以下の結果で間違いありません

最上部セル数

盤面上のストーン数

+

未使用ストーン

=

総ストーン数

判定

落札ストーン数 (プリントアウト)

落札総セル数 (プリントアウト)

-

無効セル数

=

有効セル数

判定

枠総セル数

-

空きセル数 (選手申請)

=

有効セル数

図 4-7 審判用紙

4.1.7. まとめ

本競技はコンピュータとネットワークを利用して入札を管理するバーチャルな世界と、実際に木製のピースを配布し枠上に組み立てるリアルな世界の両方で構成された。またバーチャルな世界も参加者が入札を行う入札系のネットワークと入札および落札情報を公開する情報公開系のネットワークの 2 種類が必要となり、ネットワーク的にも複雑であった。さらに、競技を途中で絶対に止めない高い可用性が要求された。これらの要求に答えるためにシステムを構築し、競技を成功させることができた。

今回の競技を通して以下の成果を確認できた。

- 入札系ネットワークと情報公開系のネットワークの 2 種類のネットワークをシステム内に混在可能とした。
- システムが一度も停止しない高い可用性が確認できた。
- 入札管理、成績入力、プリンタ管理、プロジェクタ管理など複数機能を Web サーバ経由で実現できた。
- 競技結果を正確に確認するための方法を確立できた。

また、本システムは参加者や観客にも好評で、どうやって短時間に入札結果のピースが配布されるのか不思議に感じる参加者も多かった。プログラミング能力の向上にプログラミングコンテストがどれだけ関与するのを客観的に判断することは難しいが、開催後の各メディアにおける評価は非常に高かった [49][50]。

4.2. 高セキュリティ型ネットワークシステム

全国高等専門学校第 19 回プロコンは、2008 年 10 月に福島県いわき市で開催された。18 回同様に本選では、課題部門・自由部門・競技部門の 3 部門が実施された。競技部門は、昨年と全く異なったものとなったため、新たにシステムの開発が必要であった。以下に競技の概要を示す。なお今大会は本選中にシステムの不具合が発生し、別途再試合を行う事となったため、最初に作ったシステムを改良し、よりセキュリティの高いシステムが必要となった。

4.2.1. 競技ルールとシステムの不具合

第 19 回プロコンの競技は「フラット収集車」という競技である。これは、図 4-8 のようなマップ上を各チーム 1 台のトラックでスタート地点から、ゴール地点目指して進む。途中全てのチェックポイントを通る必要がある。いかに効率よくチェックポイントを巡回するかが重要となる。またチェックポイントには集めるべき荷物が置かれており、この荷物を多く集めた方が与えられるポイントが高い。ただし、荷物には（仮想的な）重さがあるので、積載した荷物の重さに比例してトラックの燃費が悪くなる。トラックの燃料を気にしながら、

最も早くしかも多くの荷物をゴールに運んだチームが勝ちとなる。10チームが同時にスタートからゴールを目指すため、他のチームの動きも重要となり、場合によっては荷物の取り合いが発生する。

各参加チームは主催者の用意したパソコンを操作することにより各自のトラックを操作し、その内容を主催者側のサーバで集計する。このときトラックの移動に関するユーザインタフェースは Adobe Flash を使って実現した。また、他のチームの場所や残り荷物数など競技に関する情報は別途用意する情報提供ネットワークのサーバから提供される。この2つのネットワークは競技を安全に運用するためにそれぞれ分離することとした。

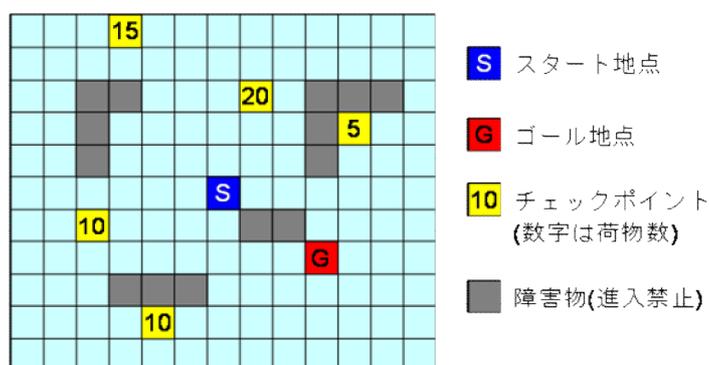


図 4-8 第 19 回競技部門「フラット収集車」

競技での勝敗判定は次の優先順位で決められる。

1. ゴール地点に到達したチームで、荷物 1 個あたりのガソリン消費量が少ないチーム。
2. ゴール地点に到達したチームで、多くの荷物を集めたチーム。
3. ゴール地点に到達したチームで、じゃんけん。
4. ゴール地点に到達しなかったチームで、荷物の置かれていた場所を多く回ったチーム。
5. ゴール地点に到達しなかったチームで、荷物 1 個あたりのガソリン消費量が少ないチーム。
6. ゴール地点に到達しなかったチームで、多くの荷物を集めたチーム。

今回の競技システムの概要を図 4-9 に示す。全体の構成は前回のシステムとほぼ同様である。ただし、今回はトラックの移動を行うユーザインタフェースに Adobe Flash を採用したため、Web サーバの部分を PHP と Adobe Flash による構成に変更している。

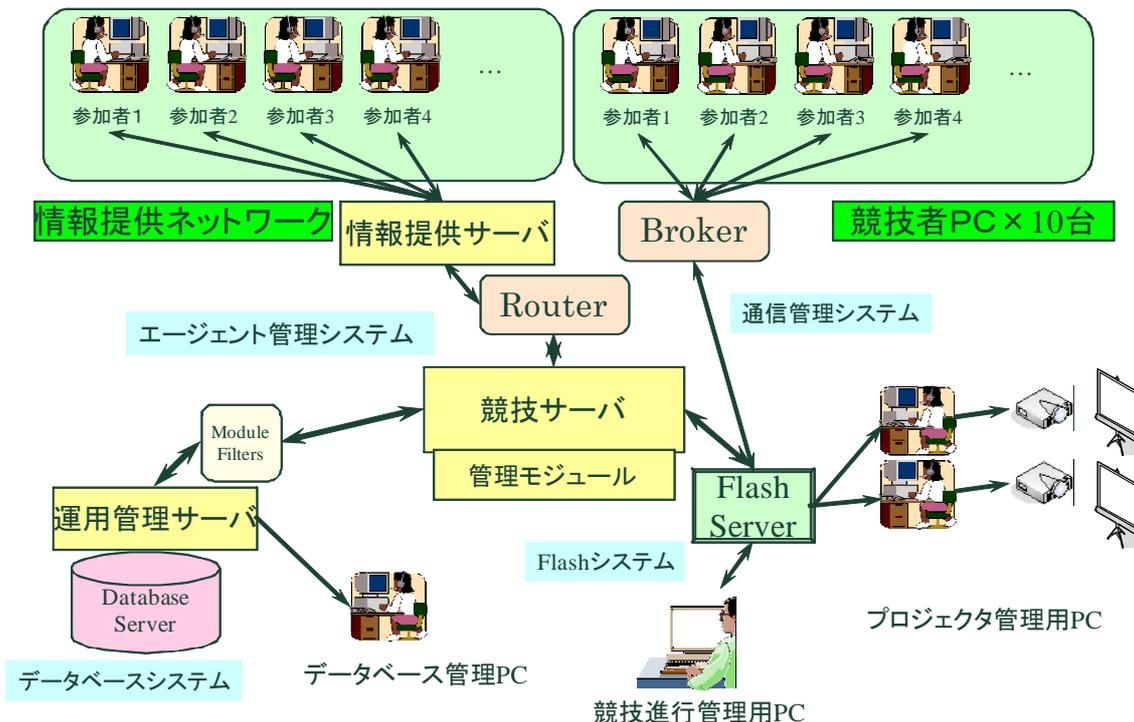


図 4-9 第 19 回大会の競技システムのネットワーク

メインとなるネットワーク部分は正常に動作したが、競技のルールを制御する部分のモジュールに不具合が発見され、本選期間中での競技の実施を断念せざるを得なかった。不具合の発生原因は、複数のチームが同時に 1 つのセルにアクセスし、荷物を奪い合った時の按分のアルゴリズムの一部に問題があり、各チーム間で公平に荷物が分けられなかった事であった。この不具合が発生する確率はかなり低かったが、競技前に十分に検証できていなかったことにより、本選になってシステムの変更を行う必要が生じた。またさらにこのシステムの変更時にデータベースを中途半端にリセットしたため、競技データでさらなる不整合が発生してしまい、結果的に競技は全面的に中止に追い込まれた。

4.2.2. 再試合の検討

本選期間中の競技の実施を断念したことに伴い、第 19 回プロコン委員会では競技参加チームによる再試合を行うことになった。

再試合の実施については、次の項目を重要と考えた。

- 本選参加の全チームが参加できるよう最大限配慮する。
- 本選で予定していた競技ルールからの変更を最小限にする。

これらを実現するために、第 19 回プロコン委員会では、本選と同様に参加チームが 1 か所に集合する場合とインターネットを経由して仮想的な環境を用意する場合について検討を行った。

実際に集まることが可能かどうかの検討と並行して、競技部門担当委員によるミーティングや実証テストを繰り返して可能性を探った。その後、実際に集まることは経費の問題もあり、参加チームのスケジュール調整に関しても困難と判断し、競技はインターネット経由で行うこととなった。

ただし、当初開発したシステムは原則として孤立した LAN 上での動作を想定しており、参加者以外のトラフィックは発生せず、第 3 者からのアクセスもないことを想定していた。インターネットを経由しての利用は、技術的には可能であってもセキュリティ対策があまりにも不十分と言えた。システム全体を刷新してセキュリティの向上が望ましいが、そのためには多くの開発コストが必要となる。時間的にも金銭的にも、新しいシステムを構築することは不可能と思われた。そこでフリーウェアをベースとした既存のネットワーク技術でこの問題を解決することとした。

4.2.3. 再試合システムの開発

インターネット経由での対戦を行うに当たり、本選用に開発された競技システムをベースに改良するか、全く新規に開発するか検討を行ったが、新規に開発した場合は、短期間の開発となり本選用システムのクオリティと同等以上のものの開発には大きなリスクがあると考え、本選用システムを改良することとした。また、全く初めての試みだったので参加チームの参加方法についても検討を行った。元々の本選用システムは、サーバクライアント形式をとっていた。この入力用パソコンを参加者側のパソコンで置き換えることも検討したが、入力用パソコンの環境が統一できないことから次のような問題が懸念された。

- 入力用クライアントソフトウェアの動作保証ができない。
- 入力用パソコンと競技サーバの時計の同期が困難である。

特に、2 番目の問題は深刻で、10 チーム同時に行う競技で時間の同期が失われると、競技を公平に実行できない。これらを解決する手法として、今回の再試合では、主催者側で入力用パソコンを用意し、参加チームはリモートデスクトップで接続して入力用パソコンを操作することにした。

しかし、リモートデスクトップ(TCP:3389)での校外への通信が許可されている高専は少ないことが予想され、またセキュリティの観点からも直接入力用パソコンにインターネットから接続することは好ましくない。そこで、VPN を利用して参加チームから競技システムへの接続を行うこととした。特殊な VPN 装置やプログラムが必要な場合は、対応できない参加チームがあることが懸念されたため、今回は通常の Web ブラウザから利用でき、Java で動作する SSL-VPN を利用することにした。SSL-VPN には、SSL-Explorer の Community Edition を用いた。

これに加えて、参加チームが試合の進行状況を確認できるように情報提供シ

システムとリプレイツールの提供を行った。情報提供システムは、元々の本選システムでの情報提供サーバをそのまま用い、リプレイツールは先に述べた新規開発の検証システムそのものである。

なお入力用パソコンについては実際にパソコンを10台用意するのは場所的にも金銭的にも難しい。さらにネットワークの攻撃やトラブルに備えて、3会場(津山高専・福島高専・有明高専)で実施できるようにしたため、合計30台のパソコンが必要なる。この問題を解決するため実機のパソコンを維持管理するのではなく、VMware Server 2.0 および VMware ESXi 3.5 による仮想化により実現し、SSL-VPN サーバおよびチャットシステムを合わせて仮想的な10台のパソコンを1台の物理サーバにより構築した。これらのシステムの概要を図4-10に示す。

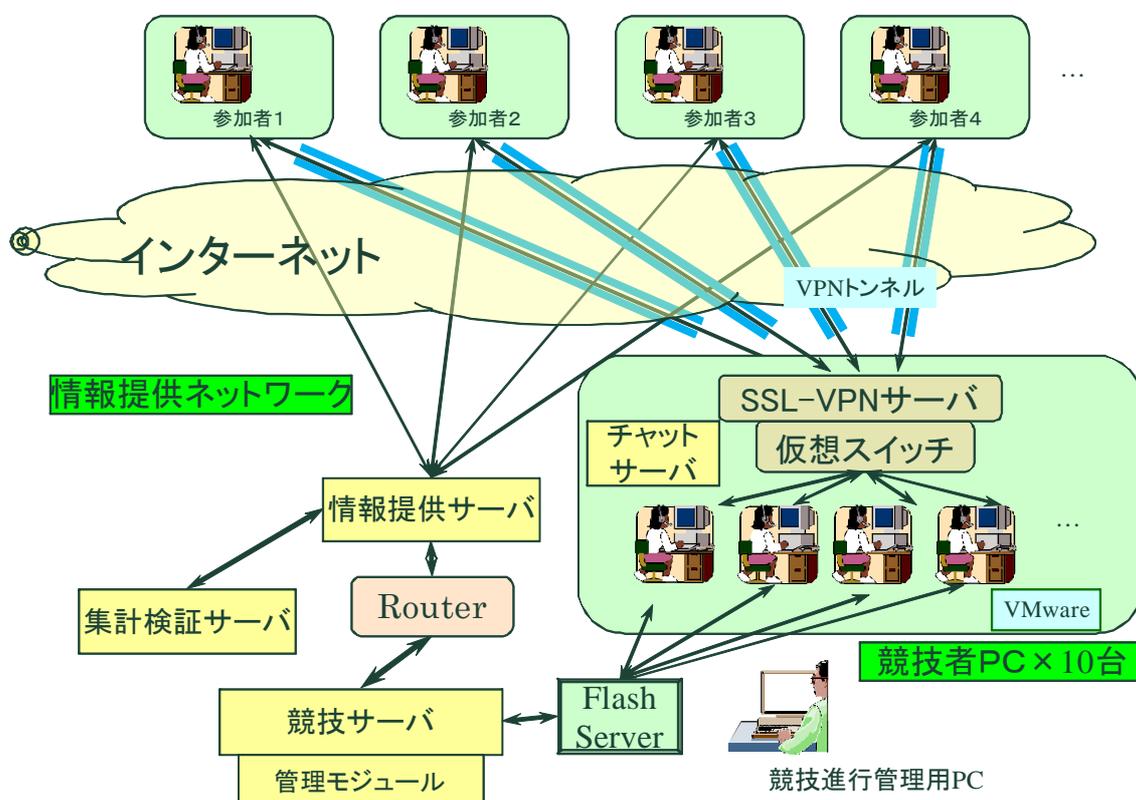


図 4-10 再試合用競技システムの構成

4.2.4. 再試合の運営

再試合の実施に先立ち、まず11月10日～17日に参加チームによる再試合システムへの接続テストを行い、各チームがインターネット経由で再試合に参加可能かどうか確認した。各校のネットワーク環境によっては、SSL-VPNを用いても接続できないこともあった。この場合は各学校のシステム管理者にネット

ワーク設定の変更を依頼した。それでも解決が困難な場合は 2008 年夏からサービスが始まった日本通信の b-mobile 3G による参加を試みるために端末を送付した。そのため、参加チームによっては何度も接続テストを行うこととなったが、結果として海外チームを含んでほぼ全チームの参加が技術的に可能であることを確認した。

実際の再試合は、次のような流れで進めた。

- 1) 試合開始 1 時間前までに試合で使用するサーバ類の URL を参加チームに連絡。
- 2) 試合開始 20 分前に情報提供サーバから問題公開。
- 3) 試合開始までに、SSL-VPN 経由のリモートデスクトップ接続およびチャットシステムへの認証・接続を完了。
- 4) 接続・動作確認のためのミニゲームの実施。
- 5) ミニゲームの正常終了を確認後、試合の実施。
- 6) 試合終了後、成績の確認。
- 7) 翌日までに参加報告書を事務局に FAX 送付。

なお、確実な試合の運営のために次のような対策を行った。再試合用システムは 3 組用意し、津山高専・福島高専・有明高専の各拠点に配置した。各試合では、1 拠点をメイン会場として用意し、1 拠点を同じ準備を行ったサブ会場とした。残りの 1 会場はチャットシステムの運用を行うこととした。これは、メイン会場が何らかの障害により機能しなくなった場合に、競技自身をサブ会場に切り換えることを想定しつつ、同時にチャットシステムが機能しなくなることを避けるためである。各試合でどの拠点のサーバを利用するかは参加チームのみに知らせることとした。

試合の途中でリモートデスクトップ接続ができなくなるチームが生じた場合に備えて、チャットシステムや電子メール、FAX 等による操作データを受け付ける準備も行った。この場合、参加チームは検証用システムを利用して送信用操作データを作成できるようにした。

4.2.5. まとめ

本競技は事情により 10 月の本選で終了することが不可能となり、インターネットを経由して 11 月下旬より再試合を行うこととなった。当初開発したシステムは原則として学内 LAN や独立した LAN での動作を想定しており、参加者以外のトラフィックは発生せず、第 3 者からのアクセスもないことを想定していた。インターネットを経由しての利用に関しては、技術的には可能であってもセキュリティ対策が不十分と言えた。そこで、システムの通信部分を高セキュリティ型に変更することで無事再試合を実現できた。

今回の競技を通して以下の成果を確認できた。

- SSL-VPN を利用しインターネット経由で全国高専および海外チームが参加した大会を開催できた。
- VMware Server 2.0 および VMware ESXi 3.5 によるクライアントパソコンの仮想化を行うことで、システムの省スペース化が実現できた。
- リモートデスクトップ接続により、参加者の環境を統一できた。
- 複数会場による開催をシームレスに実現できた。

以上のように、本システムはインターネットを経由しながら、あたかも参加者全員がその場にいるような臨場感で競技を行う事ができた。また、システムが妨害にあう可能性もあり、非常に慎重な対応を行った。また意図していたわけではないが、今回の運営を通して、教育支援フレームワークはそのシステムの内容を大きく変更することなくインターネット経由した高セキュリティ型のサービスとしても利用可能な事が実証できた。

4.3. アルゴリズム対戦型システム

第 18 回大会並びに第 19 回大会の競技は、それぞれの参加者の操作が他の参加者へ影響を与え、競技全体がダイナミックに進行するタイプであった。競技としては非常におもしろいが、競技システムにはリアルタイム性や情報提供など多くの機能が必要となる。このシステム開発の複雑性が原因で第 19 回大会はシステムの停止という異常事態を招いてしまった。そこで第 20 回の高専プロコン競技部門は、参加者同士の相互はなく、与えられた問題を解く最短手数を見つける時間を競うタイプの競技が選択された。第 20 回高専プロコンは、2009 年 10 月に千葉県木更津市で開催された。

第 19 回での失敗はユーザインタフェースをシステム側で用意し、このプログラムとサーバとが連携する部分に要因があったと判断した。この方式はグラフィカルに競技の進行状況を把握できるというメリットと裏腹にシステム開発に大きな負担をかける。そして、システム全体の整合性を維持するには多くの開発コストがかかる事が判明した。またデータベースは、正常に動けば非常に便利だが、トラブルが発生した場合に、その修正段階で管理者の誤操作における不整合発生も問題となった。

この反省から、今回のシステムではデータベースは使用せず、極力単純なプログラムだけで構成するように変更した。そのため、フレームワークのフルシステムを使うのではなく、ごく一部だけの利用にとどめた。

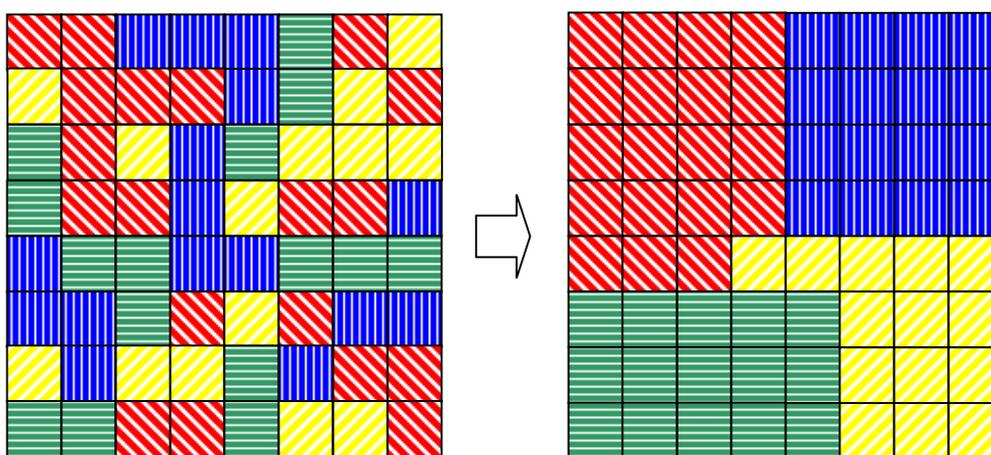
4.3.1. 競技ルール

競技のタイトルは「何色? サッと見 発見伝」と名付けられ、数色 (最大 8 色) のパネルが配置されたマス目状に区切られたフィールド上で、その一部の領域を回転しながら、色ごとに連続した領域にまとめるパズルゲームである。いか

に少ない回転数で完成状態にできるかを競い、その手数が少ない方が、また同じ手数なら解答を提出するまでの時間が短いチームが勝利となる。

競技のルールは以下の通りである。

1. 図 4-11 の左側の「初期状態」に示すような、色付きのマス目（セルと呼ぶ）が並べられたフィールドを問題として提示。
2. 競技者はフィールドの一部の正方形の領域を回転させる（図 4-12）ことを繰り返して、縦横に連結した同一色の領域（クラスタと呼ぶ）を大きくする。
3. 1 回の回転（ステップと呼ぶ）では $2n \times 2n$ (n は正の整数) の領域を、その領域の中心に対して点対称に 90 度単位で 90~270 度回転させることができる。図 4-12 に 4×4 の領域を 270 度右に回転した例を示す。
4. 目標の状態は、図 4-11 の「完成状態」に示すような、各色が一つのクラスタにまとまった配置となる。クラスタの条件を満足していれば形は、図に示した以外にも完成状態の配置は無数に存在する。
5. 初期状態から完成状態にするまでのステップ数を競い、同じステップ数の場合は解答を提出するまでの時間が速いほうが優位となる。



初期状態

完成状態

図 4-11 競技フィールドの例（左：初期状態，右：完成状態）

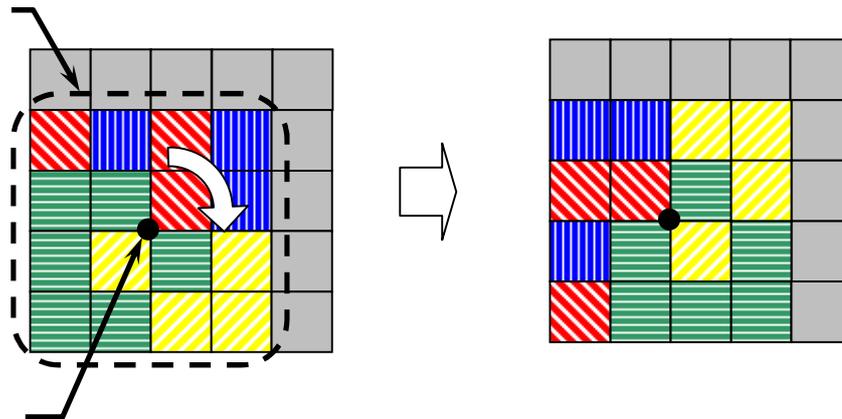


図 4-12 $2n \times 2n$ 領域の回転

4.3.2. 競技システム

本競技システムはアルゴリズム対戦であり、各チームとの相互作用は一切無い。競技システムとしては問題の提示、各チームからの解答の提出、解答時間と解答の評価がシステムに求められる機能である。昨年のシステムが、システムの肥大化によるチェック不足により競技本番でシステムのバグが見つかるという問題の反省から、今回のシステムは可能な限りシンプルな構成とした。図 4-13 に競技システムの構成を示す。

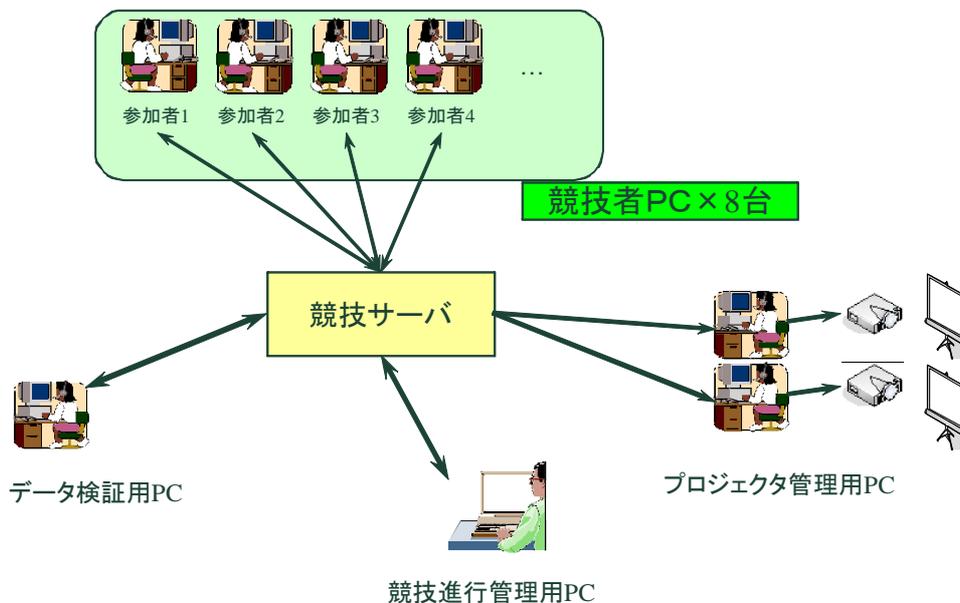


図 4-13 20 回大会の競技システム

4.3.3. 競技シミュレータ

競技では各チームから提出された解答をチェックする必要がある。昨年度は採点のチェック体制が十分でなかったために、バグを発見したのが大会本番になった。20回大会ではあらかじめ、競技用のシミュレータを作成し、解答をチェックする体制を整えた。またこのシミュレータは単体で、問題の作成、手動による解答の作成、解答の採点、そして解答のリプレー機能を搭載した。

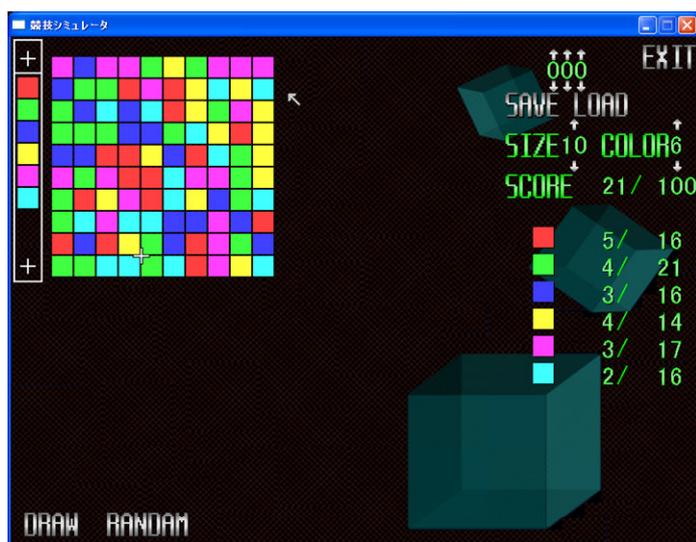


図 4-14 競技用シミュレータ（問題作成）

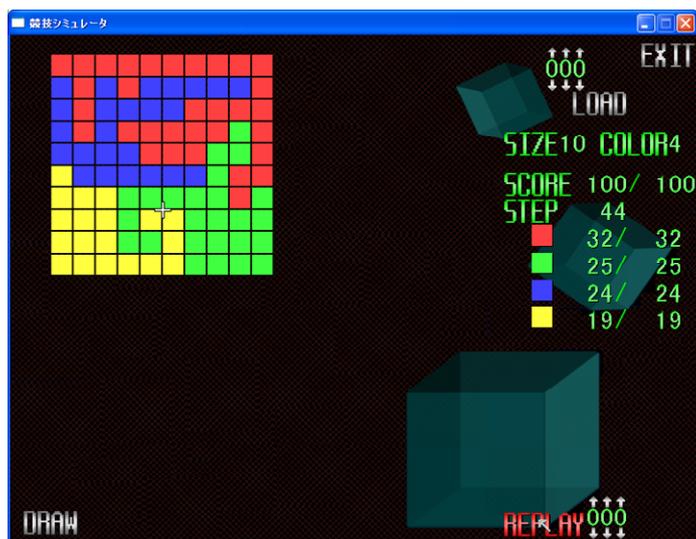


図 4-15 競技用シミュレータ（リプレー・評価）

この競技用シミュレータは大会当日、中学生たちに自由に使ってもらい、競

技の内容の理解やプログラミングのすばらしさを実感してもらうことを計画していた。10×10程度のサイズだと平均100ステップ程度で人間でも時間さえあれば解くことはできる。優秀なプログラムでは50ステップ以下で解くことができ、優勝チームでは40ステップを下回ることもあった。このようなことを体験することで人間とプログラムの違いを体験する機会を設けたかった。

残念ながら大会前に新型インフルエンザが流行してしまい、この企画は実現しなかったが、今後は是非プログラミングの良い実例として利用していきたいと考えている。

4.3.4. まとめ

本競技は4.2.1項の不具合を受けて、必要最低限のシステムだけで競技を運営する方針に変更した。そのため、新機能を追加するのではなく、サーバ1台だけで管理できるシステムとした。

今回の競技を通して以下の成果を確認できた。

- 必要最低限のシステムで競技を実施できた。
- データ検証用のシステムを導入し、競技システムの計算結果を確認できる環境を整備した。
- スタンドアロンで操作するシミュレータを開発しネットワークおよびサーバがなくても競技を楽しめるシステムを作成できた。

4.4. インターネット対戦可能な準リアルタイム対戦システム

第21回の高専プロコン競技部門は、インターネット対戦が可能なシステムを目指した。第20回大会は参加者の相互作用がない競技であったが、今回は再び参加者同士の相互作用のあるダイナミックな競技を目指した。インターネット対戦では独自アプリケーションを配布する方式ではなくWeb APIを活用して接続用のプログラムを参加者自身が作成するスタイルを採用した。第21回高専プロコンは、2010年10月に高知県高知市で開催された。

4.4.1. 競技ルール

競技内容はゲームフィールド上に設置された「水瓶」に蓄えられた水を利用し、いかに治水面積を多く取得できるかをリアルタイムで競うゲームである。各チームは自チームのロボットを3台所有しており、このロボットを使って水を各地へ配水する。ロボットはあくまでも仮想化されたものであり、本物のロボットではない。ロボットへは単純な操作命令（移動、チャージ、配水、吸い取り）を出すことができる。配水できた場所は自分のエリアとして確保され、このエリアを拡大することで各チームは生活圏を拡大する。ロボットの操作とフィールドの状況や各チームのステータスはWeb APIを通して情報交換を行う。インターネットを経由する通信の場合、完全なリアルタイム性を追求するこ

とは非常に困難である．そこで準リアルタイムという概念を導入した．これは，競技進行の最小単位としてフレームという概念を定義する．1秒あたりのフレーム数は60フレームの固定とした．ロボットへ与える各命令を実行するには，あらかじめ定められたフレーム数を消費しないと次の命令が実行できないこととし，これによりリアルタイムの時間を擬似的に再現する．たとえば10フレーム必要な命令は，最初のフレームで命令が実行され，その後の9フレームが待ち時間となる．セッションはフレームの集合体として，本競技では3秒（180フレーム）を1セッションとした．このセッション単位で各チームは命令をサーバへ送り，その1セッション分をとりまとめて，全チームのロボットの動作が決定しての競技が進行する．図4-16に競技で使用するフィールドの例を示す．



図 4-16 競技用のフィールド

フィールドは1つが六角形の最小単位（セル）が敷き詰められており，全体としても六角形の盤面（フィールド）を構成する．盤面の大きさは常に固定とし，試合のチーム数，試合時間は以下の通りである．

- 1辺のセル数は21
- 原則6チームの対戦
- 競技時間は5～15分

各チームは3台の仮想的なロボットを操作して水瓶で水をチャージし，セルに配水して自分のエリアの拡大を図る．なお，水瓶の水はいくらチャージしても減ることはない．

セルの地形には以下の5種類がある．

- 空白：移動・配水・チャージ不可能（競技進行には無関係なセル）
- 平地：移動可能、配水可能、チャージ不可能
- 壁：移動不可能、配水不可能、チャージ不可能

- 水瓶: 移動可能、配水不可能、チャージ可能
- 荒地: 移動可能、配水可能、チャージ不可能

最終的に得点エリアの多いチームが勝者となる。エリアは囲碁のように自分で配水したセルを繋げ周辺を閉曲線状に囲む事によって、より広大なエリアを一度に自分のエリアとできる。なお壁セル、空白セル、水瓶セルは配水できないため、閉曲線を構成する事はできない。囲まれたエリアの中に別のチームが入り子になっている場合は、内側のエリアは内側のチームのエリアとしてカウントする。

4.4.2. 競技システム

本競技には複数の要素が含まれる。まず、フィールドの構造を解釈して、水瓶の場所へ最短でたどり着く必要がある。そのため探索のアルゴリズムが必要となる。なお、ロボットの動作を人間が考えて操作することも不可能ではないが、競技は準リアルタイムに進行するため、命令の入力が間に合わずプログラムで操作する場合に比べ動作が遅くなってしまう。これより、ロボットの動作は基本的にすべてプログラムが自立して行うチームが大多数であった。

水をチャージしたのち、平地や荒地に給水するが、他のチームが先に配水している場合、その水を吸い取らないと配水できない。このように、他のチームとの相互作用が発生するため、常にフィールドの状態を監視し戦略を練る必要がある。競技システムとしては問題の提示、各チームからの操作命令の授受、1セッション(180フレーム)で実際に全てのロボットがどのように動作するかを計算し、フィールドの情報を更新する必要がある。

また1セッションを全チームで同期させる必要がある。過去の大会(第18回大会、第19回大会)のように、独自アプリケーションで対応する場合は、同期をとりやすいが、今回はWeb APIによって、各チームの同期を取る必要がある。そのために非同期ポーリングとも呼ばれるロング・ポーリングという技術を利用した。これは簡単に言えば各チームからロボットの命令をサーバが受け取ると、その接続をすぐには開放せず、全チームから1セッション分の操作データを受け取った後、フィールドの更新結果を戻り値として返し、そこで初めて接続を開放する。参加者は接続の解放を持って、1セッションの終了を知ると同時に、フィールドの最新状況も知ることができる。図4-17に競技システムの構成を示す。

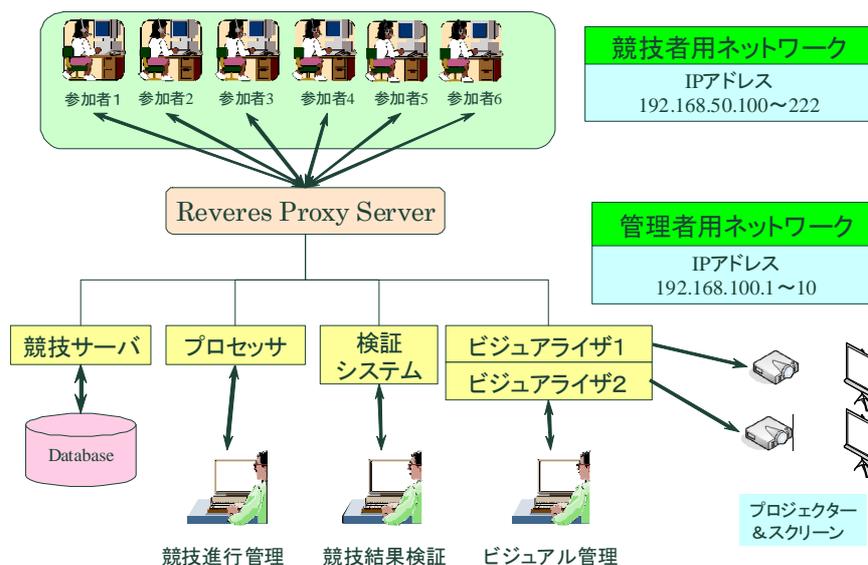


図 4-17 第 21 回大会の競技システムのネットワーク

4.4.3. インターネット対戦システム

高専プロコンの競技部門は競技の公平性と安全性の両面から、通常インターネットなど他のネットワークから切り離れた孤立した LAN 上で行う。ただし、このような環境はコンテスト当日に、コンテスト会場でしか実現することができない。そのため競技の参加者にとっては、実際に作成した自分たちのプログラムが、他のチームと戦った場合にどのような振る舞いをするのか十分に検証することができない。第 21 回大会の競技はサーバとの通信部分ならびに他のチームの情報取得が非常に重要となる競技であり、本選前に十分練習できる環境を整える必要があった。

そこで今回は SSL-VPN やブローカではなくリバースプロキシを導入し、サーバ側の安全性を保った環境を準備した。これにより各参加者は見た目上インターネットから直接サーバへ接続するのと同じ感覚でプログラミングをすることが可能となった。また、本来対戦チームや競技フィールドの設定は、競技の主催側でコントロールするものであるが、この部分もインターネット経由で、各学校が自由に競技の設定が行えるコントローラを提供して、いつでもどこからでも模擬対戦が可能な環境を整えた。このコントローラのうちフィールドを作成するゲームエディタを図 4-18 に示す。また、競技の進行を可視化する 2 次元のビジュアライザを図 4-19 に示す。

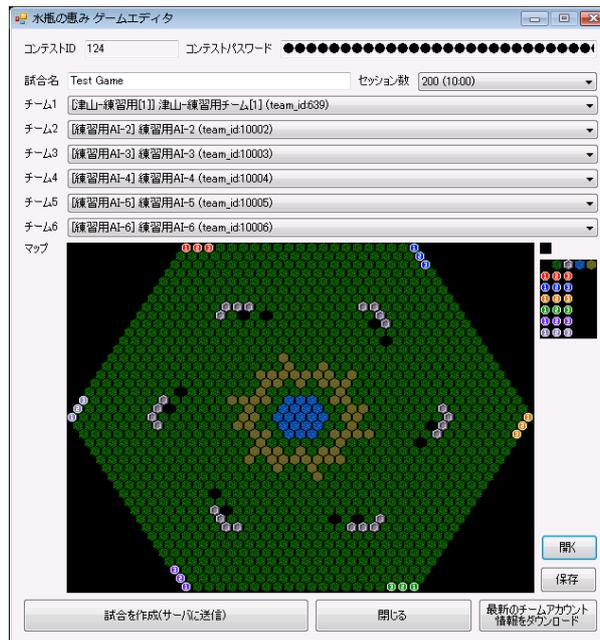


図 4-18 ゲームエディタ

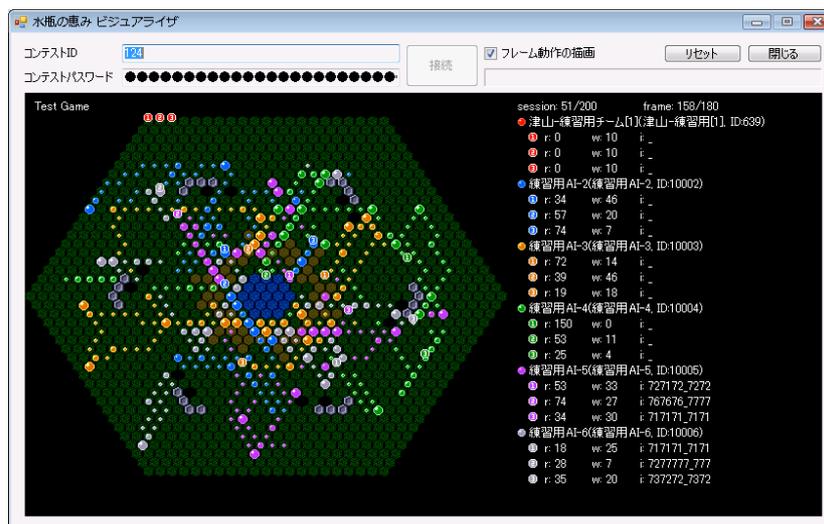


図 4-19 2次元ビジュアライザ

インターネット対戦は各参加者がゲーム作成と競技進行を行うため、それぞれが競技主催者と同様の権限を持つ必要がある。このために単純なパスワードだけによる保護では十分とはいえない。そこでコンテスト ID とコンテストパスワード、そしてチーム ID とチームトークンを以下の様に発行して安全性を確保したうえで各チームが独自に試合を作成できるようにした。

- コンテスト ID : 3 桁の整数で各校に配布
- コンテストパスワード : MD5 ハッシュ値

- チーム ID:各チームごとに配布（本選+6 チーム分）
- チームトークン：MD5 ハッシュ値

チームトークンは本選大会でも使用し，不正や誤った送信を防ぐ機能も兼ねる．また海外チーム（2 チーム）の参加を考慮し，メニューはすべて [日本語] [英語] の両方を準備した．

実際の本選の様子を図 4-20 に示す．また本選当日は競技の進行を 3 次元のビジュアライザに変更して表示した．その 3 次元ビジュアライザを図 4-21 にしめす．

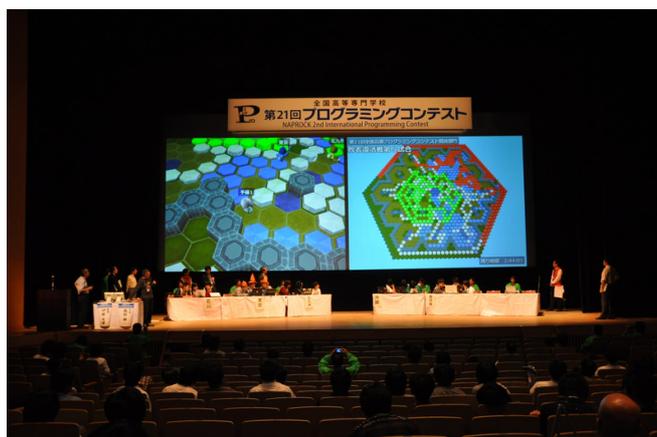


図 4-20 第 21 回大会の様子



図 4-21 3次元ビジュアライザ

4.4.4. まとめ

本競技は参加者がインターネットを経由して大会前に事前練習できるよう，インターネット対戦が可能なシステムを実現した．4.2.3 項においては

SSL-VPNによる接続形態を選択したが、今回はリバースプロキシを導入し、サーバ側の安全性を保った環境を準備した。

今回の競技を通して以下の成果を確認できた。

- リバースプロキシを導入することで競技者用ネットワークと管理者用ネットワークを分離することができた。
- インターネットを経由して参加者が練習用のサーバに接続できるシステムを開発できた。
- フィールドの作成からゲームの作成まで、管理者が使用するコントローラと同様の機能を持つゲームエディを配布できた。

また教育支援用フレームワークには無かったが、3次元ビジュアライザを開発することでエンターテインメント性の高い競技を実現できた。

5. 結論

プログラミング言語教育は、パソコンの操作方法だけでなくパソコンの環境やコンパイラの環境、そしてアルゴリズムの学習など学習者の求める要望が多く、初学者にとってはハードルの高い分野になっている。2章においてプログラミングを教える教員の問題、プログラミングを習う学生に問題に関して詳細に調査し、どのような支援が双方にとって有効か検討した。その検討を元に3章で教育支援用のフレームワークを提案し、このフレームワークを実現するネットワーク分散型の授業支援システム TSUNA-TASTE を構築した。そして授業支援システム TSUNA-TASTE を実際にプログラミング教育で利用した状況についても述べた。ここでは、授業支援システムに対する評価をもとに、本論文の結論を述べる。

5.1. プログラミング試験の評価

3.6 節で述べたプログラミング試験を支援するシステムを実際に本校情報工学科2年生(46名)の「プログラミング2」学年末試験(90分)に適用し、その効果を調査した。このとき学生に対して、本システムに関するアンケートを実施した。

図 5-1 に試験後学生に対して「授業支援システムにより公平性を保った環境下でのコンピュータを利用したプログラミングの試験は必要か」という質問に対して、「絶対必要」および「必要」と答えた学生が全体の83%もあり、多くの学生の支持を受けた。当初このように学生側のパソコンをコントロールするシステムは学生に不評かと思われたが、実際には非常に好評であった。

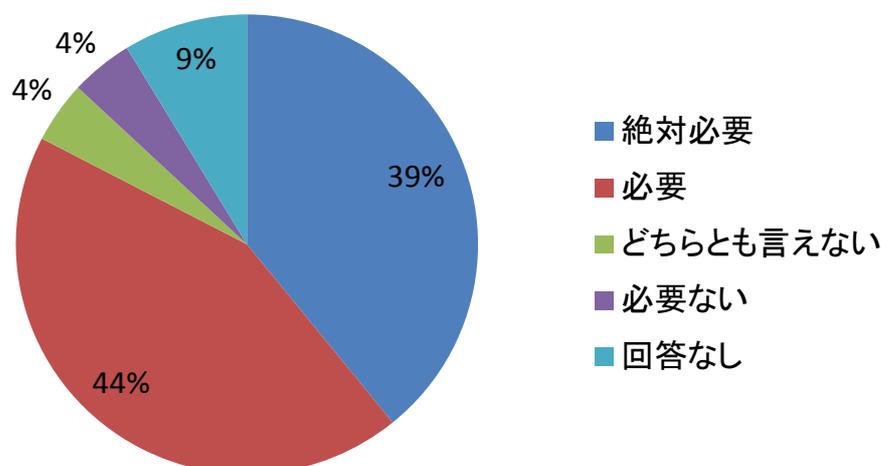


図 5-1 公平な環境下でのプログラミングの試験

このアンケート結果から、学生自身がプログラミング教育では筆記試験よりも実技試験が必要だと考えており、しかも今回のように公平に試験を実施できる環境を提供することが学生自身にとっても重要であるか確認できた。このことからシステムの必要性が確認できた。

5.2. コンパイル・エラーに対するアドバイス

2.4 節のアンケート (Q.4) が示すように、初学者がプログラミング (特にコンパイラ型言語) で最初に躓くポイントは、コンパイラの表示するエラーが理解できない事である。コンパイラが表示するエラーは、プログラミングに関してある程度知識を持つものに対しては、非常に有効なメッセージを含んでおり大変有用である。しかし、プログラミングの概念や基礎的な知識の乏しい初学者にとっては難解で意味不明の言葉でしかない。このため、コンパイル・エラーがそのままでは理解できない事が非常に多かった。

授業支援システム TSUNA-TASTE では 3.7.6 項で述べたように初心者向けのアドバイスが表示される。ただしコンパイル・エラーの文章を替えただけでは、根本的な解決にはならない。初学者がよく起こす間違いとして、「セミコロンがない」「ダブルコーテーションが閉じていない」「不要な全角スペースが入っている」などは、実際のミスとは一見関係ないエラーとして表示される。そのため、学生 (特に初学者) の起こしやすいエラーの内容を分析して、より適切なアドバイスを表示する必要がある。メッセージにはエラー番号やエラーの発生した行番号などの情報も引き継がれており、より適切にアドバイスできるよう

工夫されている。この結果を学習者にアンケートした結果を図 5-2 に示す。

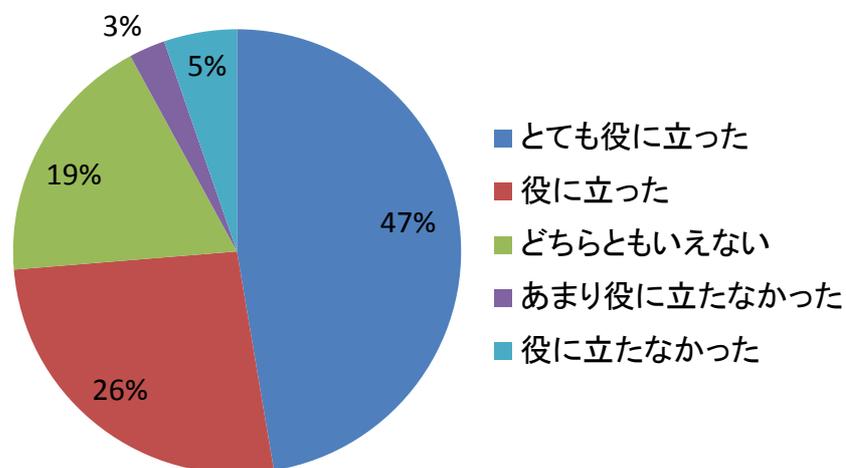


図 5-2 コンパイル・エラーに対するアドバイス (1)

この結果より約7割の学生が「とても役に立った」「役に立った」と答えており、アドバイスの表示システムが学生にとっては非常に役に立っていることが分かる。このことからシステムの有用性が確認できた。

表示されたアドバイスが適切であったかどうかを尋ねると図 5-3 の様になった。

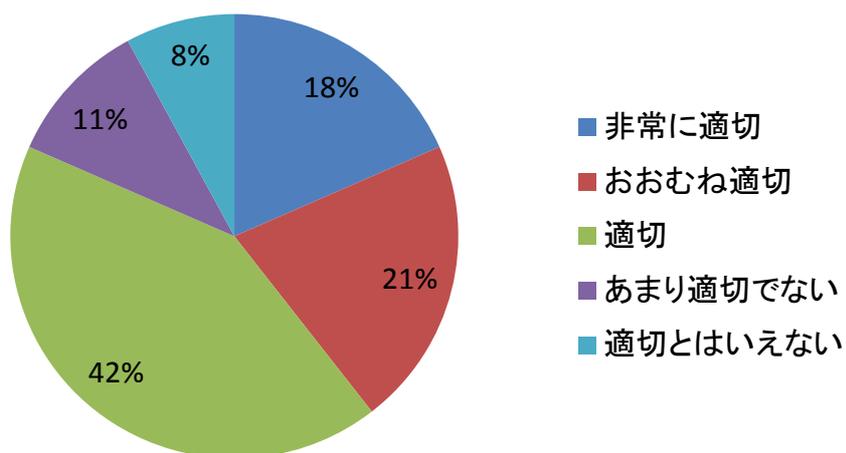


図 5-3 コンパイル・エラーに対するアドバイス (2)

さらに「メッセージが不適切と感じた理由」について自由記述で回答を求め

ると以下の回答が得られた。なお、括弧内はアンケートに自分自身のプログラミング能力を（初心者、中級者、上級者）の3段階で評価した結果である。

- メッセージ意味がわからない（初心者：2）
- 同じメッセージが繰り返される（上級者：5）
- メッセージが簡単すぎる（上級・中級者：7）
- エラーを見つける助けにならなかった（初心者：5）
- メッセージは不要（上級者：3）

このアンケート結果は中級・上級者の何人かにとってはアドバイスそのものが不要であることを示しており、この部分を自動的に判断できる仕組みが必要であることが判明した。

5.3. アルゴリズム可視化に対するアンケート

プログラムのアルゴリズム可視化ツールに関しても次のようなアンケートを行った。アンケートは2.4のアンケートと同じく、1年生から約2年間プログラミングを学習した学生であり、多くの学生がプログラミングの初心者からスタートし、アンケートを受ける頃にはプログラミングに関する一通りの説明と演習を経験した学生である。

「プログラムの動作を可視化する今回のようなシステムがあるとたすかりますか？」という問いに対して、図5-4のような結果が得られた。

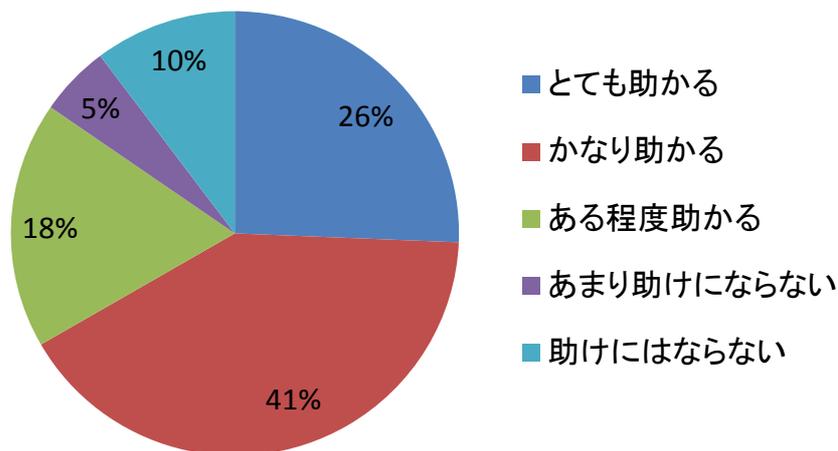


図 5-4 アルゴリズム可視化に対するアンケート結果

本結果から約7割の学生が、可視化システムがあると助かると答えている。さらにどのような可視化が一番役に立ったかに関して、記述式で回答を得た。

その一部（回答の多いもの）を表 5-1 に示す．なお回答には複数の回答を許可しているため、合計は 100%を超える値となる．

表 5-1 アルゴリズム可視化の利点

項目	割合(%)
動作が分かりやすく、処理を追っていくことができる	56
イメージしやすい 理解しやすい	27
フローチャートが分かりやすい	23
データのやり取りを確認できる	12
動きがあると助かる	11
間違いが分かりやすい	9

これらの結果からアルゴリズム可視化が学生の理解をたすけることで学習支援を行っていることが分かる．また、プログラムに少し慣れた学生にとっても、今回のアルゴリズム可視化ツールによりプログラムの動作やメモリ内のデータの遷移を確認することができ、学習効果が高いと考えられる．このことからアルゴリズム可視化の必要性と有用性を確認できた．

5.4. 教育支援フレームワークの問題解決環境としての評価

今回提案し構築したネットワーク分散型の授業支援システム TSUNA-TASTE について、使用者に対するアンケート調査からシステムの利便性と有効性が確認できた．また 4 章で示したとおり、この支援フレームワークが複数の応用事例に適用可能であることが判明した．特に 4.1 のネットオークション型システムや、4.2 の高セキュリティ型ネットワークシステムにおいては高い安全性を証明することもできた．

本研究で提案された教育支援用フレームワークは以下のような特徴が確認できた．

- プログラミング教育をはじめとするパソコンを使った学習や演習、さらに e-Learning なども含めた学習環境における問題点を明らかにし、この問題点を解決する一手法として新たな教育支援用の問題解決環境が提案できた．
- 提案した教育支援用フレームワークの実践的なシステムとして授業支援システム TSUNA-TASTE の構築を通し、さまざまな問題解決の手法を実際の教育に適用することで、その結果を多角的に確認できた．
- 学生機の遠隔コントロールやアプリケーションの制御は、現在いくつかの市販システムにも応用されているが、さらにキーボードやマウス操作

の分析、コンパイル・エラーの分析など他のシステムには無く、これら多くの特徴を持ったシステムであることが確認できた。

- プログラミング教育においてアドバイスシステムやアルゴリズム可視化ツールの導入は学習者にとって非常に有用であり必要性が高いことがアンケート結果等から確認できた。
- フレームワークの適用事例として全国高等専門学校プログラミングコンテストの競技部門に適用しフレームワークの柔軟性と拡張性が確認できた。

最後に授業支援システム TSUNA-TASTE と既存のシステムについて機能の比較を行ったのでその結果を表 5-2 に示す。ソフトウェアの機能比較のため定量的な比較になっていないが、それぞれのシステムの特徴を 4 段階で示している。「◎」は特に優れている、「○」は優れている、「△」は使用制限や対応していない部分がある、「×」は対応していないという状況をそれぞれ示している。

表 5-2 TSUNA-TASTE と他システムとの機能比較

比較項目	TSUNA-TASTE	授業支援システム (市販)	Web-ELS	プログラミング 学習支援ソフト
学生への支援	◎	○	◎	◎
教員への支援	◎	◎	○	◎
システムの拡張性	◎	△	△	△
システムの柔軟性	◎	×	×	△
学生機のコントロール	◎	◎	×	×
アドバイスの能力	○	×	○	◎
情報閲覧の容易性	○	◎	○	○
インストールの容易性	△	○	◎	○
OSの対応	△	○	◎	△
ファイル共有	×	◎	△	△

この結果より、授業支援システム TSUNA-TASTE は学生・教員への支援およびシステムの拡張性と柔軟性に優れていることが分かる。また学生機のコントロールにおいても高い機能を持っている。反面、学生に示すアドバイスや情報閲覧に関しては、他のシステムの方が優れていると思われる点もある。さらに、システムをインストールするには OS やネットワークに関する知識を必要とするなど改善すべき点もある。また、市販されている授業システムによく見られる学生と教員間のファイル共有に関しては対応していない。これはプログラミング教育においてファイル共有の機能は重要と考えなかったためである。

5.5. まとめ

本論文はプログラミング教育という分野に焦点をあて、この領域を支援する事を目的とした教育支援用のフレームワークを提案し、そのフレームワークの実践となるネットワーク分散型の授業支援システム TSUNA-TASTE を構築している。プログラミング言語教育は、パソコンの操作方法だけでなくパソコンの環境やコンパイラの環境、そしてアルゴリズムの学習など学習者の求める要望が多く、初学者にとってはハードルの高い分野になっている。2章においてプログラミングを教える教員の問題、プログラミングを習う学生に問題に関して詳細に調査し、どのような支援が双方にとって有効か検討した。

その検討を元に3章で教育支援用のフレームワークが提案されている。そしてこのフレームワークを実現するネットワーク分散型の授業支援システム TSUNA-TASTE について3.3節から3.7節で詳細に述べた。そしてネットワーク分散型の授業支援システム TSUNA-TASTE を実際にプログラミング教育で利用した状況についても述べた。さらに学生がアルゴリズムの可視化を求めている事が判明したため、3.8節ではアルゴリズムの可視化を可能にするシステムについて述べている。

教育支援用フレームワークを活用した適用事例として実践的競技システムに適用した結果を4章で述べた。実践的な場としては全国高等専門学校プログラミングコンテストの競技部門を選択し、本教育用フレームワークの仕組みを利用して競技システムの設計・構築、そして実際の競技を行ってきた。競技システムは4例あり4.1節でネットオークション型システムのシステムを、4.2節で高セキュリティ型ネットワークシステムを、そして4.3節でアルゴリズム対戦型システムを、そして4.4節でインターネット対戦可能な準リアルタイム対戦システムについてそれぞれ述べた。これらのシステムは競技のルールや開催条件により、フレームワークをベースにその都度モジュールや環境を変更し、それぞれの問題に適用できたことから本フレームワークの柔軟性をある程度実証できたと考えている。

5.1節から5.3節ではプログラミング授業支援の効果についてアンケートを行った結果を元に検討し、システムの有用性と必要性が確認できた。また、5.4節では教育支援フレームワークの柔軟性と拡張性が確認できた。

本論文では、プログラミング教育を支援する問題解決環境について、多面的な分析と実際にシステムを構築して適用を行う実践的な運用の両面から検証を行ってきた。システムにはまだ解決すべき問題やさらなる機能の充実が求められている。今後も実際の教育現場で実践的な運用を重ねて、更に優れた問題解決環境の充実を目指したい。

[参考文献]

- [1] 文部科学省「学校における教育の情報化の実態等に関する調査結果」：
http://www.mext.go.jp/b_menu/houdou/21/08/1283617.htm
- [2] C. Boonmee and S. Kawata: Computer-Assisted Simulation Environment for Partial-Differential-Equation Problem: 1. Data Structure and Steering of Problem Solving Process, Transactions of the Japan Society for Computational Engineering and Science, Paper No. 19980001, 1998.
- [3] C. Boonmee and S. Kawata: Computer-Assisted Simulation Environment for Partial-Differential-Equation Problem: 2. Visualization and Steering of Problem Solving Process, Transactions of the Japan Society for Computational Engineering and Science, Paper No. 19980002, 1998.
- [4] S. Kawata, C. Boonmee, A. Fujita, T. Nakamura, T. Teramoto, Y. Hayase, Y. Manabe, Y. Tago, M. Matsumoto: Visual Steering of the Simulation Process in a Scientific Numerical Simulation Environment -NCAS-, Enabling Technologies for Computational Science, edited by E.Houstis and J.Rice, Kluwer Academic Publishes, 2000, pp. 291-300.
- [5] A. Fujita, T. Teramoto, T. Nakamura, C. Boonmee, S. Kawata: Computer-Assisted Parallel Program Generation System P-NCAS from Mathematical Model -Visualization and Steering of Parallel Program Generation Process-, Transactions of the Japan Society for Computational Engineering and Science, Paper No. 20000037, 2000.
- [6] 川田重夫, 田子精男, 梅谷征雄, 南多善: PSE BOOK—シミュレーション科学における問題解決のための環境(応用編)—, 培風館, pp.245-253, 2005.
- [7] J07: <http://www.ipsj.or.jp/12kyoiku/J07/J0720090407.html>
- [8] 寺元貴幸, 日下孝二, 岡田正: Windows と Unix 混在環境の運用方針と設定 I, 第 17 回 (平成 9 年度) 高等専門学校情報処理教育研究委員会研究発表会情報処理教育研究発表会論文集 17, pp. 88-91, 1997.
- [9] 寺元貴幸, 日下孝二, 岡田正: Windows と Unix 混在環境の運用方針と設定 II, 第 18 回 (平成 10 年度) 高等専門学校情報処理教育研究委員会研究発表会情報処理教育研究発表会論文集 18, pp. 168-171, 1998.
- [10] 寺元貴幸, 日下孝二, 岡田正: Windows と Unix 混在環境の運用方針と設定 III, 第 19 回 (平成 11 年度) 高等専門学校情報処理教育研究委員会研究発表会情報処理教育研究発表会論文集 19, pp. 9-12, 1999.
- [11] 寺元貴幸, 岡田正: マルチメディア室を利用した新しい授業スタイル, 工学教育, 45-3,

- pp. 24-27, 1997.
- [12] 寺元貴幸, 岡田正: マルチメディア教室を利用したビジュアルプログラミング教育, 平成8年度工学・工業教育研究講演会講演論文集, pp. 87-90, 1996.
 - [13] 寺元貴幸, 岡田正: マルチメディア教室を利用したビジュアルプログラミング教育Ⅱ, 平成9年度工学工業教育研究講演会講演論文集, pp. 37-40, 1997.
 - [14] 寺元貴幸, 岡田正: 多用途なマルチメディア室の構築と運用, 論文集「高専教育」第21号, pp. 233-238, 1998.
 - [15] 寺元貴幸, 齋藤智也, 岡田正: 分散エージェントによる授業支援システムの構築, 計算工学講演会論文集, 第4巻, 第1号, pp. 231-234, 1999.
 - [16] 寺元貴幸, 齋藤智也, 青山龍一, 岡田正, 川田重夫: 分散エージェントとデータベースサーバによる授業支援システムの構築, 第3回問題解決環境ワークショップ論文集, pp. 115-120, 2000.
 - [17] 寺元貴幸, 岡田正, 青山龍一, 齋藤智也, 中村恭志, 川田重夫: データベースを活用した分散授業支援システムの構築, 計算工学会論文集, Vol.3, pp. 87-93, 2001.
 - [18] 寺元貴幸, 岡田正, 稲田英治, 齋藤智也, 中村恭志, 川田重夫: データベースを活用した分散授業支援システムの構築2, 計算工学会講演会論文集 Vol.7, pp. 123-124, 2002.
 - [19] 稲田英治, 寺元貴幸, 岡田正, 鷹取賢司: 分散授業支援システムにおける学習状況分析に関する研究, 平成14年度電気・情報関連学会中国支部第53回連合大会公演論文集, p.540, 2002.
 - [20] 鷹取賢司, 寺元貴幸, 岡田正: データ収集・分析機能を有する授業支援システムの構築Ⅲ, 平成13年度電気・情報関連学会中国支部第52回連合大会公演論文集, p.283, 2001.
 - [21] 松本寿一, 中易秀敏, 森田英嗣, 亀島鉦二: 教育支援のための教材学習履歴分析システム, 情報処理学会論文誌, Vol.40, No.9, pp. 3596-3607, 1999.
 - [22] Red Hat: <http://www.redhat.com/>
 - [23] Apache: <http://www.apache.org/>
 - [24] Apache Tomcat: <http://tomcat.apache.org/>
 - [25] Java: <http://www.java.com/ja/>
 - [26] Oracle: <http://www.oracle.com/index.html>
 - [27] PostgreSQL: <http://www.postgresql.jp/>
 - [28] MySQL: <http://www.mysql.gr.jp/>
 - [29] PostgreSQL-JDBC ドライバ: <http://jdbc.postgresql.org/download.html>
 - [30] 高度情報通信ネットワーク社会推進戦略本部 (IT戦略本部) : <http://www.kantei.go.jp/jp/singi/it2/index.html>
 - [31] 鈴木浩之, 小西達裕, 伊東幸宏: 抽象的データ構造を含むアルゴリズム表現に基づく

- プログラム評価支援システムの構築, 教育システム情報学会誌, Vol.24, No.3, 2007.
- [32] 小西達裕, 鈴木浩之, 伊東幸宏: プログラミング教育における教師支援のためのプログラム評価機構, 電子情報通信学会論文誌 D-I, Vol. J83-D-I, No. 6, pp. 682-692, Jun. 2000.
- [33] 鈴木浩之, 小西達裕, 伊東幸宏: 抽象的データ構造を用いたアルゴリズム表現に基づくプログラム評価システムの実用性評価, 教育システム情報学会研究報告, Vol. 21, No. 3, pp.25-30, 2006.
- [34] 新村晃示, 鈴木浩之, 稲垣文雄, 伊藤大輔, 小西達裕, 伊東幸宏: プログラミング言語学習を意識させないアルゴリズム作成・テスト環境の構築, 教育システム情報学会研究報告, Vol. 19, No. 5, pp.63-70, 2005.
- [35] 鈴木浩之, 新村晃示, 小西達裕, 伊東幸宏: プログラミング言語に依存しないアルゴリズム表現に基づくプログラム評価手法, 教育システム情報学会研究報告, Vol. 19, No. 3, pp. 49-56, 2004.
- [36] 飯田英祐, 新村晃示, 鈴木浩之, 小西達裕, 伊東幸宏: プログラミング言語を意識させないアルゴリズム学習支援方法, 第 37 回人工知能学会知的教育システム研究会資料, pp.77-82 (SIG-IES-A203-13), 2003.
- [37] 鈴木浩之, 酒井崇臣, 小西達裕, 伊東幸宏: 初等プログラミング教育における教師支援システムの評価, 第 28 回人工知能学会知的教育システム研究会資料, pp.27-32 (SIG-IES-9903-6), 2000.
- [38] 杉山暁, 鈴木啓志, 小西達裕, 伊東幸宏: 初等プログラミング教育におけるプログラム診断に基づく教師支援システム, 第 18 回人工知能学会知的教育システム研究会資料, pp.31-36 (SIG-IES-9701-6), 1997.
- [39] 加藤悠輔, 鈴木浩之, 小暮悟, 小西達裕, 伊東幸宏: プログラム半自動評価システムの実用性向上のための抜粋評価の導入, 教育システム情報学会 JSiSE2007 第 32 回全国大会, pp.180-181, 2007.
- [40] 鈴木浩之, 小暮悟, 小西達裕, 伊東幸宏: 抽象的データ構造を含むプログラムの半自動評価システムの実用性評価, 教育システム情報学会 30 周年記念全国大会講演論文集, pp.209-210, 2005.
- [41] 稜川友宏, 伊藤敏彦, 坂根裕, 新谷誠, 小西達裕, 伊東幸宏: 初等プログラミング教育における組み込みコード品質検証ツールの試験的活用, 教育システム情報学会 第 28 回全国大会講演論文集, pp.141-142, 2003.
- [42] T. Yoshino, J. Munemori, T. Yuizono, Y. Nagasawa, S. Ito and K. Yunokuchi: Application of Distance Learning Support System SEGODON to Exercise-type Classes, Trans. IPSJ, Vol. 40, No. 11, pp. 3946-3955, 1999.
- [43] 大川正人, 室田真男, 中山実, 清水康敬: Web ベース学習における学習履歴画面の時系列再現システムの開発, 電子情報通信学会技術報告, ET99-65, pp. 15-22, 1999.

- [44] 桑原恒夫, 玉城幹介, 山田光一, 中村善宏, 満永豊, 小西納子, 天野和哉: 個人進度別教育支援システム(MESIA)における行き詰まり生徒の支援機能とその効果, 電子情報通信学会論文誌(D-I), Vol. J83-D-I No.9, pp. 1013-1024, 2000.
- [45] 大木幹雄: プログラム動作可視化ツールにみる Program Text Markup Language の可能性, 情報処理学会ソフトウェア工学研究会研究報告, 2003-SE-142, pp.1-8, 2003.
- [46] 高橋詠子, 仙北谷祐輔, 小林宏章, 鈴木信吾: プログラミング言語教育補助システム PLESS -C 言語教育補助システム Cmaster-, 第44回プログラミングシンポジウム, pp.1-12, 2003.
- [47] 大西淳, 野田昌宏, 玉城龍洋, 内田保雄: データ操作過程を可視化する C インタプリター機能と試用結果-, 平成19年度情報教育研究集会講演論文集, pp. 414-417, 2007.
- [48] 全国高等専門学校プログラミングコンテスト: <http://www.procon.gr.jp/>
- [49] ITmedia プログラミングコンテスト津山大会「石垣工務店」:
<http://www.itmedia.co.jp/enterprise/articles/0710/13/news001.html>
- [50] ITmedia 高専プロコンレポート:
<http://www.itmedia.co.jp/enterprise/articles/0710/31/news014.html>

[業績一覧]

論文

- (1) An Education-Support PSE System: TSUNA-TASTE, Takayuki TERAMOTO, Tadashi OKADA, Shigeo KAWATA, Journal of Convergence Information Technology, Vol. 5 Number 4, June 30,2010,ISSN 1975-9320,pp216-223
- (2) プログラミングコンテスト競技部門「石垣工務店」のシステム開発と運用, 寺元貴幸, 宮下卓也, 最上勲, 岡田正, 井上恭輔, 松野良信, 2009.3 高専教育 32, pp. 921-926
- (3) 授業支援システムによるC言語初心者の学習および試験の支援, 寺元貴幸, 渡邊智之, 岡田正, 稲田英治,2005.3 高専教育 28, pp. 453-458.
- (4) A Distributed Problem Solving Environment (PSE) for Partial Differential Equation Based Problems, Takayuki TERAMOTO, Takashi NAKAMURA, Shigeo KAWATA, Syunsuke MATIDE, Koji HAYASAKA, Hidetaka NONAKA, Eiji SASAKI and Yasuhiro SANADA, 2001.5 日本計算工学会論文 論文番号 20010018
- (5) データベースを活用した分散授業支援システムの構築, 寺元貴幸, 岡田正, 青山龍一, 齊藤智也, 中村恭志, 川田重夫, 2001.5 日本計算工学会論文 論文番号 20010008

国際学会論文

- (1) Database design in a distributed educational PSE system, Takayuki Teramoto, Tadashi Okada, Shigeo Kawata, Third Asian-Pacific Congress on Computational Mechanics (APCOM'07) Eleventh International Conference on the Enhancement and Promotion of Computational Methods in Engineering and Science (EPMESC XI) 2007.12.6 (Japan, Kyoto)
- (2) A distributed education-support PSE system, Takayuki Teramoto, Tadashi Okada, Shigeo Kawata, Third IEEE International Conference on e-Science and Grid Computing 2007.12.10 (Bangalore, India)

[謝辞]

本論文の執筆にあたり、宇都宮大学工学部エネルギー環境科学専攻の川田重夫教授には多大なるご助言とご指導をいただきまして心より感謝いたします。

また本論文の執筆過程で、鈴木光政教授、杉山均教授、加藤茂夫教授、横田隆史教授の各先生方には貴重なご意見をいただきまして誠にありがとうございました。

さらに津山高専情報工学科学科長の河合雅弘教授、岡田正教授をはじめ津山高専の先生方には、宇都宮大学での研究に際し多大なご配慮をいただきましてありがとうございました。また、3.8節のアルゴリズム可視化ツールの開発にあたりましては津山高専情報工学科大西淳准教授ならびに津山高専電子・情報システム工学専攻1年の越井翔也君には大変お世話になりました。さらに、島根大学の齊藤智也教務職員をはじめ、多くの方のご協力に心から感謝いたします。

問題解決環境の研究に関しまして、田子精男先生（金沢大学元教授）、静岡大学の梅谷征雄教授、サイバネットシステム株式会社の宮地英生様、玉川大学の宇佐見仁英教授、米子高専の松本正己教授、富山商船高専の早勢欣和講師には様々なご助言本当にありがとうございました。上記の方々の励ましなしに本研究を継続することは困難だったと思います。

全国高等専門学校プログラミングコンテストに関しまして、有明工業高専の松野良信先生、弓削商船高専の長尾和彦先生、石川高専の金寺登先生をはじめ、多くのプロコン委員の方には大変お世話になりました。心から感謝いたします。またシステム開発を手伝ってくれた株式会社ミクシィの井上恭輔様、東京工業大学3年生の山下晃弘様を始め多くの学生諸君にも感謝いたします。

最後になりましたが、宇都宮大学川田研究室の皆さんには大変お世話になりました。心から感謝すると同時に、皆さんの今後の発展を期待しています。