

コンピュータサイエンスアンプラグドの分析と CS アンプラグドデザインパターンの提案

西田 知博^{†1} 井戸坂 幸男^{†2}
兼 宗 進^{†3} 久野 靖^{†4}

「コンピュータサイエンスアンプラグド」は、理解が難しいと思われる情報科学を体験的に学ぶことができる教育方法である。その実践では、ゲームなどの「体験」を通じ、生徒たちが情報科学に興味深く、意欲を持って学習できていることが検証されている。本稿では、CS アンプラグドの構造を分析し、その特長を明らかにするとともに、その分析に基づきデザインパターンテンプレート「CS アンプラグドパターン」を提案する。本パターンは教育方法を新たに考案したり、既存の教育方法を改良する際に、CS アンプラグドに見られる利点を持たせるための指針や評価方法として効果的であることが期待される。さらに、独自教材「伝言ゲーム」の紹介と CS アンプラグドパターンを用いた検証を行い、提案したデザインパターンテンプレートの有用性を評価する。

Analysis of Computer Science Unplugged and Proposal of CS Unplugged Design Pattern

TOMOHIRO NISHIDA,^{†1} YUKIO IDOSAKA,^{†2} SUSUMU KANEMUNE^{†3}
and YASUSHI KUNO^{†4}

“Computer Science Unplugged” is an educational method for learning concepts of computer sciences through hands-on activities. In general, concepts of computer sciences have been considered as being too difficult for elementary and middle school pupils/students. However, when we applied CS Unplugged methods to our classes, pupils/students could have interests on, and successfully understand such concepts through fun activities such as games. In this paper, we analysed the structures of CS Unplugged activities and clarified their features. Based on the analysis, we also proposed a design pattern namely: “CS Unplugged Pattern.” The pattern will be useful for guidelines and evaluation baseline when we develop new educational methods or revise existing ones. We also describe our experience of developing original teaching material: “Telephone Game,” and verified its structure using CS Unplugged Pattern.

1. はじめに

日本では、2002、2003年の学習指導要領の改正により、中学校では教科「技術・家庭」の「情報とコンピュータ」領域が、高等学校では教科「情報」が設置され、情報教育の拡充が図られた。しかし、その授業の多くはアプリケーションソフト（ワープロ、表計算、プレゼンテーションなど）の使用方法を中心とした内容となり、情報科学などは十分には教えられていない。これは、一般には情報科学が中高生には理解が難しいと思われることに原因があると思われる。

しかし、コンピュータはパソコンだけでなく携帯電

話やゲーム機など多くの電子機器で活用されており、情報科学の内容は生徒にとって身近で興味のある題材となりうる。一方、探索やソートなど、個々の内容を取り出した学習では、日常生活との結び付きを理解することが難しいため、何のために学ぶのかを生徒が理解することは難しいという問題もある。

この問題を解決するため、我々はコンピュータを使わずに、情報科学を体験的に学ぶことができる「コンピュータサイエンスアンプラグド」^{1),2)} (以下、CS アンプラグドと呼ぶ) という教育方法を導入し、それを用いた授業の実践を行っている。

CS アンプラグドでは、洗練された教材を使い、自分の手を動かしながら理解することで、情報科学の代表的な内容を、小学生以上の生徒が興味を持って意欲的に学習することが可能である。CS アンプラグドは

†1 大阪学院大学 Osaka Gakuin University

†2 松阪市立飯南中学校 Iinan Junior High School

†3 一橋大学 Hitotsubashi University

†4 筑波大学 University of Tsukuba

ニュージーランドカンタベリー大学の Tim Bell たちが始めたプロジェクトである。Tim Bell は約 10 年前に自分の 5 歳の子供に情報科学の楽しさを教えたいと考えて以来、教育方法と具体的な教材を改良してきた。

彼らは 1998 年に 20 章からなるテキストを出版²⁾し、その後はオンラインで 12 章からなるテキスト¹⁾を提供し続けている。また、韓国語³⁾や日本語⁴⁾にも翻訳され出版されている。

ここでは、CS アンブラグドのテキスト／教材の構造を分析し、それをを用いた学習の特長を明らかにする。また、同様の特長を持つ教材を新たに開発し充実させていくことをめざして、分析結果をデザインパターンの形にまとめた「CS アンブラグドパターン」を提案する。さらに、現在のテキストで不足している部分も同時に分析し、それを補う試みの一つとして、独自教材とそれをを用いた高等学校での実践の紹介とデザインパターンに基づいた検証を行う。

2. CS アンブラグドとは

2.1 CS アンブラグドテキストの構成

CS アンブラグドのテキストは表 1 の 12 個の章から構成され、いずれも情報科学の重要な考え方を扱っている。個々の内容は高等学校から大学の専門課程で扱われる内容だが、説明と教材を工夫することによって、小学生にも理解できるように構成されているのが特長である。

2.2 CS アンブラグド学習の例

ここでは、CS アンブラグドを使った授業の例として「カード交換の手品」を紹介する。手品は以下のような手順で進む。

- (1) 両面に白と黒の色が付いたマグネットカードを用意し、生徒にボード上に規則性のないように

表 1 CS アンブラグドテキストの構成

章	学習	内容
1	点を数える	2進数
2	色を数で表す	画像表現
3	それ、さっきも言った!	テキスト圧縮
4	カード交換の手品	エラー検出とエラー訂正
5	20の扉	情報理論
6	戦艦	探索アルゴリズム
7	いちばん軽といちばん重い	整列アルゴリズム
8	時間内に仕事を終わろ	並び替えネットワーク
9	マッディ市プロジェクト	最小全域木
10	みかんゲーム	ネットワークにおけるルーティングとデッドロック
11	宝探し	有限状態オートマトン
12	出発進行	プログラミング言語

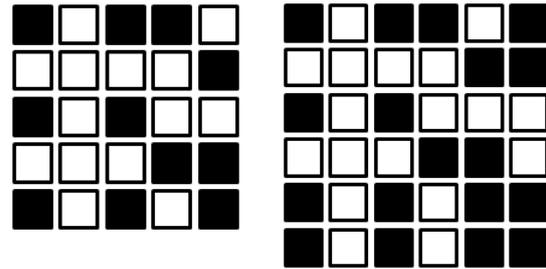


図 1 カード交換手品のカード配置例

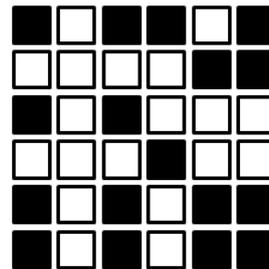


図 2 1枚だけカードを反転させたもの

ランダムに並べてもらう。(図 1(a))

- (2) 先生は「もう少し難しくしましょう」と言いながら、右端と下端に 1 列ずつカードを追加する。(図 1(b))
- (3) 先生は後ろを向き、その間に生徒にどれか 1 枚を裏返してもらう。
- (4) 先生はボードを見て、裏返されたカードを当てる。

図 1(b) のように並べたカードのうち 1 枚だけを裏返したものが図 2 である。

この手品はメモリ上にデータが正しく保存されているか否かや、データ通信が正しく行われているか否かを検証するために用いられる「パリティチェック (parity check)」の原理を学ぶためのものである。タネは、上記の手順 (2) で先生が右端と下端に加えたカードにある。

先生は縦横それぞれの黒いカードの枚数が偶数となるようにカードを加えている。一方、生徒が 1 枚のカードを裏返すと、その行と列の黒いカードの枚数は奇数になる。したがって、先生は黒いカードの枚数が奇数の行と列を見つければ裏返されたカードを当てることができるというしくみになっている。

3. CS アンブラグドの分析

3.1 CS アンブラグド教材の構成

CS アンブラグドの各章（学習）は、以下のような構成となっている。

要約

教科学習との関連、必要とされる技能、対象年齢、教材一覧を示す。

先生への説明、実演の手本

生徒の興味をひくために冒頭で行う実演などについての説明。

生徒の実習

教具やワークシートなどを用い、ゲーム等を通じて実際に「体験」させることにより生徒の意欲を引き出す。

発展の実習

応用問題など、より深い学習を行う。

実際のコンピュータでは

現実の問題への関連付けを行う。

解答とヒント

実習が問いかけになっている場合は、その解答やヒントを提示する。

この中で特に大きな役割を示しているのは、興味と意欲を引き出す先生の「実演」と生徒の「体験」である。たとえば、「カード交換の手品」では手品を見たことの驚きで興味をひき、続いてタネを見つけようと思うことから意欲を引き出している。「戦艦ゲーム」ではゲームであるということに興味を持たせ、ゲームに勝とうと思うことにより戦略＝アルゴリズムを考える意欲を引き出している。「宝探し」では「宝島」をテーマとし興味をひき、その航路を地図として書かせることによって無理なく有限状態オートマトンの考え方を学ばせることができるようになっている。

3.2 CS アンブラグド教材の特長

CS アンブラグドを学校の学習に取り入れることのメリットには以下のようなものがある。

i. 学習の中に、必ずゲームがある。

学習の中に「遊び」の要素を取り入れたことにより、遊びながら学べる教材であり、かつ、生徒の興味・関心を引き出し、意欲的に取り組める教材となっている。

学習する内容には抽象的で難しい内容もあるが、難しさを感じさせず、意欲的に楽しんで学習ができるように工夫されている。

ii. 具体物の試行錯誤から学ぶ。

カードを使って試行錯誤したり、おもりを何度も置き換えたり、具体的なものを動かしたりという体験を

伴う学習が多く取り入れられている。具体物を試行錯誤しながら学習することは、生徒の思考を刺激し、手順的に考える習慣を身につけさせていくものと考えられる。

iii. グループ（集団）で学ぶ。

ひとりで学習するのではなく、グループで学習する場面が多い。お互いに自分の考えを深めあうことができ、個人学習よりも思考の深まりが期待できる。また、生徒同士がコミュニケーションを学習するよい練習にもなる。

iv. 手軽に実践できる。

ほとんどの場合、付属のワークシートを印刷すれば授業ができる。教具が必要な単元は、安価な材料を用いて自作できる。手軽で教材費がかからないのも大きな魅力である。

また、一般的には、情報教育を行う場合、パソコン室などの設備が必要であるが、CS アンブラグドは授業の場所を限定せず、普通教室や屋外での授業も可能となる。

3.3 CS アンブラグドパターンの提案

CS アンブラグドのテキストは、そのまま授業で使えるように工夫されているが、教員によっては授業に応じて改良したり、発展させて使いたい場合がある。

そのような拡張を行った場合でも、CS アンブラグドのエッセンスを失わずに質を維持することができるように、我々はCS アンブラグドの構造を分析し、デザインパターンの形にまとめる作業を行った。

デザインパターンとは、よい設計の規範となるエッセンスを形にしたもので、1970年代後半に、C. Alexander⁵⁾が街や建物を造るときのパターンを整理したことから始まったと言われている。計算機科学分野ではGoFのデザインパターン⁶⁾、アナリシスパターン⁷⁾などが知られている。

これら以外にも、世界各地のパターン活動実践者がソフトウェア開発活動の各方面におけるパターンの収集を行っており、それらを集めたPLoP(Pattern Languages of Programming)⁸⁾のような文献も広く知られている。今回はJ. Coplienが提唱しているデザインパターンを開発するための枠組みであるパターンテンプレート⁹⁾を参考に作業を行った。

コンピュータ科学の分野では、「手順」と「アルゴリズム」は明確で、さらにそれを「適用する場面(の例)」も対応して明確に定まるため、今回検討したテンプレートのような基準を活かして教材開発を行うことは効果が大きいと考えている。図3に、作成したCS アンブラグドパターンを示す。

パターン名 (Pattern Name)

適切な名前を付ける。今回は CS アンプラグドの教材がすべて持つべき性質を扱うことから、「CS アンプラグドパターン」とした。

問題状況 (Problem)

解決しなければならない問題の説明である。今回は「コンピュータ科学を理解することが難しい」とした。

問題背景 (Context)

問題解決を適用できる場面の説明である。今回は CS アンプラグドが想定している対象 (7 歳ないし 9 歳以上) と、少なくとも中学生が理解できることを想定して、「専門知識を持たない子ども (7~15 歳程度) でも理解できる必要がある」とした。

フォース (Force)

このパターンを適用するための条件である。今回は CS アンプラグドに共通する「コンピュータを使わずに教えられる」という特徴と、学校の授業に取り入れることができるために「コンピュータ科学の専門家でない教員が教えられる」「20~40 分程度で学ぶことができる」とした。

解決策 (Solution)

問題を解決するための方策である。今回は前節で行った分析を元に、CS アンプラグドが提供する解決策を検討した。その結果、次のような特徴が必要であることが明らかになった。

「学習にゲームの要素を取り入れる」ことは、学習活動に興味を持たせる上で効果的である。

「教える対象ごとに適した教具を使う」ことは、二進法であれば各桁を表すカードを使う、画像表現であればマス目のあるワークシートを使う、ソーティングであれば天秤を使うといったように、学習活動と伝えたい学習内容を適切に結び付けるために重要である。

「最初に体験による学習を行う」ことは、正解 (最終的な学習内容) が与えられる前に「体験の中で試行錯誤しながら考えさせる」ことにつなげることができる。

「グループによる共同作業を取り入れる」ことは、自発的なコミュニケーションにより生徒の思考や発想を引き出すことにつなげることができる。

「ワークシートを用意する」ことは、教具を用いた学習活動が、学習の目的から逸れないように規定し、本来の活動に導くために有用である。

「教具を手軽で安価に作成できる」ことと「さまざまな場所で行える」ことは、多くの教員が特別

な準備や予算を必要とせず実施でき、特別な場所を必要とせずに普通教室や廊下などで行えるために必要である。

そして、体験した内容を学習内容と結び付けるために、「実際のコンピュータでの利用を解説できる」ための教材が用意されていることも重要である。

問題解決後の状況 (Resulting Context)

CS アンプラグドパターンを適用した後の状況と、さらに解決すべき課題の説明である。今回は、体験そのものは問題なく行えることから、「体験した特定の状況は理解している」を挙げ、さらに解決すべき問題となる可能性として、「体験した内容とコンピュータ科学の原理とは、漠然とした対応しにくい」「コンピュータ科学の領域のうち、一部分だけが扱われている」とした。

根拠 (Rational)

CS アンプラグドパターンが有効であることの説明である。今回は、「共同作業とゲームの要素により意欲が向上し、子どもたちがコンピュータ科学の内容を真剣に考える」ことと、「集中して考える体験により、体験した内容が学習として定着する」ことにより、学習効果が高いことを説明し、「コンピュータ科学のエッセンスを体験できるように教材が工夫されている」ことから、重要な考え方を理解できることを説明した。

CS アンプラグドのコンセプトを満たす独自の教材を開発する場合は、このパターンに沿って作成することが有効である。また、これまで独自開発してきた教材も、このパターンを用いて検証し、改良点などを検討することにより、さらによいものにできる可能性がある。また、パターンを用いることにより各教員が過去に作成した教材が CS アンプラグドのコンセプトを満たすかどうかを判別、整理することも可能となる。

次節では独自開発した教材を紹介し、その実践を CS アンプラグドパターンを用いて評価する。

4. 独自教材の開発と実践

4.1 独自教材開発の必要性

前節で述べたように、CS アンプラグドは情報科学のテーマを楽しみながら学べ、生徒の意欲を引き出す優れた教材である。しかし一方で、ゲームなどの用意されている学習活動だけでは「実際のコンピュータでは」で紹介されている実用に関する話題を十分理解できないものもある。

たとえば、4 章の「カード交換の手品」では、手品の学習活動によりパリティの有効性を実感できるが、

パターン名: CS アンプラグドパターン

問題状況:

- コンピュータ科学を理解することが難しい

問題背景:

- 専門知識を持たない子ども (7~15 歳程度) でも理解できる必要がある

フォース:

- コンピュータを使わずに教えられる
- 専門家でない教員が教えられる
- 20~40 分程度で学ぶことができる

解決策:

- 学習にゲームの要素を取り入れる
- 教える対象ごとに適した教具を使う
- 体験による学習を行う
- 体験の中で試行錯誤しながら考えさせる
- グループによる共同作業を取り入れる
- ワークシートを用意する
- 教具を手軽で安価に作成できる
- さまざまな場所で行える
- 実際のコンピュータでの利用を解説する

問題解決後の状況:

- 体験した特定の状況は理解している
- 体験した内容とコンピュータ科学の原理とは、漠然とした対応しかない
- コンピュータ科学のその領域のうち、一部分だけが扱われている

根拠:

- 共同作業とゲームの要素により意欲が向上し、コンピュータ科学の内容を真剣に考える
- 集中して考える体験により、体験した内容が定着する
- 体験する内容はコンピュータ科学のエッセンスから構成されている

図 3 CS アンプラグドパターン

なぜパリティが必要なのかということは、「実際のコンピュータでは」の中でエラーに関する説明を行っているだけで、それを体感的に理解できるようになっていない。

このようなギャップは、高校生を対象とした場合など、学習を楽しむだけではなく、情報科学の本質的な部分を理解してもらうことを目的とした場合には何らかの形で埋める必要がある。日本語訳のテキストには追補が付いており、このギャップを埋めることの助け

となっているが、その内容を生徒が理解できるよう授業に展開するのは簡単ではない。したがって、導入の学習活動と同様な体験的な教材を、そのようなギャップを埋めるために開発する必要がある。

我々は、そのような独自教材の開発も進めている。以降では、「エラー検出と訂正」の理解を促すために開発した教材「伝言ゲーム」と、高等学校における実践について紹介する。

4.2 高校での実践

大阪学院大学高等学校では、筆者の一人である大学教員が高大連携の一環で3年生を対象にプログラミングの授業を行っている。2007年4~12月に行った授業は10名の生徒が受講し、プログラミング課題の導入や展開の題材として、CS アンプラグドテキストの「子供 FAX」、「カード交換の手品」、「戦艦ゲーム」、「宝探し」を用いた。

このうち、「カード交換の手品」を用いた「エラー検出と訂正」をテーマとした授業について紹介する。この授業は100分間で、以下の流れで進めた。

- (1) カード交換の手品
- (2) 伝言ゲーム
- (3) 配布したプログラムによりパリティチェックでエラーが検出できることの確認
- (4) ISBN のチェックデジットの解説と手による計算
- (5) ISBN のチェックデジット計算のプログラミング演習

この中で挙げている「伝言ゲーム」(図 4) はテキストには含まれていない独自教材である。

この教材を開発した理由は、担当者が「カード交換の手品」により単にパリティでエラー訂正ができるということを学ぶだけではなく、通信等でエラー訂正が必要になる状況を作り、生徒がその必要性を実感することが必要であると考えたからである。



図 4 「伝言ゲーム」の様子

伝言ゲームは以下の手順で行った。

- (1) 図 5 に示した 11×9 のビットマップを 0,1 の

- 数値で表したワークシートを最初の生徒に渡す。
- (2) 隣の生徒に 0,1 の数字の列を伝えさせる。その際、伝えるのは一度だけで、やり直しをしてはいけないという指示を与える。
 - (3) 0,1 の数字の列を伝えられた生徒はさらにそれを隣の生徒に伝言する。ただし、パリティ付きのデータを伝言する場合は、おかしいと気付いた場所を修正してよいこととする。
 - (4) 最終の学生は伝えられた 0,1 の数字の列をビットマップとして構成する。

ゲームは 10 名の学生を 3 チーム (チーム A:3 名, チーム B:3 名, チーム C:4 名) に分け、1 回目にはパリティを付けないデータで、2 回目はパリティを付けたデータでゲームを行った。なお、チーム内での伝言を行う順序は、1 回目と 2 回目では逆とした。

表 2 「伝言ゲーム」の結果

チーム m	1 回目	2 回目
A	失敗 (19 箇所誤り)	失敗 (1 箇所誤り)
B	失敗 (6 箇所誤り)	成功
C	失敗 (2 箇所誤り)	成功

結果は表 2 のように、2 回目のパリティ付きの場合では 2 チームが誤りなく伝言を行うことができた。また、残り 1 チームも 1 箇所のみの誤りという結果であった。2 回目でゲームに慣れたということを差し引いても、パリティの有効性が実感できる結果となった。

これは、授業後に生徒に行った以下のアンケート結果 (表 3) にも表れており、伝言ゲームを通してパリティチェックの必要性を実感させることができるという結果が得られた。

表 3 パリティチェックの役割の理解度

パリティチェックの役割が理解できましたか?	
理解できた	6
大体理解できたと思う	3
あまり理解できていないと思う	1
全く分からなかった	0

授業期間が終了した後のアンケートとして、演習テーマを列挙し、面白いと思ったテーマを自由に選択してもらった (回答者 9 名)。その結果、「伝言ゲーム」を面白いと答えた生徒は 7 名 (78%) でもっとも多いという結果となった。また、これに次いで「戦艦ゲーム」と「地図作成と復元」を 6 名が面白かったと答えている。この「地図作成と復元」も CS アンブラグドテキストにある教材を基にした独自の教材 (授業展開) である。これは、「宝探し」で導入の学習を行った後、

生徒にオリジナルの状態遷移図を作成させ、そのプログラミングを行わせる。その後、プログラムを隣の生徒同士で交換し、地図がうまく復元できるかどうかを演習させたものである。

同様に CS アンブラグドの学習からプログラミングにつなげる授業の試みとして、「戦艦ゲーム」の後に「検索プログラム」の作成を行った。しかし、上の結果と対照的に、「検索プログラム」が面白いと答えた生徒は 0 であった。これは、「戦艦ゲーム」と実際のインプリメントとの間に大きなギャップが存在することを示す結果であると考えられる。今後は、ゲームのモデルと実現のギャップを埋めるような教材開発の手法を示し、実践していく必要がある。

4.3 CS アンブラグドパターンを用いた検証

ここでは、3.3 で示した CS アンブラグドパターンの「フォース」、「解決策」、「根拠」を用いて、今回開発した「伝言ゲーム」を検証する。

フォース:

- コンピュータを使わずに教えられる
 - － ゲーム後にコンピュータ上でプログラムを動かす検証を行うが、「伝言ゲーム」自体にはコンピュータは使っていない。
- 専門家でない教員が教えられる
 - － 今回担当した教員は、CS を専門とする大学教員であったが、ゲーム内容は易しく、非専門家でも指導可能である。これは、TT で補助を行った CS を専門としない高校教員の評価からも確認できた。
- 20~40 分程度で学ぶことができる
 - － 1 チームの人数に依存するが、30 分以内でゲームを終えることができる。

解決策:

- 学習にゲームの要素を取り入れる
 - － 今回の学習はゲームである。
- 教える対象ごとに適した教具を使う
 - － 今回の対象 (高校生) に対しては、適していると判断するが、他の対象については検討が必要である。
- 体験による学習を行う
 - － 通信時に起こるエラーを体験させるのがこのゲームの目的である。
- 体験の中で試行錯誤しながら考えさせる
 - － 第 1 回目のゲームの結果をクラス全体に示し、伝達がうまくいっていないことを実感させた。このことが、第 2 回目のゲームで生徒がパ

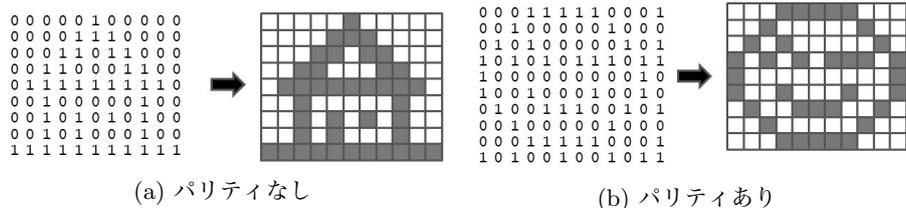


図5 「伝言ゲーム」で用いたデータ

- ティを有効に使うことにつながり、結果として間違いをほとんどなくすることができた。
- グループによる共同作業を取り入れる
 - － チーム対抗のゲームとなっている。
 - ワークシートを用意する
 - － ゲーム後に全体をまとめ、振り返らせるためのワークシートを用意した方が望ましかった。
 - 教具を手軽で安価に作成できる
 - － 用意するのはゲーム用のワークシートのみで安価である。
 - さまざまな場所で行える
 - － ワークシートに記入できる環境があればどこでも行える。
 - 実際のコンピュータでの利用を解説する
 - － ゲーム後にコンピュータ上でプログラムを動かす、検証を行った。

根拠:

- 共同作業とゲームの要素により意欲が向上し、コンピュータ科学の内容を真剣に考える
 - － 生徒は意欲を持ってゲームに取り組んでおり、ゲーム後に通信におけるパリティの重要性を認識していた。
- 集中して考える体験により、体験した内容が定着する
 - － 4.2にも示した通り、授業後のアンケートの結果、短期的にはパリティチェックの役割を10名中6名が「理解できた」、3名が「大体理解できたと思う」と答えている。また、全授業後の「面白いと思ったテーマ」を選択させるアンケートでは、「伝言ゲーム」を選んだ生徒は9名中7名と最も多いという結果となった。
- 体験する内容はコンピュータ科学のエッセンスから構成されている
 - － 通信はCSのエッセンスである。

以上のことより、今回開発した教材「伝言ゲーム」は、CSアンプラグドのパターンテンプレートにほぼ沿った内容であると考えられる。ただし、対象とする

学習者に対してどのような教具を用いるべきかの検討や、学習者に学習内容をまとめさせるために、ワークシートを用いた演習を加えていく必要がある。

今回の教材開発と授業実践はCSアンプラグドパターンの作成以前に行われたものである。しかし、パターンを元に計画を立て、PDCAサイクルに沿って教材開発と実践を進めることで、よりスムーズに効果的な教材を開発できるようになる可能性がある。「伝言ゲーム」においては評価段階になってからCSアンプラグドパターンを用いたが、今後は、パターンを利用して改善を行い、新たな実践を進めていきたい。

5. まとめ

ここでは、コンピュータを使わずに情報科学を学習できる教育方法であるCSアンプラグドを分析し、そのデザインパターンを提案した。

CSアンプラグドは題材は高度な情報科学の概念も含まれているが、教員による「実演」やゲームなどの「体験」を通じ、生徒たちが興味深く、意欲を持って学習できるような教材となっている。

また、提案したCSアンプラグドパターンを用いることによって、ここで紹介した独自教材「伝言ゲーム」の検証を行ったが、その結果は、教材開発者がその内容を振り返り、改良を考える上で非常に有用であった。このことより、今後、増えてくるであろう、その他のCSアンプラグドのコンセプトに基づいた新しい教材の開発に役立つものとなるであろう。

今後は、CSアンプラグドパターンを用いた教材の開発方法について整理し、CSアンプラグドを用いる多くの教員が独自の教材を開発できるようにしていきたい。

参考文献

1) Tim Bell, Ian H. Witten, Mike Fellows: Computer Science Unplugged – An enrichment and extension programme for primary-aged children, 2005.
<http://csunplugged.com/>

- 2) Tim Bell, Ian H. Witten, Mike Fellows: Computer Science Unplugged... off-line activities and games for all ages, 1998.
 - 3) Lee WonGyu (translation): Computer Science Unplugged (Korean Version), 2006.
 - 4) 兼宗 進監訳: コンピュータを使わない情報教育 アンプラグドコンピュータサイエンス, イーテキスト研究所, 2007.
 - 5) Christopher Alexander, Sara Ishikawa, Murray Silverstein: A Pattern Language: Towns, Buildings, Construction, 1977.
 - 6) Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software, 1995.
 - 7) Martin Fowler: Analysis Patterns: Reusable Object Models, 1996.
 - 8) Wikipedia: Pattern Languages of Programming. <http://en.wikipedia.org/wiki/PLOP>
 - 9) 中谷 多哉子: デザインパターンを活用する, 2002. http://www.s-lagoon.co.jp/object/pattern_use.pdf
-