

CS アンプラグド “オロチの食事” の提案と試行 － プロセススケジューリングを学ぶ －

福岡 久雄^{1,a)} 渡部 徹² 廣瀬 誠² 小川 仁士³

概要: 新しい CS (Computer Science) アンプラグド Activity として, オペレーティングシステムの主要機能であるプロセススケジューリングを学習するような Activity を提案する. プロセススケジューリングのアンプラグド化には出雲神話に登場するヤマタノオロチが食事をする状況を利用した. そのため, この Activity を “オロチの食事” 名付けた. アンプラグド化の妥当性については CS アンプラグドのデザインパターンによって検証した. “オロチの食事” を通して, 学習者はプロセススケジューリングの必要性を理解できるとともに, 具体的なスケジューリングを体験的に学習することができる. “オロチの食事” を高等専門学校の低学年生や大学の新生に対して試行し, アンケートによってフィードバックを得た. 本稿では, プロセススケジューリングのアンプラグド化の概要, “オロチの食事” の実施手順ならびに試行結果について報告する.

CS Unplugged “Dining Eight-Headed Dragon” － Learning Process Scheduling －

HISAO FUKUOKA^{1,a)} TORU WATANABE² MAKOTO HIROSE² HITOSHI OGAWA³

Abstract: This paper proposes a new CS Unplugged activity for learning process scheduling, one of the essential functions of operating systems. In order to make the function of process scheduling unplugged, we have utilized Yamata-no-Orochi (Eight-Headed dragon) that appears in Izumo Myths, after which we have named the activity “Dining Eight-Headed Dragon”. We have verified our process of unplugging based on the CS Unplugged design pattern. “Dining Eight-headed Dragon” enables its learners to understand the necessity of process scheduling and to experience some concrete process scheduling methods. We also describe the trials of this activity, of which targets were the students of a technical college and a university.

1. はじめに

子供向けのコンピュータ関連技術教育法の一例として, Timothy Bell 等による Computer Science Unplugged (CS アンプラグド) が提唱されている. CS アンプラグドでは, 一見してコンピュータ関連技術とは無関係の各種カードゲームなどを用いることによって, 「コンピュータを使うことなく, 各種コンピュータ関連技術の考え方を教授す

ることを目指している.

CS アンプラグドのホームページ [1] や日本語教則本 [2] によれば, 各種コンピュータ関連技術に対応して, 既に 20 種類を超える Activity (学習項目) が用意されている. また, 新たな Activity の開発が各所で行われている [3], [4], [5], [6].

本稿では新たな CS アンプラグド Activity として, オペレーティングシステム (OS) の主要機能であるプロセススケジューリング [7] について学習するためのものを提案する. プロセススケジューリングをアンプラグド化するに当たって, ヤマタノオロチが一カ所に置かれた食糧を奪い合う状況を想定した. ヤマタノオロチは出雲神話に登場する, 1 つの体に 8 つの頭を持つ大蛇であるが [8], [9], 便宜上, 8 つの頭それぞれをオロチと称し, その 8 体のオロチ

¹ 東京電機大学
Tokyo Denki University, Inzai, Chiba 270-1382, Japan

² 松江工業高等専門学校
National Institute of Technology, Matsue College

³ 県立広島大学
Prefectural University of Hiroshima

a) fukuoka@mail.dendai.ac.jp

が食糧を一斉に奪い合うような状況設定とした。この奪い合いを調停し、各オロチが整然と食事をするためには何らかの方法でオロチをスケジューリングをすることが必要となる。これによって、コンピュータにおいて複数のプロセスがCPU時間を奪い合う状況と、それを調停するためのプロセススケジューリングをうまく表現できると考えた。なお、このことから提案するActivityを“オロチの食事”と命名した。“オロチの食事”を通して、学習者はプロセススケジューリングの必要性を理解することができる。さらに、いくつかの具体的なスケジューリングを体験することによって、その多様性とそれぞれの特徴についても学習することができる。

“オロチの食事”を高等専門学校の低学年生（2年生、3年生）や大学の新生生に対して試行した。試行後のアンケートの結果、概ね前向きな評価を得たことから、その実用性ならびに有効性が確認できたと判断する。

なお、筆者らが調査した範囲ではプロセススケジューリングを学習するためのCSアンプラグドActivityは他に見い出せなかった。

以降、第2章ではプロセススケジューリングのアンプラグド化について述べる。第3章では“オロチの食事”の詳細な実施手順について述べる、これらを受けて、第4章ではCSアンプラグドのデザインパターンにより今回のアンプラグド化を評価する。第5章では3回の試行とアンケート結果について報告する。第6章では試行結果に関して考察する。第7章で今後の課題について述べ、第8章でまとめを行う。

2. プロセススケジューリングのアンプラグド化

OSの主要機能であるプロセススケジューリングについて概観し、それをアンプラグド化する方策について述べる。

2.1 プロセススケジューリング

プロセスとは動作中のプログラムのことであり、OSによってその動作状態が管理される。一般に、動作中のコンピュータにおいては同時に複数のプロセスが存在し、それらがCPU時間を奪い合うような状況にある。そこで、OSは特定のタイミングで次に実行すべき（すなわち、CPU時間を割り当てるべき）プロセスを何らかの方針（スケジューリングポリシー）に従って決定する。この機能をプロセススケジューリングと呼ぶ。この方針に関してはFCFS（First Come First Served, 到着順）、SPTF（Shortest Processing Time First, 処理時間順）^{*1}、ラウンドロビン（Round-Robin）などが知られている[7]。

^{*1} 文献[7]ではSPTFのことをSPTと称しているが、両者は同じものである。本Activityでは一貫してSPTFという名称を用いている。

表 1: 技術要素のアンプラグド化

技術要素	アンプラグド化
プロセス	オロチ
CPU 時間	食糧
プロセススケジューリング	オロチスケジューリング
プロセスの処理実行	オロチの食事
プロセスの処理時間	オロチの食事量
OS	学習者

プロセススケジューリングは、横取り（Preemption）のあるタイプと横取りのないタイプに大別される。横取りのあるプロセススケジューリングとは、ある特定のタイミングでOSが実行中のプロセスを中断し、他のプロセスの実行を開始することを許すようなものである。上述のFCFSやSPTFは横取りのないプロセススケジューリングであり、ラウンドロビンは横取りのあるプロセススケジューリングである。

2.2 アンプラグド化

プロセススケジューリングをアンプラグド化するに当たって、ヤマタノオロチが一カ所に置かれた食糧を奪い合う状況を利用した。ヤマタノオロチは出雲神話に登場する、1つの体に8つの頭を持つ大蛇であるが、便宜上、8つの頭それぞれをオロチと称し、その8体のオロチが食糧を一斉に奪い合うこととした。詳細な状況設定は以下の通りである。

- (1) 8体のオロチが食事をする。
- (2) 一度に食事ができるオロチは1体だけである。
- (3) 食糧は離散的なものとする。
- (4) オロチが1単位時間に食べる量は1単位食糧とする。
- (5) 事前に何らかの方法で各オロチの食事開始時刻とそれぞれが食べる食糧量を定めておく。例えば、8面ダイスを振って、各オロチ D_1, D_2, \dots, D_8 が食べるべき食糧の量 L_1, L_2, \dots, L_8 を決めることが考えられる。または、教授側が事前に定めた開始時刻や食糧量を学習者に与えてもよい。

設定された状況において、食糧の奪い合いを調停し、各オロチが整然と食事をするためには何らかの方法でオロチをスケジューリングをすることが必要となる。この状況は、コンピュータにおいて複数のプロセスがCPU時間を奪い合う状況に類似している。なわち、各オロチがプロセスに、食料がCPU時間に、そして食事をするという行為がプロセスの処理実行に対応付けられる。学習者は、オロチをスケジューリングするが、これはOSがプロセスをスケジューリングすることに対応する。コンピュータにおける技術要素とそれぞれのアンプラグド化について表1に整理する。

上述の状況設定の下で特定のオロチスケジューリングは

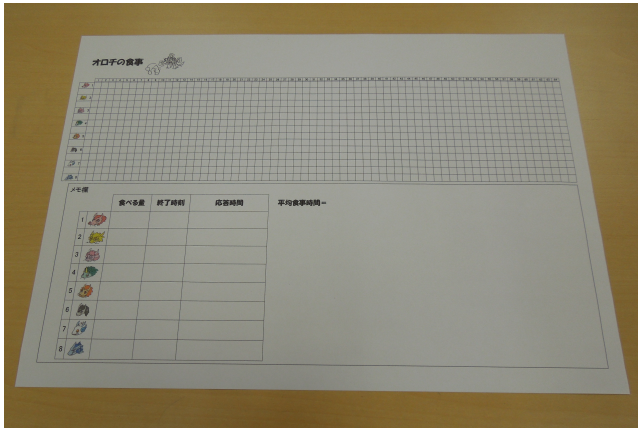


図 1: “オロチの食事” で用いるタイムチャート

次のように実行される。なお前提として、スケジューリングの進捗状況を記入していくタイムチャートを用意しておくこととする(図1)。その大きさは、机上で作業できるようにするため、A3サイズである。

- (1) 何らかの方法で次に食事をするオロチを選ぶ。何らかの方法でそのオロチが食べる食糧の量を決定する。一般的に、これらの「何らかの方法」はスケジューリングの方針に依存して決まる。
- (2) 各オロチの食事の進捗状況をタイムチャートに記入していく。
- (3) これら2つ作業を、すべてのオロチが所定の量を食べ終わるまで繰り返す。
- (4) すべてのオロチが食事を終えた後、平均食事時間(待ち時間を含む)と各オロチの応答時間(そのオロチが食事をする状態になってから、実際に食事にありつくまでの待ち時間)を計算する。

3. Activityの詳細な実施手順

“オロチの食事”の実実施手順について詳述する。なお、以下の説明では各オロチは時刻1の時点で食事可能な状態にあると想定している。

3.1 導入

“オロチの食事”への導入として、8体のオロチが食事をする状況とそこで発生する問題ならびにスケジューリングの必要性を学習者に説明する。これを平易に行うために、図2に示すような4コマ漫画を用意した。

3.2 練習ラウンド

練習ラウンドとして、ランダムなスケジューリング(以下、ランダムと略す)を行い、タイムチャートの記入方法を学ぶ。例えば、8面ダイスを2回振って、次に食事をするオロチと、そのオロチがその時食べる量を決定。食べる量に応じて、タイムチャートにマークを記入する。これをすべてのオロチが食事を終えるまで繰り返す。なお、

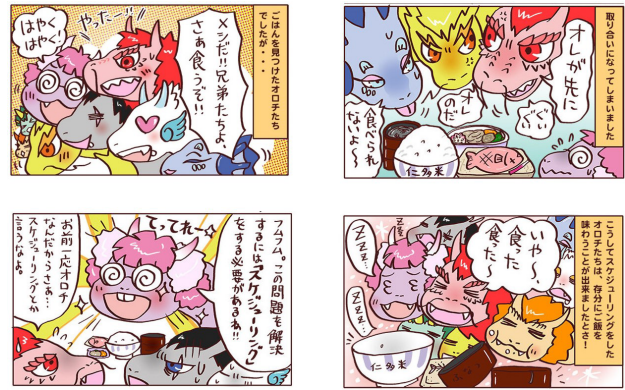


図 2: 導入に用いる4コマ漫画

ここでのポイントは特定の方針ではなくランダムに食事をするオロチとその時の食事量を定めることにあり、8面ダイスを用いることが必須ではない。

$L_1 = 6, L_2 = 8, L_3 = 3, L_4 = 7, L_5 = 4, L_6 = 5, L_7 = 8, L_8 = 5$ という条件下で、ランダムを実際に行った場合のタイムチャート記入例を図3に示す。

このランダムの結果から計算される平均食事時間は33.625である。今回用意したタイムチャートの場合、平均食事時間は各オロチの最終食事時刻の総和を8で割ることにより計算される。

応答時間は「あるオロチが食事をする状態になってから、実際に食事にありつくまでの待ち時間」のことであり、1体のオロチについて複数個存在する場合がある。例えば図3の D_4 の場合、時刻1で食事可能であるが、食事にありつくのは時刻29であり、応答時間は28である。次に時刻33でも食事可能であるが、食事にありつくのは時刻38であり、その応答時間は5となる。同様に、時刻40から時刻45の待ち時間も応答時間である。さらに、 D_7 の応答時間は23, 1, 11, 1であることが分かる。

3.3 FCFSの導入

次に、決定論的なスケジューリングとしてFCFSを導入する。これは先着順に資源を割り当てるという最も単純なスケジューリングである(オロチ $D_1, D_2, D_3 \dots$ の順に到着したと想定)。また、あるオロチが食事を始めると、そのオロチが食べるべき量をすべて食べ終わるまで、他のオロチがスケジューリングされないことを注意しておく必要がある。これは、FCFSが横取りのないスケジューリングであることによる。

FCFSが先着順であることを説明した上でスケジューリングを行わせ、その結果をタイムチャートに記録させる。ただし、各オロチの食事量は練習ラウンドと同じとする。その実行結果例を図4に示す。今の場合、平均食時間は26.125であり、応答時間は1体のオロチについて1個だけ存在し、 $D_1, D_2 \dots D_8$ の順に0, 6, 14, 17, 24, 28, 33, 41である。

オロチ	食事量	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46					
D_1	6																●	●	●	●	●																															
D_2	8			●	●	●	●	●	●	●	●													●																												
D_3	3	●	●																					●																												
D_4	7																																																		●	
D_5	4																																																		●	
D_6	5											●	●																																							
D_7	8																																																			
D_8	5																																																			

図 3: ランダムの結果例

オロチ	食事量	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46						
D_1	6	●	●	●	●	●	●																																														
D_2	8							●	●	●	●	●	●	●	●	●																																					
D_3	3																																																				
D_4	7																																																				
D_5	4																																																				
D_6	5																																																				
D_7	8																																																				
D_8	5																																																				

図 4: FCFS の結果例

3.4 問題提起と誘導

ランダムおよび FCFS の結果得られたタイムチャートにおける平均食事時間と応答時間に注意を向けさせ、スケジューリングに依存して、これらが異なることを確認させる。これを受けて以下の2つの問題を提起し、その解を学習者に考えさせる。

(1) 問題 1

平均食事時間を最小にするようなスケジューリングを考案する。

(2) 問題 2

応答時間のばらつきを極力小さくするようなスケジューリングを考案する。

問題 1 のゴールは、SPTF である。これは食事量の少ないオロチから順にスケジューリングするものであり、横取りのないスケジューリングである。

各オロチの食事量を練習ラウンドと同じとして、SPTF を行った結果例を図 5 に示す。平均食事時間は 22.000 であり、ランダムや FCFS から改善されていることが確認できる。応答時間は $D_1, D_2 \dots D_8$ の順に 17, 30, 0, 23, 3, 7, 38, 12 である。

問題 2 のゴールはラウンドロビンである。これは横取りのあるスケジューリングである。

各オロチの食事量を練習ラウンドと同じとして、ラウンドロビンを行った結果例を図 6 に示す。平均食事時間は 36.375 であり、SPTF に比べて大幅に悪化していることが分かる。一方、定性的にはあるが、FCFS に比べて応答時間のばらつきが概ね小さくなっていることが読み取れる。

いずれの問題においても、子供たちだけで解答を見出すのが困難な場合には、適宜ヒントを与え、それぞれのゴールに導く。現時点では、問題 1 に関しては「多く食べるオロチは待つのを厭わないが、少ししか食べないオロチは待ちたくない」というヒントを用意している。問題 2 に関し

ては、「特定のオロチに食べ続けさせないことがポイント。例えば、1 回の食事機会に食べる量がある（小さな）値に固定し、その量を食べたオロチは強制的に食事を打ち切っては？」というようなヒントを用意している。正解に至るかどうかにかかわらず、学習者には実際に SPTF とラウンドロビンに従ったスケジューリングを実施させる。その結果得られる平均食事時間と応答時間が、両スケジューリングの間でトレードオフの関係にあることを説明する。なお、必ずしも「トレードオフ」という用語を使う必要はなく、そのような関係にあることを学習者のレベルに合わせて平易に説明する。

3.5 フォローアップ

SPTF とラウンドロビンを体験したことで学習者の作業は終了し、後は座学形式でフォローアップを行う。フォローアップでは以下の点を強調し、スケジューリングに関する知識の定着を図る。

- 方針に応じて多様なスケジューリングが考えられること
- 平均処理時間（平均食事時間）の短縮と応答時間のばらつき抑制は両立しないこと
- スケジューリングには横取りのあるタイプと横取りのないタイプがあること
- 実際の OS（Windows, Android 等々）がプロセススケジューリングを行う機能を有すること

フォローアップは CS アンプラグドの Activity において重要な位置を占める。学習者の状況（年齢層、事前知識のレベル等）に合わせて表現に工夫を加えるなど、ふさわしいフォローアップを準備しておく必要がある。

4. デザインパターンとの対応

“オロチの食事” の設計に当たって、Nishida 等が提案

オロチ	食事量	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46							
D_1	6																		●	●	●	●	●																															
D_2	8																																																					
D_3	3	●	●	●																																																		
D_4	7																																																					
D_5	4																																																					
D_6	5																																																					
D_7	8																																																					
D_8	5																																																					

図 5: SPTF の結果例

オロチ	食事量	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46									
D_1	6	●																																																						
D_2	8		●																																																					
D_3	3			●																																																				
D_4	7				●																																																			
D_5	4					●																																																		
D_6	5						●																																																	
D_7	8							●																																																
D_8	5								●																																															

図 6: ラウンドロビンの結果例

する CS アンプラグドのデザインパターン [10] に従った。ここでは、同デザインパターンにおける、設計段階で考慮すべき項目に関する対応状況について整理する。

(1) 問題状況 (Problem)

- コンピュータ科学を理解することが難しい
OS の学習においてプロセススケジューリングは必須の学習項目である。

(2) 問題背景 (Context)

- 専門知識を持たない子ども (7 から 15 歳程度) でも理解できる必要がある
OS というものの存在については知っている学習者を前提とする。また、スキルとしては平均値の計算ができることのみを前提とするが、コンピュータ分野特有の略語 (アルファベット数文字の場合が多い) を導入することから中学生以上を対象とする。

(3) フォース (Forces)

- コンピュータを使わずに教えられる
学習者の作業はタイムチャートへの記入が主であり、コンピュータを使うことはない。
- 専門家でない教員が教えられる
学習者の作業進行を指導するだけであれば専門知識を必要としない。しかし、実際のコンピュータでの利用の解説までを考えると、OS に関する知識を有する教員であることが望ましい。
- 20~40 分で学ぶことができる
90 分授業 1 回または 50 分授業 2 回を想定して設計している。

(4) 解決策 (Solution)

- 学習にゲームの要素を取り入れる
スケジューリング作業そのものにゲーム性はない。SPTF やラウンドロビンを自ら考案する段階でチーム間での競争を取り入れ、チャレンジ性を確保して

いる。

- 教える対象ごとに適した教材を使う
試行に基づいてその要否を判断する。
- 体験による学習を行う
スケジューリングの進行過程をチャートに記入することによって、体験的な学習となっている。
- 体験の中で試行錯誤しながら考えさせる
SPTF やラウンドロビンを考案させることで、「試行錯誤しながら考える」という状況を作り出している。
- グループワークによる共同作業を取り入れる
グループでの共同作業を前提とする。
- ワークシートを用意する
スケジューリングの進行を記入するタイムチャートを用意する。これを、事後の振り返りにも用いる。
- 教具を手軽で安価に作成できる
用意するのは基本的にタイムチャートとのみであり、必要に応じて 8 面ダイス等も用意する。いずれも、安価に用意できるものである。
- さまざまな場所で行える
タイムチャートに記入できる環境があればどこでも実施できる。
- 実際のコンピュータでの利用を解説する
解説用のスライドを用意する。

(5) 根拠 (Rationale)

- 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
プロセススケジューリングはコンピュータ科学のエッセンスである。

5. 試行とその結果

本稿執筆時点までに、上述の手順に従って“オロチの食事”の試行を 3 回行い、アンケートによるフィードバック

表 2: 試行の各種諸元

	第 1 回試行	第 2 回試行	第 3 回試行
日時	2014 年 8 月 4 日 (月)	2015 年 2 月 17 日 (火)	2015 年 4 月 9 日 (木)
場所	松江工業高等専門学校 (松江市)	松江工業高等専門学校 (松江市)	東京電機大学 (千葉県印西市)
対象者	情報工学科 3 年生	情報工学科 2 年生	情報環境学部 1 年生
人数	16 名 (男子 12 名, 女子 4 名)	16 名 (男子 10 名, 女子 6 名)	5 名 (男子 5 名)
体制	4 チーム (1 チーム 4 名)	4 チーム (1 チーム 4 名)	5 チーム (1 チーム 1 名)
年齢	17~18 歳	16~17 歳	18~19 歳

を得た。各試行の諸元を表 2 に示す。

5.1 第 1 回試行

対象者の 16 名はいずれも小学校時代および中学校時代を島根県内で過ごしてきた学生たちである。

この試行では各オロチの食べる食糧量を、各チームが 8 面ダイスをを用いてランダムに決定した。従って、オロチの食事量集合 (各オロチが食べる量からなる集合) が異なることとなった。また、各オロチの食事開始時刻はすべて同じ時刻 (時刻 1) とするように、教授側が指定した。

試行全体を通して、学生たちは積極的に作業に取り組んでくれた。問題提起に対するチーム内での議論も活発であり、プロセススケジューリングに興味を持ってもらえたと判断する。

試行に要した時間は、アンケートの記入時間も含めて約 1.5 時間であった。

問題提起に対しては、各チームとも活発に議論してはいたが、やや発散する傾向にあったようで、問題 1、問題 2 ともにヒントなしで正解に至ったチームはなかった。

なお、今回は正規の授業で OS に関する科目を履修する直前の学生を対象としたため、フォローアップでは上述の内容に加えて少し技術的に突っ込んだ説明を行った。

5.2 第 2 回試行

第 1 回試行と同様に、対象者の 16 名はいずれも小学校時代および中学校時代を島根県内で過ごしてきた学生たちである。

この試行では各オロチの食べる食糧量を教授側が指定する方法を採用した。具体的には、各チームに与えるオロチの食事量集合は同じとし、各オロチと食事量の対応についてはチームごとに異なるように設定した。また、各オロチの食事開始時刻は第 1 回試行と同様である。実施済みのタイムチャート例を図 7 に示す。

問題提起に対しては、その解決に向けて 4 名が協力して取り組むことができた。各チームとも活発に議論し、問題 1 に関してはすべてのチームが短時間で SPTF に到達することができた。一方、問題 2 に関してはヒントもほとんど効果がなく、一定の時間が経過した後、正解与える結果となった。

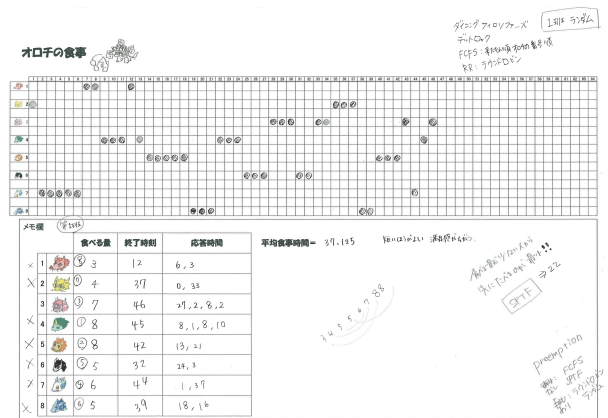


図 7: 実施済みタイムチャート (第 2 回試行)

試行に要した時間は、アンケートの記入時間も含めて約 1.5 時間であった。

5.3 第 3 回試行

この試行では、対象者を首都圏の大学で学ぶ学生を対象とした。対象者 5 名はいずれもその経歴 (特に、出身小学校、中学校の所在地) が島根県とは無関係であった。

各オロチが食べる食糧量と各オロチの食事開始時刻は第 2 回試行と同様である。

問題提起に関しては、1 人で取り組むこととなったが、問題 1 に関してはほぼ全員が短時間で SPTF を見出すことができた。しかし、問題 2 に関しては誰もラウンドロビンには至らず、教授側が正解を与える結果となった。

試行に要した時間は、アンケートの記入時間も含めて約 1 時間であった。

5.4 アンケート結果

各試行において、作業終了後アンケートによるフィードバックを得た。アンケート内容はすべての施行において同じである。アンケートは 13 個の選択型設問と 1 個の自由記述型設問からなる。選択型設問のうち 12 個は 4 択設問、1 個は 3 択設問である。

主なアンケート項目を以下に示すとともに、それらの結果を表 3 から表 6 に示す。

- スケジューリングの認知度

表 3: スケジューリングの認知度 (回答数)

選択肢	第1回	第2回	第3回
よく知っていた	0	0	0
少しは知っていた	2	0	0
詳しくは知らなかった	5	3	1
まったく知らなかった	9	13	4

表 4: スケジューリングに関する事後理解 (回答数)

選択肢	第1回	第2回	第3回
理解できた	12	9	3
まあまあ理解できた	2	7	1
あまり理解できなかった	0	0	1
理解できなかった	0	0	0

表 5: スケジューリングの表現性 (回答数)

選択肢	第1回	第2回	第3回
表現できている	13	8	2
まあまあ表現できている	3	8	2
あまり表現できていない	0	0	1
表現できていない	0	0	0

表 6: 具体的スケジューリングの理解 (荷重平均点)

名称	第1回	第2回	第3回
FCFS	3.81	3.81	3.60
SPTF	3.88	3.88	3.60
ラウンドロビン	3.81	3.75	3.40

今回の説明を受ける以前から、スケジューリングについて知っていたか。

- スケジューリングに関する事後理解
スケジューリングに関する事前知識がない場合、今回の作業を通して理解できたか。
- スケジューリングの表現性
“オロチの食事”はスケジューリングをうまく表現できているか。
- 具体的スケジューリングの理解
各スケジューリング (FCFS, SPTF, ラウンドロビン) とその特徴を理解できたか。

表 6 を除いては、各試行のアンケートにおいて得られ回答数をそのまま掲載している。一方、「具体的スケジューリングの理解」に関しては、「よく分かった」に 4 点、「まあまあ分かった」に 3 点、「あまり分からなかった」に 2 点および「まったく分からなかった」に 1 点を与え、それをそれぞれの選択肢に対する回答数によって荷重平均した得点を求めた。その結果を表 6 に示す。

6. 考察

3 回の試行に関して、試行状況に関するものとアンケート結果に関するものとに区別して考察する。また、デザイ

ンパターンに基づいた検証も行う。

6.1 試行状況に関する考察

すべての試行において、誰一人として脱落することなく、全員がスケジューリング作業を正しく実行できた。また、ランダムの実施はタイムチャート記入の練習以上にスケジューリングというものに興味を持たせるという意味でも効果があったと考える。

第 1 回試行においては、各オロチの食事をチームごとにランダムに決定したことから、スケジューリング結果をチーム間で比較することに意味がなかった。特に、SPTF を考案させる段階で考案結果のチーム間での比較ができなかった。同じ食事量集合に対しては、SPTF は同じ最短平均食事時間を与えることから、第 2 回以降では、事前に定めた食事量集合を教授側が与える方法を採用した。その結果、スケジューリング結果のチーム間での比較が意味を持つこととなり、そのためチーム間の競争意識を高める効果も得られたと判断する。

SPTF とラウンドロビンを考えさせることは、“オロチの食事”の最も重要な段階である。しかし、チームで作業をさせると議論が発散気味になり、なかなか結論にたどり着かないという状況も見られた。SPTF に関しては、3 回の試行を通して、ヒントなしでも概ね短時間で正解にたどりつけるものと考えられるが、ラウンドロビンの場合はいずれの試行においてもヒントを与える必要があった。但し、学習者の反応からは、現在与えているヒントが満足できるものとは思えなかった。適切なヒントの考案は継続検討課題である。

6.2 アンケート結果に関する考察

アンケート結果からは、ほとんどの学習者がスケジューリングに関する事前知識を持っていなかったにもかかわらず、試行の結果、ほぼ全員がそれについて理解できたことが分かる (表 3 および表 4)。また、“オロチの食事”がプロセススケジューリングをうまく表現できているかという点についても肯定的な評価を得た (表 5)。以上のことから、“オロチの食事”はプロセススケジューリングの体験的導入教育の教材として大筋で成立するものと判断する。

学習者の作業状況を観察した限りでは、ラウンドロビンの理解度に不安が感じられた。しかし、アンケート結果からは、今回扱った 3 種類の具体的スケジューリングに関して、その理解度に大差はないようにも見える (表 6)。ただし、第 2 回試行と第 3 回試行において、ラウンドロビンがやや低い点数となっている。標本数が少ないことから、この差が有意か否かは判断が難しい。ここでは、仮にこれを有意と考えることとすると、試行時の学習者の反応などから判断して、その原因はヒントの不適切さや応答時間の「バラつき」をうまく説明できなかったことにあると考え

られる。応答時間のバラつきについては、それを可視化するなどの工夫を検討したい。

6.3 デザインパターンによる検証

(1) 問題解決後の状況 (Resulting Context)

- 体験した特定の状況は理解している
アンケート結果「スケジューリングの認知度」と「スケジューリングに関する事後理解」から判断して、オロチの食事という特定の状況については理解できたと考える。
- 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
フォローアップにおいてコンピュータにおけるプロセススケジューリングとの対応について解説し、理解を得たと考えるが、その点の評価は行えていない。
- 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
“オロチの食事”で扱えていないプロセススケジューリングもある。

(2) 根拠 (Rationale)

- 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
ゲーム性がやや弱かったにもかかわらず、意欲的に作業に取り組んでいた。スケジューリングの重要性についてはしっかりと理解してくれたと判断する。
- 集中して考える体験により、体験した内容が定着する
アンケート結果「スケジューリングに関する事後理解」において高い評価を得ることができた。

7. 今後の課題

これまで、各オロチの食事開始時刻を時刻1に固定した上で、スケジューリングを行った。これは、スケジューリングを行う上で必要な待ち行列の扱いを簡単化するためである。しかし、この設定ではSPTFの変形版であるSRPT (Shortest Remaining Processing Time First, 残余処理時間順) に対応することができない。食事開始時刻を時刻1に固定した場合、SRPTはSPTFと同じスケジューリングを行うことになるからである。SRPTの導入については、それを加えた場合の所要時間も考慮に入れながら、今後の検討課題としたい。

“オロチの食事”の所要時間については、これまで試行した内容で1~1.5時間であり、想定した時間(90分ないしは50分×2回)内に収容できると考える。しかし、単純にSRPTを追加導入すると、想定時間内に収まらない可能性がある。これに対しては、各スケジューリングの実施を「部品」と考え、これらを組み合わせていろいろな実施パターンを用意するという「モジュール化」の考え方を検討中である。

8. おわりに

プロセススケジューリングを学習するための新しいCSアンプラグドActivity“オロチの食事”を提案した。“オロチの食事”では、プロセススケジューリングのアンプラグド化にヤマタノオロチが食事をする状況を活用した。高等専門学校の低学年生や大学の新生を対象に“オロチの食事”を試行し、アンケートによるフィードバックを得た。その結果、教材として成立する見込みが得られたものと判断する。

一方で、試行を通して明らかとなった改良すべき点もある。それら改良すべき点への対処、他のスケジューリング(例えばSRPT)の導入、モジュール化の考えに基づく複数の実施パターンの用意などが今後の課題である。これらの課題の解決については、その試行結果も含めて、稿を改めて報告したい。

謝辞 スケジューリングの必要性を説明するための4コマ漫画の作成に関して、イラストレータ山本ユウカ氏のご協力を得た。

本研究はJSPS 科研費 25350275 の助成を受けたものである。

参考文献

- [1] T.Bell et al.: Computer Science Unplugged. <http://csunplugged.org/> (2015.5.20 参照) .
- [2] T.Bell, I.H.Witten and M.Fellows: *Computer Science Unplugged*, <http://www.unplugged.canterbury.ac.nz> (2007). 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).
- [3] 小川仁士, 佐々木宣介, 宇野 健: 情報基礎教育におけるCSアンプラグド教材の開発-コンピュータウィルスの波及と防御を体験するゲーム制作, 情報処理学会第75回全国大会, Vol. 4, pp. 4.465-4.466 (2013).
- [4] 小川仁士, 宇野 健, 佐々木宣介: 情報基礎教育におけるCSアンプラグド教材の開発-OSのメモリ管理機能を学習するゲームの制作, 情報処理学会第76回全国大会, Vol. 4, pp. 4.369-4.370 (2014).
- [5] A.Kawakami and H.Fukuoka: A New CS Unplugged Activity for a Learning Experience in Bottlenecks, *Proc. of the 1st Symposium on Technology Sustainability(ISTS2011)*, pp. 245-248 (2012).
- [6] H.Fukuoka, A.Kawakami and Y.Wada: A Proposal of CS Unplugged Utilizing Regional Materials, *International Workshop on Informatics(IWIN)2014*, pp. 109-114 (2014).
- [7] 清水謙多郎: オペレーティングシステム, 岩波書店 (1992).
- [8] 由良弥生: 眠れなくなるほど面白い「古事記」, 三笠書房 (2013).
- [9] 出雲神話翻訳研究会: 英訳 出雲神話. <http://izumokojiki.com/izumomyths/> (2015.5.20 参照) .
- [10] T.Nishida, S.Kanemune, Y.Idosaka, M.Namiki, T.Bell and Y.Kuno: A CS Unplugged Design Pattern, *ACM SIGCSE Bulletin - SIGCSE '09*, Vol. 41, No. 1, pp. 231-235 (2009).