

## 体験的な分散アルゴリズム協調学習を支援する システムの提案

藤井 太一<sup>†1</sup> 長瀧 寛之<sup>†2</sup> 山内 由紀子<sup>†3</sup>  
大下 福仁<sup>†1</sup> 角川 裕次<sup>†1</sup> 増澤 利光<sup>†1</sup>

本研究では、分散アルゴリズムの動作の理解を目的とした、体験的協調学習システムを提案する。本システムは、複数の学習者同士が相談しながら適切な分散アルゴリズムを考える協調学習環境の支援を目的としており、小型無線端末を用いて、学習者一人一人が分散システムの各端末の動作を演じながら、試行錯誤的に分散アルゴリズムの構築と検証を行えるシミュレーション機能を提供する。分散アルゴリズムの初学段階の学習者でも容易にシミュレート可能な環境を提供することで、学習者が分散アルゴリズムの特徴や難しさを体験的に理解していくことが期待できる。本報告では、提案システムの概要、設計の工夫を、著者らが目標とする学習環境の概要とともに紹介し、また試作システムでの評価結果について議論を行う。

### A hands-on collaborative learning system for distributed algorithms

TAICHI FUJII,<sup>†1</sup> HIROYUKI NAGATAKI,<sup>†2</sup>  
YUKIKO YAMAUCHI,<sup>†3</sup> FUKUHITO OOSHITA,<sup>†1</sup>  
HIROTSUGU KAKUGAWA<sup>†1</sup> and TOSHIMITSU MASUZAWA<sup>†1</sup>

We present a hands-on collaborative learning system intended to understand behavior of distributed algorithms. This system aims to support collaborative learning where students find out effectual distributed algorithms through try-and-error and discussion between them. The system offers a distributed algorithm simulator that enables to build and verify algorithms to play a process of distributed systems using a small wireless terminal. Using the system, beginners of distributed algorithms can easily execute simulation, and understand characteristic and difficulty of distributed algorithms with trial and error approach. We show in this paper the implementation and the evaluation of the system.

### 1. はじめに

近年ネットワークの発展と普及を背景に、様々な分散システムが実用化されている。分散システムは複数の計算機（以下プロセス）が通信リンクによって結ばれたシステムであり、プロセス同士が協調しながらシステム全体でひとつの処理を行う。分散アルゴリズムは、分散システムモデル上で実行するために設計されたアルゴリズムであり、各プロセスが同じアルゴリズムを任意のタイミングで実行することで、全体として協調動作を実現する。分散アルゴリズムには、各プロセスの状態変化が任意のタイミングで起きる非同期性、プロセスがある瞬間のシステム全体の状態を把握できないという局所性、同時に複数のプロセスが動作しうる並列性という特徴がある。

分散システムの実用化に伴い、情報科学の教育においても、現在分散アルゴリズムは重要な学習項目として位置づけられている<sup>1)</sup>。しかし分散アルゴリズムは先に述べた特徴を持つことから、その動作は複雑かつ多岐にわたるため、理論的な面だけでは動作イメージを持つことが難しく、学習においてその理解に困難を伴いやすい。そのため、学習者がより効果的に分散アルゴリズムの動作原理を理解するためには、分散アルゴリズムの動作をより体験的に学ぶ仕組みが必要ではないかと考えた。

そこで本研究では、分散アルゴリズムの初学者が、様々な分散アルゴリズムの動作を試行錯誤しながら体験的に学習することを目標とした学習支援システムの実現を目標とする。提案システムでは、各学習者が分散システム上のプロセスの役割を担いグループでアルゴリズムをシミュレートする機能を提供する。また、シミュレーション実行の様子をアニメーション表示で振り返ることで、アルゴリズム全体の動作を確認しつつグループでアルゴリズムについて議論する協調学習活動の実現を支援する。

本稿の構成は以下の通りである。まず、2節で関連研究の紹介を行う。次に、3節で提案システムにおける学習法を検討し、システムの要求仕様を定める。4節では定めた要求仕様を満たす提案システムの概要、設計について説明し、5節で提案システムの内部仕様につい

<sup>†1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology, Osaka University

<sup>†2</sup> 岡山大学教育開発センター  
Center of Faculty Development, Okayama University

<sup>†3</sup> 奈良先端科学技術大学院大学情報科学研究科  
Graduate School of Information Science, Nara Institute of Science and Technology

て述べる。そして、6節では提案システムを用いた評価実験・考察を行う。最後に、7節で本研究でのまとめを行う。

## 2. 関連研究

アルゴリズムの学習に関する既存研究として、アルゴリズムの可視化、また、体験的学習や協調学習などグループ学習手法の導入といったアプローチが提案されている。

Steveら<sup>2)</sup>、Borisら<sup>3)</sup>の研究では、学習者が分散アルゴリズムの動作をイメージしやすくするために、シミュレータを用いてアルゴリズムの動作を可視化する手法が提案されている。シミュレータを用いる利点は、学習者がネットワークトポロジやアルゴリズムの動作条件などを試行錯誤しながら変更し、様々な動作例を実際に観察することで、どのように分散アルゴリズムの処理が行われているのかを理解できることにある。

また、学習者がカードなどを用いてアルゴリズムを体験的に学習した後に、体験したアルゴリズムを実際にプログラミングするものが提案されている<sup>4)</sup>。この研究の対象者はプログラミングの初学者であるが、アルゴリズムを体験的に学習することにより、自力でプログラムを作成することが可能になったと報告されている。この研究から、アルゴリズムを体験的に学習することが学習者の理解の助けになると言える。

他に、アルゴリズムを複数の学習者で協調して学習する研究として、Teresaら<sup>5)</sup>が挙げられる。Teresaら<sup>5)</sup>では、学習者がアルゴリズムの動作を説明し、複数の学習者と議論しながら学習する。Teresaら<sup>5)</sup>で、協調学習によるアルゴリズム学習は教育的手法として有効であると述べられている。

## 3. システムの提案

本節では、学習法について検討した上で、提案システムの要求仕様を定める。

### 3.1 学習法の検討

各学習者がプロセスの役割を担いグループでシミュレーションを行い、シミュレーション後、グループ内外でシミュレーション中に気付いた事項について議論するという形式の学習法を提案する。以下で提案システムにおける学習法の詳細について説明する。

まず、複数の学習者でグループを作り、与えられた問題を解く分散アルゴリズムをグループで考案した上で、考案したアルゴリズムのシミュレーションを行う。学習者はそれぞれ一つのプロセスの役割を担い、お互いにメッセージのやりとりを行って現在の自身や周囲の状況を確認しつつ、その状態でプロセスが取り得る適切な処理を、考案したアルゴリズムに

従って実行する。シミュレーション後には、シミュレーション実行の様子を GUI 上に再現し、各学習者がどのようにプロセスの状態を変化させたのかを振り返りながらアルゴリズム全体の動作を確認することで、グループで考案したアルゴリズムの正しさ、効率の評価を行う。グループ学習を行うことで、各学習者がシミュレーション中に気付いた事項を共有したり、得た知識の整理を行うことができるなどの効果も期待できる。

ここで、グループの人数が多すぎると議論の收拾がつきにくく、またシミュレーションでの学習者同士のメッセージ交換の数も単純増加するため、作業に時間がかかりがちになることが予想される。一方分散システムとしての動きを確認するには、ある程度の人数でシミュレーションを行う方が望ましい。そのため、アルゴリズムの内容や学習活動に費やせる時間にも依るが、1グループの人数は5人程度とする。

### 3.2 要求仕様

3.1節の検討結果をふまえ、以下の要求仕様を定める。

- プロセスと対応するデバイスを用いて、学習者の操作を検出できること
- 分散アルゴリズムの代表的な手法 (Lelann のアルゴリズムなど) が正しく動作すること
- シミュレーション実行後、行った操作とそれに伴うプロセスの状態変化をアニメーションとして再現することで、学習者がアルゴリズム全体の動作を確認できること

### 3.3 システムで扱う問題

提案システムでは、分散システムの基本的な問題としてリーダー選挙問題を題材に取り上げ、分散アルゴリズムの基礎概念について学習を行う。リーダー選挙問題とは、分散システムモデル上のプロセスの中から分散されたタスクの取りまとめ役として、1つのプロセス (リーダー) を選出する問題である。リーダー選挙問題に対して、これまでに多くの分散アルゴリズムが提案されている。例として、LeLann のアルゴリズム<sup>6)</sup> は、単方向リングネットワーク上で、各プロセスが固有の ID を持つことを仮定している。各プロセスの ID をネットワーク上で巡回させ、自プロセスの ID を受信したプロセスをリーダーと決定するアルゴリズムである。

学習者はグループでアルゴリズムを考える際、リーダーを正しく決定するためには ID をどのように利用すればよいか、どのような状態になればリーダーを決定できるのか、できるだけ効率がよい手法で実現できるように考える必要がある。以下に、システムでリーダー選挙問題のモデルについて説明する。ネットワークトポロジとして単方向リングネットワークを用い、リング上に配置された各プロセスを区別するために、各プロセスは固有の ID を持つと仮定する。リーダーを選出するために、各プロセスは隣接プロセスにメッセージを送

信する．ここで，送信するメッセージはプロセス ID とする．これは，メッセージをプロセス ID に限定することで操作を単純にし，学習者がアルゴリズムを考える際にメッセージの送り方に意識を集中させるため，また，代表的なリーダー選挙アルゴリズムはプロセス ID の送受信を基にしているものがほとんどだからである．メッセージを受信したプロセスは，メッセージを破棄するか，隣接プロセスに転送するかの処理を行う．この処理を繰り返し行うことにより，最終的に 1 つのプロセスをリーダーとして選出する．

#### 4. 提案システムの設計

本節では，提案システムの概要と各機能についての説明を行う．

##### 4.1 提案システムの概要

図 1 に提案システムの概略図を示す．提案システムでは，学習者がシミュレーションを行うためのシミュレーション機能と，シミュレーション実行の様子を GUI 上に表示する GUI 再生機能を提供する．シミュレーション機能では，学習者によるプロセスの動作を検出するため，分散システム上のプロセスと 1 対 1 に対応させた入出力デバイスを用いる．学習者に 1 人 1 台入力デバイスを渡し，これを用いてシミュレーションを行う．つまり，学習者がデバイスを用いてメッセージの送信や破棄といった操作を行えるようにする．また，このとき学習者はデバイスを介してのみ情報の取得・発信をできるものとする．これは，分散アルゴリズムの特徴である，各プロセスはメッセージによって通信することによってのみ，他プロセスの状態を知ることができるということを反映している．GUI 再生機能では，シミュレーション実行後に学習者がアルゴリズム全体の動作を確認できるように，GUI 上にシミュレーション実行の様子をアニメーション表示する機能を提供する．

##### 4.2 学習者用入出力デバイス

シミュレーションを行う際に学習者が用いる入出力デバイスは，メッセージの送受信といった情報の取得・発信ができること，取得した情報を表示できることが必要である．そこで本研究では，デバイスとして SunSPOT<sup>7)</sup> を用いることにした．SunSPOT の外観を図 2 に，仕様を表 1 に示す．SunSPOT は，本体の加速度を検出できるセンサデバイスで，学習者によるプロセスの動作を検出するために用いる．また，付属のセンサボードには RGB 24bit LED8 個とプッシュスイッチ 2 個がついており，学習者にメッセージの送受信を通知するために用いる．SunSPOT はフラッシュメモリに学習者が操作するためのプログラムをダウンロードすることで使用する．

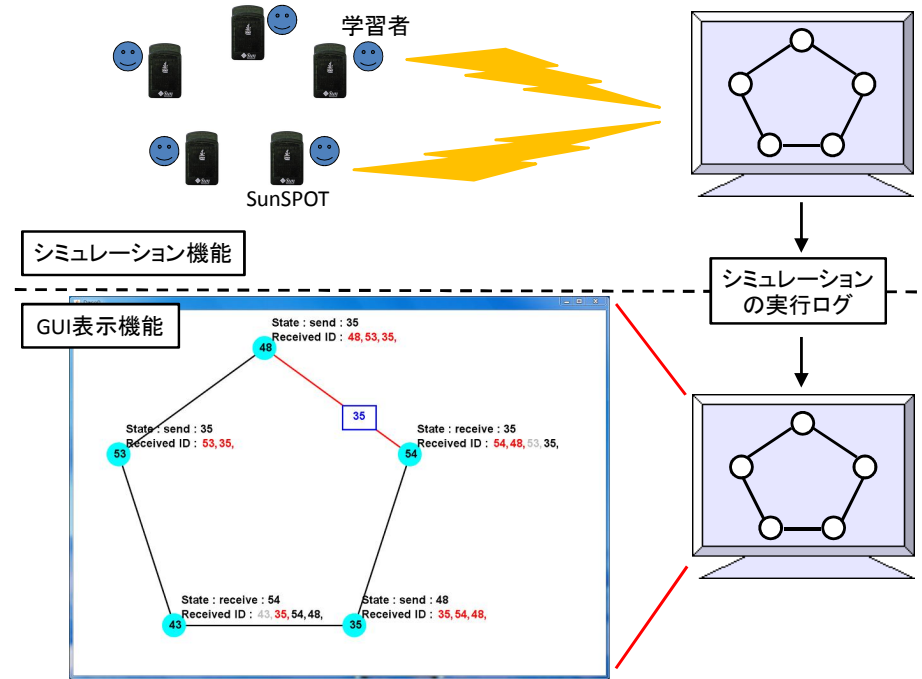


図 1 システム概略図

##### 4.3 提案システムの機能

本節では，SunSPOT 側，PC 側にシミュレーションと GUI への再生をするために必要な機能の説明を行う．

###### 4.3.1 SunSPOT 側の機能

学習者は SunSPOT からの情報だけを用いてシミュレーションを行える必要がある．そのために必要な機能は大きく分けて動作検出と情報の表示の 2 種類が挙げられる．以下で 2 つの機能について説明する．

動作検出については，学習者がシミュレーションを行う際，できるだけプロセスの振る舞いと直感的に一致するように SunSPOT の操作方法を定義する．具体的には，メッセージを隣接プロセスに転送する操作として SunSPOT を横 (X 軸方向) に振る，メッセージを破棄する操作として SunSPOT を前 (Y 軸方向) に振る，という 2 種類の操作を割り当てる．



図 2 SunSPOT の外観

CPU	180MHz 32bit ARM920T
RAM	512KB
フラッシュメモリ	4MB
利用無線周波数帯	2.4GHz IEEE 802.15.4

情報の表示については、SunSPOT の LED を用いて実現する。LED で表示する情報は、メッセージ送信を学習者に通知、メッセージ受信を学習者に通知、SunSPOT が持つプロセス ID、の 3 点とする。以下それぞれについて説明する。

#### メッセージ送信の通知

学習者がメッセージを隣接プロセスに転送する操作を行ったとき、学習者にメッセージ送信が行われたことを、LED を左から順に黄色に光らせる処理を 2 回繰り返すことで通知する。

#### メッセージ受信の通知

隣接プロセスからメッセージを受信したとき、学習者にメッセージ受信が行われたことを、LED 全てを黄色に点滅させる処理を 3 回繰り返すことで通知する。

#### プロセス ID の表示

LED には、シミュレーション開始時に自プロセスの ID を、メッセージ受信の通知後に受信したプロセス ID を表示する。また、SunSPOT の左ボタンを押している間は自プロセスの ID を表示し、ボタンを離すとボタンを押す前に表示されていたプロセス ID の表示に戻る。なお、各プロセスに割り当てられる ID は、シミュレーション実行時にランダムに決定されるものとする。

LED にプロセス ID を表示する形式は、50、10、5、1 を基準とした色分けとした。これは、8 個の LED だけで ID の数字を表示しなければいけないが、2 進数表示だと読みにくいいため、硬貨を数えるような形式の表現にした方がなじみ深いのではないかと考えたからである。表示するパターンとして、赤、緑、青、白の 4 色を使用した通知方法を利用する。具体的には、赤：50、緑：10、青：5、白：1、と LED に表示する色に数字を割り当てることで、プロセス ID を表示する。表示する際には、LED の色を割り当てた数字の大きい順に左から表示していく。この方法により、0 から 97(89 除く) までのプロセ

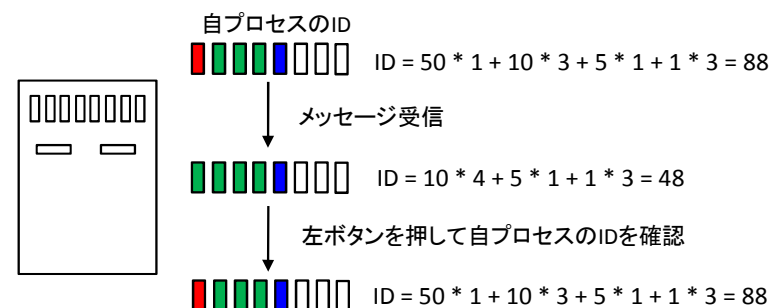


図 3 ID の読み方と操作例

ス ID を表示することが可能である。3.1 節で議論した通り本システムは 5 人程度でのシミュレーションを想定しているため、この数字の範囲でプロセス ID をランダムに割り振れば、学習者が自分のプロセス ID だけで自分をリーダーかどうか容易に推測できないためには十分であると考えられる。図 3 に ID の読み方と操作例を示す。

#### 4.3.2 PC 側の機能

シミュレーション実行の様子を表示するための GUI は、シミュレーションにおける SunSPOT の操作履歴の再生を表示するウィンドウと、操作履歴の再生・一時停止の制御を行うウィンドウの 2 種類から成る。

図 4 に履歴再生機能の GUI を示す。再生ボタンを押したときに、SunSPOT の操作履歴をシミュレーション実行の様子としてアニメーション表示する。このとき表示されるそれぞれの円はプロセスを、円を結ぶ直線はリンクを表す。円の中の数字は対応するプロセスの ID である。また、各プロセスの右上に、最後にそのプロセスが行った動作 (Behavior) と、今までに受け取ったプロセス ID のメッセージ (Received ID) を表示する。メッセージを受信したときのプロセスの動作には、send、receive、discard の 3 種類がある。send はプロセスがメッセージを隣接プロセスに送信、receive はプロセスが隣接プロセスからメッセージを受信、discard はプロセスがメッセージを破棄したことを表す。また、メッセージ送信動作については、自プロセスと隣接プロセスとの間のリンク上を、四角形のアイコンで示したメッセージが流れるアニメーションで表示する。メッセージが隣接プロセスに到着すると、Received ID の末尾に受信したプロセス ID が加えられる。Received ID は、メッセージの処理方法によって、黒 (未処理)、赤 (隣接プロセスへ送信)、灰 (破棄) で色分けして表示する。

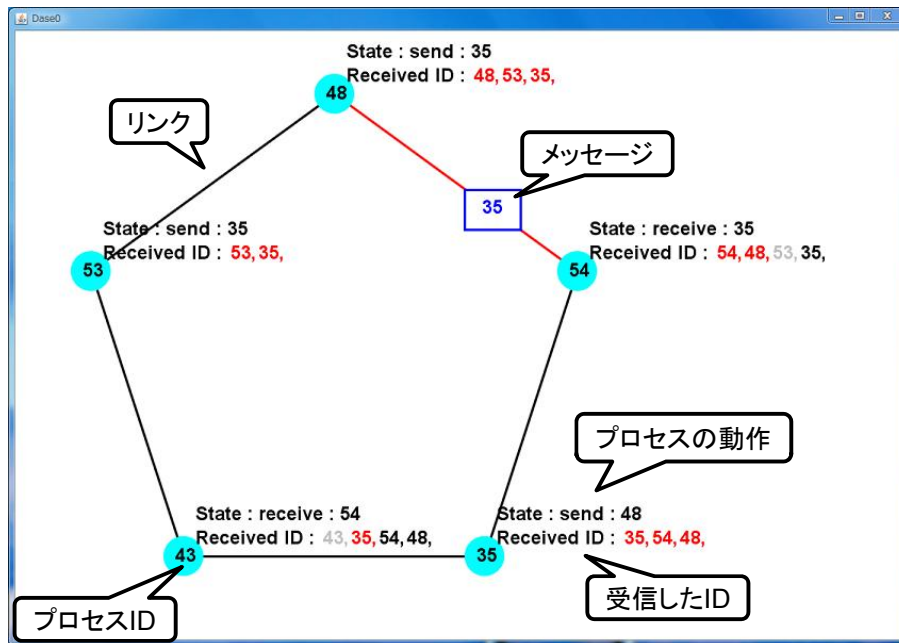


図 4 履歴再生機能

## 5. 提案システムの内部仕様

本節では、提案システムの内部処理の流れを示す。図 5 に示す通り、提案システムでは、

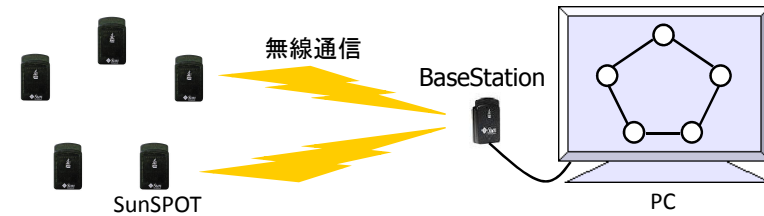


図 5 システムの機器間の関係

入出力デバイスである SunSPOT と、PC 上で動作するプログラムであるアルゴリズム制御部が通信を介してプログラムが実行される。また、SunSPOT は PC 本体と直接通信することができないので、SunSPOT と PC 間の通信を中継するための機器 BaseStation を使用する。SunSPOT の開発ツール一式は Java で提供されているため、SunSPOT 制御とアルゴリズム制御部の実装はいずれも Java 言語で行った。ソースコードの行数は 3165 行である。

SunSPOT では、電源を入れたときの初期化処理、学習者が行った操作の検出、アルゴリズム制御部へのメッセージ送信、アルゴリズム制御部から受信したメッセージの処理を行う。本システムではアルゴリズムを実行する主体は学習者自身であるため、SunSPOT 自体はアルゴリズム実行に関する処理を行わず、プロセスの動作に関する操作の授受と、受信メッセージの表示などプロセス固有の情報表示の処理だけを行う。また、仕様上は SunSPOT 同士で通信を行うことも可能だが、実環境の影響を受けないシミュレーションが可能のように、端末間の通信は全て PC 側のアルゴリズム制御部を介して行う。

アルゴリズム制御部では、シミュレーション実行時には SunSPOT からのメッセージ処理を、履歴再生時にはシミュレーション実行の様子アニメーション表示を行う。SunSPOT から動作検出のメッセージを受け取ったとき、アルゴリズム制御部はログリストへの操作履歴の記録を行う。ログリストには SunSPOT のアドレス、動作の種類、受信時刻の 3 つの情報を格納する。これは、履歴再生機能にて、実時間通りのシミュレーションを GUI 上で再現するための情報として利用する。操作履歴の記録後、動作の種類によって、隣接プロセスへのメッセージ転送、メッセージの破棄に関する処理を行う。具体的には、メッセージ転送操作の場合は、転送先プロセスに対応する SunSPOT デバイスに、メッセージにあたるプロセス ID の数字を送信する。この際、メッセージ転送先のプロセスが、前に受信したメッセージに対する処理を実行していない(そのプロセスに対応する学習者がまだ操作を実

行していない)場合は、対応 SunSPOT デバイスへのプロセス ID の転送を保留し、当該プロセスに対応する制御部内部のキューに蓄積する。これは、現在の実装上 SunSPOT 上で同時に 1 つのプロセス ID しか表示することができないため、メッセージを転送すると、SunSPOT が処理中のプロセス ID に転送したプロセス ID を上書きしてしまうためである。前のメッセージに対する操作の実行を確認し次第、キューから次に転送すべきメッセージとなるプロセス ID を取り出して、対応する SunSPOT デバイスへ転送する。

## 6. 評価実験

本節では、提案システムの評価実験について説明する。「SunSPOT の操作体系について」と「提案システムを用いたシミュレーションの有効性」の 2 点を評価するため、大阪大学大学院情報科学研究科と大阪大学基礎工学部情報科学科の学生 5 名(修士 2 年:3 名, 修士 1 年:1 名, 学部 4 年:1 名)を対象にシステムを使った疑似演習を行い、アンケートを取った。被験者は分散アルゴリズムに関する研究を行っているため、分散アルゴリズムに対する知識がある程度ある。

### 6.1 実験概要

実験の前に、システムの使用法の説明、リーダー選挙問題のモデルについて説明した。その後、学習者 5 人で 1 グループを作り 3.1 節の学習法に従って実験を行った。具体的な実験手順は以下の通りである。まず、リーダー選挙問題を解くアルゴリズムをグループで考案(5 分)し、考案したアルゴリズムを基にシミュレーションを実行する(3 分)。シミュレーションを実行する際には、学習者間で口頭でプロセス ID を教えあうことは禁止し、プロセス ID のやり取りは SunSPOT のみを用いて行うようにする。また、受信したプロセス ID など必要な事項をメモすることは許可する。シミュレーション実行後、操作履歴を再生してアルゴリズムの確認を行う(3 分)。このとき、意図した結果が得られなかった場合はアルゴリズムを再考案し、再シミュレーションと確認を行う。そして、実験後にアンケートを実施する。

### 6.2 実験結果

シミュレーション前にグループで考え出したアルゴリズムと、シミュレーションの結果は以下の通りである。

(1) 全員が全てのメッセージを隣接プロセスへ転送する。各学習者は受信メッセージを記憶していき、プロセス数だけメッセージを受信したとき、受信したプロセス ID のうち一番小さいプロセス ID をリーダーとする。終了条件は、プロセス数だけメッセージを受信

したときとする。

結果：最小 ID のプロセスがリーダーにならず、失敗

原因：ID の読み間違いによるメッセージ転送、破棄の誤操作

(2) 自プロセスの ID 以下のものだけを隣接プロセスへ転送し、自プロセスの ID より大きなものは破棄する。終了条件は、1 度受信したメッセージを 2 度目に受信し、そのメッセージを隣接プロセスへ転送した後とする。

結果：アルゴリズムが終了しない被験者が出たため、失敗

原因：グループ内でアルゴリズムの共通認識(終了条件)ができていなかった

(3) 今まで受信したプロセス ID の内、最小のプロセス ID を記憶し、最小 ID より大きなものは破棄する。終了条件は、1 度受信したメッセージを 2 度目に受信し、そのメッセージを隣接プロセスへ転送した後とする。

結果：成功

### 6.3 アンケート結果と考察

本節では、「SunSPOT の操作体系について」と「提案システムを用いたシミュレーションの有効性」の 2 点について、被験者を対象に行ったアンケート結果から考察を行う。

#### 6.3.1 SunSPOT の操作体系

SunSPOT の操作体系を評価する項目として、LED に表示された ID の読みやすさ、操作のしやすさについてアンケートを被験者に対して行った。

「LED の表示規則はすぐに理解できたか」という質問では、全員表示規則は理解できているが、5 人中 4 人は理解するまでに少し時間がかかったと回答した。また、4.3.1 節で述べた表示規則ではなく、ID は 2 進数で表示した方が分かりやすいのではないかという回答もあった。これは、被験者が 2 進数での記述方法に慣れているからではないかと考えられる。反面、「シミュレーション中に LED に表示される ID を容易に読み取れたか」という質問では、5 人中 4 人が慣れたら読み取れると回答し、1 人は表示規則を示した表を確認しながら ID を読み取る必要があったと回答し、ID を読み取るために慣れが必要であるという結果となった。

「SunSPOT の操作方法が直観的でないと感じたところはあったか」という質問では、5 人中 3 人が直観的でないと感じているという結果となった。実際、「SunSPOT の操作体系のせいでうまくシミュレーションできなかった場面はあったか」という質問で、SunSPOT を操作するときに転送と破棄を間違えて操作してしまったという回答が出ている。このようなアルゴリズムの間違いと無関係な間違いは起こらないように操作体系の改善をする必要

がある。例えば、メッセージの転送は SunSPOT を横に振る動作を用い、メッセージの破棄は SunSPOT のボタンを押すことで行うようにする、という意見が得られた。

### 6.3.2 提案システムを用いたシミュレーションの有効性

提案システムを用いたシミュレーションの有効性を評価する項目として、シミュレーションに関するアンケートを被験者に対して行った。

6.2 節の 2 回目のシミュレーション結果が失敗に終わった原因について自由記述で回答を求めたところ、「グループ内でアルゴリズムの終了条件が共通認識できていなかった」という原因が挙げられた。具体的には、「リーダーが分かった時点でアルゴリズムが終了した気分になってしまう」という意見が得られた。また、実験中に感じたこととして、「メッセージが来ないと不安になる」という意見が得られた。さらに、「実験を通して、分散アルゴリズムの基本的な考え方の理解に役立つような場面はあったか」という質問に対し、「各プロセスは自身の局所情報しか見ることができないという点は理解できそう」という意見が得られた。また、「シミュレーション実行の様子を見て何か気付いたことはあるか」という質問に対して、「どこでアルゴリズムが失敗したのか犯人が分かった気持ち」、「自分のミスが全体に波及していくのが分かった」、さらに、「人によって処理時間が違うのが非同期性を表している」という意見が得られた。以上より、被験者はシミュレーション後の履歴再生でネットワーク全体を見てアルゴリズムの動作を確認していることが分かる。

## 7. おわりに

本研究では、分散アルゴリズムの初学者が、リーダー選挙問題を題材として、体験的に分散アルゴリズムの基礎概念を学習することを目的とした協調学習支援システムを提案した。また、提案システムの機能として、アルゴリズム制御部と無線通信を介して動作する SunSPOT を用いてシミュレーションをし、シミュレーション実行の様子を GUI にアニメーション表示する機能を紹介した。これらは学習者がグループでアルゴリズムを考案し、提案システムを用いてシミュレーションをし、アルゴリズムの動作の確認、正しさや効率についての議論を行うという活動を実現するものである。

提案システムを用いて実験を行った結果、SunSPOT の操作性は改善を要するものの、提案システムを用いたシミュレーションによって、分散アルゴリズムの局所性、非同期性といった特徴を体験的に学習できるという意見が被験者より得られた。

今後の課題として、SunSPOT の操作性の改善、また、ID 破棄のアニメーションを加えるなど、GUI 上でのアルゴリズムの動作確認方法の改善が挙げられる。更に、他の分散シ

ステムモデルやリーダー選挙問題以外の問題への対応も今後の検討事項である。

## 参 考 文 献

- 1) 情報処理学会情報処理教育委員会：情報専門学科におけるカリキュラム標準 J07，<http://www.ipsj.or.jp/12kyoiku/J07/J0720090407.html> (2007).
- 2) Carr, S., Fang, C., Jozwowski, T., Mayo, J. and Shene, C.-K.: ConcurrentMentor: A Visualization System for Distributed Programming Education, *In Proceeding of the 2003 International Conference on Parallel and Distributed Processing Techniques and Applications*, pp.1676–1682 (2003).
- 3) Koldehofe, B., Papatriantafilou, M. and Tsigas, P.: LYDIAN: An Extensible Educational Animation Environment for Distributed Algorithms, *ACM Journal on Educational Resource in Computing*, Vol.6, No.1, pp.1–21 (2006).
- 4) 杉浦学, 松澤芳昭, 岡田健, 大岩元: アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズムの構築体験とその効果, *情報処理学会論文誌*, Vol.49, No.10, pp.3409–3427 (2008).
- 5) Hübscher-Younger, T. and Narayanan, N.H.: Constructive and collaborative learning of algorithms, *Proceeding of the 34th SIGCSE Technical Symposium on Computer Science Education*, pp.19–23 (2003).
- 6) Tel, G.: *Introduction to Distributed Algorithms*, Cambridge University Press, second edition (2000).
- 7) SunMicrosystems, I.: SunSPOTWorld, <http://www.sunspotworld.com/> (2010).