

ソフトウェア開発グループ演習のための チーム編成の最適化支援

橋浦 弘明¹⁾・桑原 徹²⁾・秋 玉梅²⁾
石川 達也¹⁾・山下公太郎¹⁾・古宮 誠一¹⁾

近年、ソフトウェア開発は大規模化、複雑化している。このような状況を反映し、企業ではソフトウェア開発の知識やスキルを持つ人材の育成を大学に求めている。ソフトウェア開発は集団で行われることが一般的である。芝浦工業大学でもソフトウェア開発のための知識やスキルの修得を目指したグループ演習を実施している。グループ演習ではプロジェクトチームを構成するグループのメンバー構成の良し悪しが学習効果に直接影響する。そこで、著者らはグループ演習に最適なチーム編成を自動的に行うシステムを開発した。本稿では、最適なチーム編成を行うためのモデルを提案し、システムの適用を行った結果について述べている。

キーワード

最適なチーム編成, 遺伝的アルゴリズム, CSCL, ソフトウェア開発演習, Webアプリケーションシステム

1. はじめに

近年、社会のソフトウェアに要求する機能の高度化により、ソフトウェア開発は大規模化、複雑化している。そのため企業では、プロジェクトを成功させるためにソフトウェア開発に関する知識や経験のある人材を求めている。このような要求にこたえるため、各大学ではグループ演習を通して、ソフトウェア開発に対する知識や理解を深めさせることにより、高度な人材を育成することを目指している。

芝浦工業大学（以下、同大学と記す）の情報工学科では、3年生を対象とする「高度情報演習2B」において、ソフトウェア開発のためのグループ演習授業を実施しており、ソフトウェア開発における一連のプロセスなどを体験させている。

2004年度の演習期間中に毎週提出させていた進捗報告書の内容や演習の最終成果物、演習支援システムに記録されているログ情報から「役割分担ができていない」「優れている人に多くの負担がかかる」「チームによって作業進捗の差が大きい」などの問題が発生していることが判明した。

著者らはこれらの問題が発生している原因を特定するため、演習終了後に演習に対するアンケート調査と学習

者の一部に対して演習の改善点に関する聞き取り調査を実施した。寄せられた意見から、課題の内容、演習支援システムに対する要望など現状の改善で対応できるものを除いたところ、所属するチームへの不満が多く見られ、チームメンバーの良し悪しに、学習者の理解度や満足度が左右されているとの認識するに至った。

著者らはチーム編成を最適化することにより学習者の理解度や満足度を向上させることが可能であると考え、ソフトウェア開発グループ演習に最適なプロジェクトチーム編成を行う方法を提案し、遺伝的アルゴリズム（Genetic Algorithm）を使って最適なチーム編成を自動的に行うシステム（Environment for Ultimate software Development Exercise/Group Organizer：以下EtUDE/GOと記述する）を開発した。

本稿では、遺伝的アルゴリズムを利用して最適なプロジェクトチーム編成を行うために用いたモデルを明らかにすると共に、生成された解の候補が最適であるかどうかを評価する方法を明らかにする。

最後に、本稿の構成を以下に示す。2では本研究の関連研究を示し、本研究の位置づけを述べる。3ではソフトウェア開発のためのグループ演習授業に最適なチーム編成に求められる条件を明確にし、チーム編成が最適であるかどうか評価する方法を述べる。4ではチーム編成問題の特徴と遺伝的アルゴリズムの導入の理由を挙げ、システムの実現方法を述べる。5では実装されたシステムの評価を行い、その結果を考察する。6では本稿のまとめを述べる。

¹⁾ 芝浦工業大学大学院

²⁾ 芝浦工業大学

2. 関連研究

ソフトウェア開発グループ演習は、通常の講義などと異なり学習者と教授者の頻繁なコミュニケーションが必要となるため、教授者への負担が大きい。このため、これまでも演習を支援するシステムが数多く提案されている (Chiken & Hazeyama, 2003; 松浦・相場, 2003; Jang & Steinfield & Pfaff, 2002)。

これまでの演習支援システムの多くは、演習期間中のコミュニケーションの支援や、提出物の管理、提出物の評価支援などが中心となっている。本演習においても著者らはソフトウェア開発のためのグループ演習支援システム EtUDE の開発、および運用を行うことにより、これらの支援を行っている (磯崎・山下・石川ほか, 2005; 石川・橋浦・山下ほか, 2006)。

しかしながら、ソフトウェア開発のためのグループ演習支援システムで、グループ演習のためのチーム編成を支援するという機能を持っているシステムは少ない。

岩崎・宮地・尾上 (2002) は、小学校での班学習の基礎となる学習グループの編成を行うシステムを開発している。岩崎ほか (2002) は遺伝的アルゴリズムによる解と総当り法による解を比較し、遺伝的アルゴリズムによる解が十分上界値に近く、実用において十分な有効性を持っていることを確認し、グループ編成に遺伝的アルゴリズムを用いることで処理に要する時間が大幅に短縮できる利点を述べている。

また、樫山 (2001), Hazeyama & Sawabe & Komiya (2002) は、ソフトウェア開発のような知識集約性の高い作業においては個人の能力差が大きくなりやすいため、グループ編成の良し悪しが、その教育効果に直接影響することを指摘し、学生一人ひとりの個性を考慮した最適なグループ編成を行う必要性を述べている。また樫山 (2001), Hazeyama & Sawabe & Komiya (2002) は、スキルに関する属性情報についてはシステム分析能力、システム開発への関心度、将来の進路、リーダーシップ能力、コミュニケーション能力をアンケートによって収集し、それらを学習者が有するスキルに関する属性情報とみなし、これらの属性情報を参照しながら、チーム間の能力差が最小となるような戦略に基づいてプロジェクトチーム編成を行う実験を行っている。

これらの研究では、課題を遂行するために必要となる役割の概念を持っていないため、チーム内での個人の役割分担については考慮されていない。また、個人の属性は間隔尺度を用いたアンケートで得られた評価値 (主観的な値) を使用しているため、編成結果が学習者の自己評価 (主観) に左右される。

本研究ではチーム編成時に、役割の概念を導入し、役割に対する個人の適性を学習者の成績によって定量的に

評価することによりこのような問題点を解決している。

3. チーム編成の方法

3.1 最適なチーム編成

ソフトウェア開発グループ演習では、課題となるひとつのソフトウェアを完成させるために、複数人で構成されるプロジェクトチームが作業を行う。プロジェクトチームに参加している学習者には各自に役割が設定されており、学習者はプロジェクトチームの中で役割に応じた作業を遂行することにより、演習の最終課題であるひとつのソフトウェアを完成させていく。このような形式の演習は全員が同一の課題を回答していく初歩的なプログラミング演習などと比較すると、課題に対する学習者の自由度が高い。

このため、学習者は主体的に課題に取り組む必要が生じるが、学習者が一定以上の能力を持っていないと、自分の役割を遂行することは難しいという局面が出てくる。演習において学習者がこのような局面に直面した場合、学習者は自分が遂行できない作業についてチーム内の他のメンバーの助力を得ることによって解決を図ろうとする。しかしながら、チーム内に作業を担当できる能力を有する学習者が存在しないことがある。このような場合には学習者は教授者や担当ティーチングアシスタントの助力を得ることになるが、作業そのものは演習課題となっているため、教授者やティーチングアシスタントは直接的にその作業を代行することはできない。このため、問題が発生したときにチーム内で問題解決できるチームと、教授者やティーチングアシスタントに助力を求めなければ問題が解決できないチームでは、後者の方がコミュニケーションにおけるオーバーヘッドが大きく、チーム間における作業進捗の差が大きくなる原因となっている。

このような事象を回避するために、チームには演習に必要なスキルを持った学習者をあらかじめ配置し、なるべくチーム内で問題解決を行うことができるようにプロジェクトチームを編成することが重要である。同時に、チーム編成に対する学習者の納得感を高めることを考慮すると、チーム間の能力のバラツキができるだけ小さくなることも必要である。

このようなことから、我々はプロジェクト演習における最適なチーム編成は以下の条件を満たしていなければならないと定義した。

- (a) 各チームに役割を遂行可能な適性を持った学習者を少なくとも一人以上割り当てる
- (b) チーム間の能力差をできるだけ小さくする
- (c) チーム間の人数差については最大1とする

これらの条件に従って最適なチーム編成を行うため

の、個人の能力の評価方法及び、チームの能力の評価方法を述べる。

3.2 代用特性

個人の能力の評価について述べる前に、本研究で用いている概念である代用特性について述べる。

本演習においては前述のとおり、プロジェクトチームの各メンバーはそれぞれ役割を持っている。学習者の割り当ては、演習の役割を遂行するのに必要な能力（適性）を学習者が満たしているか否かで評価される。同じ学習者でも、担当する役割によってその適性は異なる。具体的には、学習者Aがプロジェクトリーダーを担当する場合の適性と、学習者Aがシステム設計を担当する場合の適性は異なるということである。

個人の適性について評価するためには、適性に直結している属性（特性）を探す必要がある。しかしながら、このような特性が計測可能な形で見つかることは通常希である。このため、本研究では適性に直結する属性に代わり、直接測定可能な、したがって直接制御可能な属性で代用できると仮定している。このような属性を代用特性と呼ぶ。

役割、適性、代用特性の関係についての具体例を表1に示す。

表1 役割－適性－代用特性の関係（例）

役割 (Y)	適性	代用特性の例 (X)
リーダー	PM（プロジェクトマネジメント）能力	プロジェクトマネジメントに関する問題の成績
設計担当	分析・設計能力	ソフトウェア設計に関する問題の成績
コーディング担当	プログラミング能力	プログラミングに関する問題の成績
品質保証担当	品質管理能力	ソフトウェアテスト技術に関する問題の成績

ここで、リーダーにはプロジェクトマネジメント能力、設計担当には分析・設計能力、コーディングにはプログラミング能力、品質保証担当には品質管理能力を求めていることを表している。本研究では代用特性を用いてこれらの役割遂行能力を表現する。

3.3 個人能力の評価

我々は、個人がある役割を遂行する能力（適性）を有するか否かの評価を前述の代用特性を用いて間接的に評価する。

ある役割を遂行するために求められる能力と、その能力を有するか否かを表す代用特性との関係は、必ずしも1:1ではない。ある役割*i*を遂行する学生（学籍番号*l*）の能力は、ある科目*j* ($j=1 \cdots p$) の成績との関係で表せる。従って式①のように表せる。

今回の実験では目的変数と代用特性との関係や、各役割にどの程度の重みを付けるのが適当であるかについては事前に評価を行うためのデータが存在しなかったため、これらのモデルを単純化して適用を行った。

具体的には、学習者の能力値については係数 α_{ij} を1、定数 c_i を0と仮定し、代用特性の総和が学習者の能力であるものと仮定することにより、今回の実験で用いた式②が導かれる。

$$y_{il} = \sum_{j=1}^p \alpha_{ij} x_{jl} + c_i \quad \cdots \cdots \text{①}$$

i : 役割の番号 j : 科目の番号

y_{il} : 目的変数 α_{ij} : 係数

x_{jl} : 学生の成績 c_i : 定数

l : 学籍番号 p : 科目数

$$y_{il} = \sum_{j=1}^p x_{jl} \quad \cdots \cdots \text{②}$$

3.4 役割遂行可否の評価

学習者に役割を割り当てる際には、3.1で述べた「(a) 各チームに役割を遂行可能な適性を持った学習者を少なくとも一人以上割り当てる」という制約を満たす必要がある。

学習者がその役割を遂行できるかどうかについての判断は、役割を遂行するために必要な能力が一定の水準を上回っているかどうかで判断を行う。

本来、役割遂行可否の評価については、3.3で述べた式①が一定水準以上であるかどうかによって判断可能である。しかし、今回の実験ではモデルを単純化して適用を行ったため、役割の重みに差をつけず全て均一のものとして評価している。このため、今回の実験で役割遂行可否を評価する際には、割り当てられた役割ごとに、対応する代用特性を一つずつ用いて遂行可否の判断を行っている。

これを定式化すると、式①にChronicler's delta（式③）を適用し、さらに定数を0と仮定することにより、今回の実験で役割遂行可否のために用いた式④が導かれる。

$$\alpha_{ij} = \delta_{ij} \begin{cases} = 1 (i=j \text{ のとき}) \\ = 0 (i \neq j \text{ のとき}) \end{cases} \quad \cdots \cdots \text{③}$$

$$y_{il} = x_{jl} \quad \cdots \cdots \text{④}$$

3.5 チーム能力の評価

3.3で述べたとおり個人能力が算出できたので、その能力を用いてチームの能力を算出する。

チームの能力は基本的にはチームに所属するメンバーの能力の和で表すことができる。しかしながら、「リーダーさえ優秀であれば、そのチームは成功しやすい」という場合も考えられる。このような事象を表現するために、チームにおける役割の重要度を導入した。これによ

りチームkの能力はチームkに所属する個人の能力と役割の重要度を乗じたものの総和を取り、個人の能力の計算に使用したすべての代用特性の数で除算したものをチームの能力とする。

定式化すると以下の通りである。

$$O_k = \frac{1}{p \cdot n_k} \sum_{i=1}^{n_k} w_i Y_{ik} = \frac{1}{p \cdot n_k} \sum_{i=1}^{n_k} w_i \sum_{j=1}^p (\alpha_{ij} x_{jik} + C_i)$$

i : 役割の番号 k : グループの番号

O_k : グループkの能力 (平均値)

l_k : グループkに属する学生のID

n_k : グループkに属する学生数

w_i : グループkにおける役割iの重要度

r : 役割の数 p : 科目数

今回の実験では、各役割の重要度は均一であると仮定したので、すべての役割の重要度を1として評価を行った。

3.6 チーム編成の最適化

チーム編成の最適化については、3.4で述べた役割遂行可否の制約を満たしながらチームを編成し、3.5で述べたチームの能力バラツキを求め、それが最小になるように最適化を行っていく。

具体的には、役割遂行可否の制約を満たしているチーム編成案を作成する。次に全てのチームの能力を算出し、チームの能力の分散を計算する。これを複数回繰り返して、チームの能力の分散が最も小さくなるような学習者の割り当てができたとき、それが我々の考える最適なチーム編成である。

最適なチーム編成を求めるための、目的関数 O は以下のように表せる。

$$O = \min \left\{ \frac{\sum_{k=1}^m (o_k - \bar{o})^2}{m} \right\}$$

n : 学習者の総数

\bar{o} : 学習者全員の平均能力

m : チーム数

4. 遺伝的アルゴリズムによるシステムの実現

4.1 遺伝的アルゴリズムについて

遺伝的アルゴリズムは生物が環境に適応して進化していく過程を工学的に模倣したアルゴリズムである。対象となる解を遺伝的アルゴリズムで扱うために、文字列の並びで表現された染色体という形でコーディングする。この染色体の集合を母集団と呼ぶ。遺伝的アルゴリズムは前世代の母集団から親となる2つの個体を選択し、選択した個体に対して交叉もしくは突然変異という遺伝的操作を用いて次世代の個体(子)を生成する。この操作

を次世代の母集団が一定数に達するまで次々と繰り返す。最後に次世代の母集団の中から適応度により解の取捨選択を行う。このような操作を繰り返し行うことで適合度の高い解を探索する。

前述の通り、遺伝的アルゴリズムは解の候補を次々に生成し、解の候補に対して適応度関数(目的関数)を用いて評価を行い、解の取捨を決定する方法である。したがって、染色体同士の組み合わせの数が増えると指数関数的に計算時間が増大する。しかし、チーム編成問題においてすべての制約を満たす解というのはわずかな数でしかないため、高速に近似解を求めることが可能である。

加えて遺伝的アルゴリズムは解が満たすべき条件を問題における制約という形で与えるので、対象領域の知識があれば解法の知識がなくても容易に近似値を得ることができるという利点もある。

4.2 遺伝的アルゴリズムの導入理由

本演習においては2004年度までの実績より履修者の数が100人となると仮定した。また、演習では1チームあたりの人数を5人とするようになっており、これらの条件からすべてのチーム編成案の数を求めたところ $1.0E+98$ 通りの組合せが存在することが判明した。これらすべての編成案を評価し、その中から最適なチーム編成を選択することは非現実的である。

チーム編成を手で行おうとした場合、チーム編成担当者は前述の制約を満足させることを何よりも優先して編成案を練り、そのようにして作成された編成案の中から最適な案を選択するという手順になることになる。このため、制約の数が多い場合には、考慮しなければならない条件の組み合わせが多過ぎて、すべての制約を満足する案を作成することさえ困難となり、最適な案にたどり着くのは殆ど不可能である。

これに対し、遺伝的アルゴリズムは次から次へと編成案(解の候補)を生成して行き、生成された解の候補の中で制約を満たすもののみを解とする(このような方法を生成検査法と呼ぶ)。得られた解のうちでもっとも適合度の高いものを最適解とする(適合度の評価には評価関数を用いる)。このため、制約の数が多い組み合わせ最適化問題には生成検査法が断然有利なので、チーム編成問題には遺伝的アルゴリズムを導入した。

4.3 コーディング

遺伝的アルゴリズムを用いてチーム編成の最適化を行う場合には、コーディング(染色体の並びとその意味付け)及び、遺伝子操作のアルゴリズムを決定する必要がある。

本システムでは選択アルゴリズムとしてルーレットホイールアルゴリズム、交叉アルゴリズムとして一様交差、突然変異としては2点の入れ替えによるアルゴリズムを

表2 遺伝的アルゴリズムのパラメータ

個体サイズ	2000
交叉確率	0.7
突然変異確率	0.01
計算収束までの世代	4000

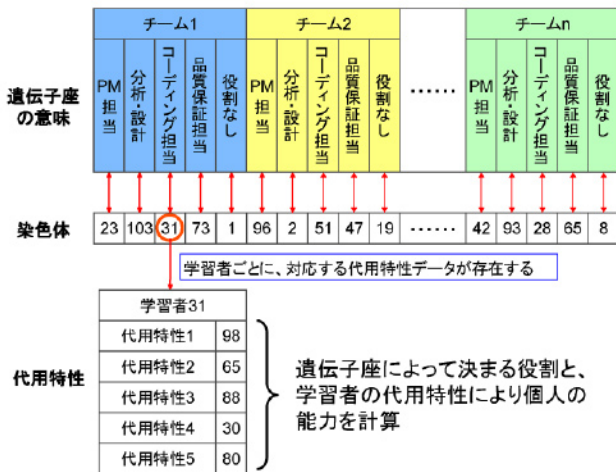


図1 遺伝子コーディング

採用した。これらのアルゴリズムはチーム編成を行う場合において致死遺伝子を発生させないので、効率的なチーム編成が可能となる。

本システムで用いている遺伝的アルゴリズムのパラメータを表2に、コーディングを図1に示す。

ここで染色体は学習者の並びを表しており、遺伝子には学習者のIDが格納されている。染色体の長さは演習を履修する学習者の数となるため固定長である。

各遺伝子座には学習者の所属チーム番号と役割が割り当てられており、遺伝子の位置（遺伝子座）によって学習者の所属チームや役割が決定する。また、それぞれの学習者には対応する代用特性のデータが別領域に格納されており、個人の能力やチームの能力を評価する際に参照することができるようになっている。

今回の実験で用いた能力計算方法の具体例を表3に示す。

計算方法については3.3で述べたとおり、個人の能力値については代用特性の総和を用いている。また、役割の遂行可否については3.4で述べたとおり、役割と最も関連が深いと思われる代用特性が一定水準を上回っているかどうかで判断している（表3の網掛け部分）。

具体的に説明すると、表3のリーダーを担当する学籍番号001の学習者の能力は276となる。また、この学習者は役割がリーダーであるため、役割と最も関連が深いと思われる代用特性の、プロジェクトマネジメントに関する問題の成績の数値が一定以上である必要がある。この学習者のプロジェクトマネジメントに関する問題の成

表3 能力計算の具体例

役割	リーダー	分析・設計	コーディング	品質保証	役割なし
学籍番号	001	002	003	004	005
プロジェクトマネジメントに関する問題の成績	84	80	90	78	67
ソフトウェア設計に関する問題の成績	69	90	75	80	76
プログラミングに関する問題の成績	75	79	88	65	78
ソフトウェアテストに関する問題の成績	48	62	棄権	93	63
個人の能力	276	311	253	316	284
チームの能力	1440/19 = 75.79				

績は84となっており、今回はこの水準を、該当の単位が60以上であれば取得可能であると設定したので、この学習者はリーダーの役割の遂行が可能であると判断できる。

しかしながら、このような方法で学習者の割り当てを行っていくと、能力の低い学習者をチームに割り当てることができない場合が出てくる。このため、「役割なし」という役割（遺伝子座）を設けて、どの役割も遂行できない学習者でもいずれかのチームに所属できるようにしている。役割を割り当てないメンバーの能力評価についてはすべての代用特性の平均を取り、これを仮想的な代用特性とみなす。なお、この代用特性については役割遂行が可能か否かの判断を行わない。

4.4 システムの実装

本システムの構成と導入されている技術の概要を図2に示す。

システムの中核となるEtUDE/GOの実装はJava言語（Sun Microsystems, 2006a）で記述したWebアプリケーションとして実装した。Webアプリケーションとしたことによって、クライアントのプラットフォームに依存せずにシステムが利用でき、システムのアップグレードについても容易に対処することができる。さらに、EtUDE/GOはJava Servlet 2.4仕様（Sun Microsystems, 2006c）及び、JSP 2.0仕様（Sun Microsystems, 2003）に準拠しているため、今回用いた動作環境だけでなく、これらの標準に準拠しているJava Platform, Enterprise

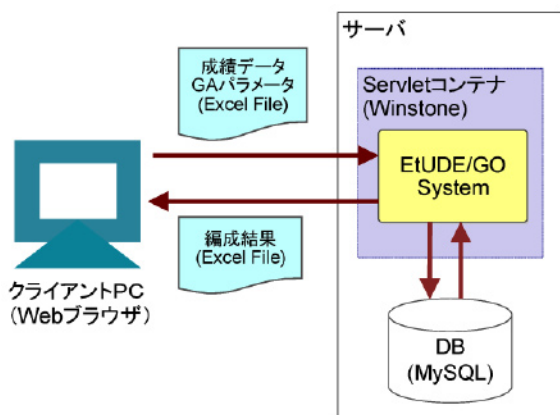


図2 EtUDE/GOシステム概要

Edition (Sun Microsystems, 2006b) 対応 Web アプリケーションサーバであれば本システムの動作させることが可能である。

また、本システムの実装にあたっては、EtUDE/GOのコアシステム以外の部分についてはオープンソースソフトウェアを積極的に採用している。近年、オープンソースソフトウェアはその機能性や信頼性が向上してきており、市販のソフトウェアとの技術的な差は縮小してきていることや、導入や運用に関する経済性に優れていることなどを考慮して採用した。

今回の実験に際しては組み込み用途に適しており、動作時のオーバーヘッドが少ないWebアプリケーションサーバであるWinstone (Knowles, 2006)を用いた。また、データベースサーバとしては同様の理由により、MySQL (MySQL AB, 2006)を採用した。

クライアントについてはOSに標準搭載されているWebブラウザ (Internet ExplorerやSafari) を使ってアクセスすることができるため、特別なソフトウェアをインストールする必要は全くない。

システムを利用する際にユーザーは多数の成績データをシステムに入力する必要があるが、教授者の負担を軽減するため、Microsoft Excelファイルインターフェイスファイルとして採用した。これにより、必要なデータを一括してシステムに入力することが可能である。

本システムでは成績データを扱うため、そのセキュリティには十分留意が必要である。このため、入力された成績データについてはメモリ内のみで保持することとし、DBには一切の保存を行っていない。また、本システムから出力された編成情報は学習者の識別IDとグループ番号のみが出力されるだけにし、編成情報から学習者の成績を知ることはできない。

5. 実験結果とその評価

5.1 チーム編成結果とその評価

EtUDE/GOに実際の成績データを入力してチーム編成

を行い、同大学の「高度情報演習2B」に対して適用した。演習の概要を表4に示す。なお、2004年度については、チームにおけるメンバー数の上限を超えない範囲で学生同士が任意にチーム編成し、演習を実施した結果である。EtUDE/GOによってチーム編成を行った結果を表5に示す。

表5中のチームの能力については3.5で述べたチーム能力の評価式によって求められた値を正規化したものである。表5中の数値が大きいくほどチームの能力が高いことを表しており、数値が0を上回っている場合にはその

表4 演習の概要

	2004年度	2005年度
演習授業名	高度情報演習2B (本学情報工学科授業)	高度情報演習2B (本学情報工学科授業)
演習期間	2004年12月9日～ 2005年1月13日	2005年11月24日～ 2006年1月12日
対象	55人 1チーム3～4人の学生 ×15グループ	98人 1チーム4～5人の学生 ×24グループ
課題	会議室予約システムの開発	

表5 編成結果

チームNo	チームの能力
1	0.185119428
2	0.094237485
3	0.060064946
4	-0.065403569
5	0.011069905
6	0.092516021
7	0.038672404
8	0.129560758
9	0.104781865
10	-0.165508255
11	0.11599564
12	0.024061426
13	-0.027259409
14	-0.054279482
15	0.185479665
16	-0.097264909
17	-0.263507054
18	-0.054867172
19	-0.037484953
20	-0.003858067
21	0.073745431
22	-0.101143654
23	-0.030188755
24	-0.165594005
平均	0.01285761
分散	0.002039404

チームは学習者全員の平均能力と比較して、それ以上の能力があることを表している（正規化したデータの平均は0であり、分散は1となるため）。この編成結果より、チームの能力の平均及び分散が小さな値を示しており、本システムによる編成が適切に行われたと見なすことができる。

5.2 演習結果とその評価

ここでは演習終了後の結果について述べる。2004年度においては全15チーム中、最終成果物の提出に至らなかったチームが3チーム存在したが、2005年度においては全25チーム中そのようなチームは皆無であった。

演習終了後に演習に関するアンケート調査を行った。演習全体に対する学習者の満足度については、80%以上の学習者が演習の内容に対し、「とても充実していた」「充実していた」との回答を行っている（図3）。この結果から、本演習に対する学習者の満足度は非常に高く、演習全体としては質の高い演習が提供できたことが確認された。

さらに、チーム編成についての学習者の実感を調査するために、チームのメンバーとどの程度協力できたかという質問を行ったところ、「全員で協力できた」「一部のメンバーで協力した」という回答の割合が90%に上った。同様の質問に対する2004年度の回答の割合は55%であり、グループ編成の満足度が飛躍的に向上している（図4）。

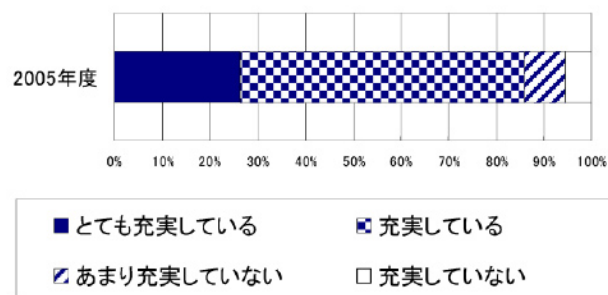


図3 演習全体の充実度

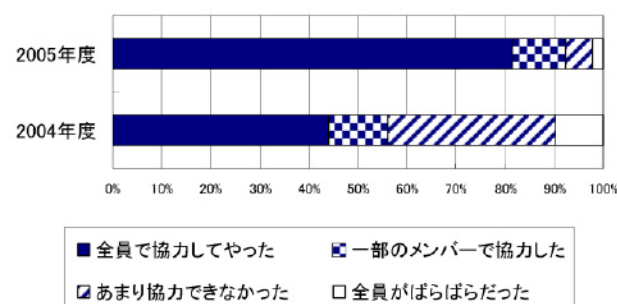


図4 チームに対するアンケート

表6 最終成果物のクラス数

2004年度		2005年度	
チーム	クラス数	チーム	クラス数
1	5	1	6
2	19	2	5
3	4	3	3
4	8	4	6
5	10	5	5
6	8	6	3
7	4	7	7
8	11	8	9
9	6	9	3
10	10	10	7
11	4	11	7
12	9	12	4
13	3	13	6
14	8	14	2
15	9	15	3
平均	7.866666667	17	2
分散	15.04888889	18	2
		19	5
		20	5
		21	3
		22	7
		23	2
		24	7
		25	4
		平均	4.708333333
		分散	4.052930057

この結果より、最適なチーム編成が学習者の実感にも現れていることが窺える。

また、演習の結果について定量的な評価を行うために、成果物のクラス数を計測し、学習者のモデリング能力のバラツキを評価した。クラスとはオブジェクト指向ソフトウェア設計における、ソフトウェアモジュールの単位を表している。学習者は演習において、クラスをいくつ作り、また必要とされる機能を各クラスにどのように割り振るのかを決定する必要があるが、この作業を適切に行うのは初学者にはかなり難しく、本演習において理解度による差が現れやすい部分である。その結果を表6に示す。

その結果、2004年度に比べて、2005年度は分散が小さくなっており、チーム間のバラツキが軽減されたことが確認できた。

これらの結果から、本システムはソフトウェア開発演習授業のための最適なチーム編成を実現できていることが、学習者の実感と最終成果物より確認された。

6. まとめと今後の課題

本稿ではソフトウェア開発のためのグループ演習における最適なチーム編成を行う方法について提案し、チーム編成を自動的に行うシステムを構築した。また、構築

したシステムを実際の同大学の授業科目である高度情報演習2Bに適用し、評価を行い、システムの有効性を確認した。

今回の実験では代用特性として用いる授業科目の選択やそれらの間の重み付けについては十分な検討を行っていない。今後、このシステムの有効性を高めるために、蓄積されたデータを多変量解析によって解析し、役割と代用特性の関係を明らかにしていく予定である。しかし、役割と代用特性の間にどのような関係式が得られたとしても、システムはプログラムをまったく変更せずに利用できる設計となっている。

最後に、本論文で述べたチーム編成の問題については、本論文が扱っているソフトウェア開発グループ演習のためのチーム編成に留まらず、メディア教育を含めた様々な分野で起こりうる問題であるため、本論文にて著者らが提案したチーム編成手法は、様々な分野への応用が可能であると考えている。

参考文献

- Kunihiko Chiken, Atsuo Hazeyama (2003). Awareness Support in Group-based Software Engineering Education System Proceedings of the 10th Asia-Pacific Software and Applications Conference (APSEC2003), IEEE Computer Society Press, pp.280-289.
- 植山 淳雄 (2001). Research on Optimal Group Organization System in Group Learning ソフトウェア工学研究財団高度情報化支援ソフトウェアシズ育成事業平成12年度成果発表会。
- Atsuo Hazeyama, Naota Sawabe, Seiichi Komiya (2002). Group Organization System for Software Engineering Group Learning with Genetic Algorithm IEICE Transactions on Information and Systems, Vol. E85-D, No. 4, pp.666-673.
- 石川達也, 橋浦弘明, 山下公太郎, 桑原 徹, 秋 玉梅, 古宮誠一 (2006). ソフトウェア開発グループ演習支援システム EtUDE 電子情報通信学会技術報告, vol. 105, no. 653, KBSE2005-48, pp.7-12.
- 磯崎友香, 山下公太郎, 石川達也, 橋浦弘明, 古宮誠一 (2005). ソフトウェア開発プロジェクト演習支援システム EtUDE 第67回情報処理学会全国大会講演論文集, vol. 4, 4D-4.
- 岩崎彰典, 宮地 功, 尾上誉幸 (2002). 学習グループ編成におけるGAによる解と厳密解との比較 電子情報通信学会技術報告, vol. 102, no. 330, ET2002-40, pp.47-52.
- Chyng-Yang Jang, Charles Steinfield, Ben Pfaff (2002). Virtual group awareness and groupware support: an evaluation of the TeamSCOPE system, International Journal of Human-Computer Studies, Volume 56, Issue 1, Academic Press, pp.109-126.
- Rick Knowles (2006). Winstone Servlet Container, September 11, 2006, <<http://winstone.sourceforge.net>> (2006年11月15日現在)
- 松浦佐江子, 相場 亮 (2003). グループワークによるソフトウェア工学教育の試み情報処理学会研究報告, Vol.

- 2003, No. 13, 2003-CE-68-1, pp.9-16.
- MySQL AB (2006). MySQL Community Server 4.1, July 19, 2006, <<http://www.mysql.com/>> (2006年11月15日現在)
- Sun Microsystems, Inc. (2003). JSR-000152 Java Server Pages 2.0 Specification, November 24, 2003, <<http://jcp.org/aboutJava/communityprocess/final/jsr152/>> (2006年11月15日現在)
- Sun Microsystems, Inc. (2006a). The Source for Java Developers, 2006, <<http://www.java.com/>> (2006年11月15日現在)
- Sun Microsystems, Inc. (2006b). Java EE at a Glance, 2006, <<http://java.sun.com/javaee/>> (2006年11月15日現在)
- Sun Microsystems, Inc. (2006c). JSR-000154 Java Servlet 2.4 Specification, October 3, 2006, <<http://jcp.org/aboutJava/communityprocess/final/jsr154/>> (2006年11月15日現在)



橋浦 弘明

2002 芝浦工大・工・工業経営卒, (株)野村総合研究所入社。2003 上山日通販売(株)入社。現職。主として新規事業開発及び、ソフトウェアの設計開発に従事。2005 同大専門職大学院・工学マネジメント・専門職学位課程(技術経営修士[MOT])了。現在、同大大学院・工・博士(後期)課程(機能制御システム)在学中。主としてソフトウェア工学及び、ソフトウェア工学教育に関する研究開発に従事。ソフトウェア設計方法論に関心を持つ。



桑原 徹

2006 芝浦工大・工・情報工卒, (株)日立システムアンドサービスに入社。保険業界を中心としたIT基盤設計・構築に従事。



秋 玉梅

2006 芝浦工大・工・情報工卒, 日本電気(株)入社。現在、流通業向けシステム開発に従事。



石川 達也

2004 芝浦工大・工・工業経営卒。2006 同大大学院・工・修士課程(電気電子情報工)了, 三菱電機(株)入社。主に社内のソフトウェア開発標準化の活動に従事。ソフトウェア開発方法論に興味を持つ。



山下公太郎

2005 芝浦工大・工・情報工卒。現在、同大大学院・工・修士課程(電気電子情報工)在学中。主にソフトウェア開発の教育支援システムに関する研究に従事。



こみや せいいち
古宮 誠一

1969 埼玉大・理工・数学卒, 1970 (株)日立製作所入社。1984 (特別認可法人) 情報処理技術者センター (略称IPA) に出向し, 自動プログラミングシステムをはじめとする各種CASEツールの構築技術, ソフトウェア設計方法論とそのメタ理論, CAI 及び知的CAIなどの研究に従事。1988~2000 IPA 技術センター 特別研究員。1991~7 IPA 新ソフトウェア構造化モデル研究本部長付を兼務。1993 徳島大学客員教授。1995より千葉大学情報工学科非常勤講師。1997年より芝浦工業大学客員教授(兼)同大学大学院非常勤講師。2000年3月信州大学博士(工学)。2001より芝浦工業大学教授。2003より同大学専門職大学院 (MOT) 教授を兼務。1992~3/1994~5/1996~7 知能ソフトウェア工学研究会幹事/副委員長/委員長。1996~7 電子情報通信学会情報・システムソサエティ運営委員。1994~7 電子情報通信学会論文誌編集委員。1998~9 電子情報通信学会論文誌編集委員。1998~9 電子情報通信学会論文誌編集委員会幹事。

Optimizing Project Team Formation for Exercises in Software Development Conducted in Units of Groups

Hiroaki Hashiura¹⁾ · Toru Kuwabara²⁾ · Yumei Qiu²⁾
Tatsuya Ishikawa¹⁾ · Koutarou Yamashita¹⁾ · Seiichi Komiya¹⁾

Recently, software to be developed becomes large-scale and increasingly complex. Most of the companies require universities to develop talented human resources capable of software development. It is usually planned a project for software development. Students in Shibaura Institute of Technology make many small groups and conduct exercises in units of groups in order to acquire knowledge and skills for software development. Each member of each group assumes respectively a specified role for software development. The authors call such an exercise a “project exercise”. Success and failure of a project exercise depend on whether or not project team formation of each group is good. Therefore, the authors developed a system to create automatically such project team formation of each group as each one of the members of each group can respectively undertake his/her role and variance in ability of software development among groups is as small as possible. In this paper, the authors propose a model for creating project team formation of each group to meet the above condition by using Genetic Algorithm, show project team formation of each group created by using a Genetic Algorithm based system developed on the basis of the model, and describe success and failure of the project exercise based on the created configuration

Keywords

Optimizing Project Team Formation, Exercises in Units of Groups, Exercise for Software Development, Genetic Algorithm, Web Application System

¹⁾ Graduate School of Engineering, Shibaura Institute of Technology

²⁾ Faculty of Engineering, Shibaura Institute of Technology