

ソフトウェア開発におけるチ - ムの協調と競争

玉井 哲雄, 西山 聡, 堀 敦史 (三菱総合研究所)

概要

チ - ムや組織でソフトウェアを開発する際、そのチ - ム内の協調のあり方に改めて関心が向けられ、研究が活発化しつつある。一方、ソフトウェア開発には、チ - ムや組織間の競争も重要な役割を果たす。本論文は、ソフトウェア開発における協調と競争の意義を探るため、1チ - ム3名からなる3チ - ムに同一課題を与えて、チ - ム内の協調とチ - ム間の競争を見た実験結果につき報告する。

1 はじめに

ソフトウェア開発作業における協調 (coordination) の役割が改めて見直され、その理論や協力体制を支援するシステムに関する研究が活発化している。とくにこれまで、協同作業は日本に比べあまり得意でないといわれる米国にこうした動きが活発なのは、注目される。T. Winograd と F. Flores[3] は、従来の合理主義アプローチ (rational approach) と、その具体例としての言語の情報コミュニケーション的解釈、問題解決による意思決定というモデル、などに哲学的な批判を浴びせるとともに、新しいコンピュータ利用の考え方の例として協調システム (Coordinating System) をあげて、人間同士や人間とコンピュータとの間の“行動のための会話 (conversation for action)”モデルの重要性を示した。また 1986 年 12 月には、テキサス州オースチンで MCC 主催によるコンピュータ支援協力作業に関する会議 (Conference on Computer Supported Cooperative Work) が開かれ、30 編の論文が寄せられた [1]。1987 年 4 月のソフトウェア工学国際会議 (International Conference on Software Engineering) では、“ソフトウェア・プロセスの実験的研究”というパネルが開かれたが、そこでもチ - ムによる設計作業でどんな点が重要かという問題につき、実験などによって得られたデータの分析結果が報告された。

TQC のようなグループ活動の盛んな日本では、どうであろうか。よくいわれるように、我が国ではどのような大きさの単位でチ - ムであるかと、一般にその構成メンバーの資質、考え方、経験が均質化している上に、各人が自己の個性を主張するよりは、集団としての合意形成に努める傾向が強いとすれば、協同作業を行う上での障害は、少ないように見える。しかしこのことは同時に、協同作業の問題点が表面化するのを妨げ、協力体制を分析しその向上を工夫するという意識的な努力に関心が向かない、という傾向を生み出す可能性を示唆するものかもしれない。

ソフトウェア工学で生みだされてきた様々な技法も、実際に適用される際には組織やチ - ムという体制の中で用いられる。そこでは役割の分担、役割と役割との間のコミュニケーションと協調が重要であり、手法やツールはその中に動的に位置づけられてその使命を果たす。したがって、ソフトウェア工学の研究と実践に際しては、ソフトウェア開発の協同作業の側面にも注目する必要がある。

協調と対立する概念として競争 (competition) がある。ソフトウェアの世界でも、協調とともに競争関係が大きなインパクトを与える。端的に現れるのが、ベンダ - 間の商品開発競争である。ワ - プロのように売れるソフトウェアは、機能、操作性、スピード、などを競う。競争の赴くところが、ユーザ - の利便性とは無関係のところでなくても行ってしまう場合も間々あるものの、多くの場合このような競争はより優れた製品をより安価に提供するという結果をもたらす。

ソフトウェアの開発過程に、この競争原理を意識的に取込む場合もある。複数の独立したチ - ムに同一の目標をもったソフトウェア開発の課題を与え、その結果もっとも優れた動作を示すものを採用する、といった方法である。評価基準が性能の場合、この方法はとくに明快である。

同じような方法を、ソフトウェアの信頼性や安全性を高めるために用いる場合もある。ハードウェアの耐故障設計の方法の一つとして、部品やサブシステムに冗長性をもたせるやり方がある。これは多重化された要素間で、故障の発生仕方が互いに独立であることを前提としている。同様に同じ機能を持ったソフトウェアを複数のチ - ムで独立に開発させ、それらを多重化してソフトウェア・システムの信頼性を高めようとするものである。本報告では、このようなソフトウェア開発過程でのチ - ム内の協力とチ - ム間の競争について、ささやかな実験を行った結果を述べる。

2 チ - ムによる開発実験

2.1 課題

チ - ムによるソフトウェア開発のテ - マとして、住宅金融公庫の融資に関する相談システムを取り上げた。住宅金融公庫の融資は、申込者の属性、融資対象の土地や住宅などによって、融資を受けられるかどうかの資格、融資可能額、返済条件などが決まる。その仕組みはかなり複雑である上に、規則の変更も頻繁に起こる。申込みは各種の金融機関で受けつけるが、その窓口担当者を教育するのも金融機関にとつてかなりの手間だという。

説明書としては、住宅金融公庫が発行している小冊子「マイホ - ム新築融資のご案内」[4]がある。これは約 60 ペ - ジの詳細なもので、たとえば、次のような記述がされている。

「ご融資が受けられる住宅は、住宅部分の床面積が 60 以上 180 以下のもの、公庫の建設基準にあてはまるもの、建築費が公庫の決めている限度以内のもの、店舗併用住宅などの場合は住宅部分の床面積が 1/2 以上のもの、敷地面積が 100 以上のもの。」

さらにこれらの各項について、詳しい定義や例外事項などが附随する。

課題としてこの問題を取り上げた理由は、1) 実際的な問題であること（実際、この問題はある都市銀行の方から示唆された）、2) 3 か月程度で開発可能な、実験に適当な規模の問題であること、3) 知識工学的なアプロ - チになじむ問題であること、などによる。

知識処理という観点では、この問題の知識源としてすでに挙げた小冊子を利用できるので、知識獲得作業が簡便になる。もちろん実用的なシステムを目指すには、この業務の担当者の知識を収集し、加えたほうがよいが、今回の実験の目的からは、共通の資料を知識源とできることでよしとした。なお、開発期間中に一度、銀行の担当部を訪れ、開発チ - ム共通の疑問点につき専門家からの教示を得た。

2.2 チ - ムの編成

チ - ム編成は 1 チ - ム 3 名とし、3 チ - ム作った。各チ - ムには、小冊子「マイホ - ム新築融資のご案内」と申込書のコピーを渡し、それに基づいて融資相談システムを作ること、内容・形式は自由とし、成果として動くプログラムとドキュメントを出すように指示した。そのほか、次の点に注意した。

1. 目的の一つは、チ - ムによるソフトウェア開発における協調と競争の意味を探ることにある。したがってチ - ム内では協同作業の進め方とくに意識するとともに、チ - ム間では情報の交換を行わないこと。ただし、一堂に会する進捗ミ - ティングは 2 週間に 1 度行い、そこでは進捗状況を報告しあう。
2. ソフトウェア工学の実践というねらいもあるので、開発経過を示す統計量（ミ - ティング、設計、プログラミングなどに要した投入時間、など）、設計および開発で意識的に用いたソフトウェア技法については、報告すること。
3. 知識工学的なアプロ - チになじむ課題なので、知識表現など知識の取扱いに、設計上とくに注意を払うこと。

開発期間は、1987 年 6 月の初めから 3 か月間とした。ただし、この間、各チ - ム員は他の主要なプロジェクトに従事し、当作業はそのあい間をぬって行っている。

チ - ムの構成員は筆者等の所属する室の室員である。たまたま室員が 9 名なので、3 名 × 3 チ - ムとした。技術や経験によるバランスには一応配慮したものの、使用した方法論、ツ - ルなどによって、生産性や成果の質を実証的に比較評価できるほどの、周到な実験計画ではない。

チ - ムはそれぞれ MacArthur, SKY, KIT と自称した。以下ではこれを、M チ - ム、S チ - ム、K チ - ムと呼ぶことにする。

3 実験結果

3.1 各チ - ムの開発状況

表 1 に各チ - ムが開発に使用した機種とプログラミング言語を示す。

LISP マシン、UNIX ワ - クステ - ション、MS/DOS パソコンと適当な組合わせに分れたのは、ある程度作為的である。とくに、K チ - ムが開発環境としては不利なパソコンを選んだのは、全体のリ - ダ - が K にいて、実際の運用環境としてはもっとも使われそうなパソコンを、少なくとも 1 チ - ムは採用するようにと、意図したものである。

各チ - ムのプロフィ - ルを、簡単に紹介する。

表 1: 使用機種と言語

チ - ム	機種	言語
M	Symbolics	ART
S	NEWS	C
K	PC98/LT	BASIC

表 2: プログラム行数と総投入時間

チ - ム	プログラム行数	総投入時間 (人時間)
M	約 3000 行	220
S	約 4000 行	240
K	約 2000 行	160

1. M チ - ム

AI 関連のソフトウェア開発にかなりの経験のある中堅技術者 2 名と、入社 2 年目の新人からなる。中堅のうち 1 名は ART (米国 Inference 社が開発販売する、高級知識処理言語) をよく知っており、このチ - ムの中心人物である。もう 1 人の中堅は、他の仕事でとくに忙しかったこともあり、参加した時間がかなり少ない。したがって、このチ - ムの作業の進め方は、1 人のリ - ダ - による新人の教育という色彩を多分に帯びた。

2. S チ - ム

3 名の経験は 5 ~ 7 年で、比較的平均している。内 2 名はとくにソフトウェアが専門で、そのうちの 1 名がこのチ - ムのリ - ダ - 格である。残る 1 名は、調査分析型の仕事に主に従事しているが、ある程度ソフトウェアの開発経験もある。このチ - ムは、システムの機能を 3 分割し、それぞれの要求定義、設計、開発を 3 名でロ - テ - ションして担当するという方法を取ったこともあり、3 名が平均的に作業をした唯一のチ - ムとなった。

3. K チ - ム

このチ - ムにいるプロジェクト全体の企画者は、チ - ム内では、前半の問題分析および設計方針の決定まではリ - ダ - としての役割を果たしたものの、後半はほとんど作業に寄与しなかった。他の 2 名はソフトウェアを専門とし、経験 4 ~ 5 年である。

表 2 に各チ - ムの開発したプログラムの行数と、総投入時間を示す。ここで行数とは、言語の違いを無視し、機械的にテキストとしての行の数を数えたものである。

この数字から、BASIC より C の方がプログラム規模が大きくなり開発に手間がかかるとか、逆に、1 時間当りの行数で生産性を測ると、C が 1 番よいなどというのは、もちろん意味がない。各チ - ムの実現したシステムの機能はかなり異なり、しかも実現範囲の大小を比較するのもかなり難しい。M が K より総投入時間が多いのは、トレーニングに近い時間が多分に含まれているからでもあるし、S と K の時間差は、3 人による開発と実質 2 人による開発の違いという点が影響していよう。

3.2 システムの比較

3 チ - ムによって作成されたシステムは、いずれもそのまま実用に使えるものではなく、プロトタイプ の性格を持つものである。以下、各システムの特徴を述べる。

(1) M (システム名: MacArthur) M チ - ムのシステムは、融資申込者が申込書を作成するのを助け、必要に応じたアドバイスをするを想定している。そのために、LISP マシンと ART の特徴を生かした柔軟で使い易いユ - ザ - インタフェ - スを目指し、かなりのレベルでそれに成功した。

この問題は、申込み本人、土地、住宅などに関する種々の属性から、融資資格、融資限度額、返済条件などが定まるという構造を持っている。そこで M チ - ムは、入力となる属性項目を、出力となる融資資格などの検査項目に応じて分類し、出力項目に対応するメニュー - の構成要素とするという設計方針を取った。トップレベルのメニュー - は、資格、融資額、返済条件であり、各々のメニュー - 内で任意の順序で入力項目を入れることができる。複数のメニュー - で共通に使われる入力項目は、1 個所で入力または修正されれば、他のメニュー - 画面にも反映される。

メニュー - やフィールドの選択は、マウスで自由に行なえ、エラ - 処理は入力直後にエラ - ウィンドウ が現れてただちに修正できるようにし、ヘルプ機能もある程度備えるなど、実用面からの完成度は 3 シス

テムのなかでももっとも高い。ただし、時間の制約により、手引きに書かれている項目で組入れられていない部分もある。

このシステムの最大の欠点は、表示がロ - マ字表記になっていることである。Symbolics と ART の組み合わせでは、漢字の使用は不可能ではないにしても困難で、やむをえなかったが、もとの問題が日本語の文書で表されているだけに、その効果を著しくそぐこととなった。

S (システム名:SKY) このシステムはM と違い、金融機関の窓口で申込書を受取った際、窓口担当者が行う整合性のチェックに用いられることを、想定している。チェック項目としては、手引きにあるほとんどのものが組入れられており、プログラム・コードの行数が他チ - ムより多い理由の一つもここにある。

想定している使用条件のせいもあるが、このシステムの処理形態はあまり柔軟性がない。入力項目は定められた順序で一直線に入力処理され、最後に整合性のチェック結果が出力される。一度入力されたものの部分的な修正すら現バ - ジョンではできないのは、時間的な制約にもよる。

入出力にはかな漢字が使われている。入力方法は、単純な質問応答型である。

(3) K (システム名:家ある子) このチ - ムの設計方針は、M にかなり近い。利用形態はM と同様、融資申込者の申込書作成支援である。入力の各項目は、一般に複数の出力検査項目で用いられ、1 つの検査項目は逆に複数の入力項目によって定まる。そこで入力は、申込書にほぼ準拠した画面上から入れられるが、すべての項目を一度に入れる必要はなく、適当にとぼしてよい。融資資格、融資額、返済条件などの出力項目は自由に選べ、その検査のための入力で未だ入れられていないものがあつたり、不適格なものがあれば、適当なエラー - メッセージを出して、対応する入力画面に戻る。

ART のようにデ - タ指向型プログラミングが自然に支援されている開発環境ではないから、できたものの柔軟性もM とくらべれば劣る。ただ、かな漢字画面による入出力は、このシステムの最大のとりえである。

3.3 協調と競争の成果

各チ - ムはそれぞれ、チ - ム内の協調を工夫した。M では、チ - ムに6 つの役割を定めた。すなわち、マネ - ジャ、ドメインエキスパート、ナレッジエンジニア、システムエンジニア、プログラマ、ユ - ザである。これを、3 名がそれぞれ2 つづつの役割をこなすように、割当てた。実際は、マネ - ジャ兼システムエンジニア役の指導のもとに、ドメインエキスパート兼プログラマ役の新人が、問題自身と ART の勉強に努めるといった形となったが、それなりに機能したといえるだろう。この役割割振りという発想は、MCC の P.Marks のいう“ パフォーマンス - ミング・アートとしてのシステム開発 ”[2] を連想させる。

S はシステムを、申込書入力、融資額算定、返済額決定の3 モジュールに分け、それぞれの作業を、要求定義、設計、開発の3 つのフェーズに分けて、その結果できる9 個の作業単位に、3 名をロ - テ - ションさせながら、各自がすべてのモジュールとフェーズに関係するように割当てた。作業単位の間は、形式を定めた文書で受渡しをするという、きわめてフォーマルな方法を取っている。その結果、作成された文書の量は、3 チ - ムの中でめだつて多い。できたシステムが硬いきらいがあるのは、あるいはこのような協調方法をとったせいかもしれない。K チ - ムは、協調に関しあまり工夫をしていない。互いのコミュニケーションのためには、ミーティングの議事録、設計過程でのメモ、などを、意識的に残すようにした程度である。プロジェクト運営はS と対照的にきわめてインフォーマルであり、残したドキュメントも少ないが、その僅かな文書は設計過程をよく示すものではある。

チ - ム間の競争の成果については、評価が難しい。チ - ムといっても、同じ室に所属するもの同士であり、真の意味での競争意識は薄く、互いの情報も必ずしも遮断されていたとは言い難い。しかし、片手間仕事で3 か月という期間の中で、とにかく3 チ - ムとも一応のプログラムを作りあげたのには、互いの競争がなんらかのインパクトになっていたと、言うことができよう。近くで他チ - ムのシステムが動きだしているようすを見れば、自分たちも頑張らねばならないと思うのは、自然なことである。

4 まとめ

ソフトウェア開発における協調と競争をテーマに、3 チ - ムに同一課題を与えて、開発実験を行った。チ - ムによって協調の仕方に違いがあり、それが成果物の性格にも反映している。また、競争はある程度、プロジェクト進捗の駆動力となっている。

今回の実験は、実験計画も結果の分析も、決して十分なものとはいえない。今後、このような問題意識のもとに、多方面で研究が行われるきっかけとなれば、幸いである。

最後に、この実験に参加した三菱総合研究所人工知能開発室の室員全員に、感謝したい。

参考文献

- [1] KRASNER, H., Ed. *Proceedings of Conference on Computer-Supported Cooperative Work* (Austin, Texas, 1986).
- [2] MARKS, P., 岸田孝一 (訳) . パフォ - ミング・ア - トとしてのシステム開発. *bit* 19, 11 (1987), 45–62.
- [3] WINOGRAD, T., AND FLORES, F. *Understanding Computers and Cognition: A New Foundation for Design*. Ablex Publishing Corporation, 1986.
- [4] 住宅金融公庫 (監修) . マイホ - ム新築融資のご案内. 住宅金融普及協会, 1987.