

教育利用講義報告：

「実践的シミュレーションソフトウェア開発演習」

高橋 英男

東京大学大学院工学系研究科非常勤講師

1. はじめに

本稿は、2009年度より実施している演習科目「実践的シミュレーションソフトウェア開発演習」についての実施報告である。本演習は、物理シミュレーションプログラムを、ソフトウェア工学の知識や手法を活用して開発するスキルを身に付けることを目的としたものである。指導に当たるのは大学教員と、企業在籍ないし企業出身の講師からなる混成チームである。大学教員はシミュレーションの題材についての指導を担当し、企業側講師はシミュレーションの数式は理解した上で、プログラム設計やプロジェクトマネジメントの指導を担当している。

本演習については、本広報誌「スーパーコンピューティングニュース」で何度か報告済みであり、科目の狙いや教育の内容についてはそちらを参照していただきたい^{1,2,3,4,5}。今回執筆を担当する筆者は、企業在籍の講師として、本演習の開発プロセスの策定を担当した。そこで本稿では、本演習の特徴的なコンテンツの一つでもある、シミュレーションプログラム向けの開発プロセスに焦点を当てつつ、本演習について改めて概要を紹介する。加えて、2019年度の演習の様子と、現状の課題認識について述べる。

2. 科目の概要

本演習の主な力点は、シミュレーションプログラムの開発に、ソフトウェア工学的なアプローチを適用することである。これにより、シミュレーション研究現場で生み出されるソフトウェア資産の質・量の向上に寄与することを目指している。ソフトウェア工学を適用する分野は大きく3つある。プログラムの設計プロセスと、開発プロジェクトの進捗管理、それに、開発ツールである。これらに関して、既存の手法や知見を、講師陣なりにアレンジして適用している。

演習で取り組むシミュレーションの題材として、受講者は分子動力学と流体力学から選ぶことができる。課題が過度に複雑にならないように、分子動力学では式が単純な Lennard-Jones ポテンシャルに基づく分子間力モデル、流体力学では有限要素法による二次元の非圧縮流体の非定常解析をそれぞれ扱う。

科目の前半、5月第2週くらいまでは、予備知識の講義と小演習の日が続く。この間に計算機アーキテクチャ、MPI、OpenMP、C++の定石、オブジェクト指向設計の考え方と作図法のエッセンスを教える。C++の文法については原則として学生の自習に委ねている。その後2つのシミュレーション題材のそれぞれの理論を各1日ずつ説明する。前半の講義と並行して、MPIとOpenMPに慣れてもらうために、行列積を並列処理により計算するプログラムを作成する課題を基礎演習として課している。

科目後半では、いよいよグループに分かれてシミュレーションプログラムの開発に入る。しかし、ここまでの

¹ <https://www.cc.u-tokyo.ac.jp/public/VOL20/No6/11.Lec201811-satof.pdf>

² <https://www.cc.u-tokyo.ac.jp/public/VOL19/No5/14.Lec201709-satof.pdf>

³ <https://www.cc.u-tokyo.ac.jp/public/VOL18/No5/11.Lec201609-satof.pdf>

⁴ <https://www.cc.u-tokyo.ac.jp/public/VOL17/No5/12.Lec201509-satofumi.pdf>

⁵ https://www.cc.u-tokyo.ac.jp/public/VOL14/No6/10_201211education-2.pdf

講義内容に基づいて学生にプログラムを独力で設計させようとする、多くの人はどこから手をつけてよいか分からず、立ち止まってしまふ。科目の開設計画が実際にそうであったため、講師によるプログラム設計例の文書を半完成の形で与える方法に切り替えた。この設計文書は、本演習で教える設計プロセスとセットになっており、かかるプロセスに即して書かれている。設計プロセスについては次節にて解説するが、複数のステップからなり、各ステップではステップに応じた観点で対象を考察し、所定の文書ないしは図面を作成する流れとなっている。

本来であればこの設計プロセスに従った設計を学生にさせて、設計文書を書かせたいところではあるが、出来上がる文書の量が多いため現実的ではない。講師がこの文書を演習の題材に即して実際に執筆してみたところ、A4のワープロ形式で50~70ページと言った分量になった。これでは、演習科目の提出課題としては多すぎると判断した。この設計に基づいてC++で実装すると2000~3000行程度のソースコードになる。この分量も多いため、シミュレーション計算のハイライト部分以外については、実装済みのソースコードを与えている。

もちろん、文書もソースも学生が自分で全て書くのが一番勉強にはなるが、作成する成果物の形式に不慣れなうちは、完成品を読んで学ぶことにも大いに意義があると考え、このようにしている。ところで、筆者が執筆した2題材に対する設計文書は、本科目の教科書の実践編 [佐藤 加藤, 2015] に収録されているので、シミュレーションプログラムの設計文書の例を見たいという方は参照いただきたい。

受講者は、与えられた設計文書とソースコードの内容をまずは把握し、その後半完成の設計文書とソースコードのそれぞれを完成させるべく、グループで手分けをして作業を進める。この時の作業は、小さな開発プロジェクトの形を取り、作業計画を立て、進捗管理や不測事態への対処をしつつ開発を進める。

開発にあたり、様々なツールを利用する。ソースコードはバージョン管理ツールGitで保管・共有し、プロジェクト内の作業項目はTrelloでチケット管理しながら、プロジェクトを進める。開発の途中に、何度かレビューを設けており、最終日には成果発表会がある。

3. 本演習での設計プロセスと表記法について

本演習での設計プロセスは、物理シミュレーションプログラムの設計に即して、本演習で独自に定めたものであり、業務システム向けの設計プロセスを単純に流用したものではない。業務システム向けの設計プロセスそのままでは物理シミュレーションプログラムの特性にマッチしない点が多いためである。筆者はIT分野の商用ソフトウェア開発を本業としており仕事では物理計算とは縁がないが、どのようなプロセスが適当なのかを検討するために、本演習の題材を元にC++のプログラムを設計・実装してみた。そこでの体験に基づき、必要となる思考過程や表記法について考察し、プロセスとして定式化した。以下では、本プロセスで採用した表記法を紹介する。思考過程については、本演習教科書の入門編 [佐藤 加藤, 2014] を参照いただきたい。

ソフトウェア工学の常識の一つとして、自明ではない工夫は、他人に伝達できるように文書化する必要がある。実装してみてよく分かったことであるが、シミュレーションプログラムには、自明ではない工夫を込める余地や必要性がたくさんある。それらの工夫を文書に書き留めるためには表記法を適切に決めておく必要がある。

まず、いくつかの標準的な表記法は、そのまま役に立つことが分かった。IT分野では広く知られている作図法の体系として **Unified Modelling Language (UML)** がある。UMLで規定されたクラス図やシーケンス図はシミュレーションプログラムでも有用である。**クラス図**はモデルに存在する変数相互の関係性を表現できる強力な手段である。また、**シーケンス図**は、並列に進行する計算処理と要所要所で発生する通信の様子を検討する上で役に立つ。言葉だけで表現しているときは大変そうに見えた流れも、シーケンス図として整理すると案外とシンプルな図になる。UMLの図ではないが、**データフロー図**はシミュレーション計算の核心となる時間発展計算の1ステップの流れを整理するために有用であることも分かった。処理手順を階層的に表現する図である **Problem Analysis Diagram (PAD)** は、MPIなどで並列動作させるプロセスの中のどれか一つのプロセスが起動してから停止するまで

の間に実行すべき処理の流れを設計する上で役立つ。

一連の標準的な図が活用できることが分かった一方で、標準化された表記法が使えないものもあった。シミュレーション対象の幾何学的な位置関係に即した図である。物理的な計算の意味を図示しようとする、対象の幾何学形状を簡素化した図が必須である。しかし、そのような図には標準的な表記法が確立していない。そのため設計者の知識に基づいた臨機応変な作図が求められる。検討プロセスを定式化しようとする立場からすれば、この点は表記法の考案や標準化が望まれる点である。

4. 2019 年度の演習の様子

本演習は昨年まで、時間割の関係で主な対象としている機械系の学生にとって受講しにくい事情があった。今年はその点が解消され、例年よりも多い 14 名の学生が最終日を迎えた。アンケートの結果を見ると満足度は全般的には高いが、一方で演習時間外にも課題に取り組む必要があることの負担が高いことを訴えたコメントもみられた。プロジェクト型の演習である以上、演習時間外の活動がある程度は想定する必要があるため、その可能性を早期に学生に案内されるように改善を図りたい。

5. 現状に対する課題認識

本科目についての初回の報告で詳述されているように、本科目の開講前に前提知識として想定していた内容の一部を、典型的な受講生がマスターできていないという問題がある。科目開設 11 周年の今年も、状況はあまり変わっていない。それらの事項は必要に応じて科目内で教えることとしているが、そのために、相応の時間を引き当てている。特に unix のシェル操作や、Reedbush へのログインと ssh の設定、のジョブ投入・MPI 関数の使い方に関して手厚い説明が必要となっている。

これらの内容に関して、受講生に利用しやすい自習教材の拡充が望まれる。現状提供されている資料は、それぞれ手間のかかった労作であると思うが、本演習の受講生の様子を見てみると、説明が難しかったり、操作の意味の説明が不足しているように見受けられる。初心者でもとりやすいチュートリアルや、ダウンロードして実行可能なコード例が拡充されれば、本科目に限らず、多くの科目にとって大変有用な資産となると思われる。

6. おわりに

本演習の受講者は、特定の物理現象のシミュレーション理論だけでなく、シミュレーションプログラム一般の設計の仕方についても学ぶことができる。読者の周囲に、本格的なシミュレーションプログラム設計の仕方を学びたいという大学院生がいたら、ぜひ本演習を紹介いただきたい。

企業勤めの筆者にとって毎年夏学期の本演習は、自分が得た知見を若い学生に伝えられる貴重な機会であると共に、学生から新しい情報や刺激をもらえる場でもある。このような機会を与えて下さった先生方、ならびに高度な計算機環境を整備いただいている情報基盤センターのスタッフの方々、そして科目を履修した受講生の皆さんには大変感謝している。

参 考 文 献

佐藤文俊, 加藤千幸編. (2014). ソフトウェア開発入門: シミュレーションソフトウェアの設計理論からプロジェクト管理まで. 東京大学出版会.

佐藤文俊, 加藤千幸編. (2015). ソフトウェア開発実践: 科学技術シミュレーションソフトウェアの設計. 東京大学出版会.