

Literate Computing for Reproducible Infrastructure による 属人化させない運用の実践

谷沢智史¹ 政谷好伸² 石坂徹³ 桑田喜隆³ 中川晋吾⁴
長久勝⁵ 合田憲人²

概要: 情報インフラの構築・運用において自動化は主要なアプローチの一つであるが、自動化においてはしばしば属人化の問題が指摘されている。我々は、コードと説明テキストを組み合わせた Jupyter Notebook をインフラ構築・運用に用いることで、知識移転がしやすいインフラの実現を目指している。

本報告では、Literate Computing for Reproducible Infrastructure(LC4RI)と名付けた、Jupyter Notebook をはじめとするツール及びアーキテクチャに基づくインフラ構築・運用手法について整理し、LC4RI の実践として (1)約 5 年間にわたる Jupyter Notebook を用いた運用 (2)Notebook 作成者と運用主体が異なる状況での Jupyter Notebook を使った運用 の 2 つの事例に関して報告する。これらの事例から、LC4RI を実運用に適用することによって得られた効果と見えてきた課題について報告する。

キーワード: 情報インフラ運用, DevOps, Jupyter Notebook, コミュニケーション可視化

Literate Computing for Reproducible Infrastructure as Practical Knowledge Transferring on DevOps

SATOSHI YAZAWA^{†1} YOSHINOBU MASATANI^{†2} TOHRU ISHIZAKA^{†3}
YOSHITAKA KUWATA^{†3} SHINGO NAKAGAWA^{†4} MASARU NAGAKU^{†5}
KENTO AIDA^{†2}

Abstract: Automation is one of the major approaches in the development and operation of information infrastructures, but the problem of individualization is often pointed out in automation. Our goal is to create an infrastructure that facilitates knowledge transfer by using Jupyter Notebooks, which can combine code and explanatory text, for infrastructure development and operation. In this report, we describe a method for building and operating infrastructure based on tools such as Jupyter Notebook and specialized architecture, which we named Literate Computing for Reproducible Infrastructure (LC4RI), and report on two practices of LC4RI: (1) an operation using Jupyter Notebooks for about five years, and (2) an operation using Jupyter Notebooks in a situation where the author and the operator of the Notebook are different. From these cases, we describe the effects obtained by applying LC4RI to actual operations and the issues that have been identified.

Keywords: Management of information infrastructure, DevOps, Jupyter Notebook, Visualization of Communication

1. 背景

近年の情報インフラの大規模化、複雑化に伴い情報インフラ構築・運用作業もソフトウェア開発プロセスと同様に効率化が求められており、この要求に対するアプローチの一つとして DevOps がある[1]。DevOps は開発者と運用者の連携を緊密にすることで、開発からサービスデリバリーまでを迅速に行う考え方である。DevOps は人間同士の協働と自動化の双方を重視するが、ツールの多くが自動化に偏っており、人間同士の協働に注目したツールは少ないことが指摘されている[2]。そのため、DevOps の実践において自動化へと注意が偏ることで、人間同士の協働が疎かになり、結果的に自動化コード作成担当者などの特定のメン

バーしかシステムを理解できない状況に陥ってしまうといった、属人化の問題が生じることが考えられる。そのため、DevOps では CALMS[3]として Culture を先頭に置き、異なる背景を持ったスタッフの相互連携を重視しており、個々の技術者の自律性と責任感の文化を醸成することで、技術者の継続的な学習を行うための動機付けをするような事例もある[4]。自律的かつ継続的な学習は必要不可欠なものであるが、これらの基本的な能力を持った技術者を育成する教育プログラムもまた重要である。しかしながら、このような教育プログラムの実現にあたっては提出物の評価、フィードバックなどの自動化が困難であるため、大規模に実施することは困難であろうことが指摘されている[5]。一方で DevOps の実施には学術的な報告が乏しいという問題も

¹ 株式会社ボイスリサーチ
Voice Research Inc.
² 国立情報学研究所
National Institute of Informatics
³ 室蘭工業大学
Muroran Institute of Technology

⁴ 有限会社カラビナシステムズ
Carabiner Systems, Inc.
⁵ ライフマティクス株式会社
Lifematics Inc.

ある。例えば DevOps の中でも Infrastructure as Code (IaC) と呼ばれる方法論に関する調査では、これらの多くは学術論文ではなくインターネット記事等のグレー文献であり、IaC のコードの保守、進化、継続的な改善に関してまとめられた学術論文は多くはないことが報告されている[6]。

我々がおこなっている、研究・教育のための情報インフラ運用においては、先進的かつ複雑なハードウェア・ソフトウェアの利用を求められるが、これらの複雑な要求をカバーする潤沢な資源を確保するのは困難であるため、特に利用するソフトウェアについてはベンダーサポートに頼らない独力による情報インフラの運用が求められる。このような情報インフラ運用においては、運用者全員が常に試行錯誤を行いながら継続的に学習をし、日頃から互いに情報共有を行うことが重要であると考えた。このような状況を作り出すための方法論として Literate Computing for Reproducible Infrastructure (LC4RI) を提案し、我々の研究・教育向けクラウド基盤の構築や運用において実践を行ってきた[7][8]。LC4RI では Jupyter Notebook[9]を用いることでスクリプトによる自動化の恩恵を享受しつつ、説明文や見出しなどの文書的な構造を付与することで、自己説明的なコードを実現することができる。

本報告では、LC4RI の実践として (1) 約 5 年間にわたる Jupyter Notebook を用いた運用: 運用対象は国立情報学研究所の研究者向けベアメタルプライベートクラウド[10] (2) Notebook 作成者と運用主体が異なる状況での Jupyter Notebook を使った運用: 運用対象は学生向け教育環境 CoursewareHub[11] の 2 つの事例に関して報告する。これらの事例から、Jupyter Notebook がチーム内の全ての運用者に利用され、かつ運用者間で Notebook を介したノウハウの伝達が行われていることを、運用者間での Notebook の転記を追跡、可視化する機構により示すことができたので報告する。

2. Literate Computing for Reproducible Infrastructure における知識共有

LC4RI では Jupyter Notebook を用いることでコードと説明文の両方を含んだ手順書を作成、実行しながら情報インフラの運用を行う (図 1)。Python スクリプト、シェルスクリプト等のコードはコードセルという形で記述することができる。シェルスクリプトを用いることで Jupyter Notebook が動作している環境でのコマンド操作や、Ansible[12]コマンドを通じた他のホストに対する操作を記述、実行することができる。ここで得られた出力文字列、画像等はコードセルの配下に記録される。また、説明はマークダウンセルという形で記述することができ、必要に応じて見出しや画像などを貼り付けることができる。これらのセルを活用することで、説明が付与された実行可能な手順書を作成する

ことが可能となる。

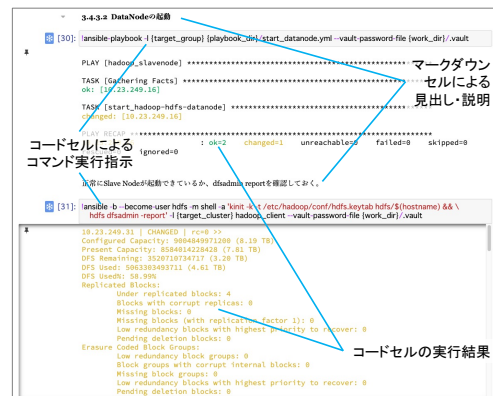


図 1 Jupyter Notebook におけるコード・説明文・出力
Figure 1 Code, description, and output in Jupyter Notebook.

2.1 Jupyter Notebook を使った試行錯誤の奨励

LC4RI では手順書の作成者と実行者は原則として分離しない。運用者は過去に実施した手順書を再利用することが多いが、自身の判断に基づいて既存のセルの内容を変更したり、新規にセルを追加したり、異なる既存の手順書からセルを転記しても良いものとする。ここでは、ソフトウェアは常にアップデートされており[13]、情報インフラは常に変化する対象であると捉えている。このような変化する対象に対して柔軟に対処するためには、運用者自身が手順書の内容を理解し、必要に応じて適切に変更し状況に対応することが不可欠と考えるためである。

Jupyter Notebook を用いて作業する範囲を広げるため、運用者に対して全ての場面で Jupyter Notebook を利用することを奨励している。構築・アップグレードなどのコードでの自動化がされやすい場面以外にも、障害調査のような従来コード化されづらい場面についても Jupyter Notebook の使用を推奨している。また、バージョンアップのような状態の変更を伴う場面での失敗を減らすため、試行錯誤をしやすような配慮も行っている。試行錯誤の奨励には、手順書は完全な手順でなくても良いことを運用者に対して伝える活動のほか、試行錯誤が可能な環境となるよう、ハードウェアは一定数同型のものを複数揃え、これらを接続するネットワークは API やコマンドライン等で容易に変更可能な装置により接続している。この配慮については実践環境において後述する。

2.2 証跡つき手順の保存と共有

LC4RI では全ての作業に関して、できる限り結果を Notebook として残すことを推奨する。従来のようにターミナルで作業した場合でも、その実行コマンドと実行結果をマークダウンセルとして貼り付けたものを保存することを推奨することで、運用に関わる全ての行為が Notebook ファイルとして保存されることを目指す。このように日々の運

用作業において作成, 実行した Notebook はセル実行結果の出力とともに証跡 Notebook として残すものとし, 日付_通し番号_Notebook 名の形式で Jupyter Notebook 環境に残すよう指示している。

Jupyter Notebook を操作する環境は, 後述する OperationHub 環境によりユーザごとに分離された形としているが, 各運用者は他の全ての運用者の Jupyter Notebook を閲覧することができるよう設定を行った。Notebook は個々の事例に特化した証跡の形で記録されるため, 運用者間の情報共有の形式は次のいずれかになる。(1) 運用者がある作業を開始する際, すでに実施した例があるか他の運用者に質問をし, 他の運用者は事例がある場合はその証跡 Notebook の場所を指し示す。(2) 運用者が意図しないエラー等で作業を中断した場合, そのエラーが記録された証跡 Notebook を他の運用者に示して相談を行う。(3) 運用者が自ら, 他の運用者の Notebook ディレクトリ内を所望の Notebook があるか探す。このように証跡 Notebook を介して情報共有することにより, 運用者が手順を別の運用者に伝える際に, 手順だけでなく期待する出力結果の情報も容易に共有できるよう配慮している。

2.3 セル単位での利用の追跡

運用者は作業を実施する際, 過去類似した作業を実施した際に使用した証跡 Notebook をコピーして作業を開始する。また, 作業の中で別の Notebook からセルを流用したり, 新規にセルを作成し記述したりすることもできる。この方式により状況に合わせた手順を柔軟に作成, 実施できる一方, 証跡として残される Notebook は個々の事例に特化した形になりやすい。結果としてある目的に対して単一の Notebook がメンテナンスされることはないため, Version Control System のようなファイル単位のバージョン管理は困難である。そこで我々は, nblineage と呼ぶ, セル単位での利用追跡が可能な拡張機能を開発した。

nblineage[14]は Jupyter Notebook の拡張機能として実装し, 新規にセルが作成されると MEME と呼ぶランダムな UUID を生成して挿入する (図 2)。この MEME はセルのメタデータとして保存され, セルがコピーされる際にこの内容もコピーされる。この MEME 情報は Notebook 配布の際などは専用ツールによりリセットをすることも可能である。

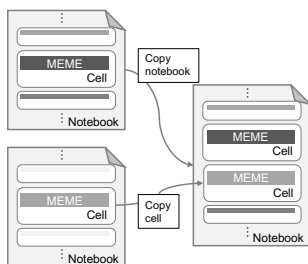


図 2 nblineage による MEME の付与・継承

Figure 2 Assigning and transferring MEME by nblineage.

3. LC4RI の実践環境

LC4RI の実践にあたり利用した環境の模式図を図 3 に示す。LC4RI を実践するにあたっては, Jupyter Notebook が動作する環境を整備すること, Notebook 中のコードによって制御しやすいハードウェア環境を整えることが重要である。

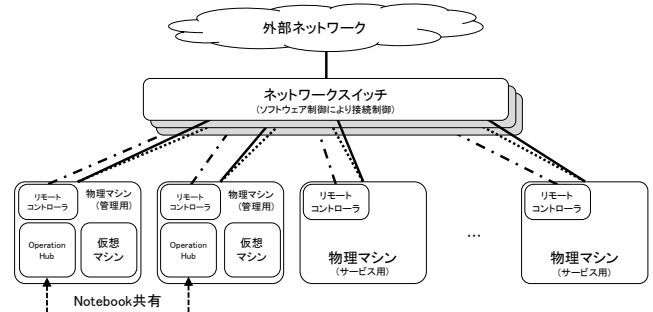


図 3 LC4RI の実践環境

Figure 3 Practical Environment for LC4RI.

3.1 操作環境: OperationHub

運用者が利用する Jupyter Notebook 環境には OperationHub[15]と名付けたマルチユーザ Jupyter Notebook 環境を利用する。OperationHub は JupyterHub[16]をベースに情報インフラ運用に必要な機能を拡張したものである。OperationHub はホスト Linux ユーザと JupyterHub ユーザの対応関係を管理する機能のほか, ユーザ間の Notebook 共有機能を提供する。OperationHub により, ある運用者が作成した Notebook は全て別の運用者から参照できるようになる。共有されるのは Notebook ディレクトリの内容のみで, 各運用者の SSH 秘密鍵等の情報は共有されない。

OperationHub において運用者は nblineage を含むインフラ運用に特化したツールを一通りインストールした Jupyter Notebook イメージ[17]を利用することができる。ここで運用者は Ansible 等のコマンドを利用しながらインフラ運用を行う。Jupyter Notebook 環境には運用者が任意のソフトウェアをインストールすることができるが, 原則としてイメージ内にプリインストールされていないツールを利用する場合は, このようなツールを利用する Notebook 中にインストールコードもしくは説明文を明記するものとする。

3.2 操作対象インフラ: Symmetric Architecture

情報インフラは試行錯誤により柔軟に変更できるよう, LC4RI では原則としてできる限りサーバーの接続は同スペック・同形となるようにする。この方針を Symmetric Architecture と呼ぶ。Borg[18]のような同形のシステムを多数揃えたアーキテクチャを採用することも考えられたが, ソフトウェアを独自に開発する体制は想定しておらず, 方針と位置付けるにとどめた。例えば, 管理用マシン, ユーザ提供用マシンといった役割を定義するが, この役割の種

類をなるべく増やさないようにし、各役割を担うハードウェアや物理接続はできる限り同形となるようにした上で、役割ごとに必ず複数のハードウェアを用意できるように配慮する。このようにオンプレミス環境でもできる限りソフトウェアによる制御を可能にし、かつクラウドサービスのようなオンデマンド性を擬似的に確保することで、臨機応変な試行錯誤の促進を狙っている。

4. ケーススタディ

4.1 国立情報学研究所による所内クラウドの運用

(1) 運用対象システム

国立情報学研究所では所内研究者向けベアメタルプライベートクラウドを運用している。ベアメタルクラウドは OpenStack[19]にて実現する。現在4世代目の環境の運用試行を行っている段階で、OpenStack のバージョンアップやマシンの最新化を行っている。運用当初から、ベアメタル貸し出し機能の実現のため、独自のソフトウェア拡張を含んでおり、この開発は運用とは別に開発を発注して拡張を行なっている。貸し出し可能なマシンは常時100台前後あり、ログ収集・分析を目的とした Hadoop[20]環境の構築・運用や、利用者が使用するデータ処理クラスターの構築・運用の支援も行う。

(2) 運用体制

定常的な運用・利用者支援は2つの企業からそれぞれエンジニア2名の計4名（のちに1名別の企業から参加）、OpenStack の機能開発として1業者4名が開発期間のみ参加した。

4.2 室蘭工業大学による CoursewareHub の運用

(3) 運用対象システム

室蘭工業大学ではプログラミング教育を目的とした学習環境を CoursewareHub により構築・運用している[21]。CoursewareHub は JupyterHub をベースに教育向け独自拡張を施したソフトウェアであり、開発は国立情報学研究所で行ったものである。配備環境は Amazon Web Services 上の仮想マシン5台および学内に配置した計算機8台である。

(4) 運用体制

構築・運用は室蘭工大教員1名で実施した。CoursewareHub の構築・運用サポートとして国立情報学研究所側から業者1名が参加し、Jupyter Notebook の提供を通じて支援を行なった。

5. 結果

5.1 Jupyter Notebook を使った運用の定着

国立情報学研究所におけるプライベートクラウド運用において LC4RI の実施状況を Notebook の作成頻度及びセ

ルの観点に着目してまとめた(図4)。Notebook数(a)は2015年12月から2022年1月までで18171個あり、Notebookの作成日時を Notebook ファイルの mtime 属性であるとみなし、3ヶ月ごとの Jupyter Notebook 作成頻度を集計した。なお、システム監視のために自動的・定期的に作成される Notebook はここから除外した。運用開始直後は多くの試行錯誤がなされ、2017年ごろから徐々に増加量が増えている。2020年ごろの急激な増加は新規クラウドの増設に伴う開発作業によるものであり、2021年1月ごろの減少はクラウド装置の設置場所移転にともなう Notebook による作業の減少を示し、その後移転後のサービス再開のための各種作業により Notebook 数が増加している。この変化から、運用作業を Jupyter Notebook により行うことが定着していることが確認できた。

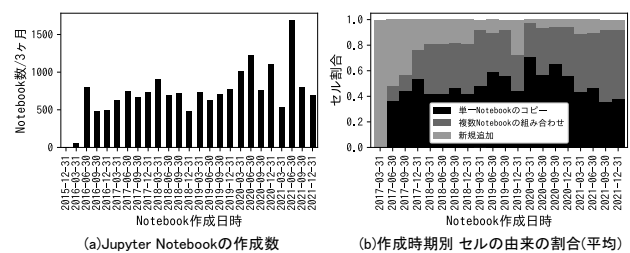


図4 Jupyter Notebook の作成頻度の変化

Figure 4 Frequency of creating Jupyter Notebooks

また、Notebook を構成するセルの由来の割合(図4(b))は、MEME が記録されている各 Notebook に関して、Notebook 内セルの各 MEME と同一の MEME が現れる過去の Notebook を抽出し(複数の Notebook があれば直近のものを利用)、Notebook が単一の Notebook を再利用したものであればそのセルの割合、あるいは複数の Notebook を組み合わせたものであれば再利用されたセル全体の割合、MEME が過去に現れないセルについては新規セルとみなし新規追加セルの割合を計算している。作成時期別にこれら3種類のセル割合の平均値を取ることで、作成時期別に業務がどの程度定型化されているかの指標とすることができる。MEME 機能を運用開始したのが2017年1月ごろであり、この頃作成されたセルは新規に MEME が付与されたため、新規に追加されたセルとして扱われる。その後、徐々に MEME が付与されることで定型的なセルの使用率が識別されはじめ、2020年ごろまで単一 Notebook を由来とするセルの割合が増えており、これは業務がある程度定型化されていることを示唆している。また、2020年ごろからの新規クラウド増設、移転作業にともなって単一 Notebook を由来とするセルの割合が減り、複数の Notebook からセルを組み合わせることで業務に当たっている。これは増設、移転のような作業に対して過去のノウハウを流用しながら柔軟に対応していることを示唆している。このように、MEME を用いることで手順をどのように再利用して

いるかを詳細に追跡可能であり、運用者は過去の手順書を活用することで新規の作業に対して対応していることが確認できた。

5.2 運用者間の知識の共有

国立情報学研究所でのプライベートクラウド運用における運用者別の Notebook 再利用状況をまとめた (表 1)。

表 1 運用者別の Notebook 再利用状況(2017-2022)

Table 1 Reuse of notebooks by operators (2017-2022)

| 運用者 | 作成 Notebook 数 | 単一 Notebook 再利用率 (%) | 複数 Notebook 再利用率 (%) | 新規セル追加率 (%) |
|--------------|---------------|----------------------|----------------------|-------------|
| Operator1@B | 2452 | 28.2 | 55.4 | 16.3 |
| Operator2@B | 262 | 36.0 | 53.6 | 10.4 |
| Operator1@C | 899 | 45.3 | 48.9 | 5.7 |
| Developer1@D | 46 | 48.6 | 45.1 | 6.3 |
| Developer4@D | 552 | 48.7 | 46.0 | 5.3 |
| Developer2@D | 424 | 50.7 | 43.4 | 5.9 |
| Developer3@D | 546 | 52.2 | 38.0 | 9.8 |
| Operator2@A | 7090 | 63.1 | 28.3 | 8.6 |
| Operator1@A | 1980 | 66.6 | 21.3 | 12.0 |

作成 Notebook 数は運用開始からこれまでの累計 Notebook 数であり, Notebook 再利用率として単一 Notebook 再利用率, 複数 Notebook 再利用率, 新規セル追加率の 3 種の数値の平均を運用者ごとに算出した。

運用者の所属ごとに単一 Notebook 再利用率と複数 Notebook 再利用率に特徴が現れている。特に A 社メンバーは単一 Notebook 再利用率が多いが B 社メンバーは複数 Notebook 再利用率が高い。これは、A 社はライフサイクルが短い研究環境やイベント向けデータ分析環境を中心に担当しており、B 社は情報インフラ基盤部分を担当していることに由来しているためと考えられる。A 社は試行錯誤が可能なタイミングが多いため、多様性、複雑さのある程度抑えての業務の定型化が可能であるが、情報インフラ基盤部分は同一の環境を作るには、同形ハードウェアがあるとは言えそれなりの手間がかかり、結果的に試行錯誤が実施しづらく、業務の複雑さを抑えるような対処が困難であったものと考えられる。

次にセルの再利用情報に基づいてどの運用者がどの運用者のセルを再利用したかを集計し、運用者間のネットワークとして可視化した (図 5)。

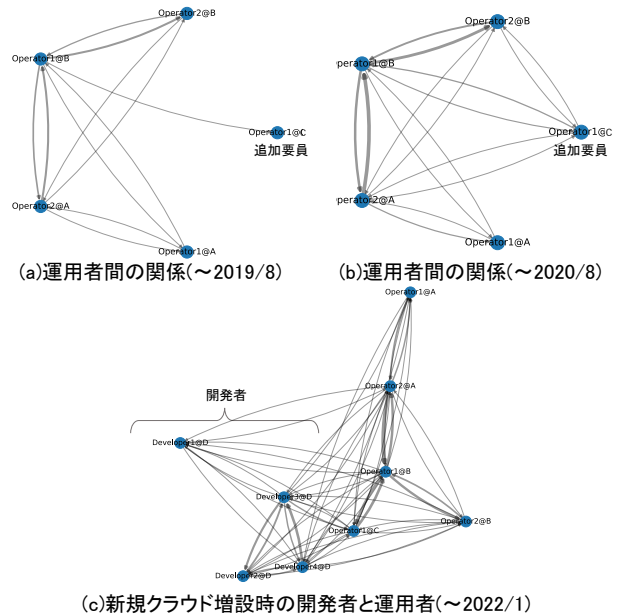


図 5 運用者-開発者のコミュニケーション可視化

Figure 5 Visualization of operator-developer communication

(a)及び(b)は新規運用者が参加した際のネットワークの変化を示したものである。(a)は新規運用者の参加直後であり、他の運用者との関係はほとんど見られないが、1年後(b)は業者 B の運用者 2 名の Notebook の再利用関係が構築されている。新規運用者は業者 B と同様に情報インフラ基盤を主に担当しているためこのような関係が構築されたものと思われる。また、(c)は最新のネットワーク構成であるが、新規クラウド構築時に参加した開発者 4 名が運用者とネットワークを構成していることがわかる。これは、クラウド構築時の成果物として Jupyter Notebook 形式の手順書を求めたため、構築前の検討作業から Notebook を使い作業をしていたためであると考えられる。このように、Jupyter Notebook を日常の運用業務に使用するだけでなく、開発や環境構築にも利用することで、一時的に入る関係者間の連携状況も把握することが可能なことが確認できた。

5.3 新規利用者への知識の伝播

ここまでは国立情報学研究所における事例であったが、国立情報学研究所で開発した CoursewareHub を室蘭工業大学にて利用することになったため、ここでも LC4RI を用いて環境構築・運用を実施した。室蘭工大側の教員は Jupyter Notebook を使用したことはあるが LC4RI のような情報インフラ運用への適用は未経験であった。室蘭工大での Notebook の作成状況とその由来を MEME によって分析したグラフを示す (図 6)。

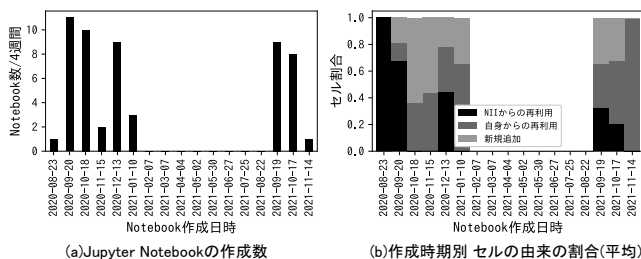


図 6 室蘭工大による Jupyter Notebook 再利用状況

Figure 6 Reuse of Jupyter Notebook by new user

Notebook は CoursewareHub の構築・運用での利用に限られているため、国立情報学研究所に比べ Notebook 作成頻度は低い。利用開始は全て国立情報学研究所(NII)から提供された Notebook の再利用から始まり、運用フェーズでは自分が作成したセルの再利用や、新規セル作成が多くなることが確認できた。これは、トラブルシュート時の状況調査など、定型的ではない作業が多かったものと考えられる。その1年後、再度講義を行う際にはNIIから新機能に対応した Notebook を提供し環境再構築を行なっている。そのため一部新規にNIIからのセルの再利用が表れているが、自身の Notebook の再利用も同じようにみられる。この傾向は新規運用者が LC4RI に慣れつつあり、作業に必要な Notebook を選び出し再利用できていることを示唆していると考えられる。

6. 議論

6.1 LC4RI における知識の転移

結果より、国立情報学研究所においては LC4RI として Jupyter Notebook を情報インフラ運用に適用し始めてから Notebook を用いた作業を定着させることができた。また、室蘭工業大学においては国立情報学研究所が提供した Notebook を利用するところからスタートし、独自の Notebook を作成しながらの作業へと変遷が生じていることが確認できた。LC4RI では Notebook としてコードを作成するが、従来の自動化アプローチとは異なり配備だけではなく障害の原因調査など不規則の作業にも利用することができる。そのため、状況ごとに固有の Notebook が作成され一貫したバージョン管理が困難であるという問題があるものの、情報インフラ運用における運用者の振る舞いを記録するツールとして有用であることを示すことができた。

また、それぞれの運用者・開発者の MEME の記録から、開発のようなバックグラウンドを持たない者も含め全ての運用者がそれぞれセルを組み合わせたたり、セルを追加したりしている状況を確認することができた。完全な自動化のデメリットとして、セキュリティパッチの自動適用場面において利用者がセキュリティシステムについて正しく理解できていないといった指摘[22]のような例が挙げられるが、

運用者全員が運用の中で試行錯誤を継続的に行うことで、システムを理解しながら運用できる可能性がある。また、外部のシステム開発業者に委託する場合でも、「Jupyter Notebook による手順書を納品物とする」という条件を課すことで、委託元が持つ Notebook をベースに外部業者が調査・開発し、委託元に手順が共有されるという状況を実現することができる。このように、LC4RI は開発者・運用者間のギャップを埋めることが可能であり、DevOps の思想を実現する方法論の一つであると言える。しかしながら、本報告では運用者・開発者間の知識共有の度合い・程度にのみ言及しており、各々の運用者がどのような内容の Notebook を共有しているのかの分析については今後の課題である。

6.2 MEME を用いた可視化の妥当性

MEME を用いることで運用者間の Notebook の共有状況を可視化することが可能であることを示したが、このような利用者間の関係性の可視化にはビジネスチャットの会話履歴を用いる方法[23]やメーリングリストの履歴を用いる方法がある。しかしながら、オープンソースソフトウェアの開発メーリングリストに対する解析[24]では、開発に関するトピックの全てがメーリングリストで議論されているわけではないため、課題リポジトリなどの他のチャンネルの包括的な分析が必要であることが示唆されている。また、コードのコミットログを用いた分析[25]ではコード自体のバージョンの変遷の分析が重視されており、開発者間のコミュニケーションに着目したものは少ない。nblineage のように Notebook ファイルを利用して運用者の振る舞いを可視化する方法は、自然言語によるコミュニケーションを分析する方法に比べ直接的に開発・運用行為に関係する振る舞いのみを包括的に抽出できる可能性がある。

一方で LC4RI ではソースコードリポジトリのコミットログのように明示的に履歴が保存されるわけではない。今回の分析方法ではファイル更新時刻 mtime をファイル利用日時とみなし、MEME が各運用者へどのように拡散したかを推定しているが、mtime を使う場合 Notebook 実行後に数日経ったのち参照、誤ってファイルを再保存することで本来のファイル利用時刻からずれてしまうおそれがある。このように、保存する MEME をコピーさせ伝播する方式では複数回再利用されるセルに対しては再利用順序を正確に追跡するには制限がある。そのため、MEME がコピーされるたびに付加情報を付与する枝番つき MEME[26]を実装した。現在この枝番つき MEME を記録するようにして MEME の変化を記録している。今後、枝番つき MEME による運用者間の変化の分析を進めていきたい。

7. まとめと今後の課題

我々は柔軟な情報インフラ運用を行うための方法論として LC4RI を考案し、国立情報学研究所と室蘭工業大学における研究環境・教育環境の情報インフラ運用での LC4RI の実践を通じて、運用者・開発者が互いにコミュニケーションを取りつつ Notebook を再利用しながら日々の運用作業を実施できることを、実際に運用に使われた Notebook ファイル群から示すことができた。従来の手順書に基づく運用ではこのような各運用者の貢献を可視化することは困難であるが、LC4RI が持つ試行錯誤を許容し証跡を残す運用と nblineage により記録される MEME により可視化を実現することができた。本報告で示した可視化により各運用者のさまざまな特徴が見えることがわかったため、今後は MEME による分析をどのようにチームにフィードバックするかを検討していきたい。例えば運用者によって Jupyter Notebook の再利用度が異なるため、ここから各業務がどれだけルーチン化できているか、各業務の負荷などを定量化できる可能性がある。これらのデータを活用しつつ、柔軟さを維持しながら情報インフラ運用をするためのノウハウ収集、共有を進めていきたい。

謝辞 国立情報学研究所のクラウド運用チームのメンバーには日々の運用で様々なチャレンジをしていただいた。ここに感謝する。

参考文献

- [1] Bass, L.J., Weber, I.M. and Zhu, L.: DevOps: A software architect's perspective, SEI Series in Software Engineering, Addison-Wesley, pp. I-XVIII,1-332 (2015).
- [2] Leite, L., Rocha, C., Kon, F., Milojicic, D. and Meirelles, P.: A Survey of DevOps Concepts and Challenges, ACM Comput. Surv. 52, 6, Article 127, pp. 35 (2019).
- [3] Wiedemann, A., Forsgren, N., Wiesche, M., Gewald, H., and Krömer, H.: Research for practice: the DevOps phenomenon, Commun. ACM 62, 8, 44-49 (2019).
- [4] Feitelson, D.G., Frachtenberg, E., and Beck, K.L.: Development and Deployment at Facebook, IEEE Internet Computing, 17, 8-17 (2013).
- [5] Christensen, H.B.: Teaching DevOps and cloud computing using a cognitive apprenticeship and storytelling approach, In Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITICSE'16). ACM, 174-179 (2016).
- [6] Kumara, I., Garriga, M., Romeu, A.U., Nucci, D.D., Palomba, F., Tamburri, D.A., and van den Heuvel, W.-J.: The do's and don'ts of infrastructure code: A systematic gray literature review, Information and Software Technology, Volume 137, 106593 (2021).
- [7] 政谷好伸, 谷沢智史, 横山重俊, 吉岡信和, 合田憲人: インフラ・コード化の実践における IPython notebook の適用, 信学技報, 115 巻, 72 号, 27-32 (2015).
- [8] Masatani, Y.: Collaboration and automated operation as literate computing for reproducible infrastructure, JupyterCon 2017, Available online: <https://conferences.oreilly.com/jupyter/jup-ny-2017/public/schedule/detail/59995.html> (accessed on 26 Jan 2022).
- [9] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., et al.: Jupyter Notebooks - A publishing format for reproducible computational workflows, Position. Power Acad. Publ. Play. Agent Agenda, 87-90 (2016).
- [10] 横山 重俊, 桑田 喜隆, 吉岡 信和: アカデミッククラウドアーキテクチャの提案と評価, 情報処理学会論文誌, 54 巻, 2 号, 688-698 (2013).
- [11] 長久勝, 政谷好伸, 合田憲人: Notebook による講義・演習環境の開発, 研究報告教育学習支援情報システム (CLE), 27 巻, 20 号, 1-4 (2019).
- [12] Red Hat: Ansible is Simple IT Automation, Available online: <https://www.ansible.com/> (accessed on 26 Jan 2022)
- [13] 独立行政法人情報処理推進機構: ソフトウェア等の脆弱性関連情報に関する届出状況[2021 年第 2 四半期 (4 月~6 月)], Available online: <https://www.ipa.go.jp/security/vuln/report/vuln2021q2.html> (accessed on 26 Jan 2022)
- [14] 長久 勝, 政谷 好伸, 谷沢 智史, 中川 晋吾, 合田 憲人: Notebook を介した作業ノウハウの継承・移転を分析するための基盤, 研究報告インターネットと運用技術 (IOT), 44 巻, 16 号, 1-6 (2019).
- [15] NII クラウド運用チーム: OperationHub, Available online: <https://github.com/NII-cloud-operation/OperationHub> (accessed on 26 Jan 2022)
- [16] Project jupyter team: JupyterHub, Available online: <https://jupyterhub.readthedocs.io/> (accessed on 26 Jan 2022)
- [17] NII クラウド運用チーム: LC4RI 全部入り Docker イメージ, Available online: https://github.com/NII-cloud-operation/Jupyter-LC_docker (accessed on 26 Jan 2022)
- [18] Verma, A., Pedrosa, L., Korupolu, M.R., Oppenheimer, D., Tune, E. and Wilkes, J.: Large-scale cluster management at Google with Borg, Proceedings of the European Conference on Computer Systems (EuroSys), ACM, Bordeaux, France (2015)
- [19] OpenStack project: OpenStack, Available online: <https://www.openstack.org/> (accessed on 26 Jan 2022)
- [20] Apache Software Foundation: Hadoop, Available online: <https://hadoop.apache.org/> (accessed on 26 Jan 2022)
- [21] 桑田 喜隆, 石坂 徹, 政谷 好伸, 長久 勝, 横山 重俊, 浜元 信州: クラウドを利用した対話的なプログラミング教育環境とその評価手法の提案, 人工知能学会第二種研究会資料, 2018 年 2018 巻 KSN-023 号 p. 05- (2018).
- [22] Wash, R., Rader, E., Vaniea, K., and Rizer, M.: Out of the Loop: How Automated Software Updates Cause Unintended Security Consequences. 10th Symposium On Usable Privacy and Security (SOUPS 2014), Menlo Park, CA, 89-104 (2014).
- [23] 野中 賢也, 山下 遥, 後藤 正幸: ビジネスチャットの会話履歴データに基づく社員間のネットワーク分析モデル, 人工知能学会全国大会論文集, JSAI2020 巻, 第 34 回, p. 114GS204 (2020). DOI:https://doi.org/10.11517/pjsai.JSAI2020.0_114GS204
- [24] Guzzi, A., Bacchelli, A., Lanza, M., Pinzger, M., and van Deursen, A.: Communication in open source software development mailing lists, In Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13). IEEE Press, 277-286(2013).
- [25] de F. Farias, M.A., Novais, R., Colaço, J. M., da S. Carvalho, L.P., Mendonça, M., and Spínola, R.O.: A systematic mapping study on mining software repositories, In Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16). Association for Computing Machinery, New York, NY, USA, 1472-1479.
- [26] Yazawa, S.: Sharing Notebook within Small Team, JupyterCon 2020, Available online: <https://cfp.jupytercon.com/2020/schedule/presentation/147/sharing-notebook-within-small-team/> (accessed on 26 Jan 2022)