

4 グリーンソフトウェア開発・運用支援

環境負荷低減に寄与する ソフトウェア開発技術の取り組み

開発・運用技術プロジェクトでは、ソフトウェアの開発から運用までライフサイクル全体を俯瞰したCO₂排出量の可視化・削減に取り組み、環境負荷低減に寄与するソフトウェア開発技術の実現を目指しています。今回、ソフトウェア業界におけるCO₂排出量の削減を推進する取り組みと、CO₂削減効果の観点で開発自動化を図る取り組みについて紹介する。

全体背景と取り組み

開発・運用技術プロジェクトでは、ビジネス進化につなげるシステム構築・運用技術の確立に向けて、ソフトウェア分野のカーボンニュートラルやNTTグループの環境エネルギービジョンの実現に向けて環境負荷低減に寄与するグリーンソフトウェア開発・運用技術の研究開発に取り組んでいる。

①CO₂排出量の可視化・削減：ソフトウェア業界におけるCO₂排出量の算定ルールの合意形成を図り、CO₂排出量の削減を推進する。

②開発自動化技術の高度化：「テスト効率化」「自動プログラミング」など開発自動化技術に対してCO₂削減効果の観点で高度化を図り、普及展開を推進する。

CO₂ 排出量の可視化と削減

近年、世界各国にて2050年までのカーボンニュートラル達成を目標に掲げ、企業活動による会社単位のCO₂排出量について算定および開示が強く求められている。更に、製品単位のCO₂排出量に着目しサプライチェーン排出量



NTTソフトウェアイノベーションセンタ
開発・運用技術プロジェクト
(左) グループリーダー 村本 達也氏
(右) グループリーダー 西澤 幸久氏

全体を把握し削減アクションを促す動きも活発になっている（参考文献(1)）。

一方、ソフトウェアについては、直接的にCO₂を排出しないため削減に向けた取り組みが遅れており、ソフトウェア開発・運用における排出量の実態は把握されてこなかった。当プロジェクトでは、ハードウェアのエネルギー効率がソフトウェアにより左右されること、さらに今後ソフトウェア製品に関するCO₂排出量の開示等が調達条件となり得るグローバル・産業界の動向を踏まえ、ソフトウェア開発時の

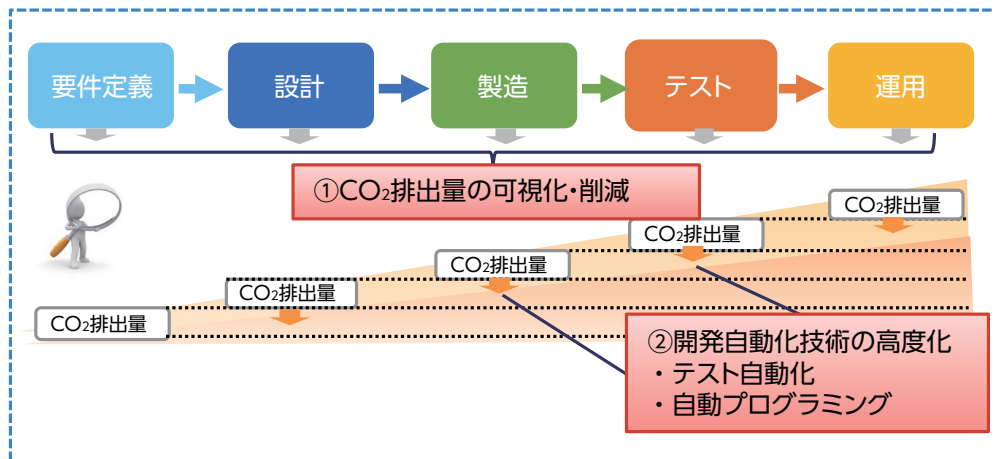


図1 グリーンソフトウェア開発・運用支援技術

CO₂ 排出量の可視化に取り組んでいる。

ソフトウェア開発に関する排出量算定方法には開発費用を基に算出する方法があるが、開発費用にはソフトウェア開発時に取り得る様々な削減努力を反映出来ない点が問題であり、ソフトウェア開発時に取り得る様々な削減努力を反映出来る排出量算定の実現が必要である。

これまで、ソフトウェア開発における CO₂ 排出量の実態把握と算定方式検討のため、ウォーターフォール型の開発手法 / オンプレミスな開発環境を利用した開発プロジェクトにおいて、IT 機器等の消費電力計測と開発者の作業ログを取得し、CO₂ 排出量を算定・可視化する手法を確立した。本手法は、ツール整備などによる手順効率化を図り、ソフトウェア開発のプロジェクトリーダーや開発メンバが自らの開発活動による CO₂ 排出量を算定し削減努力を定量的に評価するためのガイドラインとして展開することを目指している。

開発自動化の高度化による 開発時の排出量削減

ソフトウェアにおける CO₂ 排出量に関しては、製造期間の排出量も無視できない。SIC はソフトウェア開発の各工程の作業効率を向上させることによって工期を短縮、または、携わる人数を削減することにより排出量の低減に貢献する研究を実施している。

具体的な研究内容の1つとして、ソフトウェア開発の全体工数の約1/3を占めるテスト工程の抜本的な稼働削減に挑戦し、フィードバック

指向テスト技術「LatteArt」(参考文献(2)、(3))を開発している。この技術はあらかじめ詳細なテスト項目を作成することなくテスト対象を探索的に検証することを特長としており、テスト項目作成に掛かる工数を大幅に削減することができる。また、LatteArtはテストを実施する際のテスターの操作を蓄積・分析する機能を有しており、不具合検出効率が高く、回帰テスト用の自動化スクリプトの保守が容易という特長も有している。本技術はオープンソース化しておりコミュニティサイトから誰でもダウンロードして利用できるので是非ソフトウェアテストの稼働削減に活用していただきたい(参考リンク(4))。

次の研究内容として、人間ならではの創作工程であるプログラムコード生成を人間の代わりに機械が自動的に実施する研究を進めている(参考文献(5)、(6))。一般的にプログラミングを行う際は要件を決めてその内容を段階的に詳細化しながら設計して、設計内容に沿ったプログラムコードを生成するといった手順をとることが多いと思われるが、我々は具体的に要件を定義することなく機械と対話的にやり取りを行うことでプログラムコードを生成することにチャレンジしている。実験では複雑なデータの加工を実施するプログラムを生成する際に、加工結果を機械に例示することで機械がプログラムと加工結果を出力し、それに対して人が間違いを指摘することを繰り返すことで正解プログラムを生成できることを確認している。

今後

今後、①の取り組みについては、算定・可視化技術の適用領域拡大と実案件適用による基礎データ収集を進め、NTTグループ算定モデルの確立を目指すと共に、ソフトウェア業界における開発時のCO₂排出量算定ルールの合意形成にむけ、製品カテゴリルールのコミュニティ提案や仲間作りを推進する。更には、ソフトウェア運用時の排出量可視化や削減に寄与するソフトウェア開発・運用技術の確立を目指していく。

また、②の取り組みについては、実際のシステム・サービスを実装する大規模で複雑なプロミシングを人と機械が対話しながら完成させる世界を目指していく。

これらにより、環境負荷低減へ寄与していく。

参考文献：

- (1) https://www.meti.go.jp/shingikai/energy_environment/carbon_footprint/index.html
- (2) Hiroyuki Kirinuki, Masaki Tajima, Haruto Tanno: "LatteArt: A Platform for Recording and Analyzing Exploratory Testing", 16th IEEE International Conference on Software Testing, Verification and Validation (ICST2023) [Testing Tools],
- (3) 丹野 治門, 切貫 弘之, 川口 敬宏, 但馬 将貴, 生沼 守英, 村本 達也: "繰り返し型の効率的なテストを実現するテスト活動データ分析技術", NTT技術ジャーナル, 2022, Vol 34, No.2, pp.18-21, 2022年2月
- (4) <https://github.com/latteart-org/latteart/>
- (5) 倉林 利行, 丹野 治門, 切貫 弘之, 大島 剛志: "プログラム合成技術の動向調査", コンピュータソフトウェア, 2022, to appear.
- (6) 倉林 利行, 丹野 治門: "非エンジニア向けのProgramming by Examplesによるデータ加工支援の試み", ソフトウェア工学の基礎 29 日本ソフトウェア科学会FOSE 2022 (レクチャーノート/ソフトウェア学) pp. 197-198, 2022年11月