

---

# ソフトウェア・デファインド・ウォーフェア

## Software-Defined Warfare

### *DOD のデジタル時代への移行を設計する*

*Architecting the DOD's Transition to the Digital Age*

CSIS 戦略的技術プログラムの報告書

A Report of the CSIS Strategic Technologies Program

2022年9月

ナンド・ムルチャンドニ      ジョン・N・T・「ジャック」・シャナハン

Nand Mulchandani

米国中央情報局 (CIA)

初代最高技術責任者 (CTO)

John N.T. "Jack" Shanahan

米国空軍中將 (退役)

訳：木村 初夫

株式会社エヌ・エス・アール 上級研究員

**CSIS** 戦略国際問題研究所  
Center for Strategic & International Studies

---

#### About Japanese translation right

We are happy to grant one-time, nonexclusive permission for the requested use. This permission is granted without charge, but with the stipulation that the author, title, publisher, and date of our material be cited in a customary manner in all reuses/reprints of the work.

© 2022 by the Center for Strategic and International Studies.

## はじめに

2011年8月20日、マーク・アンドリーセン (Marc Andreessen) は「Why Software Is Eating the World」(なぜソフトウェアは世界を食い尽くしているのか)<sup>1)</sup>を発表した。この重要な論文が書かれてから10年、テクノロジー業界では永遠ともいえるこの期間に、動きの鈍い企業を追いやる新世代の「デジタルネイティブ」企業が出現した。テクノロジーは世界を食い尽くし、また自身をも食い尽くしている。

米軍は世界で最も優れた軍隊と考えられている。国防総省 (DOD) は、二位以下10ヶ国の軍隊の合計をしのご巨額の予算で、競争国を圧倒し、優れた装備を持ち、訓練を行っている。また、DOD は、戦車、艦船、航空機、その他あらゆるものを含む「最も大きく、最も多く、最高の」軍事資本投資を行う、工業化時代のハードウェア中心の組織である。残念ながら、今日の世界では、ハードウェアは「旧式」となった存在であり、利益率の低い商品として製造、保管、輸送、消費されている。減価償却後、あるいは一定以上の損耗が起きると、ハードウェアは部品としてスクラップされるか、あるいは廃棄される。

軍のハードウェアシステムは、銃撃や極度のストレスにさらされながら機器類を操作し、パワーポイントやエクセル画面で情報を処理し、計画や命令を発する、脆弱な人間の集団の周りを、何重もの金属と保護で包み込むように作られている。彼らは、驚くべきことを成し遂げているが、DOD が多くの点で優れている反面で、他のほとんどの連邦機関と同様に、バックオフィス機能の自動化から、オペレーターへのデジタル戦闘サービスの提供まであらゆる点で、商業ソフトウェア業界の最先端技術に大きく遅れをとっている共通の特徴を抱えている。

通常であれば、DOD のような組織は、起業家や投資家による「ディスラプション」(「廃業」の婉曲表現)の巨大なターゲットとなるはずである。DOD は、国の戦闘機能を独占しているため、幸い通常の市場競争とは無縁である。しかし、DOD が新しい技術を採用せず、レガシーとなったワークフローやプロセスを変更せず、新しい作戦構想による設計や実験を行わなかった場合、この独占状態こそが最悪の問題の多くの根本原因でもある。米国政府内で、DOD は、資本主義的で厳しいダーウィンの旅を経験することはない。ハングリーな新興スタートアップと対決し、こうしたスタートアップがまったく新しい技術で既存の組織にとって変わり、「新 DOD」となるようリスクに直面することはありえない。同様に問題なのは、巨大な DOD の官僚機構が、商業産業の成功を後押しする定期的な「技術革新」をすすめるににくいことである。DOD は米国経済内の市場競争からは隔離されているが、政府外で起きている革命的、長期的、広範囲な技術的变化から免れることはできない。

---

1) Marc Andreessen, “Why Software Is Eating the World,” *Wall Street Journal*, August 20, 2011, <https://www.wsj.com/articles/SB10001424053111903480904576512250915629460>.

また、世界中の他の軍隊との競争の脅威からも免れられない。DOD が自ら変わるか、あるいは競争国から強いられる形で変化が起きるか、いずれかであろう。

米国が将来にわたって支配的な地位を維持するためには（保証された結果ではないが）、DOD では、はるかに柔軟で、必要に応じて拡張でき、状況の変化に動的に適応できる新しい設計とアーキテクチャを必要としている。これらは重要なサービスであるため、劇的に低いコストで、これらを実現する必要がある、情報および意思決定のループと、エンドポイントにおける人間の役割を変える必要がある。つまり、速度、精度、規模、技術を高め、そして、敵対者を混乱させ、認識不能にし、圧倒するとともに、敵対者が米軍に対して同じことができないようにする。DOD のシステムは、飛躍的に速い意思決定と実行速度をサポートし、システムの迅速な更新と修正を可能とし、システムの構築と配備のコスト構造を下げ、そして、新しい機能の提供の限界費用と速度を向上させる必要がある。

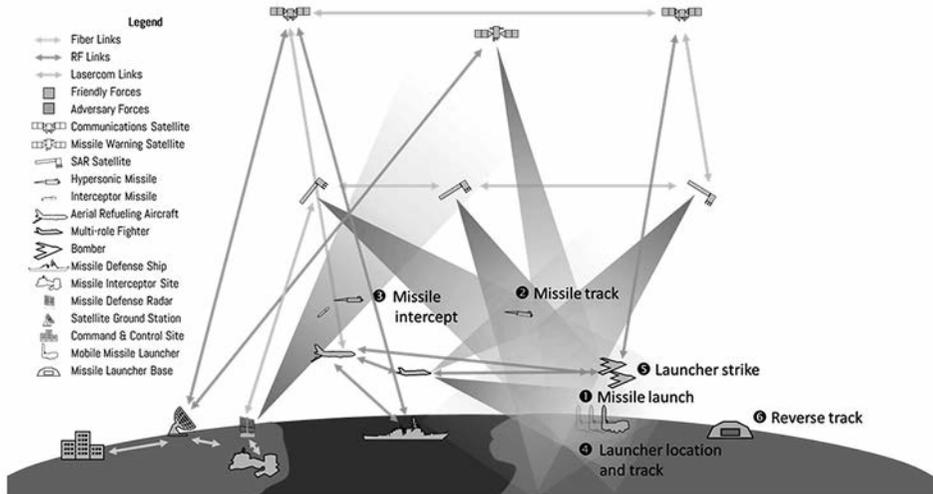
DOD の戦闘システムの構築方法を劇的に変えなければならないという発想は、まったく目新しいものではない。これまでも説得力のある事例を持って、抜本的な変革を訴える論文やシンクタンクの論評、書籍は何百と出ている。最近の代表的な例としては、クリス・ブロス（Chris Brose）の『キルチェーン』がある。彼は、DOD に対して、「自律システムの大規模ネットワーク」と、異なる兵器システムの組み合わせを含む「軍事のモノのインターネット（MIoT）」を構築するよう提言している<sup>2)</sup>。同様に、ボブ・ワーク（Bob Work）前国防副長官は、第三のオフセット戦略の概念を構築する上で、DOD 全体を大きく変える必要性、特に、自律システムや AI 対応能力の広範な採用を長年主張してきた<sup>3)</sup>。その他、国防政策界では定期的に、ネットワーク戦、モザイク戦、次世代戦闘ネットワークといった概念を中心としたアイデアが提唱され浸透している。このような研究の多くは、問題の説明はできたものの、具体的な技術的解決策を提案するレベルには至っていない。極めて困難な技術的問題を解決する場合、商業技術のリーディングカンパニーは、通常、システム全体を構築するための一連の主要なアーキテクチャ標準の概要を示すことから始める。本論文では、次世代戦闘システムに必要なアーキテクチャの概要を示す。

このアーキテクチャの鍵は、ソフトウェアにある。ソフトウェアは、産業全体を破壊し、再構築する。特に手作業に依存し、限界費用が低いかゼロであるということを活用できない産業は、ソフトウェアに破壊される。これは、企業や業務モデルを「魔法のソフトウェアの粉」で成功させるということではなく、どのような業務でも、永続的な競争優位を得ることを望むなら、すべての業務

---

2) Christian Brose, *The Kill Chain: Defending America in the Future of High-Tech Warfare* (New York: Hachette Books, 2020).

3) Bob Work, "Remarks by Deputy Secretary Work on Third Offset Strategy," U.S. Department of Defense, speech delivered April 28, 2016, <https://www.defense.gov/News/Speeches/Speech/Article/753482/remarks-by-deputy-secretary-work-on-third-offset-strategy/>.



出典：Todd Harrison, “Battle Networks and the Future Force: Part 2: Operational Challenges and Acquisition Opportunities,” CSIS, CSIS Briefs, November 3, 2021, [https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/211103\\_Harrison\\_Battle\\_Networks\\_Part2\\_0.pdf?vsuBpGNyDDOwNE\\_hMzckmGEfb8fq13dx](https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/211103_Harrison_Battle_Networks_Part2_0.pdf?vsuBpGNyDDOwNE_hMzckmGEfb8fq13dx).

と業務モデルの中核はソフトウェアである必要があるのだ。DOD も同じである。

## 第1章 キルチェーンを閉じ、敵対者のOODA ループ内に入ること

「キルチェーン」とは、ブросの著書よりも古くからある言葉だ。情報を吸収して知識（行動可能インテリジェンス）に変え、意思決定し、決定に基づき行動し、決定の結果を理解し、適切に今後の行動を修正する多段階のプロセスを意味する。同様に、ジョン・ボイド (John Boyd) の「OODA ループ」の概念は、個人、指揮官、またはチームが、観察、状況判断、意思決定、および行動という同じ段階を経て進行するプロセスを指している<sup>4)</sup>。プロセスの最終段階である「行動する」または「キルチェーンを閉じる」は、一般に何かを発射して破壊することを意味するが、今日の世界では、標的に対するサイバー攻撃を実行する、OODA ループの敵の観察段階を混乱させ、自軍を守るため、誤情報を流布するといった、非運動エネルギー効果も含むことがある。一般に、OODA ループを短く、緊密に、かつ強靱に一貫して示すことができる側が優位に立つ。

結局のところ、キルチェーンやOODA ループは、人間の認識力に本質的に支えられたワークフローである。今日、これらのワークフローは、手動、半手動(電話を含む)、および電子プロセスの混乱した寄せ集めとして実装されている。戦闘指揮所の作戦センターの様子は、刺激的であると同時に衝撃的でもある。平時には、これらの司令部とその隷下部隊が、比較的限られた数の航空機や

4) Frans Osinga, *Science, Strategy and War: The Strategic Theory of John Boyd* (Delft: Eburon Academic Publishers, 2005), [http://www.projectwhitehorse.com/pdfs/ScienceStrategyWar\\_Osinga.pdf](http://www.projectwhitehorse.com/pdfs/ScienceStrategyWar_Osinga.pdf)

艦船の動きの処理を見るのは容易である。しかし、戦時には、同じ既存の手動プロセスで、数千とはいかないまでも、数百の動く物体、つまり、海底から宇宙、サイバー空間、そしてその間にあるあらゆる場所で、運動エネルギー効果と非運動エネルギー効果の任務を負ったすべてを追尾する場合、どのように機能しているかは想像に難くない。意思決定が依然として人間の指揮官やオペレーターの管理下であっても、プロセスの各ステップとデータの流れからなる、これらのワークフローをソフトウェアで実行することにより、敵対者に対して決定的な優位性をもたらすことができる。OODA ループのすべての段階で、より緊密に、より高速に、より十分な情報を得ることができるようになる。逆に未来の戦場において、デジタル時代のソフトウェア中心のアプローチを取らなければ、戦術的、作戦的、さらには戦略的な敗北をさえ意味する可能性が非常に高い。



出典：U.S. Marine Corps photo, licensed under CC BY 2.0

このような「ソフトウェア応援論」は、ハードウェアの終焉を意味するものと曲解されやすいが、それはまったく的外れだ。軍用ハードウェアと兵器システムは、依然として極めて重要であるが、インターネット規模のソフトウェアシステムの末端に、意思決定、ターゲティング、資源調達複雑さをすべて処理する、つまり、キルチェーンを閉じる意思決定を実行する最終段階を除いたすべてをつなぎ止めておく必要があるのだ。ソフトウェアは、望ましい効果をもたらすプロセスをより速く、より安く、より効率的に、より拡張可能に、そしてより正確にすることで、ハードウェアの価値を根本的かつ回復不能なまでに変化させるだろう。

## 第2章 データセンターアーキテクチャからの教訓

他の領域からの教訓や経験を移植することは、DOD や他の連邦政府組織における大きな問題に対する画期的な解決策を導き出すために役立つ。最小のコストで最大の性能を引き出し、適切な場合には、コンピュータソフトウェアとシステムの設計でプロセスとコスト構造の中で最も高価で拡張性がなく、遅い部分を取り除く方法に関する最高の原則と実践が、長年にわたる大規模なソフトウェアシステムの構築で明らかになった。これらの同じコンセプトは、DOD にも適用でき、またそうすべきである。

コンピューター業界では、「ソフトウェア定義」(Software-Defined) とは、いくつかの異なる中核的な設計上の決定を促す広範なアーキテクチャの概念である。このような設計上の決定により、

バラバラに配置されたハードウェア製品の束が、単一のプラットフォームとして運用・管理できる統合された統一体になる。また、通常至る所に分散している制御と複雑性を、簡素化、管理、拡張が可能な場所に集中化する。軍事用語で言えば、これは中央集権的な指示と分散的な実行であり、両者の長所を併せ持っているのである。

テクノロジー産業で使用されている大規模なコンピューティングシステムやプラットフォームのほとんどは、ソフトウェア定義である。Amazon Web Services(AWS)やMicrosoft Azure といった最大規模のクラウドコンピューティングプラットフォームは、いずれも「ソフトウェア定義ネットワーク」(SDN) に依存する「ソフトウェア定義コンピューティングプラットフォーム」である、また、第5世代無線通信システム(5G)でも、プロビジョニングとルーティングがすべてソフトウェアで行われるなど、今やストレージも「ソフトウェア定義」となっている。これらの大規模なプラットフォームは、コモディティハードウェア(場合によっては自社製)上で動作し、大規模なオーケストレーションソフトウェアによって、プラットフォームの下層から中央のシステムに複雑さを移している。

今日のコンピューターシステムで、戦場の規模と大きさに対応できるかどうかという疑問は、妥当であるが、その答えは「イエス」である。Apple、Meta、Amazon、Alphabet、Microsoft、Netflix といったインターネット規模の企業は、毎日何十億ものユーザーとシステムを日常的に接続し、コンテンツやビデオをリアルタイムでやり取りしている。これらの企業では、通常データセンターが物理破壊されるリスクに直面することはないものの、サイバー犯罪者や、国家によるサイバー攻撃に、さらにはリスやネズミのケーブル線の食い破りといった問題にも対処を行っている。これらの優れたアーキテクチャは、障害が発生した場合にも強靱であるように構築されている。これらの企業がこのような巨大なスケールでシステムを運用できるのであれば、同じキーコンセプトを戦闘に適用し、何千ものハードウェアオブジェクトを一つのソフトウェア定義システムにまとめられない理由はまったくない。

### コンセプト：スケーリングのための設計

初期のコンピューターシステムとソフトウェアは、一つのコンピューターシステムで一つのアプリケーションが動作するように設計されていた。コンピューターシステムの容量が足りなくなった場合、より高速なプロセッサ、大きなディスクスペース、高速なネットワークカードなど、より多くの資源を問題に投入することが解決策となった。この方法は、「垂直スケーリング」と呼ばれる。システムに追加できる資源には、物理的な限界があるため、サーバーの容量が一杯になった時点で終わりであったが、このような進化を経て、業界にとって最も大きなシステムの一つであるメインフレームにつながった。

本論文の著者の一人は、大学を卒業してネットワークコンピューティングに革命を起こしていた

サン・マイクロシステムズという注目の新興企業に最初に就職した。サンのコンピューターは、見ただけから「ピザボックス」と呼ばれる、巨大なメインフレームに対抗するものだった。メインフレームの「スケーリング単位」である別のメインフレームではなく、アプリケーションの資源が限界に達するたびに、「ピザボックス」1個分のコンピューターとメモリを追加するだけで、スケールアップができたのである。このような限界費用の選択肢は、何百万ドルもする巨大なコンピューターをもう1台購入することを強いられるよりも、はるかに魅力的である。このような、小規模で限界的な追加を行う方法は「水平スケーリング」と呼ばれ、垂直スケーリングの限界を克服するために生み出されたものである。

水平スケーリングでは、単一の大型コンピューターシステムでアプリケーションを実行するのではなく、アプリケーションの計算負荷を複数の小型コンピューターシステムに分散させる。残念ながら、垂直方向に拡張されたアーキテクチャから水平方向に拡張されたアーキテクチャへの移行は、無償で行えるものではなく、この新しいシステムトポロジーをサポートするため、ソフトウェア自体を再設計する必要が生じる。データベースストレージ(複数のシステムへの分散は困難)、ファイルシステム、キャッシュとデータの一貫性、ソフトウェアキュー、ロードバランサーといった問題が出てくる。企業は、レガシーアプリケーションを、オンプレミスのデータセンターからクラウドに移行する際、クラウドコンピューティングを最大限に活用するために、アプリケーションをリファクタリング(訳注:プログラムの外部から見た動作・機能・仕様を変えずに、保守性・可読性を高めるためにソースコードの内部構造を整理すること)したり、書き直したりしなければならないことに気付いた。

古いデータセンターのコンピューターシステムは「ペット」と呼ばれた。これは、各コンピューターシステムが固有の名前を持っており、通常より広い命名スキームの一部であったためである。一般的な命名スキームは、太陽系の惑星や、飼い猫によくある名前のグループ(たとえば、「Fluffy」)であった。コンピューターに問題があれば、エンジニアは、ペットが病気になると獣医に診せるように、必死でシステムを直そうとした。これに対して、現在の大手クラウドベンダーで稼働しているコンピューターは、ほぼ意味のない文字列(たとえば、「i-0123456789abcdef」)が名前であり、残念ながら匿名の家畜と同じである。クラウドコンピューティングのインスタンスは、常に生成と破壊が繰り返されており、ほとんど感傷的な価値はない。

これを米国の戦闘システムのアーキテクチャに当てはめることは簡単である。DODは、安価で、使い捨ての、製造が容易なエンドポイントシステムを大量に必要としている。DODはその集中、分配、スケールアップ、スケールダウンを必要に応じて行うことができる。多くの場合、これらのシステムはほとんど、あるいはまったく保守・整備を必要としない。故障しても、同一のクローン機と交換するだけでよいため、後方支援と保守・整備の全体の複雑さを軽減することができる。DODの一部はすでにこのコンセプトについて検討中だが(たとえば、小型無人機)、DODは、これから、

このアーキテクチャの考え方を、構築または購入するハードウェア、車両、兵器システムの多くの側面に適用する必要がある。

### コンセプト：単一障害点の排除

現在の米軍の設計ポイントは、ハードウェア中心であり、人間中心であり、大型である。大きさは重要であり、巨大な空母は、このパラダイムを体現している。何十億ドルもの建造費をかけ、何千人もの水兵を乗せ、何十種類もの戦闘機を搭載している。空母打撃群（CSG）の中央には、空母が位置し、多くの艦船や潜水艦が、この空母を護衛する役割を担う。CSG が提供する護衛機能により、米国の空母を撃沈するのは非常に困難である。しかし、空母が撃沈されてしまった場合、CSG の構造全体が破壊され、CSG の設計意図である世界中への戦力投射が、ほとんど不可能になる。

空母は、その大きさとフットプリントから、非常に大きなメインフレームまたは大規模なデータベースシステムに似ている。空母は、垂直スケーリングであり、システムのスケーリング単位は、別の空母となる。これらの垂直スケーリングのシステムの多くは、単一障害点でもある。このためシステムは非常に高価であり、通常、二つのシステムを同時に稼働させることによる冗長性はない。メインフレームあるいはデータベースシステムがクラッシュすると、バックアップがないため、システム全体がクラッシュしてしまう。このような単一障害点を持つアーキテクチャは、明らかに非常に脆く、最終的には故障してしまう。

このような単一障害点を特定し、排除することは、民間企業においても DOD においても、強靱性のあるアーキテクチャを構築する上で非常に重要である。空母と、CSG のコンセプトは、中国が空母キラーのミサイルを開発するより前の時期に作られたものである。一発1,000万ドルのミサイルを数発発射し、100億ドル以上の価値を持つ米国の空母を攻撃できるなら、目を見張るような投資収益率 (ROI) が得られるだろう。あるいは、ウクライナでの戦闘が示唆するように、小型で攻撃力の高い、自律武装ドローン群数百機（数千機はないとしても）で CSG を攻撃すれば、さらに素晴らしい ROI を得ることができよう。

### コンセプト：仮想化

仮想化技術は、コンピューターサーバーをコピー、クローン化、スナップショット、移動が可能な仮想マシンに変換することで、データセンターに革命をもたらした。仮想化技術によって、コンピューターを分単位でレンタルし、非常に安価にリモートで操作できるようになり、クラウドコンピューティングと「サービスとしてのソフトウェア」の革命が起きた。

仮想化とは、ハードウェアとソフトウェアのような二つの異なるシステム間に抽象化と変換層を作成すること、または古いシステムをより大きなデジタルワークフローの一部として操作し管理できるようにする仕組みであり、当初、仮想化の最も一般的な特徴は、Windows 2000 といった古い

Windows OS を、新しいハードウェアでも実行できることと、システムを中央データセンターに統合できることであった。

レガシーの軍用のプラットフォーム、センサーとワークフローの場合では、システムの仮想化に、これらを手動プロセスの中断なしに、あらゆる電子ワークフローに参加できる、一連の呼び出し可能なアプリケーションプログラミングインターフェース (API) で包むことが必要である。API とは、プログラマーがデータを受け取り、処理を開始するためにコードから呼び出すことができる、明確に定義された、十分に文書化されたソフトウェアのエントリポイントを持つ標準化された、ソフトウェア層である。キルチェーンがエンドツーエンドの電子ワークフローに変換されると、このワークフローの終端にある兵器システムの結果は、API コールとして実装される。理想的には、これらの API は双方向であるべきである。つまり、システムの制御と実行を可能とするインバウンドのもの、状態、容量、使用統計、そして計画、再供給、資源管理のためにオーケストレーションシステムにフィードバックできるものを提供するアウトバウンドのものである。システムが一部の API (たとえば、状態と供給レベル情報を提供するアウトバウンド API) しか使用できないとしても、その計算がより広範なオーケストレーションシステムにネットワーク化できるため、実質的な改善となる。



出典：U.S. Navy photo by Mass Communication Specialist 2nd Class Michael H. Lehman/Released

この機能は、デモンストレーションまたは実験で一度使用するだけでなく、生産現場や現場に移植する必要がある。個々のハードウェアシステムや単発のソフトウェアプログラムに依存するものから、DOD の考え方を、API の役割を中心としたシステムの使用に変えることの重要性は、どれほど強調してもし過ぎることはない。

仮想化の第二の側面は、エンドポイントハードウェアをセットアップしたうえですべてが論理的に互いに完全に分離され、かつ基盤となるハードウェアプラットフォームのみを共有する、複数の

以前の DOD の統合全領域指揮統制 (Joint All-Domain Command and Control [JADC2]) のデモンストレーションの際、米軍は API ベースの兵器システムへのアクセスを実証した。これは重要な第一歩である。事実上、このことは、これらのシステムを電子キルチェーンに統合し、全体的ソフトウェアオーケストレーションエンジンの管理下に置く可能性があることを意味する。しかし、こ

ソフトウェアスタックを実行させることである。固定的で一時的なハードウェアコストが支配的であるケースは数多くある。たとえば、時間帯に応じて、まったく異なる負荷スタックを動的に再プログラムまたは切り替えることができれば、ハードウェアの使用率を高めることができる。小さな例では、一つのデバイスで複数の電話番号（SIM カードに紐づく）を使える携帯電話がある。より複雑で高価な例としては衛星がある。衛星は、システムの構築と打ち上げの初期コストが非常に高いことも多い一方で、運用にかかる変動コストはかなり予測しやすいものである。このような高価なシステムで最大限の効用を引き出すことが、投資が利益と損失のどちらを生むかの分かれ目となる。さらに、衛星コンステレーションを、複数の動的な負荷を展開できる「コンピューティンググリッド」として考えれば、衛星が固定的な機能を持つ従来のパラダイムを根本的に変革でき、代わりに柔軟性のあるクラウドコンピューティングシステムと同じように使用することで、実際の「クラウドコンピューティング」を実現できるのである。

### コンセプト：低価格コモディティハードウェアとステートレスエンドポイント

映画『トップガン』のオリジナルでは、ピート・ミッチェル (Pete Mitchell) の飛行機の「MAVERICK」という刻印を見逃すわけにはいかない。それは非常にクールであるが、故障したら何が起るのかという疑問が湧いてくればそうともいってられない。エド・ハリス (Ed Harris) (最新作の提督役) のこんなセリフが思い浮かぶ。「マーベリック、悪い知らせだ。おまえの飛行機は保守・整備で停止する。いい知らせは、アイスマンが同意すれば、アイスマンのジェットを使えることだ」。マーベリックにとっては嬉しいことかもしれないが、彼はアイスマンと対決しなければならない。アイスマンというのは、ジェットを操縦するチャンスをあきらめるくらいなら死を選ぶような人物であり、マーベリックが座席設定、芳香剤、無線のプリセットを弄ることを快く思わないだろう。

マーベリックが自分の名前がついた飛行機以外飛ばせないというこの考え方は、「ペットか家畜か」のシステムの概念に立ち戻るものである。1台のハードウェアは1人のパイロットと結びつき、「ジャストインタイム」のソフトウェアの展開でパイロットを別のハードウェアに動的に切り替えることはできない。垂直スケーリングから水平スケーリングへのアーキテクチャの転換の核となるのは、実際のエンドポイントシステムをステートレス（専有にしない）かつ使い捨てにし、ジャストインタイムのソフトウェアの展開と構成の可能性を導入することでもある。このアプローチの利点としては、時間の経過とともに劇的に低価格化するハードウェアに移行できる（つまり、コモディティ化曲線に乗る）、必要に応じてスケールアップ/ダウンできる、新しいシステムを迅速に用意できる、緊急時にボトルネックを回避できる（問題が発生しても、新しいシステムに移行し後で修正すればよい）、そしてエンドポイントシステムには常に最新のソフトウェアが搭載され、状態が保持されないため、新しいセキュリティ上の利点があることなどが挙げられる。データはシステムソフトウェアやアプリケーションソフトウェアから明確に分離され、クラウドに保管され、ローカル処理のために必要に応じてエンドポイントに同期される。

これらの明らかな利点に加え、DODは、人工知能(AI)のような新しい技術を既存のシステムに統合する方法と、新しいシステムを一から異なる方法で設計する方法についても検討を始める必要がある。現在、DODは「グローバルな」ハードウェアプラットフォーム、たとえば無人航空機(UAV)を複数の責任範囲(AOR)にわたって配備することを検討している。AI以前の時代には、これは大きな問題ではなかったが、DODがこれらのシステムにAIベースのソフトウェアをより多く組み込むようになると、各AORのアルゴリズムが学習するデータが、大きく異なる可能性がある。たとえば、雪原やぬかるみのある東欧と、開けた海域と散在する島々のある南シナ海では、データが大きく異なる可能性がある。AIソフトは、作戦現場で収集した新しいデータから再訓練することで、最適な性能を発揮する。デジタル時代には、実戦配備されたAIモデルを定期的に更新しなければならない。CI/CD(継続的インテグレーション/継続的デリバリー)は商業産業ではかなり前から日常化されていたものであるが、DODでは、Project MavenとDOD統合AIセンター(Joint AI Center)以外には、このアプローチにより実戦配備されたAIソフトウェアの運用例はほとんど存在しない。実戦配備されたAIソフトウェアは、決して完成せず、継続的な更新が必要となる考え方は、少数派の発想ではない。ある時点で、ほとんどすべてのソフトウェアがAI対応となり、ハードウェアシステムに組み込まれる場合は、継続的な、さらには無線によるリモート更新をできるように設計する必要がある。国防総省がこれを大規模に運用するようになると、任務に合わせてカスタマイズされたローカルシステムにジャストインタイムのソフトウェアをよりスマートに配備する必要がある。

同様に、ひとたび危機や紛争が始まれば、実戦配備されたシステムのバグや問題は迅速に修正が必要になる。これは、商業製品が「最低限実現可能な製品」から実証済みの高性能な生産製品へと急速に移行するときと同じである。ソフトウェアのパッチや「ドット・リリース」を回し、実戦配備されたハードウェアシステムに展開する能力は、将来の紛争では、重要な差別化要因になるだろう。AppleがmacOS®、iOS®(iPhone用)、iPadOS®といったオペレーティングシステムを実行するように、DODが、TankOS®、FighterOS®、ShipOS®といった個別のハードウェアシステムを実行することを想定しておく必要がある。各軍は、それぞれのOSに無線でパッチや新しいアプリケーションを配信し、機能改善やバグフィックスを行えるはずである。どのような戦争でも、最初の段階では、米国の敵は、兵器システムの脆弱性に対して、多くの「ゼロデイ」サイバー攻撃を仕掛けてくることは間違いない。米国が戦いに復帰できるかどうかは、DODがこれらの脆弱性にパッチを当て、エンドポイントをクリーンアップし、システムを迅速に更新し再び起動できるかどうかにかかっている。DODがジャストインタイムのソフトウェア配備モデルを採用する場合、このループを閉じるために必要な時間は、ステートフルなソフトウェア/ハードウェアシステムからマルウェアを一掃しその駆除を確認するよりも、劇的に時間が短くなる。

## コンセプト：計装設計

誰でも自分の携帯電話でハンバーガーやピザの配達状況をリアルタイムに追跡できるのだから、

国防総省の戦闘指揮官が管理下にあるエンドポイントシステムをリアルタイムに可視化できないとすれば、理解しがたい話である。今日、使用されているほぼすべての製品は、エンドポイントシステムのリアルタイム制御、エンドポイントの状態、(開発側では)エンドユーザーがどのような機能を活用しているかの把握など、さまざまなデータを測定するために高度に計装化が行われている。使用状況を計測・追跡する優れた製品は何百とあり、DODのエンドポイントハードウェアシステムは、当初から計測機能を組み込んで設計・構築する必要がある(DODにおけるユーザーインターフェース(UI)とユーザーエクスペリエンス(UX)の評価は、本論文の範囲外である。残念ながら、UI/UXのベストプラクティスは、ほとんどのDODハードウェアおよびソフトウェアシステム開発者にとって異質な概念のままである)。

最も重要な設計要素とは、リモート制御やオーケストレーションに関連する可能性のあるシステムのすべての部分と、システム全体の運用に重要なその他の構成要素にセンサーを含めることである。センサーは、オン/オフの状態や単位などの情報を読み取れるものでなければならない。最もシンプルでわかりやすい例は、すべての飛行機と戦車に搭載されている、資源と弾薬の多くが、まだ自動では追跡できないことである(ほとんど例外なく、リアルタイムで追跡されることはない)。

特定の構成要素に一つ以上のセンサーが含まれている場合、次の問題は、センサーと構成要素を自動化する機能を組み合わせるかどうかということである。読み取り専用のセンサーを持つ構成要素は自動化できないが、他の構成要素は自動化できるはずであることは明らかである。これには、実戦配備されたシステムが期待通りに動作していない場合に報告する試験・評価(T&E)センサーが含まれる。いわゆる「ブラックボックス」的なAI能力を統合したシステムでは、これは特に重要である。

計測にとってリアルタイム接続と報告は密接な関係がある。リアルタイム接続が不可能な場合の次善策は、システムが接続を回復したときに優先順位をつけ、データを同期させる機能である。現在、DODのiPhoneバージョン同期作業では、ハードディスクを、ヘリコプター、飛行機、またはドローンから取り出し、国土を横断して輸送し、安全なデータセンターでダウンロードと処理が行われる。軍ではこのような方法の運用を続けることはできない。

### コンセプト：シミュレーション、試験、検証

計画立案、演習、およびウォーゲームは、常に戦争準備の不可欠な部分であった。軍事指導者と政策立案者が、大きな地図の付いたテーブルを囲み、米国やその同盟国、同志国、および敵対国の人員、戦車、航空機、および艦船を表すボード駒を覗いている場面は、十分に見慣れたものである。各陣営が一手を打ち、次の一手の前に、結果を公平な立場のオブザーバーが評価する。このような軍事ウォーゲームは、定期的に行われているが、今日のデジタル環境の急速な変化のペースを考慮すると、十分とは言えない。どの軍隊も、さまざまな種類のシナリオや緊急事態計画を持っている

が、これらの計画のうち、事前に試験されたものはほとんどない。それはなぜだろうか。

まず、シミュレーションシステムは、何をシミュレーションするかによって大きく異なる。シミュレーションに影響を与える主な要因は、そのサイズとスケール（一般的に扱われるオブジェクトの数）、各オブジェクトの忠実度（どのレベルの詳細をシミュレーションするか）、そしてオブジェクトに投入しようとする資源である。たとえば、ファーストパーソン・シューティングゲームは、比較的少数の兵士を、非常に高い忠実度でシミュレーションとして動作させることができる。銃の動作も、非常にリアルな小規模シミュレーション環境とともに、極めて正確に再現することが可能である。一方、インド太平洋戦域全体で、何千もの兵器システムとオブジェクトがすべて動的に相互作用するシミュレーションを実行しようとするれば、複雑さは指数関数的に増大する。このようなシミュレーションは、非常に複雑で高価であり、おそらく実現不可能だろう。

第二に、シミュレーションの設定そのものである。ほとんどの場合、実運用システムはシミュレーションに使用できないため、意味のある実運用データがシミュレーションシステムに移植されることはほとんどない。その場合でも、データ転送は複雑であり、データサイズや機密情報の保護の障害になる。このため、参入障壁が非常に高く、乗り越えることは非常に困難である。このような例として、過去10年間、各軍がライブ・バーチャル・コンストラクティブ(Live, Virtual, and Constructive [LVC])兵器システムを統合する方法について取り組んできたことが挙げられる。たとえば、米国防空軍のバーチャルフラッグ・プログラム(Virtual Flag Program)は、当初からLVCの統合・共同演習として設計された。過去数年の間にはかなりの進歩があったものの、バーチャルフラッグは、規模、複数レベルの機密システムの統合、忠実度という複合的な課題に依然として悩まされている。一つの有望な選択肢は、低忠実度・高オブジェクト数、または高忠実度・低オブジェクト数で設計されたシミュレーションシステムに、運用システムから情報を簡単にエクスポートまたは引き出す方法を見つけることであり、これは、これらの選択肢間のトレードオフを管理するダイヤルのようなものである。いったん構築してしまえば、これらのシステムの上で試験や検証を行えるだけでなく、「Chaos Monkey」(訳注：故意にサーバーをオフラインにしてクラウド環境の耐障害性を試験するツール)のような概念を取り入れることもできるだろう。

運用計画と緊急事態計画(分冊や続編を含む)も、システムの一部として記述し、検討することができる。敵対的生成ネットワーク(GAN)やデジタルツインなど、これらのプラットフォーム上で構築・実行できる極めて有利な技術が数多く存在する。このようなアプローチにより、DODはAI対応技術やその他の能力を使用して、さまざまな敵対的な戦略や技術を試験し、対抗する方法を探ることができる(この分野で最も有望な研究として、ゲーム理論的競争がある)。問題は、これらの開発には、国防総省に、適切なAPIで設計された適切なプラットフォームが必要となることである。今のところ、これは現実というより夢物語のようなものである。

## コンセプト：失敗を前提に設計し、Chaos Monkey を放つ

Netflix は、複雑で可用性の高いソフトウェアシステムの試験に使用する「Chaos Monkey」というオープンソースの製品を開発した。ソフトウェアシステムの一部を、まるで狂った猿が楽しく遊ぶようにランダムに破壊し、システムが回復し動作を継続できるかどうかを確認するものである。この試験の概念は現在「Simian Army」ファミリーツールにも拡張されている。

Chaos Monkey の実行は、物理データセンターにおける「電源コード試験」に相当する。つまり、稼働中のコンピューターシステムまたはルーターの電源コードを、ランダムに引き抜き、サイトの高可用性機能がすべて稼働し続けるかどうかを確認するのである。

DOD は、Chaos Monkey を二つの形態で検討する必要がある。一つは、DOD が戦闘ソフトウェアシステムのバックボーンを試験するために使用できる従来の試験製品として、また二つ目は、ランダムで制御不能な変化（現実世界で起こる不可避の摩擦）が起きるような状況で、戦闘計画が実際にどう展開するかを確認するシミュレーション環境での試験である。どちらのアプローチも、決して存在しない完璧な条件下での性能を最適化するのではなく、DOD が強靱性と抗脆弱性を重視した設計を始めることを可能にする。これはもちろん、DOD が、実戦配備されたシステムを現実的な動作条件の全範囲にわたって正確に表現するソフトウェアシミュレーションシステムを構築できてはじめて可能である。

## コンセプト：エッジにおける自律性と自動化の構築

どのようなシステムでも、ソフトウェア定義の場合、制御パスとデータパスは一般に分離される。制御は集中化され、実際のエンドポイントシステムは複雑な制御ロジックで任務を実行し、コストと複雑さが増大する。小さな制御の決定をすべて集中的に行うことは、明らかに現実的でなく、また最適でもない。エンドポイントと中央制御ロジックの間の遅延が大きすぎるため、通信回線が切断されただけで、エンドポイントの機能は停止してしまう。また、軍事的な状況で見られる、一般的なコンピューターネットワークよりも種類がはるかに大規模で複雑なエンドポイント（航空機、車両、艦船など）にも当てはまる。

このアーキテクチャは、DOD で大規模なシステムと小規模なシステムの両方に手動管理を必要とする場合、単純計算で、より多くのシステムに拡張するためにより多くの人手を必要とするため、反対論が出るであろう。実際のところ、今日のドローン飛行の制限要因は、実際のハードウェアではなく、機体を操縦するクルーの不足である。たとえば、MQ-1 プレデターまたは MQ-9 リーパーの遠隔操縦機（RPA）システムには、パイロットとセンサーオペレーター、保守整備要員やインテリジェンス分析官など、かなり大規模な裏方チーム（最大で合計100人程度）が必要である。業務モデルとしては、たとえ比較的安価なハードウェアを使う場合であっても、システム運用の人的コストはかなり高く、これではまったくスケールしない。現在の一人のパイロットが一台の RPA を操縦

するアーキテクチャに UAV を追加しても、コストや運用面で大きなメリットはない(つまり、規模の経済効果は限定的である)。それどころか、DOD は規模の経済が結局マイナスで終わる。より多くのシステムを飛ばすほど、各機体の運用は難しくなり、コストも高くなる。無限の資金がなければ、民間企業ならマイナスの規模の経済の影響ですぐに倒産してしまう。これを回避するには、AI 対応システムも含む自動化と自律化によって、人間が関与する高価なオーバーヘッドを取り除く方法を考え出すことである。

これらのエンドポイントシステムに自律性を与えることは、オーケストレーションによる集中制御の議論全体に反するように思われる。ここで正しい考え方は、特定のシステムが持つべき指揮範囲を考慮することである。たとえば、個々のドローンが、離陸、着陸、飛行、ルート計画立案といった関連する自律機能に責任を負う必要がある。ドローンが特定の物体を識別して追尾するようにプログラムされているのであれば小さなタスクであろうと、人間の大規模な介入や中央制御システムがなくても、機能を自ら実行できるようにする必要がある。しかし、1機のドローンが他の何百機ものドローンの機能を調整したり、ドローンの群れと地上の装甲車との間の調整を行ったりすべきではない。これらの指揮機能は、上位のソフトウェアが行うべきである。単一のエンドポイントが、指揮を受けるすべてのシステムのより広い文脈の中で適切な役割と機能を果たさなければならぬ。

#### コンセプト：開発環境と本番環境の統合

現在、DOD の中核的なコード開発環境は、このコードのエンドポイントへの配備と緊密には統合されていない。DOD には、同様の問題に対しさまざまな開発環境があるが、これらの特定の領域では、コード開発から本番環境への簡単な経路は存在しない。実際のエンドポイント（ドローン、艦船、車両、または有人航空機といったハードウェア）にコードを配備するのは、さらに困難なことだろう。このようなタイプの展開は、展開前にかなり多くの試験と評価が必要であることを考えると、決して簡単ではないが、このような開発－製造－配備－維持のサイクルを加速することは、将来の成功と戦場での競争優位の獲得に不可欠である。DOD はかつて、世界で最も大きな存在を主張し、商業的なベストプラクティスを拒むことを正当化できたが、今日、世界最大のハイテク企業は、DOD の常識をはるかに超えた、目を見張る速度と規模を扱っている。

コードを予測可能な方法でエンドポイントに「封印」して展開できるよう適切にパッケージングするコンテナ化は、重要な推進役としての役割を担うため、エンドポイントシステムの開発者は、新しいシステムでコンテナを扱い、さらにこれらのシステムを開発、セキュリティ、および運用プロセスの受容的なターゲットとして構築することが必要になる。

これは、このようなシステムに AI のような機能が統合された場合、さらに重要になる。従来のソフトウェアとは異なり、AI ソフトウェアは決して完成することはない。運用に関連するデータを集

めるほど、そして AI ベースのシステムが運用環境でより多く機能するほど、再訓練と更新が多く必要になる。このデータ収集、再訓練、コードの再展開のループの習熟が重要であり、開発環境が最終的な本番環境を考慮した正しい方法で構築されている場合にのみ可能となる。

### 第3章 すべてを統合する

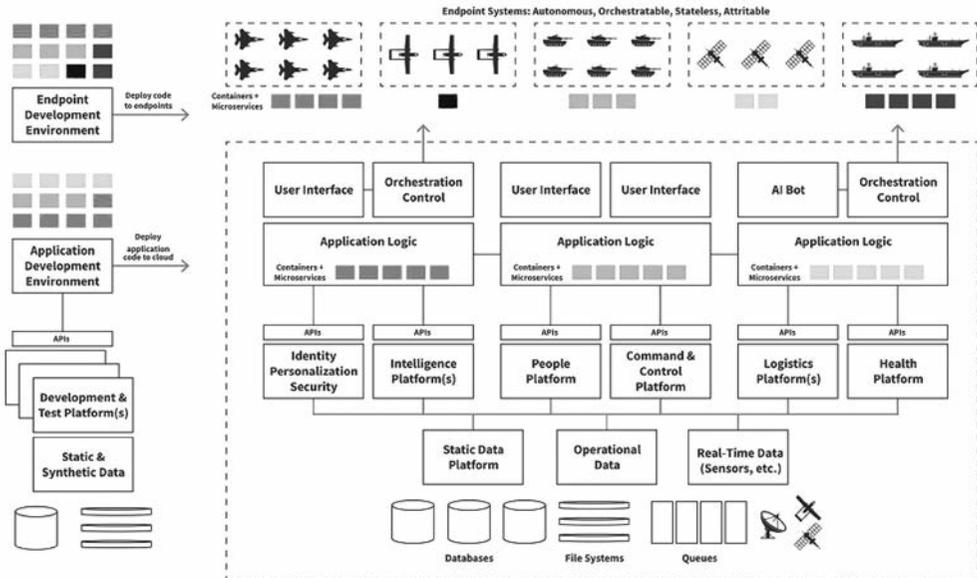
これらの基本原則を DOD 全体に適用することは、不可能に思え、レガシーな工業化時代の組織における大規模な技術変革は、一見、乗り越えられない障害に見える。実際、前述のソフトウェアのベストプラクティスを採用した組織の成功例のほとんどは、レガシーシステムに対処する必要のない、デジタルネイティブな企業であり、DOD のような複雑さに直面したことはほとんどなかったが、DOD では、何百もの異なるベンダーに構築された何千ものハードウェアおよびソフトウェアシステムが存在しているが、商用ソフトウェアのベストプラクティスとみなされるようなコアアーキテクチャの標準を強制されてきたベンダーはほとんどいない。

本報告書は、今後進むべき道についての青写真を提供するものである。DOD は、すべての兵器システムとハードウェアを設計、開発、実戦配備、維持する際に、これらのコアコンセプトを採用すべきである。ソフトウェアの「コア」と、外側のハードウェアの「シェル」、つまり、論理、計画立案、最適化をソフトウェアプレーンで実行し、ハードウェアを、ソフトウェアの長い連鎖の末端に繋ぐと考える。この理想的なアーキテクチャでは、エンドツーエンドのソフトウェアアーキテクチャが、関連するすべてのセンサー（入力）と出力（兵器システム）をつなぎ、予測インテリジェンスシステム、後方支援、ターゲティング、射撃などとともに、すべての正しいワークフロー（キルチェーン）のモデル化が容易になる。これらのすべてが、少数の大規模なコアプラットフォーム上に構築された単一のアーキテクチャに統合される。

これらの大規模なネットワークシステムの中心には、自律制御であれ手動制御であれ、すべてを統合する複雑なインターネット規模の高可用性ソフトウェアが置かれることになる。基盤となるソフトウェアシステムは、DOD の人、兵器、およびインテリジェンスをつなぐのに役立ち、また重要なこととして、より迅速なコードの開発、展開、および維持を可能にするため、DOD の開発環境および本番環境を統合することが必要になる。

このような変革を成し遂げるには、二つの重要なステップが必要である。まず、大きな問題を、小さな独立した部分に分解し、構造化されたインターフェースと API を作成する必要がある。これは、各軍が独自のインフラとアプリケーションを実行しているため、特に重要である。おそらく、最初の切り出しはこの部分で、その下にあるセグメントが続くだろう。第二に、DOD はシステムの他の部分を、時間をかけて「接着」できるコアの足場を構築する必要がある。このような「疎結合かつ高度に連携した」アーキテクチャは、問題のすべての部分を一度にアップグレードし書き換えるのではなく、より多くの部分をシステムに取り込み統合することで、徐々に拡張できる。

## ソフトウェア・デファインド・ウォーフェアの意欲的なコアシステムアーキテクチャ



出典：Nand Mulchandani による設計

先に強調したように、このプロセスの最も重要な部分は、全体的なアーキテクチャを適切なインターフェースと抽象化により定義することである。さらに、これらのタイプのインターフェース (API、REST)、データ交換フォーマット (JSON、XML)、ストレージと表現方法 (時系列データベース、イベント/パイプライン管理システム)、アプリケーションパッケージ (コンテナ)、および実行時アーキテクチャ (マイクロサービス) の構築には、DOD がよく確立され実証済みの業界標準およびオープンソースのベストプラクティスを使用する必要がある。同時に、このアプローチでは、データプラットフォームとアプリケーションロジックを切り離し、ユーザーインターフェースとプレゼンテーションサービスをバックエンドデータとアプリケーションロジックから (API などのインターフェースを通じて) 切り離す必要がある。これにより、高度に相互接続されている脆弱なスタックを構築する必要がなくなり、システムの各層は独立して進化させることができる。相互接続されている脆弱なスタックでは、小さな変更であっても、アーキテクチャを通じてシステムのあらゆる部分に波及し、コードの更新が非常に困難であり、多額の費用を要し高価になる。

DOD のソフトウェアアーキテクチャの中心には、省内で構築・運用されるさまざまなソフトウェアプラットフォームがある。これらのプラットフォームは、データ/コア/ロジック層とアプリケーションの間の必要な境界を提供する。この二つを分離することで、データの集約、クリーニング、統合といった手間のかかる作業はただでさえプラットフォームレベルで行われるため、新しいアプリケーションの構築が劇的に単純化された環境が提供される。さらに、DOD は、プラットフォームと API を実装するすべてのベンダーが、十分に文書化され、バージョン管理され、レンタルフリー

で、すべてのベンダーが自由に利用できるオープンインターフェースとしてこれらを構築するよう主張する必要がある。

こうなれば、アプリケーションの作成者は自分のアプリケーションロジックにのみ責任を負えばよく、データや一部の共通ロジックについては、これらのバックエンドプラットフォームサービスを自由に利用できるようになる。データ環境全体を一から再インポート、クリーニング、再構築し、共通のビジネスロジックを再コード化する必要がなく、新しい機能を迅速に劇的に低いコストで構築できる。また、複数のベンダーがこのプラットフォーム上で競合する製品を構築し、コストを削減し、DOD の顧客向けアプリケーションの品質を向上させることもできる。最後に、DOD がインターフェース(API、実際のアプリケーションそのものではない)を所有することで、ベンダーのロックインが減少する。どのベンダーも、顧客(DOD)のデータをシステムに閉じ込め、エクスポートまたはアクセスの方法を奪いデータを人質にとることはできなくなる。これは「ゴキブリモテル」(Roach Motel) 問題と呼ばれ、「ゴキブリはチェックインしてもチェックアウトしない！」という害虫駆除に特化した深夜の宣伝文句にちなんでいる<sup>5)</sup>。この場合の問題は、データをチェックアウトできないことである。このアーキテクチャは、外部ベンダーが新しいアプリケーションを構築できるようにすることに加えて、DOD 内部のコーダーやデータサイエンティストが小規模なアプリケーションの迅速な構築、既存のデータ上の新しい分析が劇的に容易になる。小規模なベンダーのためには、存在するデータのクリーニング、インポート、正規化の同じ障壁は、個々の内部開発者にとっては、さらに大きくなる。これを広い内部の開発者コミュニティに即座に開放することで、高コストと参入障壁のために現在開発されていない「マイクロアプリケーション」が劇的に増加する可能性がある。

## 第4章 結論

戦闘は、非常に管理が複雑で、常に技術と科学の難解な組み合わせと見なされている。事件の発生、継続、さらには規模、スケール、速度の無秩序の変化は、技術と科学のどちらもマスターが不可能に近いと思わせる。将来、戦闘はより複雑化し、さらに混沌とし、より高速化するだけだろう。技術と科学の方程式における技術の側面は、これまでと同様に重要である。しかし、新しいテクノロジーの普及とデータ中心の世界への移行に伴い、方程式の科学的側面はますます重要性を増している。将来の危機と紛争により速く適応し、ソフトウェアやAIモデルを実戦配備し、迅速に更新し、普及させるなど、最大の敏捷性を発揮することで、大きな競争優位性を獲得できる可能性がある。

DOD がこの新しい戦闘環境で競争力を維持するための唯一の方法は、最も強力な兵器であるテク

---

5) See recordman33, "Black Flag Roach Motel Commercial-1983," YouTube video, uploaded May 23, 2012, <https://www.youtube.com/watch?v=d55RPr5W0ys>.

ノロジー、より具体的にはソフトウェアを確実に利用することである。商用コンピューター業界は、DOD がこれまで戦闘に用いてきた方法でデータセンターを管理していたが、拡張は可能ではないことに気付いた。そこで業界は、データセンターの仮想化、自動化、監視、拡張する新しいツールや技術を考え出し、ダウンタイムをほとんどなくし、わずかなコストでサービスを提供できるようになったのである。

DOD をはじめとする米国連邦政府機関が直面している最大の課題の一つは、政府システムが、工業化時代のハードウェア中心の組織として一から構築されていることにある。デジタル時代のソフトウェア中心のリスク許容度の高い組織への移行は、非常に困難であるが、将来の成功への唯一の道でもある。テンプレートは存在し、あとは実行あるのみである。

ソフトウェア業界におけるこれらのコンセプトの多くは、軍用ハードウェアと戦闘に完全には類似しないかもしれないが、本論文で概説した中核的なコンセプトと、大規模で複雑なシステムのアーキテクチャに与える影響を理解することは重要である。重要なことは、軍用オブジェクトとハードウェアから、ソフトウェアの織り成すより大きなシステムの一部へと思考をシフトさせることである。これにより、極めて退屈でつまらないレガシーワークフロープロセスで最も高価で希少な資源を処理する事態を防ぎ、情報と意思決定のサイクルを劇的に加速させることもできる。同時に、現在あまりにも多くの時間が消費される、日常的でありふれた機能のほとんどを自動化し、ソフトウェアのベストプラクティスを実施することで、軍の指導部は、必要かつ重要な意思決定により多くの時間を使うことができるようになるだろう。

ソフトウェア・デファインド・ウォーフェアは、未来の方法である。もちろん、これは人間が冗長であったり、関与したりしていないことを意味するものであってはならない。それどころか、本論文で紹介したアプローチは、人間と機械のチームワークに、驚くほど新しい可能性をもたらし、人間と機械の役割は、それぞれが最も得意とすることに基づいて最適化されるのである。業務の目的、目標、そして全体的なシステムのアーキテクチャは、これまで通りすべて人間が設定する。実際の運用は、人間が設定したパラメーターに基づき、ソフトウェアが処理する。例外的な状況または問題が発生した場合は、これまで通り人間に任される。この変化を特徴付ける最良の方法は、エンドツーエンドのプロセスを電子的に接続することで、人間が、最も重要な最終意思決定に集中する時間を確保し、「人間からロボットのような作業を取り除く」ことである。完全に AI 化された未来では、UI/UX は、ユーザーがスマート機械を訓練、システムが個人の能力、認知の発達のペース、さらには過去の行動に適応できるようになるところまで進化するだろう。DOD のハードウェアとソフトウェアの現状では、現在これを実現できる方法はない。

われわれは、米国のために最良の結果を目指し、DOD 在職中は、米国をこの新しい方向に向かわせるためのベストプラクティスを制定し、実施するために最善を尽くしてきた。したがって、この

研究が継続され、これらのアイデアが次世代戦闘システムの考えと設計に反映され続けることを望んでいる。

---

## 著者について

**ナンド・ムルチャングニ (Nand Mulchandani)** は、米国中央情報局 (CIA) の初代最高技術責任者 (CTO)。CIA 入局以前は、国防総省の統合人工知能センター (JAIC) の CTO およびディレクター代理を務めた。また、Oblivion (Oracle が買収)、Determina (VMWare が買収)、OpenDNS (Cisco が買収)、ScaleXtreme (Citrix が買収) といった複数のスタートアップ企業の共同設立者であり、CEO として成功を収めた。コーネル大学でコンピューターサイエンスと数学の学士号、スタンフォード大学で経営学修士、ハーバード大学で行政学修士を取得。

**ジョン・N・T・「ジャック」・シャナハン (John N.T. “Jack” Shanahan) 中将 (退役)** は、米国空軍の中将として退役。2018年から2020年まで、国防総省統合人工知能センター (JAIC) の初代所長を務めた。JAIC を率いる以前は、国防総省インテリジェンス担当次官室の国防インテリジェンス (戦士支援) 担当ディレクターとして、アルゴリズム戦横断機能チーム (別名：プロジェクト MAVEN) の創設を主導した。空軍での36年間の勤務で、2,800時間以上の飛行時間を蓄積し、F-4 D/E/G、F-15E、RC-135の航空機を飛ばした。ミシガン大学、海軍大学、国立戦争大学、ノースカロライナ州立大学を卒業。