

ロボットのオペレーティングシステムを選ぶ際に考慮すべきポイント

November 2019

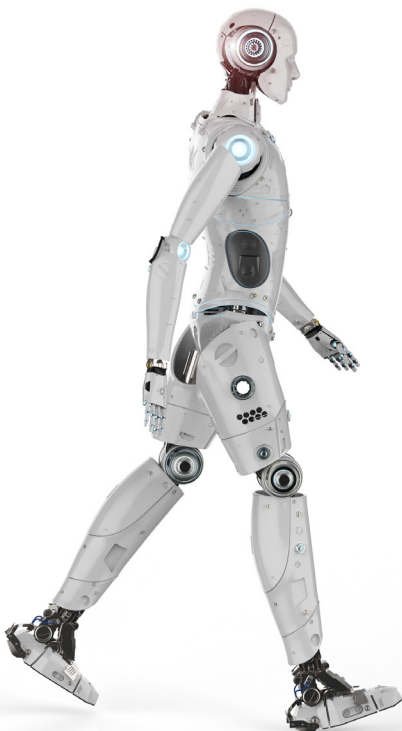
前書き

ロボットはこれまで何十年の間、SF映画のテーマになってきましたが、インターネットに接続されたデバイスなどの技術的な進歩により、ロボット革命が現実になり始めています。それは人工知能や機械学習の発展とともに、消費者関連、産業、ヘルスケア、小売など、あらゆる分野でロボットが採用されるようになっていることから明らかです。[IDC](#)によると、ハードウェア、ソフトウェア、関連サービスなど、ロボットに対する投資は、2021年までに世界で2,307億米ドルに達すると見込まれています。

市場の将来性と需要は相当大きく、技術の進歩に伴い、各企業はロボット生産の商機を捉えようとしています。しかし、IoT業界における他の多くの取り組みと同様に、持続可能かつ、将来においても有効で、セキュアなロボットを作成するために、ロボットメーカーが開発段階から決定すべき重要なポイントがいくつかあります。

そのなかで特に重要なのが、ロボットのオペレーティングシステムです。しかし、多くの場合開発が進んで変更が難しくなるまでこの重要性に気づかず、その結果数々の遅れや問題に対処する必要が生じることになります。例えば、即興での作業に最適化されたオペレーティングシステムは、ロボットが本稼働段階に入ると、保守が困難になる場合があります。同様に、独自に作成した場合には、ロボットの寿命においてオペレーティングシステム「全体」（アップストリームのセキュリティ更新のバックポートなど）を保守しなければなりません。また、堅実で安定したオプションには、結果的に古くて使用できない依存関係が含まれていたり、ニッチな分野のスタートアップが作った斬新なシステムは、すべての要件を満たしていたとしても、会社自体が突然倒産してしまうかもしれません。

こうしたさまざまな選択肢を前に、最善の選択をするには何を考慮すべきでしょうか？フットプリントの制約、性能要件など、下層の技術検討も確かに必要ですが、多くの要素は、開発プロセスだけでなく、それ以降の工程に大きく影響します。オペレーティングシステムは、企業の収益化計画、サポートサービス、セキュリティ戦略に貢献することができます。このホワイトペーパーでは、このような決定事項がどのように、より幅広い戦略を決定づけるのかを説明し、さらに製品が開発段階から本稼働段階および保守段階に移行するうえで変化していく重要な考慮ポイントを解説します。



開発に関する要点

ロボットの初期開発は、ハードウェアとソフトウェア両方のテクノロジーを選定するための重要なタイミングです。その会社にお金を無駄にする習慣でもない限り、この段階でロボットに使用されるオペレーティングシステムは、本稼働段階で使用するものと同じか、少なくとも関連性の高いシステムである必要があります。ここでは微妙なバランスが必要です。当然エンジニアリングチームは自分

Ubuntu -

Ubuntuは、デスクトップPC、クラウド、ロボットなどのIoTデバイス向けのオープンソースのオペレーティングシステムです。Ubuntuは、Azure、AWS、Google、およびIBMの大半のパブリッククラウドにおけるワークロードを実行する、主要なクラウドゲストOSです。RightScaleによる最新のState of the Cloud Report (クラウドの現状レポート) によると、クラウド市場の大半(64%)がAmazon Web Service (AWS) で稼働しています。AWSのクラウドインスタンスのうち、56%がUbuntuであり、さらに、Eclipseの2018年開発者調査によると、Linuxユーザーの40%がIoTソリューションにおいてUbuntuまたはUbuntu Coreを使用しているのです。



Ubuntuとその用途に関する詳細はウェブサイトをご覧ください。



が開発しているものを最適にサポートするシステムを求めますが、他にも考慮すべき要素が数多くあるのです。

ソフトウェアスタックの互換性

そのオペレーティングシステムの採用を決定した他の理由が何であれ、必要となる技術(ライブラリやフレームワーク)と互換性のないシステムを選択すれば必ず大きな失敗を招きます。そうなるとエンジニアリングチームが足踏みをしている間に、競合が先に製品を市場に出してしまいます。通常は、必要なライブラリが特定のLinuxディストリビューション向けに作成されているか、開発を早めるためにエンジニアリングチームはミドルウェアを使用します。たとえば、当社が聞き取り調査をしたロボット技術者の49%がRobot Operating System (ROS) を使用しています。これを考慮して候補となるオペレーティングシステムを評価すれば、開発プロセスを大きく効率化することが可能となります。



世界で初めてテニスボール収集ロボットを開発したTennibot社CEOのHaitham Eletrabi氏は、開発時にこのことを考慮しています。

「ROSやOpenCVのようなソフトウェアとの互換性があれば、Tennibotのアルゴリズムの実装やテストは簡単なタスクになります。さらに、Ubuntuはさまざまなセンサーやコンポーネントに使えるため、非常に優れた選択肢です。」

ハードウェアの互換性

ハードウェアの互換性もソフトウェアの場合と同じ理由から重要な問題と言えます。ロボット自体を進化させる前に、各コンポーネントを確実に連携させるために、莫大な時間がかかります。たとえば、特定のLinuxディストリビューション用に作成されたドライバを必要とするハードウェアを目の当たりにしたり、Linux全般について限られた知識しか持たないベンダーとの協業も珍しくありません。



顧客サービスロボットの限界を押し上げるBotsAndUs社CTOのAdrian Negoita氏は、早くからこの点を意識していました。

「Ubuntuはすべての要件を満たしていました。Ubuntuのハードウェアサポートは、まさに当社が求めていたとおりでした。市場シェアも高く、ほとんどのベンダーがUbuntuベースのシステムとの統合において豊富な経験を持っているため、私たちにとって追加のコンポーネントやソリューションの選定が非常に容易です。」

開発チームとの親和性

製品開発においてスピードは重要です。誰よりも速く目的地に到達することで、市場のリーダーとなるのです。チームが新規プロジェクトのプログラミング言語を選択する際、チームがその言語にどれだけ精通しているかが大きく影響します。これはチームが言語を変えながらないからではありません。チームは、使い慣れた言語を使用することで、短時間で高品質の製品を生み出せることを自覚しているからです。オペレーティングシステムの選定においても同様の考慮が必要なのです。エンジニアリングチームがそのオペレーティングシステムに精通していない場合、詳細に学び直す必要があり、当然、製品化までには時間を要することとなります。

SnapおよびSnapcraft

Snapはクラウド、デスクトップ、およびIoTデバイス上のLinuxシステム向けに設計されたコンテナ化されたソフトウェアパッケージです。Snapは安全な実行、依存関係のバンドル、機能の自動更新さが可能です。Snapcraftは、ソフトウェアをSnapとして書いたり、公開するためのコマンドラインツールです。



[Snap とSnapcraft](#)に関する詳細はウェブサイトをご覧ください。

システム統合の容易さ

ロボットがスタンドアロンデバイスであることは稀であり、通常は、他のデバイスとのシームレスな相互作用が必要です。他のデバイスは、ハードウェアインザループ試験のデジタルツインのように単純な場合もありますが、一般的にロボティクスにおいては、デバイス外での計算が主流になってきています。クラウドロボティクス、音声処理、機械学習などは、リソース制約のあるロボット上ではなく、サイバーファームで情報処理をおこなうことにより恩恵を受けている事例です。できることならロボットにおいてもクラウドと同じオペレーティングシステムを使用することは理にかなっています。そうすれば専門知識の分断を防ぎ、同じプロセスを用いて、クライアントとサーバー両方のコンポーネントの開発時間を短縮できます。

たとえば、[Small Robot Company](#)では、多くの農業用ロボットの製造を手掛けています。同



社は、さまざまな作業をこなせる高価な1台の大型ロボットではなく、特定の作業に特化した小型の専用ロボットを複数作っています。これにより、互いに通信し、連携する複数のロボットが協調しながら農地を整備することが可能になります。システム間の円滑な統合プロセスの重要性を理解している同社のチーフロボット技術者Joe Allnutt氏は次のように述べています。

「Small Robot Companyは農業用ロボット、ロボットの農場内基地（小屋）、および当社の開発マシンでUbuntuを使用し、AWSのクラウドでインスタンスを実行しています。Ubuntuは当社のシステムに対して一貫性があり、セキュアであり、よく研究された基盤を提供してくれます。」

Ubuntu Core

Ubuntu Coreは、ロボティクスなどのIoTデバイス向けの軽量な、トランザクショナルバージョンのUbuntuです。従来のUbuntuと同じカーネル、ライブラリ、およびシステムソフトウェアを使用し、Ubuntuベースの開発プロジェクトを簡単にUbuntu Coreに移行し、本稼働できます。

ゼロからSnapで作成されたUbuntu Coreは、自動更新により、無人デバイスであっても確実に最新のセキュリティパッチを受信できます。更新に失敗した場合は、ダウンタイムを最小限に抑えるために、最後に動作したバージョンにロールバックされます。

詳細は[Ubuntu Core](#)を参照してください。



サポートの有無

問題解決とは、エンジニアリングの醍醐味の一つです。しかし、どのエンジニアにもヘルプが必要なときがあります。そうした時、エンジニアは何を頼りにすれば良いのでしょうか？ もちろん、それは問題によって異なります。オペレーティングシステムに関するヘルプが必要な場合、そのディストリビューションに関するコミュニティがよく利用されます。多くの場合、そこには同じ問題を抱えている人が集まっています。たとえば、Ubuntuは[セキュリティ通知リスト](#)、[AskUbuntu](#)、[Ubuntuフォーラム](#)、[Launchpad](#)、さまざまな[IRCチャネル](#)および[メーリングリスト](#)を通じてコミュニティサポートを提供しています。





[BootsAndUs社](#)のリードソフトウェアアーキテクトのMatt MacLeod氏は、コミュニティサポートを活用する利点について次のように述べています。

「当社の技術チームはUbuntuの広範なコミュニティサポートを大いに活用しています。アドバイスを求めたり、問題の解決策を迅速に見つけることができるため、開発プロセスをさらに加速させることができます。」

コミュニティサポートはLinuxの長所の一つですが、常に完璧というわけではありません。誰も返事をしない、誰も知らない、解決する時間のある人が今はいないといったこともあります。このようなときはどうすればいいのでしょうか？ 選定したオペレーティングシステムに関して、製品の設計に対する支援から広範なソフトウェアサポートまで、プロフェッショナルによる信頼性の高いサポートが確実に得られるということは重要です。

IoTアプリストア

CanonicalのIoTアプリストアにより、ロボットメーカーはパブリックまたはプライベートのマーケットプレイスを作成できます。マーケットプレイスでは、アプリケーション（Snap）を管理し、更新、サポートサービス、新機能に加え、新しいアプリケーションの提供により収益化が可能です。



[IoTアプリストアに関する詳細はウェブサイトをご覧ください。](#)

一方で、問題がオペレーティングシステムとはまったく関係ないとしたらどうでしょう。それでも、オペレーティングシステムは、サポートを得る上で重要であることに変わりありません。たとえば、ROSはDebianおよびUbuntuのパッケージリポジトリのみをホストしています。つまり、ROSに関して誰かがヘルプを求めた場合、基本的にこれらのオペレーティングシステムの使用が前提になります。

製品に関する要点

ロボットが開発段階から本稼働および保守段階に移行すると、新たな要因が作用し始めます。この段階にいたるまでにこのことを考慮しなかった場合、事態は深刻で、会社はあるものをそのまま出荷するか、本来は解決できているはずの問題の解決策を必死で探すことになります。こうなると、市場への投入が遅れ、利益を失い、サポート費用がかさみ、多くの技術的問題を抱えることになるでしょう。

ソフトウェア更新プロセス

更新の必要性を考慮せずにデバイスを出荷してしまう企業はいくらでもあります。IoT市場はそのような企業で溢れています。デバイスの市場投入を急ぐあまり、ハードコーディングされた資格情報、開発キー、さまざまなセキュリティ脆弱性を持ったデバイスや、更新パスのないデバイスはめずらしくありません。



Miraiボットネットは象徴的な例です。あまりにも多くのデバイスに感染したため、2016年にはTwitterやPaypal、Spotifyといった大手サービスをダウンさせるほどの帯域幅を使用しました。どのようにデバイスに感染したのでしょうか？ それは、デフォルトのハードコーディングされた資格情報の使用を通しておこなわれます。では、デバイスを作った企業は、安全を確保するために迅速に対応してすべてのデバイスの更新をリリースしたのでしょうか？ いいえ、更新の計画はなく、デバイスは強制的にリコールされたのです。正しい更新手段を前もって見つけ出す費用は、大量リコールの費用に比べると微々たるものです。

これほど大事件ではなくても、ユーザーがバグを見つけたり、メーカーが新機能を有効化したい場合もあります。ユーザーに迷惑をかけずにこれらに迅速かつ低コストで対応できれば、それは素晴らしいことです。では、その方法をいくつかご紹介しましょう。



- まず、顧客数が少ない場合は、現場エンジニアが顧客まで出向いて更新することが可能です。この場合は詳細な対応が可能ですが、規模の拡大が難しく、更新数や顧客数が増加すると費用はすぐに天文学的な数字になってしまいます。
- 次に、より一般的な方法としてファームウェアのダウンロードがあります。そうすることですべての更新を1つのダウンロード可能なイメージにバンドルしてデバイスに適用されます。この場合、更新数や顧客数に合わせて拡張が可能です。更新をホストするインフラの整備と、更新を確実に適用するためのソフトウェアが必要になります。さらに、更新したらロボットが動かなるという場合はどうでしょう。復旧プランの開発は必要になります。これらすべてにお金と時間がかかってしまうのです。
- 最適な解決策は、**Snapcraft**をサポートしているオペレーティングシステムのように、こうした機能が組み込まれたオペレーティングシステムを選ぶことです。自社のソフトウェアにSnapを使用する企業は、これらの問題をすべて一度に解決することができます。Snapは高速で、かつ操作が簡単であり、既存のインフラ（Snap Store）を使用し、更新がリリースされると自動で更新され、ロールバック／障害復旧機能が組み込まれています。しかし理想的には、こうした利点がソフトウェアだけでなく、ソフトウェアが動作するオペレーティングシステム全体にもたらされるべきなのです。オペレーティングシステム自体の内部動作を含むすべての動作にSnapを使用しているのは**Ubuntu Core**だけです。

長期サポート

オペレーティングシステムの更新の配布方法に加えて、それらの更新が配布される期間も考慮しなければなりません。特定のバージョンのオペレーティングシステムは、通常はサポート期間が限られています。たとえば、Android Thingsの長期サポートは**3年サポート**です。一方、UbuntuおよびUbuntu Coreは**5年間**（**Extended Security Maintenance** [延長セキュリティメンテナンス]）ならさらに長い期間）サポートされます。オペレーティングシステムのサポート寿命が生産するロボットの予想寿命より短い場合、いずれ更新がなくなり、今後現れるMiraiロボットネットのようなマルウェアの犠牲になる可能性があります。少なくとも、企業はセキュリティ更新のために新しい計画を立案しなければならず、開発時には考慮していなかった追加の費用と時間が発生することになります。

収益性の高い寿命の確保

ロボットを出荷して保守するだけでいいと言う人もいるでしょうが、それでは、顧客を維持してロボットの寿命を延ばすための戦略にはなりません。競合相手が新しい製品を投入しても優位性を保てるでしょうか。

このケースではiPhoneが良い例になります。2007年にリリースされた最初のiPhoneは、iPhone OS 1を実装していましたが、このOSは他社のアプリケーションをサポートしていませんでした。このやり方を続けていたとしたら、iPhoneはいつまで優位性を保てたでしょうか。新鮮さがなくなれば、大多数のユーザーは他の機種に乗り換えていたことでしょう。しかし、早い時期にiPhone OS 2が発売され、アプリストアが利用できるようになりました。以降の評判は皆さんご存知のとおりです。

このことがすべてのロボティクスプラットフォームに当てはまるわけではありませんが、考慮すべきであることは明らかでしょう。ロボットの目的に応じて、完全にオープンにして、APIを介した制御やセンサーの使用を可能にし、**Snap Store**のような他社アプリストアをサポートすることもできるでしょう。このように、実質的に第三者がロボットに対する独自の展望を持てるようにすることで、ロボットの寿命は伸びるでしょう。ただしこれには、ロボットの汎用性が高くなければいけません。

ロボットの汎用性が低くても（低い場合がほとんどですが）、アプリストアで有料またはサブスクリプション方式で新しい機能を提供することで収入源を増やすことができます。企業が独自のストア（IoTアプリストアなど）を採用した場合でも、パートナーを自由に選択することで、コラボレーションと相互利益のチャンスが生まれます。

Rigadoの例を説明します。同社は大規模な商用IoT展開のためのエッジゲートウェイを開発しています。顧客によって要求が少しずつ異なるため、Rigadoのゲートウェイは、同社のアプリストアで厳選されたアプリケーションを活用してカスタマイズすることが可能です。これにより、デバイスをさまざまな業界で使用できるようになるだけでなく、Rigadoの提携企業がさらにカスタマイズされたアプリストアを顧客に提供できるよう、この機能を拡張することを計画しています。



結論

ロボティクスの分野は、IoTを構成する他のデバイスと同じように、よりスマートになり常時接続される方向に進んでいます。このようなプラットフォーム上で稼働するオペレーティングシステムの重要性は軽視できません。ロボットの寿命全体を通じて、オペレーティングシステムは有効にもなり、妨害にもなります。そのどちらになるかは、企業が意思決定する前に全体像を考慮するか否かで決まります。

Ubuntuでのロボットの運用に関する詳細については、[こちらからお問い合わせください](#)。

参考情報（英語）：

- ウェビナー： [Building a commercial robot with Open Source](#)
- ブログシリーズ： [ROS prototype to production on Ubuntu Core](#)
- 5部構成のブログと動画のシリーズで、ロボティクスにSnapとUbuntu Coreを使用すべき理由とその方法について、技術的に解説します。
- ケーススタディ： [BotsAndUs](#)
- ケーススタディ： [Robotcheers](#)
- ブログ： [Tennibot](#)
[SeaMachines](#)