

航空宇宙向け Software Defined アーキテクチャの設計

青柳 理紗子[†] 堀内 隆志[†] 西川 和宏[†] 福島 信純[†] 小川 大佑[†] 鶴崎 健介[†]

三菱電機株式会社 鎌倉製作所[†]

1. はじめに

ソフトウェア(S/W)によって機能を定義・実現する Software Defined Anything の導入が組み込みシステムで加速している。Software Defined を実現することで、システムの大部分が S/W により制御可能となる。

一方、航空宇宙向けシステムは、ハードウェア(H/W)によって主な機能を定義・実現し、かつ、H/W と S/W が緊密に連携する構成であり、S/W による機能実現や拡張・変更が困難である。そのため、航空宇宙向けシステムにおける Software Defined 実現には、S/W アーキテクチャを一新する必要がある。

そこで、航空宇宙向けシステムで Software Defined を実現するための S/W アーキテクチャについて検討を行ったので報告する。

2. 課題

Software Defined は、機器やシステムの機能を S/W にて定義、すなわち S/W にて機器/システムの機能を柔軟に拡張・変更できるようにするものである。この対象として、機器ではネットワーク(Software Defined Networking:SDN[1])、やデータ記憶装置(Software Defined Storage: SDS[2])、システムでは車(Software Defined Vehicle:SDV[3])などに適用が広がりつつある。

これら Software Defined を実現するために S/W に要求される機能を整理すると、以下となる。

1) H/W と S/W の分離

S/W にて柔軟な機能拡張・変更を可能とし、S/W の再利用性を高めるため、S/W を可能な限り H/W 非依存とする必要がある。

2) S/W 部品の疎結合化

1)と同様に機能の拡張・変更を容易にするため、極力 S/W を部品化するとともに部品間の依存度を極力下げる必要がある。

3) 動的アップデート

運用中に Over The Air(OTA)による S/W の追加・拡張・更新を実現するため、機器運用中に S/W を更新できる仕組みが必要である。

4) コネクティビティ

移植容易性・動的アップデートの実現や、機器/

システムが連携動作するためには、機器/システムが外部と繋がる必要があり、これらを S/W で柔軟に対応できる仕組みが必要である。

以上は、機器/システム搭載 S/W のアーキテクチャに要求されるが、S/W アーキテクチャのみの検討では実現できず、S/W 開発環境も含めた検討が必要である。また、機器/システム運用時に S/W の動作状況を監視・管理し、必要に応じて S/W 更新ができる運用環境の仕組みも必要である。

3. 全体アーキテクチャ

これらの課題を解決するための提案アーキテクチャを図1に示す。提案アーキテクチャは、搭載 S/W、搭載 S/W を開発する Dev(開発)環境、運用時に搭載 S/W を管理・監視する Ops(運用)環境の3つの要素からなり、これらが連携して動作することが重要と考える。

以下、各要素で必要な機能を述べる。

3.1. 搭載 S/W

搭載 S/W では、2章に示した1)~4)を実現するため、以下のレイヤ構造を備える。

1-1) ハイパーバイザ

H/W 仮想化機能を備え、これより上位の S/W に対し H/W を隠蔽する役割を担う。

1-2) OS

計算機資源の管理を行い、これより上位の S/W に対し、計算機を隠蔽する役割を担う。ハイパーバイザと OS により、H/W と S/W の分離を実現する。

1-3) SDx エンジン

下記アプリケーションに対し、様々な共通サービスを提供する。本 S/W に、アップデート機能やサービスバス機能を配し、アップデートやコネクティビティを実現する。

1-4) アプリケーション

サービス単位・疎結合に構成され、製品の機能を実現する。拡張性・柔軟性は、上記 1-1)~1-3)と合わせて実現する。

3.2. Dev(開発)環境

搭載 S/W の Software Defined 化には、搭載 S/W の開発環境にも柔軟かつ迅速さが必要である。そこで、以下の機能・仕組みとする。

2-1) アジャイル開発対応環境

DevOps や DevSecOps 的な S/W の継続進化に迅速

Design of Software Defined Architecture for Aerospace Systems
Risako Aoyagi[†], Takashi Horie[†], Kazuhiro Nishikawa[†],
Nobusumi Fukushima[†], Daisuke Ogawa[†], Takeyuki Tsurusaki[†]
Kamakura works, Mitsubishi Electric Corporation[†]

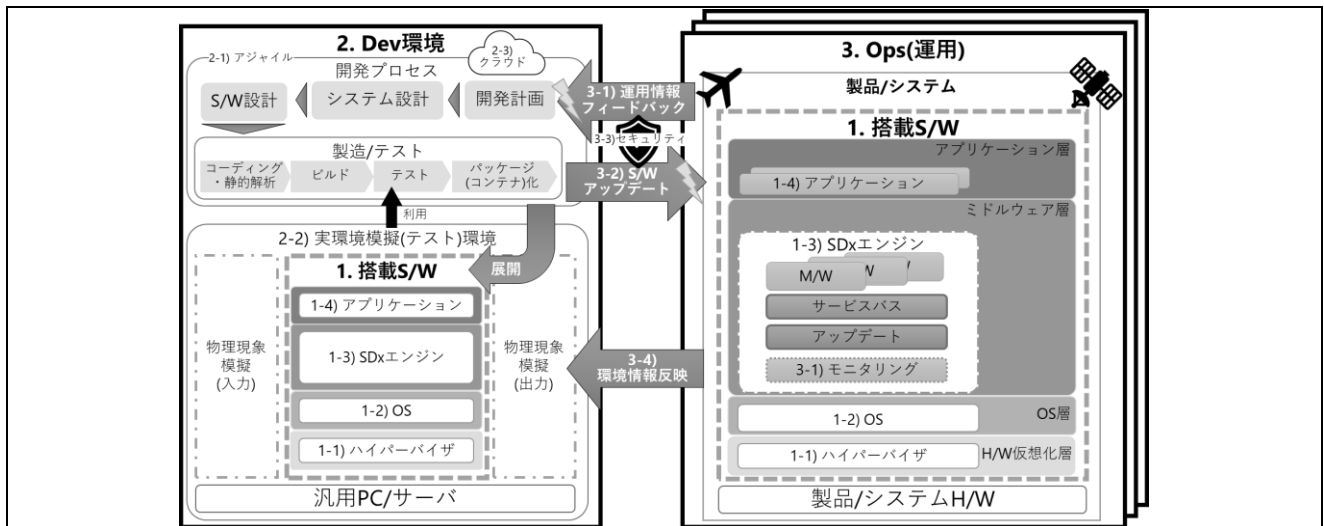


図1 航空宇宙向け Software Defined アーキテクチャ

に対応できる仕組みとする。

2-2) 実環境模擬(テスト)環境

S/W の継続進化へ迅速に対応しつつ、品質の高い S/W を実現するため、実(Ops)環境に近い S/W テスト環境とする。

2-3) クラウド環境

S/W の継続進化において、開発規模の変化、開発環境の速やかな提供に対応するため、可搬性・拡張性を備えるクラウド環境を活用する。

3.3. Ops(運用)環境

搭載 S/W を運用する Ops 環境は、搭載 S/W 及び Dev 環境と連携し、製品の Software Defined 化を実現するため、以下の仕組みとする。

3-1) モニタリング

搭載 S/W の状態を監視し、修正・改善が必要な場合や、製品に対するユーザ評価を速やかに Dev 環境に情報が伝達できる仕組みとする。

3-2) オンラインでの S/W 更新

Dev 環境で開発された搭載 S/W を Ops 環境に速やかに適用するために運用中に S/W を更新する仕組みとする。

3-3) セキュリティ対応

上記 3-1)、3-2)はネットワークを介して行うため、セキュリティ対策を施す。

3-4) テスト環境との連携

前記の通り、テスト環境と Ops 環境を近い環境とするため、Ops 環境の変更や状況を速やかに反映させる仕組みとする。

4. 航空宇宙機器での検討課題

前記にて、Software Defined を実現するためのアーキテクチャについて述べたが、これを航空宇宙機器へ適用するには以下の課題があり、解決していく必要がある。

4.1. 搭載 S/W の安全性

航空宇宙機器は、S/W の不具合により製品の機能を失うことが許されない。そのため、S/W の信頼性やセキュリティなど安全性を考慮した開発が必要であり、Software Defined 化で必要な迅速・柔軟性と相反する部分が存在する。

4.2. リアルタイム性

航空宇宙機器 S/W は、外部からの入力に対し一定時間内で応答する必要がある(リアルタイム性要求)。提案アーキテクチャは、H/W 抽象化などリアルタイム性が阻害される部分が存在する。

4.3. 長期運用・保守

航空宇宙機器の運用は、10 年以上の長期間にわたる。そのため、運用期間中の搭載 S/W 及び Dev 環境、Ops 環境の維持・管理が必要である。

5. おわりに

本稿では、航空宇宙向け Software Defined を実現するための S/W アーキテクチャを提案した。提案アーキテクチャは、搭載 S/W と Dev 環境、Ops 環境の3つのコンポーネントからなり、これらが連携して動作することが重要と考える。

今後は、航空宇宙機器における課題を解決すると共にアーキテクチャの詳細化を行う予定である。

参考文献

- [1] ONF, Software-Defined Networking(SDN) Definition, <https://opennetworking.org/sdn-definition/>
- [2] Macedo, Ricardo, et al. "A survey and classification of software-defined storage systems." ACM Computing Surveys (CSUR) 53.3 (2020): 1-38.
- [3] SBD, ソフトウェア・デファインド・ビークル, <https://www.sbdautomotive.com/reports/the-software-defined-vehicle>