サイバーフィジカルシステムをソフトウェアデファインド化するWebAssembly応用技術

Constructing Software Defined Cyber-Physical Systems by Applying WebAssembly

ソフトウェアで，顧客の要望するサイバーフィジカルシステムを迅速に構成でき，柔軟に変更できる技術を実現

ハードウェア依存からの脱却とシステム拡張性の向上が可能な，ソフトウェアデファインド化（以下，SD化と略記）が注目されています。東芝は，WebAssembly（WASM）を活用し，仮想マシンやコンテナよりも高いポータビリティー（可搬性）と安全性を持つソフトウェアコンポーネントのモジュール化を実現しました。これにより，サイバーフィジカルシステム（CPS）は，WASMの最小機能単位のソフトウェアをコアにした柔軟なシステム構築で，サイバー空間からの迅速な構成変更・更新が容易になりました。更に，将来は生成AIと組み合わせることで，コーディング不要のシステム開発にすることを目指しています。

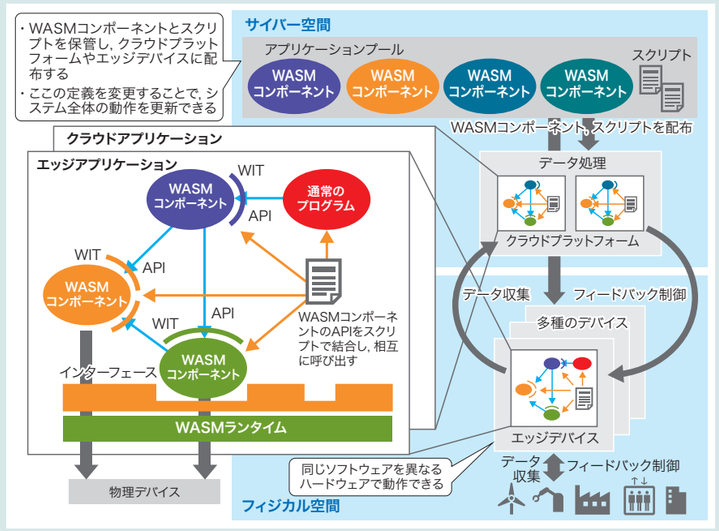
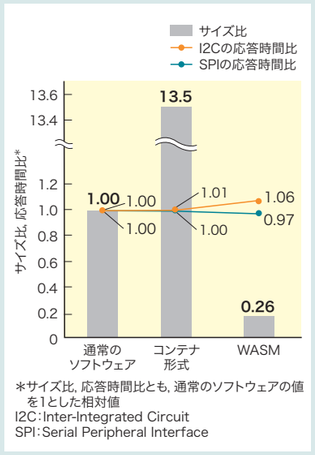


図1．WASMを用いたSD化システム構成

サイバー空間側の構成定義により，WASMのAPIを結合し，CPSを構成します。ハードウェアに依存せず動作可能です。



＊サイズ比，応答時間比とも，通常のソフトウェアの値を1とした相対値
I2C：Inter-Integrated Circuit
SPI：Serial Peripheral Interface

図2．WASMと，通常のソフトウェア，コンテナ形式の応答時間・ソフトウェアサイズの比較

WASMは，応答時間の遅延は6％以内，サイズは通常のソフトウェアの約1/4となり，軽量で高速実行できます。

背　景

実世界に存在するデータは，電圧・電流・温度・圧力などの物理量や，生産台数・品質指標などの統計量に至るまで多種多様であり，その種類や形式によって処理の仕方も異なります。処理に使用するソフトウェアは，個別に開発するのではなく，仮想マシンやコンテナを用いてモジュール化し，流用可能にしています。しかし，従来方法では，ソフトウェアを実行するプラットフォームのハードウェアやアーキテクチャーに依存し，また仮想マシンやコンテナのサイズが大きくなることで可搬性が低下し，ビジネスの変化に柔軟に対応できるシステムの構築や変更が困難でした。

東芝は，従来はハードウェアに依存していた機能を，SD化によりソフトウェアで定義・制御することで，システムの柔軟性や拡張性を向上させ，製品やサービスの価値を高めるアプローチを進めています。この実現には，ハードウェアとソフトウェアを効率的に使うための抽象化と可搬性の向上，そして迅速に仕様の定義や変更が行える統合管理や構成自動化が不可欠です。これらに合うソフトウェアコンポーネントとしてWASMに注目・活用し，SD化を実現しました。

WASMの特長である高い可搬性と安全性

WASMは，様々なWebブラウザーで動作できる可搬性を持ち，安全・高速に動作できるように設計された機械語（コンピューターが理解するバイナリー命令）形式のソフトウェアです。2019年にWorld Wide Web Consortium（W3C）でWeb標準規格となりました。WASMコンポーネントは，Webブラウザーだけではなく，ランタイムと呼ばれる画面表示がないWebブラウザーに代わるソフトウェア上でも動作できます。ランタイムによって，Webブラウザー以外のプラットフォーム上で，ハードウェアやアーキテクチャーによらない可搬性を持ち，安全・高速に実行できます。

最小単位のモジュールを複数結合したシステム構成

ソフトウェアを効率的に活用するためには，機能ごとにコンポーネント（部品）化され，最小限で実行可能な製品であるMVP（Minimum Viable Product）の形が理想的です。MVPとして設計されたソフトウェアは，比較的小さく，機能も単純なため，API（Application Programming Interface）を介して複数のコンポーネントと組み合わせやすくなります。

そこで，WASMの各コンポーネントをMVP化し，そのAPIをスクリプトで連結します。これらのコンポーネントとスクリプトをサイバー空間からネットワークを通じて，CPSの物理空間との接点である各種エッジデバイスや，サイバー空間のクラウドプラットフォームに配布し，それぞれ動作させます（**図1**）。サイバー空間に登録されているアプリケーションプールのWASMとスクリプトを追加・更新することで，CPS全体の機能の追加・アップグレードが容易にできるようになります。また，常に最新のモジュールを参照して実行することで，安全性を向上させます。

リアルタイム性とインターフェースの検証

WASMの利用は，Webブラウザーなどのデスクトップでは普及していますが，それ以外ではまだ発展途上です。特にCPSでは，リアルタイム性，安定性，サポートされるインターフェースの種類の多さが重要です。

そこで，工場などの産業用途で利用されるネットワークプロトコルや，ロボットシステムを制御するソフトウェアを実際にWASM化し，エッジ装置でのリアルタイム性，ソフトウェアのサイズ，及びインターフェースの不足や互換性の問題がないかどうかを検証しました。

リアルタイム性の評価は，センサーデータを定期的に取得し，WASMで処理・出力した応答時間と本来の処理すべき時間との遅延を測定しました。応答時間比は，通常のソフトウェアやコンテナ形式に比べ，インターフェース2種類について遅延は6％以内とほぼ同等であり，ソフトウェアのサイズは約1/4になりました（**図2**）。インターフェース検証の結果，多くのソフトウェアでは従来のOS（基本ソフトウェア）のシステムインターフェースが使われますが，WASMではそれら全ての互換性を保証する標準化は未完了と確認できたため，未実装のインターフェースを新たに追加して，標準化を提案しています。

生成AIによるコーディング不要化と今後の展望

WASMは，C/C++やスクリプトなどの多種の言語で記述されたソフトウェアからの呼び出しが容易です。また，インターフェース仕様は，WIT（WebAssembly Interface Type）という定義書に記載され，API利用時に読み取られます。そこで，顧客が要望するシステム要件とWIT情報から，必要コンポーネント選択と結合コード（スクリプトを含む）・テストコード作成を生成AIで行い，テスト実行したログ情報から再度，生成AIでリファインし，正確にシステムを構成，動作させることを目標に研究しています。

現行の生成AIでは，システムの全てのソフトウェアコードは作成できません。作成したコードを過剰に信頼するとバグや脆弱（ぜいじゃく）性を含んだままとなり危険です。しかし，品質の高いMVPであるWASMモジュールをコアにして，生成AIで作成した結合コードでAPIを結合するのであれば，正確かつ信頼性の高いシステムが，コーディング不要で構築できると考えられます。

今後，WASMと生成AIを活用した新たなSD化を推進していきます。

中嶋　宏
デジタルイノベーションテクノロジーセンター
先端ソフトウェア技術室　オープンソース技術部