

ホリスティック・クラウド指向デバイス・アーキテクチャ

Architecture for Devices based on Holistic Cloud-oriented Configurations (ADHoCC)

大橋 正¹

Tadashi Ohashi¹

¹アイリクト

¹iLICT

Abstract: The device's architecture for an Internet of Things (IoT) can be configured (upgraded or downgraded) by hardware functions of FPGA (Field Programmable Gate Alloy) software products or of application software products in device storage. These FPGA's objects and application software objects are gathered by Open Linked Data from various objects sources on Cloud. Architecture for devices based on Cloud-oriented Configurations (ADHoCC) is proposed so that FPGA's software, system software and application software and are deployed by ADHoCC under the basis of current device configuration components. A given necessary architecture is performed on cloud. Finally those FPGA's software and application software in ADHoCC are reconfigured (downloaded) into FPGA or storage components in devices.

1 はじめに

IoT は全ての物(Things)がインターネットに繋がれることを標榜しているが、例えば物が電子部品であってもソフトウェア部品であっても、単に繋げるだけでなくある一定のルールに則って、全てに開放されることが重要である。この物と物が繋がりあうルールにセマンテックスがあるかを検討することは重要なテーマである。このテーマを深める為に自ら提唱するホリスティック・クラウド指向デバイス・アーキテクチャ(ADHoCC)を採り上げその実現性について検討する。

IoT デバイスのアーキテクチャは所与の要求仕様を満たすハードウェア及びソフトウェアコンポーネントにより構成されたシステムによって実現してきた。しかし今日ではアーキテクチャの高機能化に伴いユーザは操作性や利便性を失いつつあり、更に高密度実装などにより消費電力の増大も散見されるようになってきた。そこでこの解決のためにデバイス、例えばモバイル端末等にはクラウドとの共同作業の為に通信処理部やセンサ等の入力処理部とデバイスに特化した処理部のみに限定し、必要に応じてハードウェア(FPGA[1]ソフトウェア)やOS やドライバやライブラリ等のシステム・ソフトウェア及びアプリケーション・ソフトウェアを追加、変更、削除を

行いユーザの利便性と拡張性をもたらす機能の強化や弱体化の考え方を導入する[2]。このユーザの観点での機能強化又は弱体化のアーキテクチャを実現するコンポーネントの構成をクラウド上で展開し、各構成要素を LOD(Linked Open Data) [3]で結合し、再構成[4]後にオブジェクトを SPARQL[5]で収集しデバイスへコンフィギュレーション(ダウンロード)するADHoCCを検討する。

2 現状の課題

従来のデバイスのアーキテクチャは個々のデバイス構成情報を満たすハードウェア及びソフトウェアコンポーネントを製造部門でシステムジェネレーションをそのままデバイスへ実装してユーザに提供してきた。しかし以下の課題を列挙できる。

2.1 開発元の多様化とユーザの管理

FPGA ソフトウェア、システム・ソフトウェアやアプリケーション・ソフトウェアの開発元が多様化しており、オブジェクトの所在が把握され難くなってきた。その結果ソフトウェア・プロダクトの機能や版数の最終管理はユーザ自身が実施しなければならない状況にある。

2.2 デバイス仕様の高機能化

ユーザには最新且つフルスペックのデバイスが提供されるのでユーザは自ずとデバイスに追従したフルスペックの使い方に従わざるを得ない。以上に加えてフルスペック稼働に伴う消費電力の増大や非経

連絡先:大橋 正, アイリクト, <http://www.ilict.jp/> メール・アドレス: ohashi@ilict.jp

済性の問題がユーザ側に生じると考えられる。

2.3 クラウド側の極大化とデバイス側の負荷

FPGA ソフトウェアのライブラリ群やコンポーネント類も統合開発環境 IDE(Integrated Development Environment)を用いて個別に構築し、必要なコンポーネントを添付 CDROM より実装するか若しくはネットからダウンロードするか自ら HDL(Hardware Description Language)を用いて回路設計し、ソフトウェアとしてライブラリに登録する。従いデバイス側の開発工程において FPGA ソフトウェア、システム・ソフトウェアやアプリケーション・ソフトウェアをクラウドから収集し、デバイスへ書き込む手順の複雑さが増大し、これが開発手番の増大化につながり FPGA のライブラリやコンポーネントの再利用性が十分に生かされていないと考える。

3 課題の解決

前記の課題の解決のためには、デバイス側でシステム生成を行うのではなく、クラウド側で所与のデバイスに関わるハードウェア(FPGA ソフトウェア)、システム・ソフトウェア及びアプリケーションの構成を展開し、クラウド側で個々のデバイスのシステム生成する。その為に構成要素の表現をセマンティック web 表現に置き換えて、各々のオブジェクトを LOD で結合させるクラウド指向型デバイス・アーキテクチャを実現させて問題の解決をはかる。

3.1 ユーザを中心としたアーキテクチャ

IoT のアーキテクチャの基本はユーザの使い勝手に基づいた設計思想を貫徹することである。即ちユーザの使い勝手をクラウド上で設計して高位合成(HLS: High Level Synthesis)を極めることである。この思想の根幹をなすものが LOD として捉え、その役目が大きいと考える。

3.2 コンポーネント要素表現

ユーザ側のデバイス構成要素を RDF(Resource Data Framework)若しくは OWL(Web Ontology Language)等による知識表現を導入し FPGA ソフトウェア若しくはアプリケーションソフトウェアの名称、版数、オブジェクトの格納ロケーション、稼働前提及びセキュリティや課金情報を表現しこれらをまとめて構成要素群とする。

3.3 FPGA のクラウド開発環境 HDE

FPGA ソフトウェアのライブラリ類若しくはコンポーネント類も前述と同様な知識表現で具象させる。更に統合開発環境 HDE(Holistic Device Environment)を個別に構築し必要なコンポーネントをクラウド上で共用させる。

4 ADHoCC の概念

以上の解決案をクラウド指向デバイス・アーキテクチャ(ADHoCC)としての概念を図1に例示する。

4.1 ユーザを中心としたアーキテクチャ

デバイス・アーキテクチャはユーザが所持しているデバイスの構成情報をクラウド側で所持し、ユー

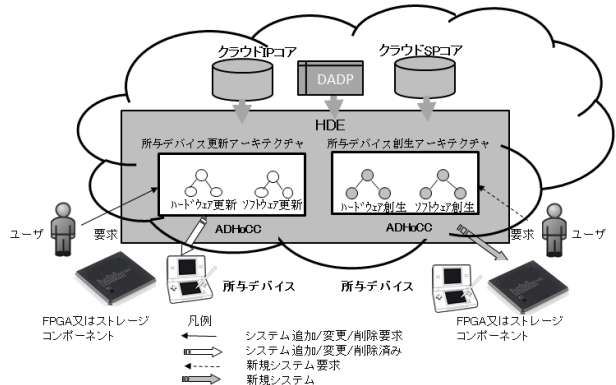


図1 クラウド指向デバイス・アーキテクチャ概念図

ザの全体機能若しくは部分機能の強化若しくは弱化的要求を受け取り、その要求に従い構成情報を変更する。一方クラウド側では幾つかのデバイス・アーキテクチャ・デザイン・パターン(DADP)により新システムを創生してデバイス内のFPGA若しくはストレージ・コンポーネントへコンフィギュレーション(ダウンロード)する方式となる。無構成の状態からシステムを生成することは現時点では難しいのでDADPを用いてユーザに選択させる方法である。ここでシステム構成を図2に例示す。ここでは

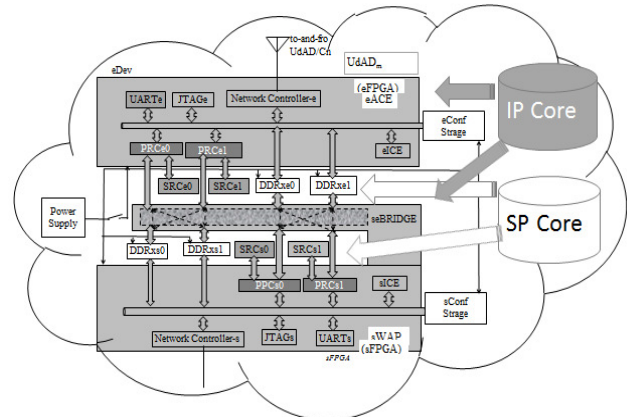


図2 ADHoCC デバイス・アーキテクチャの一例

eDev はデバイス処理部であり PRCe はプロセッサ、SRCs は記憶コンポーネントである。

4.2 コンポーネント構成要素の知識表現

所与デバイスの構成要素となる CMD(Component Meta Data)形成メタ情報と、これ等が集められて CMD 形成群情報をなし部品のコンポーネント単位と構成要素群を一致させる。この構成要素に RDF 若し

にシステムを創生する方式が考えられる。

4.5 クラウド指向デバイス・アーキテクチャの概要

デバイスの開発プロセスでは統合開発環境によって提供され、システムの更新又は創成の際に利用される。FPGA ソフトウェアとしてのコンポーネント・ライブラリについての特殊性を維持するのではなく一般のソフトウェアと同一レベルにする必要がある。この環境でクラウド指向デバイス・アーキテクチャを更新若しくは創生することが可能となる。クラウド上で展開された所与デバイスのストレージは FPGA ソフトウェアで実現されるが DDR (Double Data Rate) で転送する大容量ストレージ・コンポーネントは現時点では物理的なコンポーネントとして存在するので ADHoCC には利用できないが、近い将来には代替コンポーネントが実現するのではないかとと思われる。

4.6 システム・コンフィギュレーション

更新若しくは創生されたデバイスの FPGA ソフトウェアやアプリケーション・ソフトウェアのオブジェクトは図 5 の如く各社とも多種多様である。各社若しくは個人で開発されたソフトウェアプロダクト SP コアは LOD により相互接続されており更新コンフィギュレーション若しくは創生コンフィギュレーションを基に SPARQL で検索されたオブジェクトがデバイスへの連続したアドレス空間に展開し、そのまま所与のデバイスに書き込めば良い。この提案によりコンポーネントの流通性は飛躍的改善される。

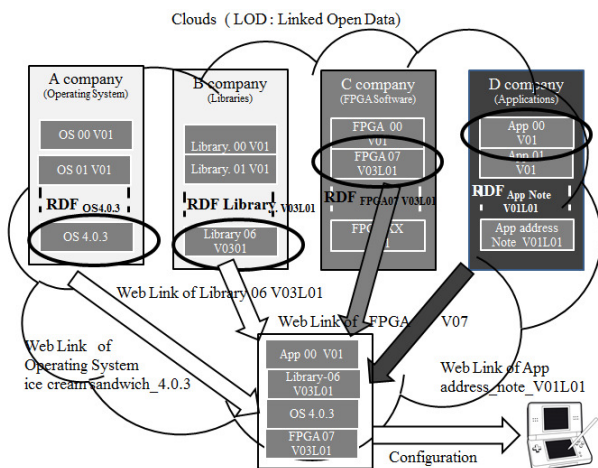


図 5 デバイス・コンフィギュレーション

5 期待される効果

実験して確認するには至っていないが提唱する ADHoCC により以下の効果が期待される。

5.1 FPGA の IP/SP プロパティの流通性の向上

従来の FPGA ソフトウェアは各開発者の PC レベ

ルの限定されていた閉作業空間の壁をぶち破りクラウドなる超上流階層で行うためにコンポーネントの利用度が格段に改善することが期待される。この結果 HDL の相乗効果と相まって HLS も大幅な改善が期待できる。

5.2 ADHoCC の拡張性

本論ではデバイスだけに限定したが ADHoCC のアーキテクチャはあらゆる情報処理の分野に適用できる可能性が大きい。今まではデバイスに FPGA のソフトウェアをコンフィギュレーションしてからストレージ・コンポーネントをローディングして運用していた方式から ADHoCC でそのままストレージにローディングさせる方式で大幅な利便性と拡張性も期待できる。

6 今後の課題

従来の開発プロセスで蓄積したアイデアをまとめて理想的な ADHoCC を検討してみたが実際に運用するには様々な今後の課題が挙げられる。

6.1 IoT 間の LOD 結合でのセマンテックス

ソフトウェア・プロダクトを IoT として捉え、システムをコンフィギュレーションする要素間に果たしてセマンテックスは存在するのかを実現性の観点で再考する必要がある。クラウド上にはビッグデータとして何億という天文学的な数値のオブジェクトが存在する。これらの相互関係にセマンテックスを付与する ADHoCC に実現性はあるのかを研究する必要がある。

6.2 ハードウェアの FPGA ソフトウェア化

一般的なプロセッサや信号処理プロセッサ (DSP) 等は既に FPGA ソフトウェアで実現しているが GPU や ARM 等も FPGA ハードマクロとしてではなく FPGA ソフトウェアで実現する研究や DDR3/4/5 等の大容量ストレージ・コンポーネントの FPGA ソフトウェア化とそのストレージ・コンポーネントを ADHoCC のレベルで格納する技術も必要である。

参考文献

- [1] 天野英晴 [編著]: FPGA の原理と構成, オーム社刊, (2016)
- [2] 大橋 正: ユーザ・ドリブン・アーキテクチャ・デバイス (UdAD) の検討, 第 15 回情報科学技術フォーラム, C-035, (2016)
- [3] T. Heath, C. Bizer: *Linked Data Evolving the Web into a Global Data Space*, Morgan & Claypool Publishers, (2013)
- [4] 末吉敏則・天野英晴 [編著]: リコンフィギュラブルシステム, オーム社刊, (2005)
- [5] B. DuCharme: *Learning SPARQL Querying and Updating with SPARQL 1.1*, (2013)