

Web Linked Data の Android 系組み込みシステム設計への適用

大橋 正^{†1}

初期のコンピュータのハードウェア設計段階では文書に記載された回路図を基として論理ゲート間を配線でリンクした Linked logic が適用されていた。そのコンピュータは人とワイヤーを介して操作パネルでコミュニケーションを行い運用・保守を行った。その後インターネットを介し、コンピュータと人がコミュニケーションをなして運用・保守をした。一方、コンピュータの設計図面等の文書は紙媒体で閲覧・管理されて以来、HTML などのハイパーメディアで閲覧・管理をするようになった。更に今日ではクラウドの普及と Android のスマートフォン等の出現により、利用者が固定しているか否かに関わらず多種多様な電子機器に関わるハードウェアやソフトウェアの運用、保守、閲覧、管理等の情報を Android はインターネットにより膨大なビッグデータとして取込める様になってきた。この膨大なビッグデータから、必要な Android のハードウェア・データやソフトウェアデータを Android で入手し、運用、保守、閲覧、管理等を従来よりも高機能化する方法として Web Linked Data を導入し、最適なシステム設計に役立たせる期待が生じてきた。本論は Android を情報システムと組み込みシステムの両面から捉えて設計する eALD (embedded Android with Linked Data) システムを念頭に置いて、Web Linked Data の Android 系設計への適用を考察する。

Application of Web Linked Data to Android Embedded Systems' Design

TADASHI OHASHI^{†1}

In first stage, a computer is comprised of discrete components as well as devices which are linked by wires based on linked logics. The computer in maintenance by personnel thru linked panels from the computer. On the other hand, linked logic is drawn on the papers and maintained as paper media. In the next stage, this drawing documents including linked logic were substituted by HTML documents on the web. In process of time, this web includes web data of various kinds of data and become big data. An eALD (embedded Android with Liked Data) System imports web big data such as Android's hardware data and firmware (embedded software) and Android application software to operate, maintain, refer, and manage systems. For this purpose, an eALD are expected to be applied web linked data. This paper takes eALD in two aspects of (1) information process system and (2) an embedded sys into consideration.

1. はじめに

1.1 情報システム技術の背景

初期のコンピュータのハードウェア論理設計では各種の演算素子等、の電子部品を紙媒体上に記し、回路図や仕様書等の文書を作成し、その回路図を基に配線等でリンクした論理回路(Linked Logic)の設計をしていた。そのコンピュータの運用や保守に作業者が操作パネルにより直接介入をしていたが、後にミニコンがコンピュータと人間との間に介在した間接的な運用・保守に変わった。やがて Linked Logic の一部は記憶素子を用いて、ソフトウェア(ファームウェア)で代替させる組み込みシステムへと発展した。更にインターネットの出現によりコンピュータと人間がネ

ットワークでリンクをして運用・保守するようになった。この結果、運用・保守の作業能率を飛躍的に向上させた。一方、仕様書等の紙媒体は HTML(Hyper Text Markup Language)等の電子文書に置き換えられ、インターネットを介して世界の各工場へ提供されるようになった。

この HTML は 1993 年にスイスの CERN(欧州原子核研究機構)にて勤務をしていた Berners-Lee⁰¹⁾が考案したもので、CERN の研究者同士が効率よく情報共有をさせるために文書間をリンクで関係つけた方式であった。

1.2 組み込みシステム技術の背景

一方コミュニケーション・ツールの携帯電話は Android²⁻⁴⁾を搭載したスマートフォンに置き換わった。やがて HTML⁵⁻⁶⁾の出現により通常の PC 等でブラウザ動作することに加えてスマートフォンの Android 上でも動作するハイパーテキストとなり、インターネットを介してコミュ

^{†1} アイリクト iLICT Co.
(Homepage URL <http://www.ilict.jp/>, Mail Address ohashi@ilict.jp)

ニケーションを飛躍的に向上させる可能性が出てきた。

2007年にGoogleなどの数社が規格団体(Open Handset Alliance :OHA)を立ち上げ、Androidを携帯電話用プラットフォームとすることをOHAが発表し、初版のAndroid 1.0からAndroid4.2.1(2012.11.27現在)にまで至っている。詳細は別紙²⁾を参照されたい。

Androidはインターネットのクラウド上に存在する膨大な設計情報、顧客情報等のビッグデータをウェブ上で意味づけをしてセマンティックや⁷⁾にリンクさせたLinked Data⁸⁻⁹⁾として捉えることにより、グローバル空間でビッグデータのアクセスを容易にならしめる「情報システムとしてのAndroid」とあらゆるハードウェアまたはソフトウェアの束縛に柔軟的に対応できる「組込みシステムとしてのAndroid」の両面でも普及する可能性があるかと察する。

本論では、以下の6項目に絞って考察を進める。

- (1) 情報システムと組込みシステムとしての現状課題を列挙する。(第2章)
- (2) 上記(1)の各々の課題解決を検討する。(第3章)
- (3) 課題の解決に向けてeALDシステムを提唱する。(第4章)
- (4) eALDシステムへの適用を検討する。(第5章)
- (5) 日本アンドロイドの会「4周年記念講演」¹¹⁾になされたAndroidへの4つの期待を基として「Androidシステム設計に於けるeALDシステムへの適用」をまとめる。(第6章)
- (6) 最後に今後の課題を述べる。(第7章)

2. 現状技術と現状課題

現状技術と現状課題を考察する上で、情報システムと組込みシステムの二面からアプローチする必要がある。Androidは携帯電話の機能やインターネット接続デバイスの役割を果たす広義の情報システムとAndroidが各種センサを具備した携帯機器としての広義の組込みシステムの二面を捉えて論ずる。

2.1 情報システムとしての課題

Androidを情報システムと捉える上では、インターネットの情報授受に注目をする必要がある。インターネットから各種マルチメディア情報を携帯機器がダウンロードし、又はアップロードを行う。この場合は各種のプロトコルを用いるがHTTP(Hypertext Transfer Protocol)などを使用する機会が多い。この際に利用するメディアがHTMLで記述されたものが多く、このHTMLについて現状技術を知る必要がある。この技術の延長線に存在するものが本論の主題でもあるLinked Open Data⁹⁾である。

2.1.1 HTMLの現状技術

先に述べたBerners-Leeが考案したHTMLは企業などで、文書作成における様式として、広く普及した。私自身、設計情報の文書類をHTMLで記述し、ウェブで開示を行った。また設計文書の管理にもHTMLは大変役に立った。このHTMLはDTD(Document Type Definition)によりHTMLのタグの定義づけが行われた。この考えを拡張して、XML(Extensible Markup Language)¹⁰⁾の標準化がなされ、HTML内のタグを設計者がXMLスキーマとして自由に定義出来るようになった。

このXMLは従来の文書を電子化するという枠を越えて家電を含む組込みシステムのデータを管理する役割を担えた。やがてオブジェクト指向の到来とあらゆる対象物クラスとその構成要素であるインスタンスを相互に区別でき、対象物とその事象の特性と関係を表現できるメタ記述言語としてRDF(Resource Data Framework)¹¹⁾が出現しweb linked dataの出現へと導いた。

2.1.2 Web linked dataの出現

このXML表現ではシステムの対象とする自然界を十分に表現できないので、更にRDFに付随する語彙記述言語としてRDFs(Resource Data Framework

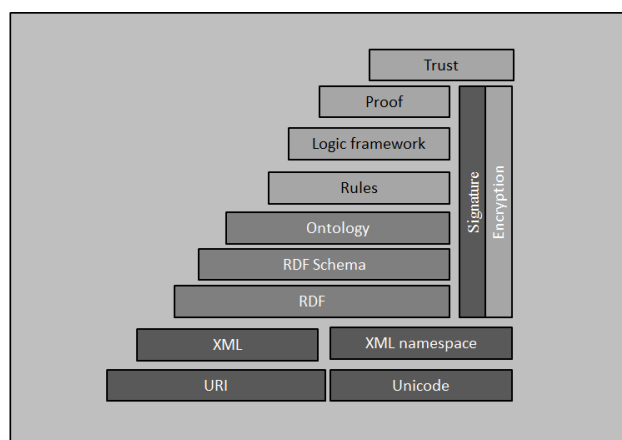


図1 web linked dataの階層構造

Figure 1 Structural Layers of Semantic web.

Schema)¹²⁾が出現した。二つのクラスが同じインスタンスを持っていても異なるクラスであることがあるこの解釈の補完を行うのがRDFsである。更にはオントロジー記述言語としてOWL¹³⁾が追加された。この様にしたクラス間の結合関係をオントロジー¹⁴⁾と呼ばれた。このオントロジーを用いて万象の実世界をどのように関係しているかを表現することが可能となった。以上の関係を図1にWeb Linked Dataの階層構造として示す。

2.1.3 HTML5 の概要

HTML の登場後はウェブの技術進歩に伴い、HTML のグレードアップを余儀なくされてきた。更に将来に向けた改訂版が必要になり HTML5 の WHATWG(Web Hypertext Application Technology Working Group)によって2008年にドラフトが提出⁶⁾された。現在、2014年までの正式勧告を目指して策定中。改訂の主要目的のひとつとして「人間にも読解可能でコンピュータやデバイス(ウェブブラウザ、構文解析器など)にも矛盾せず読解されるとともに最新のマルチメディアをサポートする言語に向上すること」である。この点は本論で注目する点である。尚 HTML5 では HTML4 だけでなく XHTML1¹⁵⁾ や DOM2HTML¹⁶⁾ (特に JavaScript) も加える予定である。

2.1.4 HTML5 の特徴⁰⁶⁾

W3C の HTML5 Logo¹⁷⁾ では以下のカテゴリを HTML5 に含めている。本論では特にセマンティクス及びモバイルアプリケーションに注目して考察をする。

(1) セマンティクス

HTML5 の新タグとして RDFa¹⁸⁾、マイクロデータ¹⁹⁾、マイクロフォーマット²⁰⁾ が追加された。RDFa は先に述べた RDF を更に簡単にさせてメタ記述を XHTML1 に組み込め可能とさせた。マイクロデータは機械に理解させる為のメタ情報を直接書き込みを可能とさせた。マイクロフォーマットは先に述べた RDF を更に簡単にさせてメタ記述を XHTML1 に組み込め可能とさせた。マイクロデータは機械に理解させる為のメタ情報を直接書き込み可能とさせた。マイクロフォーマットは HTML で使われていた class や rel の属性を用いてメタ情報を直接書き込み可能とさせた。この結果、Android に於いて、HTML5 は既にセマンティクスが行える環境にある。

(2) モバイルアプリケーション

モバイルアプリケーションで「Web アプリケーションでは様々なハードウェア仕様に対応する必要がないが Android では1つのアプリで様々なハードウェア仕様に対応する必要がある。この点は Android を後述の組み込みソフトウェア(ファームウェア)の実装を行うことでこの問題を解決できるのではないかと察する。一方で各端末向けの公式アプリストアで配布されるネイティブ・アプリにもメリットはあるため、2012年現在では置き換えが大きく進んでいるわけではない」とのコメントに着目し、Web Linked Data の Android への適用の検討課題になると考える。

2.1.5 Linked Data の現状技術

前述 2.1.2 で敷衍したウェブ上の Linked Data はビッグデ

ータ若しくはクラウドのオープンデータとして利用者に提供される。前述した HTML は設計ドキュメントなどをウェブ上で扱い、利用者がアクセスした。この場合はドキュメント間が単にハイパーリンクされて結合された web グループ結合であった。相互の接続は URL (Uniform Resource Locator) で接続されていた(図2の左)。相互の関係付けは HTML の製作者の意図したデータ結合構造であった。一方ハイパーリンクで接続された RDF data 群に意味づけを付与することで(図2の右)、ハイパーリンクに直接接続関係がなくても RDF data 構造群と同じ意味を持つデータを紐解くことで予測しえなかった、未知の新たな情報、技術、データ等を入手し、利用することができる方式がウェブの Linked Data である。次期の web3.0 はこの拡充を目指したものである。

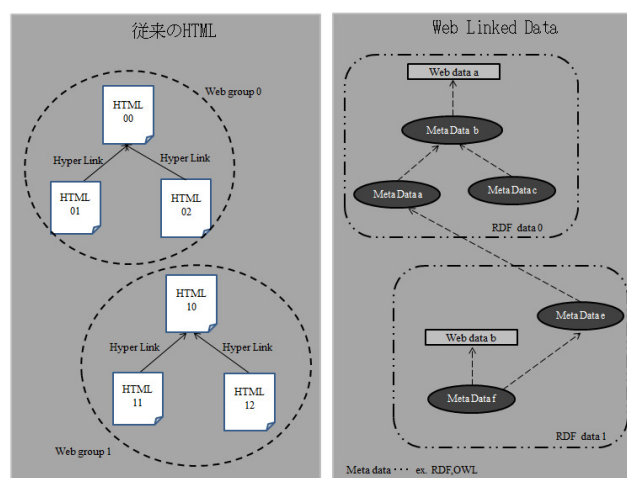


図 2 HTML 構造と Web Linked Data 構造
 Figure 2 Structural Architecture of HTML and Web Linked Data.

バーナズ・リーはグローバルなデータ空間の一部としてデータを公開する際の規則として以下の4項目を列挙している⁹⁾。

- (1) ものの名前として URI (Uniform Resource Identifier) を使用すること。
- (2) 名前について調べられるように HTTP URI を使うこと。
- (3) URI を調べた時に、RDF や SPARQL²¹⁾等の標準技術を使って有益な情報を提供すること。
- (4) 他の URI を含めることによって、より多くのものを発見できるようにすること。

2.1.6 Linked Data の課題

情報システムとしての Linked Data の課題は、Android システムを情報システムとして捉えておらず、Linked Data の適用に Android の世界に受け入れられていないか、研究案件の創出に至っていないことである。Linked Data の中

心的理論は機械に理解させる意味(セマンティック)を対象オブジェクトに付随することに原因の端を発している。Androidにはセマンティックが無いし、考にも及びつかないことであろう。

本論で扱う Linked Data の具体的課題は LOD であるが、期待される LOD の今後の技術はウェブ上の天文学的膨大な LOD が公開された場合の最適オントロジーの LOD 検索エンジンが必要となる。この検索エンジンは誰もが容易に Android のハードウェア・ソフトウェアの設計に役立つツールまたは API 化されて提供する技術が必要となる。更には不必要にリンクされてきたデータを排除する技術も必要になる。更に加えてオープンといえどもオープンデータに冗長性ができて相反する linked data 結果やバッティングするデータがリンクされた場合どのリンクを信じるのかの確信度又は信頼度の重みづけのリンクを扱う技術も必要となると考えられる。

2.2 組込みシステムとしての課題

前述 1.2 で述べた Android の出現時には携帯電話の OS として出現した。そして今日では Android がスマートフォンに実装されることによりウェブにぶら下がったデバイスという概念が定着している。しかし Android システムが単なるデバイスでなく、固定型若しくは移動型のユビキタスな各種計測システムの役割を担ってきた。この点を踏まえ Android を組込みシステムとして捉え、課題を列挙する。

2.2.1 Android システムの現状技術

今までに自分が設計した組込みシステムには real time OS が実装された。当時は自らがアセンブラで設計し、組込みシステムのカーネルとして実装した。その後、小型のリアルタイム OS が市販され、システムに実装することができた。その延長線上に組込みシステム用に OS として Android が実装できるようになった²²⁾。Android はスマートフォンの OS としては一般的に知られているが、この分野以外では家庭電化製品、各種計測制御システム、入出力周辺機器、航空宇宙機器等の組込みシステムとして Android が使用される。

2.2.2 Android システムの課題

現時点での Android システムは OS やアプリケーションが、既に携帯電話などの携帯機器に予め実装しているか又はインターネットからアプリケーションをダウンロードして追加実装する。しかし現時点ではアプリケーションが OS によって分類され、または携帯機器のハードウェア構成により種別しなければならないという課題が存在する。従い、

以上の課題を解決しなければならない。

3. 課題の解決に向けて

3.1 情報システムとしての解決

3.1.1 Android と HTML5 の解決上の注意事項

日本 Android の会 2012 年月定例会で述べられた「アプリ開発の動向」²³⁾で、web が以下の変遷と将来に向かうことが述べられた。

- (1) 第一世代 Static web
- (2) 第二世代 Dynamic web
- (3) 第三世代 Structured web
- (4) 第四世代 Real-time Responsive web

この講演会での助言として「Facebook アプリが、iOS、Android とともに、HTML5 からネイティブ・アプリに移行したことが話題になっている。ただ、HTML5 がネイティブかというふうに、単純化して問題をたてると、現在進行中の変化のポイントを見失うと思う」に注意しながら以下の Android と Linked data について述べる。

3.1.2 Linked Data と Android への解決

今まで述べてきた Web Linked Data の単なるインターネット上の文書やデータを意味づけして結び合わせることで Linked Data の考えを実現できる。先に述べた RDF 等を用いて対象物の実体の所在(URI)とその意味と関係をリンクして表象させるテクノロジーである。

この結果、従来の HTML では固定された情報源を組み合わせただけでしかなかったが、linked data を使った Android アプリケーションや Android 組込システム(ファームウェア)は制限のないグローバルでダイナミックなデータ空間上で Android システムが動作することが可能とさせて問題の解決を図る。

Android では個別にアプリケーションを Java で走行させるための Dalvik が実装されていることである。Dalvik がソフトウェアで実現されようが FPGA のハードウェアで代替させようが、今まで Java を中心とした web の情報システムでしか運用されていなかった。しかし LOD を始めとするセマンティック処理が Java 走行のできる Android により携帯機器で動作する LOD 運用の可能性が拓けてきたので問題の解決を試みる。

前述の 2.1.6 で述べた Linked Data の課題の解決にはかなり研究期間が必要と思われる。現時点では図 2 の RDF メタ・データに着目して Android に必要なウェブ・データを RDF を辿って意味づけ検索を行い、所与の

Android 組込ソフトウェア(ファームウェア)やアプリケーションデータを入手する為に SPARQL 等を利用し, Android に取り組んで利用させる解決手段を講ずる.

3.2 組込システムとしての解決

OS の種別や携帯等の機種多様性を吸収するには組込みシステムをファームウェアで実装し, 固定のハードウェアに多種のファームウェアを書き込むことでユニバーサル・OS ハブ(エミュレーション)を構築し多種の OS 実行の実現を目指し, FPGA 等を実装して解決を図る. 以上の解決をなす eALD システムを提唱する²⁴⁾.

4. eALD システム

第3章の課題の解決に沿って本論の提唱する web linked data を基とした Android 組込システム, 即ち eALD システムのアーキテクチャについて述べる. 但し eALD システムは種々の形態をとると考えられるが, 2.2 項の組込みシステムの課題を念頭に置き, 本論ではスマートフォン等の携帯電話の例を挙げて記述する.

4.1 eALD のアーキテクチャ

eALD システムの設計思想は Yield(イールドと発音し, その意味は「・・・を産する」とか「生じる」という語彙の発音を充てたところにある. つまり eALD システムはウェブ上のグローバル空間上の LOD を融合し, 今までに思いもよらなかった情報や知識を Android へ取込み, かつ書き込み可能な Android の組込みソフトウェア又はアプリケーション・プログラムが所与の携帯機器のバーサチルな OS とで全く新たな機能をイールドさせることを可能とさせるシステムを目指している.

図1に eALD のアーキテクチャを図示し, 各機能について列挙する.

(1) LOD(Linked Open Data)

政府, 官公庁, 地方自治体, 各種事業団体等が提供するオープン・データをビッグデータ又はクラウドとして利用者に提供される.

(2) Sensors

携帯機器に実装される又は外付けされる各種センサである. 実装されるセンサは加速度センサ, 地磁気センサ, マイクロフォン等がある.

(3) 出力デバイス

スピーカ, 液晶画面 等がある.

(4) 組込みソフトウェア

Android, ライブラリーズ, Android ランタイム, ライブ

ラリ, アプリケーション・フレームワーク及びアプリケーションから構成される. 詳細は 4. 3 を参照されたい.

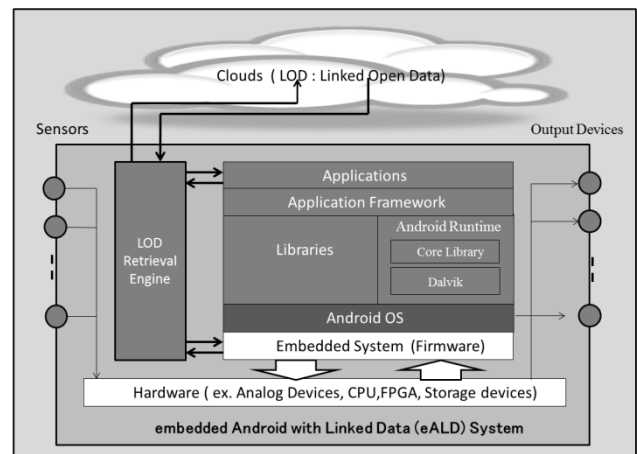


図3 eALD システムのアーキテクチャ

Figure 3 Structural Architecture of eALD.

(5) LOD Retrieval Engine

前述 2.5.1 でバーナーズ・リーが web Linked Data で URI を調べた時に, RDF や SPARQL²¹⁾等の標準技術を使って有益な情報を提供する必要性をルールとして述べている.

(6) Embedded System

eALD システムの中核をなすものでスレージ・デバイス等に格納されて CPU や FPGA で実行されるダウンロード書き込みのできる組込みソフトウェア(ファームウェア)である.

4.2 eALD のハードウェア

上記 4.1 の図3「eALD システムのアーキテクチャ」と 4.3.1 の図4「eALD システムのソフトウェア」に於いて白塗りボックスでハードウェアを図示した. 携帯機器の収納ケースである本体(ボディ)に実装される各種部品から構成される. 例えば CPU, FPGA, ストレージ・デバイス等があげられる.

4.3 eALD のソフトウェア

以下に Android ソフトウェアのアーキテクチャ³⁾を述べる.

4.3.1 基本ソフトウェアのアーキテクチャ

図4の如く Linux カーネルをベースに多種の機能を具備したネイティブライブラリ群や Dalvik と呼ばれる仮想マシンが実装され, この仮想マシン上で Android のアプリケーションが実行されるアーキテクチャである. アプリケーションからデバイスを制御するシーケンスはアプリケーションからフレームワークの API(Application Programming Interface)

をコールし、そこからライブラリ層のミドルウェアがコールされ、ミドルウェアから Linux カーネルを通して、デバイスドライバからデバイスが制御される

以下、図. 4 について解説³⁾をする。

(1) アプリケーション(Applications)

Android プラットフォームに標準装備されている電子メールクライアント、ブラウザ、オーディオアプリケーション、地図アプリケーション、電話帳などの基本アプリケーション、更にはバンドル・アプリケーション、ユーザがダウンロードして実装するダウンロードアプリケーションなどがある。

(2) アプリケーション・フレームワーク (Application Framework)

アプリケーションを起動したりウィンドウを表示したりするためのライブラリである

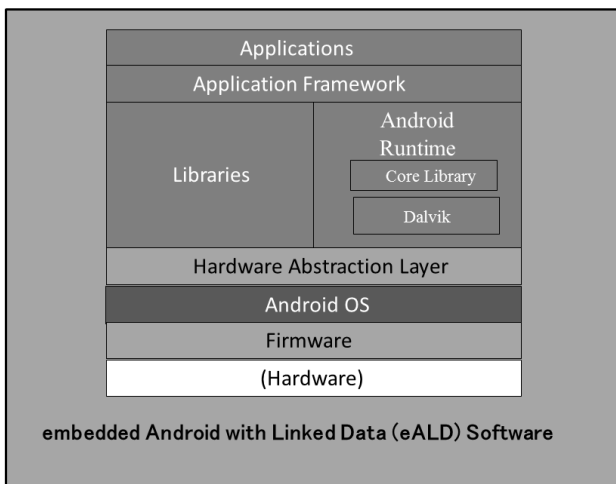


図 4 eALD システムのソフトウェア

Figure 4 eALD System Software.

例えば Active Manager, Window Manage 等がある。更に Android アプリケーションを容易に作成するための API が用意されている。Android アプリケーションはこのアプリケーション・フレームワークの API から、ライブラリ層以下のミドルウェア群の機能を使用することができる。

(3) Android ランタイム(Runtime)

Android は Java SE(Standard Edition) を元に作成されており、Java SE で利用できるコア・ライブラリの大部分は含まれている、しかし GUI(Graphical User Interface)に関する Java AWT(Abstract Window Toolkit)パッケージなどは削除されている。Android 固有の GUI を補う Android のパッケージや、センサやカメラなどのハードウェアを制御するためのパッケージが追加されている。

(4) Dalvik 仮想マシン(Virtual Machine)

Java の仮想マシンと同様の機能を提供しているがバイトコードそのものとは互換性はない。Android アプリケーシ

ョンは Java で記述され、コンパイルされて Java バイトコードが生成されて、dx のツールで Dalvik Executable(.dex) 形式へ変換される。Dalvik 仮想マシンは実行形式のファイルをプロセス単位で実行する。尚、Dalvik 仮想マシンはハードウェアでも代行できるが²⁵⁻³⁰⁾、本論ではソフトウェア方式を考える。

(5) ライブラリ (libraries)

Java 言語や C 言語、更には C++ 言語で作成されたライブラリで構成されている。Linux の libc や SQLite, webkit などのオープンソースのライブラリを Android 仕様にカスタマイズし、各種機能を実現するための汎用的なミドルウェア群として構成される。

このライブラリには SQLite, Webkit, OpenGL ES, 等がある。

(6) HAL (Hardware Abstraction Layer)

Hardware Abstraction Layer はカメラや GPS, 携帯電話の無線関係の処理部分をラップ(wrap)して、デバイス・ドライバとライブラリ間のやり取りを仲介する。

(7) Linux カーネル(Kernel)

Android は Linux ベースのプラットフォームであるために、Linux カーネル層が存在する。Linux カーネルの版数はバージョン Android 4.2.1(前述) が採用されている。セキュリティ、メモリ管理、プロセス管理、ネットワークスタック、デバイス・ドライバなどのコアシステムのサービスを提供している。このカーネルはハードウェアと上位階層との抽象化レイヤとしても機能している。

(8) ファームウェア (Firmware)

ストレージ・デバイス等に格納された組込ソフトウェアであり、CPU や FPGA で演算を行い、OS のエミュレーション、論理ゲートの制御や種々のネーキッド(Naked)なハードウェア制御を行う。

(9) ハードウェア

物理的なハードウェアの層であり、CPU, FPGA(Field Programmable Gate Allay), メモリ, ディスプレイデバイス, Bluetooth デバイス, GPS(Global Positioning System) デバイス, オーディオデバイス, バッテリ等から構成される。

4.3.2 ハードウェアのファームウェア化

ここで携帯機器の一例としてスマートフォンを図. 5 に挙げ、最上位にはスマートフォンの実体を示した。白抜きボックスはハードウェアを示し、黒塗りボックスはソフトウェアを示している

この例ではハードウェアとソフトウェア(組込みソフトウェア)の両面から構成される。オントロジーを常にこの両面で捉え検討することを持論としている。ここでハードウェアとソフトウェアの両面を同時に表現することでソフトウェアの機能がハードウェアで置き換えられる可能性を暗

に示している。

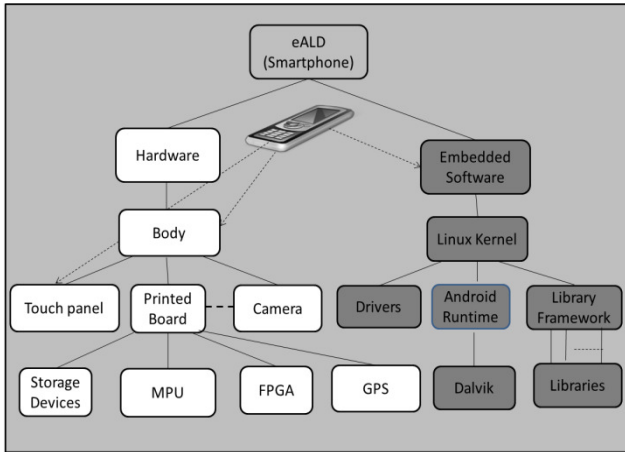


図 5 web linked data のオントロジーの例

Figure 5 Example of ontology of web linked data.

例えばソフトウェアで実行していたものを FPGA と CPU のハードウェアの動きにエミュレート出来る可能性が在ることを示している。

4.3.3 LOD 検索エンジン

当該 LOD 検索エンジンの機能は以下の各項を充足するものである。

(1) 2. 1. 5 項で触れたバーナズ・リーの要求するグローバルなデータ空間で URI を調べた時に、RDF や SPARQL²¹⁾等の標準技術を使って有益な情報を検索できるものとする。

(2) 不必要にリンクされてきたデータを排除する技術を具備している。更にオープンデータに冗長性ができて相反する linked data の検索結果が得られ、バッティングするデータがリンクされた場合にどのリンクを信じるのかの確信度又は信頼度の重みづけのリンクを扱うことができる。

5. eALD システムへの適用検討

5.1 情報システムとしての Android への適用

(1) 既に HTML5 が適用される時代になって、セマンティックが即ち Android にも適用環境にある。その適用は文書管理分野が考えられる。特にハードウェア及びソフトウェアの設計図面から出てくる BOM(Bills of Materials)³¹⁾のハードウェア及びソフトウェア部品表は Web Linked Data を最も早く適用できる分野と考える。

(2) ウェブ上のデータソースをオープンにすることで LOD(Linked Open Data)として官公庁や諸団体の公文書やデータの開示し、モバイルコマースに於ける web linked data

のオープンデータの適用が考えられる。特に商業行為は法制度、消費税の段階的適用等の管理下で営まれるので Android に有効であると考えられる。

(3) TPP(環太平洋経済協定)の段階的自由貿易協定施行に伴うモノづくり等の分野に適用できると考える。相手国の基準値に従った Android システムの動作に関して中央政府や所轄官公庁から提供される Web Linked Data に基づく LOD を取り込むことでオープンデータに準拠した Android の動作が可能になると考える。

以上詳細に述べることができないが多種多様の方面で Web Linked Data が Android へ適用できると考えられる。

5.2 組込みシステムとしての Android への適用

(1) Android に於いても積極的にハードウェアの制御やファームウェア又はソフトウェアの版数管理、更にはエミュレーションの管理などに適用できると考える。BOM は何も部品管理だけではなくたとえば Android の版数に応じたデータ類の自動ダウンロードや修正データの自動適用も可能できると考えられる。

(2) Android では1つのアプリで様々なハードウェア仕様に対応する必要がある。この点でセマンティックスが利用できないかと考える。一つのアプリが全ての Android ハードウェアに適用できるユニバーサル・OS ハブの役目ができるのではないかと考える。

(3) Android はモバイル環境にて動作することから GPS の位置情報に基づく Android アプリなどの運用環境に Web Linked Data が利用できると考えられる。モバイルユーザの目的指向と位置情報から生じる定常若しくは非定常プロセスのモデリング³²⁾にセマンティックウェブ³⁾の効果が期待できるのではないかと考える。

(4) 更には Android を OS として用いた FPGA などのシステムでは、ゲートロジックデータをインターネットから提供される Linked data で制御させるのも有効な方法といえる。

6. まとめ

(1) 「アプリ開発の動向」²³⁾の中で述べられた「HTML5 の単純化による本質の見落とし」に注意すれば、既に Web Linked Data が Android のハードウェア及びソフトウェア設計に適用できる分野がある。

(2) Web Linked Data は文書管理だけでなくデータの管理に適用できる。

(3) 従来から組込みシステムのハードウェア制御を C 言語系であったが Dalvik のハードウェア化により Java 言語でのハードウェア制御が可能となってきた。

(4) Android のアプリが Java 系が中心となることで従来の PC で動作していたセマンティックな各種 Web プログラミングが Android へ今後更に浸透していくと予測できる。つまり Android で web2.0 として HTML5 や JavaScript の動作が進展し、更に今後の web3.0 でセマンティック処理が Android で実現し易くなると考えられる。

(5) ハードウェアのゲートアレーの制御をある条件に応じて制御する分野にも使える。

7. 今後の課題

「Web Linked Data の本質は人間が読み操作する static な web を脱皮して機械が読めて判断し行動できる web」であるから、Android の研究ミッションに合致するのではないかと察する。これからの Android マシンは、ウェブ上のセマンティックを解釈し、eALD システムが自発的に行動できるシステムの実現を視野に入れて Android ハードウェア・ソフトウェアの設計を行うことがミッションであると察する。このミッションに基づき私は「アンドロイドの夢を見て」³³⁾を座右の銘として、eALD システムの実現に向かって活動して行きたい。

参考文献

- 1) 斎藤信男, 萩野達也, 他監修: semantic web 入門, p. 2 オーム社 (2003).
- 2) Android:
<http://ja.wikipedia.org/wiki/Android>
- 3) 木島貴志, 石丸宗平: アンドロイドプログラミング入門, 情報処理, Vol. 52, No. 4・5, pp. 527-539 (2011).
- 4) 塚本昌彦: スマートフォンプログラミング 2, 情報処理, Vol. 53, No. 1, pp. 6-7 (2012).
- 5) HTML 5.1 A vocabulary and associated APIs for HTML and XHTML: W3C Working Draft 17 December 2012” <http://www.w3.org/TR/html51/>
- 6) HTML5
<http://ja.wikipedia.org/wiki/HTML5>
- 7) Semantic Web
<http://www.w3.org/standards/semanticweb/>
- 8) Tom Heath, Christian Bizer “Linked Data: Evolving the Web into a Global Data Space” Morgan & Claypool Publishers (February 2011)
<http://www.morganclaypool.com/>
- 9) Christian Bizer, Tom Heath, Tim Berners-Lee 萩野達也 訳: Linked Data の仕組み, 情報処理, Vol. 52, No. 3, pp. 284-292, (2011)
- 10) Extensible Markup Language (XML), <http://www.w3.org/XML/>
- 11) Resource Description Framework (RDF) <http://www.w3.org/RDF/>
- 12) RDF Vocabulary Description Language 1.0 RDFS Schema
<http://www.w3.org/TR/rdf-schema/>
- 13) Web Ontology Language (OWL)”
<http://www.w3.org/2004/OWL/>
- 14) Vocabularies
<http://www.w3.org/standards/semanticweb/ontology>
- 15) XHTML™1.0 The Extensible Hyper Text Markup Language (Second Edition) A Reformulation of HTML4 in XML1.0
<http://www.w3.org/TR/xhtml1/>
- 16) Document Object Model (DOM) Level 2 HTML Specification
<http://www.w3.org/TR/DOM-Level-2-HTML/>
- 17) HTML5 Logo
<http://www.w3.org/html/logo/>
- 18) RDFa
<http://www.w3.org/TR/xhtml1-rdfa-primer/>
- 19) HTML Microdata Nightly
<http://www.w3.org/html/wg/drafts/microdata/master/>
- 20) Microformats in HTML5
<http://microformats.org/wiki/html5>
- 21) SPARQL 1.1 Query Language
<http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>
- 22) 金丸隆志: 組み込み OS としての Android の利用, 日本 Android の会 Android Bazaar and Conference 2011 Summer
- 23) 丸山不二夫: アプリ開発の新しい動向, 日本 Android の会 2012 月定例会 http://www.youtube.com/watch?v=1GEE456_QKk
- 24) 大橋 正: Android システム設計に於ける Semantic web の検討, Android Bazaar & Conference 2013 Spring/Tokyo (2013).
- 25) 小池恵介他: Android における Java アプリケーションの FPGA アクセラレーション, 情報処理学会論文誌, Vol. 53 No. 12 pp. 2740-2751 (2012).
- 26) Arnold, M., Fink, S. J., Grove, D., Hind, M. and Sweeney, P. F.: A Survey of Adaptive Optimization in Virtual Machine, Proc. IEEE, Vol. 93, No. 2, pp. 449-466 (2005).
- 27) Lattanzi, E., Gayasen, A., Kandemir, M., Narayanan, V., Benini, L. and Bogliolo, A.: Improving java performance using dynamic method migration on fpgas, Proc. 18th International Parallel and Distributed Processing Symposium, p. 134 (2004)
- 28) 太田 淳, 三輪 忍, 中条拓伯: Dalvik アクセラレータ: Android 端末における Java アプリケーションの高速実行機構, 組み込みシステムシンポジウム (ESS2010), pp. 13-22 (2010).
- 29) 太田 淳, 三輪 忍, 中条拓伯: Android 端末におけるハードウェアによる Java の高速化手法の提案, 情報処理学会論文誌 コンピューティングシステム Vol. 4, No. 3, pp. 115-132 (2011).
- 30) Christiaens, M. and Stroob, D.: Interfacing java with reconfigurable hardware (2004).
- 31) 川村隆浩, 長野信一: ビジネスに使えるオントロジー関連ツール, 人工知能学会誌, Vol. 25, No. 3, pp. 345-353 (2010).
- 32) 来村徳信, 笹島宗彦, 溝口理一郎: 目的指向プロセスのオントロジー的共通性に基づいた人工物機能とモバイルユーザ行動のモデリングとその応用, 人工知能学会誌, Vol. 25, No. 4, pp. 526-536, (2010).
- 33) 黒崎政男: 哲学者はアンドロイドの夢を見たか, 哲学書房 (1987)