

# 組み込みサーバにおける車載アプリケーション間通信機構の実装とその機構を利用した位置情報の取得手段

石渡 要介\* 福田 和真† 峯村 治実‡ 鷹取 功人§ 田中 聡¶  
三菱電機株式会社

## 1 はじめに

車載情報システム上のアプリケーションで扱われる情報には様々なものがある。その中でも、車 (移動体) に特徴的な情報である位置、移動情報を扱うことは重要である。本稿では、弊社で開発している OSGi Release2 準拠組み込みサーバ TSUBASA 上への、OSGi Release3 仕様に含まれる位置情報を扱うための機構の実装について述べる。

## 2 OSGi Service Platform 及び OSGi における位置情報仕様

### 2.1 OSGi Service Platform

OSGi (Open Services Gateway Initiatives) Service Platform [1] は、サービスやデバイスの管理をネットワーク越しに行なうための規格で、アプリケーションのインストール・起動・停止・更新などを行なうための Java API を規定したものである。

現在では Release3 (以下、R3) が定められている。

### 2.2 位置情報クラスと位置情報のやりとり

R3 では、OSGi Service Platform で位置情報を扱うためのクラスである Position クラスが定義されている。このクラスでは、緯度・経度・高度・速度・進行方向の値を含み、それぞれの値が Position クラスで定義されたメソッドを用いて取得できる。

それぞれの値は、同様に R3 で定義された Measurement クラスを用いて表現される。Measurement クラスは、double で定義された値と誤差と、単位を示す Unit クラスから成る。Unit クラスは SI 単位系を元に定義さ

れたクラスで、これにより、個々の値の誤認識 (単位系を間違えることによる、値の判定間違いなど) を防ぐ。

この Position クラスは、OSGi Service Platform 下で動作するアプリケーションの間でやりとりするために WireAdmin サービスを使うことを前提としている。

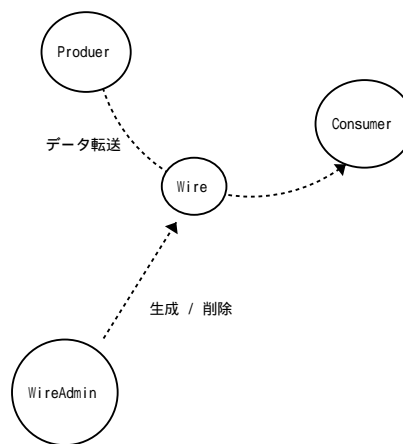


図 1: WireAdmin

WireAdmin サービスも同様に R3 で規定されているサービスで、図 1 はその概念を示したものである。WireAdmin サービスはアプリケーション間を Wire と呼ばれるオブジェクトで結びつけ、情報の一方向への流れ (情報の生成元アプリケーション (Producer) から消費先アプリケーション (Consumer) への情報の転送) に関する管理を行なうものである。ここで行なう管理とは、Producer から Consumer への値の転送要求や Consumer から Producer の値要求に対しての情報の転送、Producer から送られた複数の値 (WireAdmin では複数のクラスをまとめて送ることができるために、値の識別を行なうための Envelope インターフェースが定義されている) から Consumer が要求する情報の選択、転送された値のフィルタリングなどを含む。

この WireAdmin を利用することで、アプリケーションが Position オブジェクトを取得し、利用することが

\* isiwata@isl.melco.co.jp  
† kfukuda@isl.melco.co.jp  
‡ mine@isl.melco.co.jp  
§ takatori@isl.melco.co.jp  
¶ stanaka@isl.melco.co.jp

出来る。

### 3 実装

OSGi R3の仕様に従い、Position クラスと Measurement クラス、WireAdmin サービスを TSUBASA 上に実装した。

仕様では、前章で記述した機能の他に、Wire での処理を行なった際に発生する WireAdminEvent 及びこれを受けるためのインターフェースである WireAdminListener、Producer/Consumer が値を転送/受信するためのパーミッションをチェックするためのクラスである WirePermission、Envelope インターフェースのデフォルト実装である BasicEnvelope が決められている。

Producer, Consumer, WireAdminListener の各インターフェースは、WireAdmin を利用するアプリケーションで実装するものである。特に、Wire の接続対象となるアプリケーションでは、Producer もしくは Consumer のどちらかは必ず実装していなくてはならない。

### 4 動作例

TSUBASA 上に、Producer として、一定時間毎に方向を変化させながら一定速度で進む移動体の Position オブジェクトを生成するアプリケーションを作成、また Consumer として、その Position オブジェクトを WireAdmin 経由で受け取り、その移動軌跡を描画するアプリケーションと、Position オブジェクトを受け取りその中の要素を表示させるアプリケーションを作成した (図 2)。

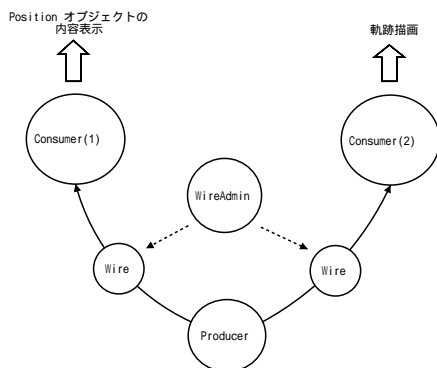


図 2: 動作例における構成

利用 JavaVM は J2SE 1.4.1\_02 で、Linux kernel 2.4.23, glibc-2.2.5 の環境下で動作させた。



図 3: 軌跡描画を行なうアプリの画面

Wire 経由で情報を扱う時、Producer と Consumer は 1 対 1 に接続するが、これらと他の Wire で接続している他の Consumer や Producer とは直接影響はしない。これにより、独立に作成された複数のアプリケーションが同じ情報を同時に扱うことを可能としており、これが WireAdmin サービスを利用する特徴となっている。

本稿では、1 つの Producer に対して 2 つの Consumer をそれぞれ独立に作成/実行し、それぞれが同じ内容の Position オブジェクトを同じ Producer から受け取っていることを調べることで、WireAdmin が目的通りの動作をしていることを確認した。

### 5 おわりに

OSGi Service Platform R3 には、位置情報を扱うためのクラスが定義されている。このクラスと、アプリケーション間での情報伝達のためのサービスである WireAdmin サービスを組み合わせることにより、アプリケーション内で位置情報を取得することや、そのようなアプリケーションを任意数作成することが可能である。本稿では、WireAdmin サービスと Position クラス、またそのために必要な Measurement クラスと Unit クラスを実装し、それらを用いたアプリケーションを作成して、実際に動作することを確認した。

### 参考文献

[1] OSGi Alliance, <http://www.osgi.org/>