C-026

OSGi Frameworkにおける Bundle共有方式の提案

A Proposal of a Bundle Sharing Method for OSGi Framework

小玉 哲平 † 高山 洋史 ‡ 小板 隆浩 † 佐藤 健哉 ‡
Teppei Kodama Youji Takayama Takahiro Koita Kenya Sato

1 はじめに

近年，OSGi Framework[1] を用いた機器連携が実現されている．OSGi Framework は，OSGi Alliance により仕様が策定されており，Java 言語を用いたサービス指向アーキテクチャベースのミドルウェアである．機器連携とは，複数の機器間で互いの機能を利用しあうことである．OSGi Framework は Bundle というプログラムモジュールを管理し，他の OSGi Framework がインストールされた機器にネットワークを介して Bundle を提供することで機器連携を実現している．OSGi Framework を用いることでサービスの更新や機能の追加や削除が容易になる．しかし，機器連携を実現する際には複数の Bundle を用いることが多い．また，既存の OSGi Framework は自身が所持している Bundle しか利用できない．この課題の有効な解決策の 1 つに複数の OSGi Framework 間での Bundle 共有が挙げられる．本稿では，LAN 内における複数の OSGi Framework 間での Java RMI(Remote Method Invocation) 技術を利用した Bundle 共有方式の提案を行う．

2 既存技術の問題点

OSGi Framework は，ネットワークに接続された機器間での連携を行う際に機器間の技術仕様の差異を吸収するミドルウェアである．例えば，OSGi Framework を用いることで UPnP[2] 対応機器と Jini[3] 対応機器でも機器連携することが可能である．OSGi Framework は，Bundle というプログラムモジュールを管理し，機器連携を実現している．Bundle には，OSGi Alliance 規定の標準 Bundle とアプリケーション Bundle がある．標準 Bundle は，UPnP 対応の Bundle や HTTP 対応の Bundle など機器の仕様の差異を吸収するために不可欠な Bundle であり，全ての OSGi Framework が所持している．一方，ネットワークに接続された機器の種類により必要なサービスが異なるため，OSGi Framework ごとに所持しているアプリケーション Bundle は異なる．例えば，エアコンの制御に必要な Bundle とネットワークの制御に必要な Bundle は異なる．OSGi Framework では，Bundle の提供するサービスを OSGi サービスレジストリとして登録・リスト化し，サービスの管理を行っている．OSGi Framework を用いる利点として，Java 言語を利用しているため OS やハードウェアに非依存であることや Bundle の再利用性を確保していることが挙げられる．OS やハードウェアに非依存であるために，今後様々な機器に搭載することも可能であり，Bundle の再利用性を確保しているため新しい機器への移行もスムーズに行える．しかし，既存の OSGi Framework では，自身が所持している Bundle しか利用できず，複数の OSGi Framework 間での Bundle 共有は実現されていない．Bundle 共有を行うことにより他の OSGi Framework が所持している Bundle を利用可能となる．また，利用する全ての Bundle を所持しなくても利用できるため，メモリ消費量削減も可能である．さらに，メモリの消費量削減ができれば，携帯電話などに OSGi Framework を搭載し機能の更新やアプリケーションの共有などが可能となる．

3 提案方式

本稿では，LAN 内における複数の OSGi Framework での標準 Bundle の共有方式の提案を行う．OSGi サービスレジストリから Bundle 共有するための Bundle の提供するサービスのリストである RMI サービスレジストリを作成する．この RMI サービスレジストリを複数の OSGi Framework 間で共有・同期することによって Bundle 共有を実現する．各 OSGi Framework は RMI サービスレジストリから利用したい Bundle のサービスを発見し，Java RMI 技術を用いてその Bundle のサービスを利用する．Java RMI 技術は，他のコンピュータの Java オブジェクトのメソッドをネットワークを介して利用できる技術である．Java RMI 技術を用いる際には 4 つのクラスを生成する．メソッドを呼び出す側にはクライアントのメインクラス，スタブクラスを生成し，メソッドを提供する側にはリモート Java オブジェクトクラスとスケルトンクラスを生成する．クライアントのメインクラスとリモート Java オブジェクトクラスは通信せず，メソッドの呼び出しや返り値を返すだけで，実際に通信を行うのはスタブクラスとスケルトンクラスである．以下に，Java RMI 技術の動作モデルを図 1 に示す．

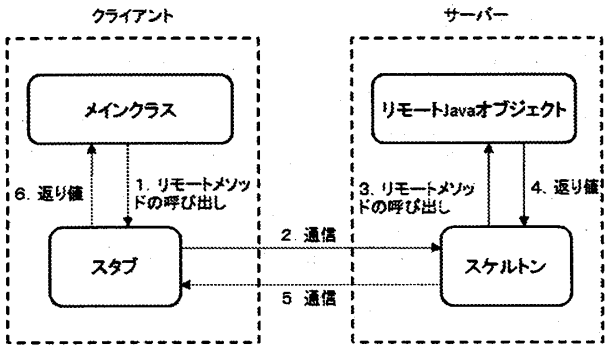

図 1 Java RMI の動作

1. メインクラスがリモートメソッドの呼び出しをクライアント側の通信を行うを行うためのスタブクラスに渡す．

† 同志社大学理工学部情報システムデザイン学科
‡ 同志社大学大学院工学研究科情報工学専攻

2. スタブクラスはネットワークを介してサーバー側の通信を行うためのスケルトンクラスと通信する．
3. スケルトンクラスはメインクラスが呼び出したメソッドのリモート Java オブジェクトクラスからメソッドを呼び出す．
4. リモート Java オブジェクトクラスはメソッドを実行し，返り値をスケルトンクラスに返す．
5. スケルトンクラスはスタブクラスと通信する．
6. スタブクラスはメインクラスに返り値を返す．

OSGi サービスレジストリ利用のために OSGi Framework は，サービスの取得，サービスの登録などシンプルなアーキテクチャを API として提供している．この API を通じて Bundle から他の Bundle がもつサービスを呼び出すことが可能となる．提案方式ではこのシンプルなアーキテクチャを維持しながら，他の OSGi Framework 上にある Bundle のサービスを利用可能にする RMI サービスレジストリの設計を行い，Bundle 共有を実現する．共有する Bundle は全ての OSGi Framework がもつ標準化された Bundle を対象とする．Bundle 共有とは他の OSGi Framework 上にある Bundle のサービスを呼び出すことである．Bundle のサービス実行のためには Java オブジェクトを通じてサービスを呼び出すことが必要となる．そこで，他の OSGi Framework にある Java オブジェクトを取得し，メソッドの呼び出しが可能な Java RMI 技術を用いる．標準 Bundle がもつ Java オブジェクトを RMI サービスレジストリに登録し，他の OSGi Framework の RMI サービスレジストリと同期を行う．以下に Bundle A が Bundle B のサービスを利用する場合の動作の流れを図 2 に示す．

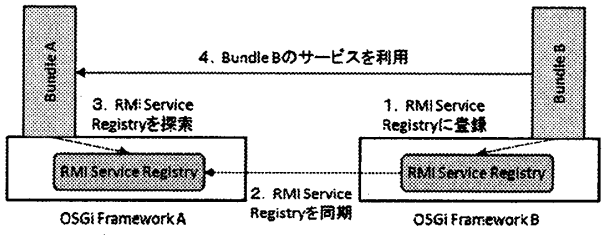

図 2　Bundle 共有の流れ

1. 新しい Bundle をダウンロードした時は RMI サービスレジストリに Java オブジェクトである Bundle の情報を登録する．
2. RMI サービスレジストリをネットワークを介して他の OSGi Framework と共有する．
3. 他の OSGi Framework 上にある Bundle のサービスを利用したい時は自身の RMI サービスレジストリを探索する．
4. 利用したい Bundle が RMI サービスレジストリに登録されている時は JavaRMI 技術を用いて利用する．

4　既存技術との比較

Bundle 共有のメリットについて考察する．まず，既存の OSGi Framework と比較した場合，提案方式を用いた OSGi Framework は，他の OSGi Framework の Bundle のサービスを利用できるので，実際には所持していない Bundle のサービスも利用することが可能となる．本提案方式のアプリケーション Bundle への応用も考えられ，より効率的な Bundle 共有が可能となる．さらに，サーバ・クライアント方式や P2P 方式での Bundle 共有方式も考えられる．

次に，利用する全ての Bundle を所持する必要が無くなるため，メモリ消費量を削減できる．OSGi Framework を携帯電話などのモバイル機器に搭載し，メモリの容量の小さな機器での利用も想定される．モバイル機器などに搭載させると機能の更新や削除が容易になる．

その他の Bundle 共有方式には，CORBA（Common Object Request Broker Architecture）[4] を利用した方式がある．本稿の Java RMI 技術を利用する場合の共有方式と類似しており，OSGi サービスレジストリの共有により Bundle 共有を実現している．この方式では，独自のインターフェース記述言語を使って外部インターフェースを作成する必要がある．Java 言語以外のプログラミング言語にも対応できるという利点もあるが，外部インターフェースを作成する部分は Java RMI 技術を用いる場合と比べてオーバーヘッドになる．また，これらの Bundle 共有方式と異なるアプローチで Bundle 共有を実現する方式に R-OSGi(Remoting-OSGi)[5] がある．R-OSGi では，プロキシを用いて Bundle の次元で Bundle 共有を行う．このプロキシが他の OSGi Framework の所持するプロキシと通信することで Bundle 共有を行う．また，このプロキシを利用するとき，自身の所持する Bundle を利用するときと区別なく利用することができる．

5　まとめと今後の課題

本稿では OSGi Framework における Bundle 共有方式について提案を行った．既存の全ての OSGi Framework は標準 Bundle を利用しなければいけない．OSGi Framework は今後，組み込み機器などのリソースの限られている機器に搭載される．また，さまざまなネットワーク技術などの登場により標準化が活発になり標準 Bundle も増加すると考えられる．組み込み機器への搭載，標準 Bundle の増加など，既存の Bundle 利用方法ではメモリなどのリソース消費の観点から見て，適切ではない．そのため全ての OSGi Framework が持つ標準 Bundle を共有する効率の良い Bundle 利用方法を提案した．今後の課題は，提案方式の実装と評価及び Bundle 共有する OSGi Framework の発見方法の検討である．Java RMI 技術を用いた Bundle 共有の有用性を示すことを目指す．また，Bundle 共有する OSGi Framework の発見方法に関しては，サーバ・クライアント方式と P2P 方式での共有の検討・評価を行う．

参考文献

[1] OSGi Service Platform Core Specification,Release4,OSGi Alliance,2005.
[2] Universal Plug and Play device Architecture Version 1.0,2000. http://www.upnp.org/resources/documents/CleanUPnPDA101_20031202s.pdf.
[3] Jini Network Technology,Sun Microsystems. http://www.sun.com/software/jini/.
[4] J2SE 5.0 での CORBA サポートの公式仕様. http://java.sun.com/j2se/1.5.0/ja/docs/ja/guide/idl/compliance.html
[5] R-OSGi:Distributed Applications through Software Modularization. Jan S.Rellermeyer,Gustavo Alonso,and Timothy Roscoe. http://www.iks.inf.ethz.ch/publications/files/rellermeyer-middleware07.pdf