

アドホックデバイスネットワークにおける仮想デバイスの実現

馬 天峰[†] 水口 孝夫[†] 永宗 宏一[†] 毛利 公一^{††} 大久保 英嗣^{††}[†]立命館大学大学院理工学研究科 ^{††}立命館大学理工学部

1 はじめに

近年、ハードウェア技術とネットワーク技術の進歩により、ユーザの周りのさまざまなデバイスが情報端末に接続されるようになった。近い将来、それらのデバイス自体がアドホックにネットワークを形成し、ユーザが自分の情報端末からネットワーク上に存在するデバイスを自由に操作し、利用することが考えられる。我々は、このような環境をアドホックデバイスネットワークと呼んでいる。このような環境においては、アプリケーションがデバイスの位置や、特徴、利用状況、デバイスを制御しているOSといった構成要素に関係なく、ネットワーク上のデバイスを利用可能とする必要がある。また、ユーザの周りに存在するデバイスやネットワークの構成が頻繁に変化し、ユーザが利用するデバイスが一定でない場合も考えられる。環境の変化に適応し、デバイスを自由に切り替えながらアプリケーションを継続することが望ましい。ネットワーク上のデバイスやサービスを容易に利用可能にするための研究として、STONE [1] や VNA [2] をはじめ、ユビキタスネットワーク [3] やウェアラブルコンピューティング分野で数多く行われている。しかし、それらの研究では、既存のアプリケーションをそのまま適応可能なものは少ない。そこで、我々は、デバイスの視点から、アドホックデバイスネットワークのためのアプリケーションを容易に作成可能とするためのプラットフォームを開発している。また、本プラットフォームは、既存のアプリケーションでも利用可能である。

本研究では、ミドルウェアにおいてデバイスを仮想化し、アプリケーションに提供する。また、ミドルウェアは、ネットワーク上のデバイスの発見や管理を行う。これによって、デバイスをアプリケーションから隠蔽し、アプリケーションはデバイスを意識せずに利用可能となる。また、デバイスを仮想化することによって、アプリケーションは多様なデバイスを統一的に扱うことが可能となる。本稿では、我々が開発しているミドルウェアプラットフォームにおけるデバイスの切替えの機能について述べる。

2 プラットフォームの全体構成

我々が開発しているプラットフォームの全体構成を図1に示す。各構成要素の機能は、以下の通りである。

仮想デバイスインタフェース デバイスの種類ごとに、アクセスするためのインタフェースをアプリケーションに提供する。

デバイスマネージャ デバイスを管理し、アプリケーションとデバイス間のデータ転送を行う。

デバイスセレクタ ユーザの要求や、ユーザの周りのコンテキストの変化に適応し、アプリケーションが利用するデバイスを決定する。

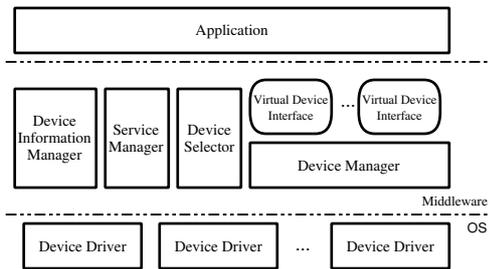


図1 プラットフォームの全体構成

サービスマネージャ 環境に応じて、ユーザに利用可能なサービスを提示する。

デバイス情報マネージャ ユーザの周りにあるデバイスを発見し、デバイスを利用する際に必要な情報を提供する。

以上の機能によって、ユーザは環境に応じて利用可能なサービスを把握することが可能となる。また、アプリケーションは、ネットワーク上のデバイスを適切に選択し、利用することが可能となる。

3 デバイスの切替え

既存のアプリケーションは、実行中のデバイス切替えを想定していないため、デバイスを切り替える際に実行を中断する必要がある。アプリケーションが利用するデバイスを切り替える度に再起動することは、ユーザの作業継続の観点から望ましくない。本プラットフォームでは、デバイス切替えによるアプリケーションへの影響を仮想デバイスで隠蔽する。これによってアプリケーションはデバイス切替えを意識せずに継続することが可能となる。

3.1 概要

本研究におけるデバイス切替え手法の概要を図2に示す。仮想デバイスインタフェースは、仮想的なデバイス

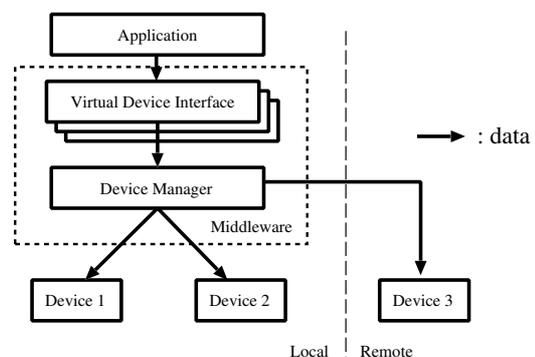


図2 仮想デバイスによるデバイス切替え手法

をアプリケーションに提供するものであり、アプリケーションからの要求を受け付ける。アプリケーションは、仮想デバイスインタフェースを通常のデバイスと同様に

Tianfeng Ma[†], Takao Mizuguchi[†], Koichi Nagamune[†], Koichi Mouri^{††}, and Eiji Okubo^{††}

[†]Graduate School of Science and Engineering, Ritsumeikan University

^{††}Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University

扱う。アプリケーションからデバイスに送信するデータは、仮想デバイスインタフェースで受信される。

デバイスマネージャは、情報端末ごとに存在し、仮想デバイスインタフェースによって必要な機能が呼び出される。デバイスマネージャは、アプリケーションから受信したデータを、実際に利用するデバイスのドライバに送信する。アプリケーションの実行中にデバイスを切り替える必要がある場合、デバイスマネージャがデータの送信先を変更することによって、デバイスの切替えを実現する。アプリケーションは、常に同一の仮想デバイスインタフェースにアクセスしているため、デバイスの切替えを意識する必要がない。

3.2 サウンド入出力デバイスにおける機能構成

サウンド入出力デバイスの例を用いて、デバイス切替え手法の詳細を述べる。Linux におけるデジタルサンプリングおよびデジタル録音デバイスとして、DSP デバイスがある。アプリケーションは、DSP デバイスを open し、デジタルサンプルデータを read/write することによって、サウンドの入出力を行う。また、アプリケーションは、デバイスファイルに対して ioctl リクエストを発行することによってデバイスの制御を行う。サウンドデバイスをミドルウェアによって切り替える場合、次の問題点が挙げられる。

- 切替え元デバイスの録音 / 再生の中止や、切替え先デバイスに対してのデータフォーマットの指定といった操作が必要となるため、ミドルウェアはアプリケーションの代わりにデバイスへの制御要求を発行する必要がある。
- デバイスからのデータの送受信速度が変化することが考えられる。また、デバイスの性能差によって、処理速度が異なることが考えられる。データの書込みが遅くなると、サウンド出力に音落ちといった問題が発生する。したがって、ミドルウェアは、データの送受信を監視し、必要に応じてデータの送受信速度を調整する必要がある。
- 切替え元デバイスに送信したデータがまだ再生されていないことが考えられる。切替え元デバイスに送信したデータの再生が中断され、続きのデータを切替え先デバイスに送信すると、データの損失が発生し、シームレスな切替えが困難となる。デバイスに送信したデータを残し、デバイス切替えの際に、残していたデータを切替え先デバイスに送信することが望ましい。

このような問題点を解決するために、サウンドデバイスに対応したミドルウェアを次のように構成する (図 3 参照)。

- 仮想デバイスインタフェース
デバイスにアクセスするためのインタフェースをアプリケーションに提供する。仮想デバイスインタフェースは、アプリケーションから音声データを受信し、バッファに保存する。アプリケーションからデバイス制御要求があった場合、これをデバイスマネージャに転送する。また、仮想デバイスインタフェースは、デバイスセレクタから受信したデバイス ID をデバイスマネージャに送信する。
- デバイスマネージャ
データコントローラとデバイスコントローラから構成される。データコントローラは、アプリケーショ

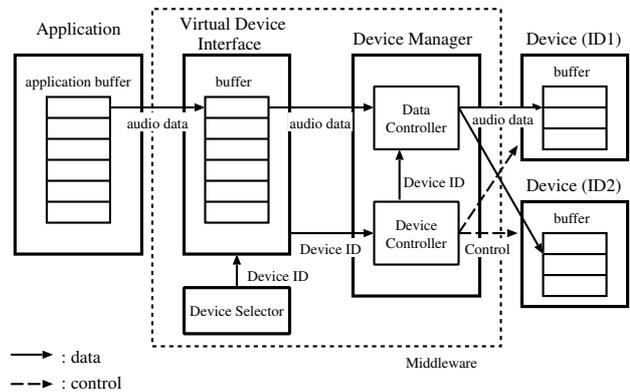


図 3 サウンドデバイスに対応したミドルウェア構成

ンとデバイス間のデータ転送を行う。データコントローラは、デバイス切替えの際にネットワークにおけるデータ転送速度の変化に対応するために、アプリケーションとデバイス間のデータの送受信を監視する。必要に応じて、アプリケーションからのデータの送受信速度を調整する。また、データコントローラは、デバイスに送信したデータのキャッシュを持つ。デバイスを切り替える場合、キャッシュに存在するデータを切替え先デバイスに送信する。一方、デバイスコントローラは、アプリケーションが利用するデバイスを監視し制御を行う。デバイスコントローラは、仮想デバイスインタフェースからデバイス ID を受信することによってデバイス切替えを判断する。デバイス切替え要求が発生した場合、デバイスコントローラは、切替え先デバイスを open し、必要に応じてデバイスの初期設定を行う。デバイスコントローラは、データの送信先デバイスをデータコントローラにセットし、切替え先デバイスのバッファの大きさに応じてデータコントローラのキャッシュサイズを調整する。また、デバイスコントローラは、切替え元デバイスの出力中止や、デバイスの解放を行う。

このような構成をとることにより、アプリケーションを継続しながらのデバイス切替えが可能となる。

4 おわりに

本稿では、アドホックデバイスネットワークを実現するために、アプリケーションの実行中にデバイスを切り替える手法について述べた。今後の予定として、他の種類のデバイスについて詳細を検討し、実装および評価を行うことが挙げられる。

参考文献

- [1] 南 正輝, 森川 博之, 青山 友紀: “ユビキタス環境におけるサービス合成支援のためのインタフェース指向ネームサービス,” 電子情報通信学会論文誌, vol.J86-B, no.5, pp.777-789, May 2003.
- [2] Jin Nakazawa, Yoshito Tobe, Hideyuki Tokuda: “A Pluggable Service-to-Service Communication Mechanism for VNA Architecture,” The 22nd International Conference on Distributed Computing Systems (IEEE ICDCS2002), July 2002.
- [3] Mark Weiser, “The Computer for the Twenty-First Century,” Scientific American, pp. 94-100, September 1991.