

超分散オブジェクトモデルを用いた機器間のアドホックな連携

川上賢一郎*
ken-ichi@sdl.hitachi.co.jp

鮫嶋茂稔*
samesima@sdl.hitachi.co.jp

河野克己*
kawano@sdl.hitachi.co.jp

計算機技術およびネットワーク技術の発展により、様々な機器を通じて実世界とコンピュータシステムを密接に接続することが可能となり、その上でのコンテキストウェアサービスが注目されている。ユビキタスコンピューティングを用いたサービス提供の実現を考える上での課題のひとつは、非均質な機器群およびソフトウェアアプリケーション群をいかにして相互に接続するかということにある。本稿では、それらの機器およびソフトウェアコンポーネント群の統合的な管理を可能とする新しいオブジェクトモデルである *Super Distributed Objects(SDO)* について述べる。また、*SDO* モデルの一実装例として、アドホックネットワークを利用した *SDO* 管理ミドルウェアについて述べる。

Super Distributed Objects Model for Ad-Hoc Coordination among Devices

Ken-ichiro Kawakami

Shigetoshi Sameshima

Katsumi Kawano

Growth of processor and networking technologies are enabling to connect real-world and computer systems through devices and much attention is being paid to the context-aware services. One of the issues to provide services using ubiquitous or pervasive computing is how to connect heterogeneous hardware device and software component to provide services. This paper proposes a new object model named Super Distributed Objects (SDO) for unified management of them, And a SDO management middleware on the ad-hoc network as an implementation.

1. はじめに

計算機プロセッサの高性能化・低コスト化により、いわゆるコンピュータ以外にも様々な機器（携帯電話、PDA、インターネット家電など）に、コンピュータに劣らない高度な計算機リソースを備えさせることが可能となった。同時に、無線 LAN、IPv6、プラグアンドプレイ技術などの新たなネットワーク技術の開発やそれらを含めたネットワーク技術全般の発展により、機器群を簡易かつアドホックに相互接続し、ユーザに対して様々なサービスを提供するための機器群による動的なネットワークを構成することが可能となった。近年、これらの技術を利用したユビキタスコンピューティングシステム[1]が注目されている。このようなネットワーク基盤技術の目指すところは、分散的に存在する機器、ソフトウェアコンポーネントおよびそれらの集合が、相互に情報共有し、必要なサービスを動的に構成することを可能とすることにある[2]。

このようなユビキタスコンピューティングを実現するために、近年 Universal Plug and Play(UPnP¹)[3]、HAVi²[4]、OSGi³[5]、ECHONET⁴[6]、Jini⁵[7]など、の相互接続技術が提案されている。

しかしこれらは、技術毎に使用できるプラットフォーム、ネットワークプロトコル、プログラム言語などが規定されており、あるいは家電や AV 機器などといった限定されたアプリケーション領域にのみ焦点を絞っており、異なる基盤技術同士での連携運用については言及していない。そのため、様々な機器およびソフトウェアコンポーネント（以下、本稿ではリソースと呼称する）を統合的に運用・管理するためには、これら既存の基盤技術を一段上位の階

¹ UPnP は the UPnP™ Implementers Corporation の登録商標。

² HAVi は GRUNDIG AG, Hitachi, Ltd., Matsushita Electric Industrial Co., Ltd., Royal Philips Electronics N.V., Sharp Corporation, Sony Corporation, Thomson multimedia S.A. and Toshiba Corporation の登録商標。

³ OSGi は the Open Services Gateway Initiative の登録商標。

⁴ ECHONET は ECHONET CONSORTIUM の登録商標。

⁵ Jini は Sun Microsystems の登録商標。

* (株)日立製作所 システム開発研究所、
神奈川県川崎市麻生区王禅寺 1099

層で抽象化してリソースの情報モデルを定義し、このモデルに基づいて相互利用可能なオブジェクトを生成することが必要となる。

これに対し我々は、多数のリソースが分散的に存在する環境において、様々な基盤技術を抽象化し相互変換してリソースの相互接続を可能とするオブジェクトモデルとして Super Distributed Objects (SDO) を提案してきた。

SDO は、なんらかの機能やその機能を利用したサービスを提供するリソースの論理表現である。SDO としてオブジェクト化されるリソースの例としては、携帯電話、PDA、インターネット家電などが挙げられ、テレビセット、冷蔵庫、電灯スイッチなど多種多様な機能を持つリソースを抽象化する。また、機器のみではなくソフトウェアコンポーネントもまた SDO としてオブジェクト化される。これにより、多様なリソースが SDO として相互に連携し、必要に応じてサービスを構成することを可能とする[8]。

一方、このような多種多様なリソースが動的に構成を変化させながら、また動的に位置や状況を変化させるユーザに対してサービスを提供するユビキタスコンピューティングシステムにおいては、リソース同士およびリソースとユーザ（すなわちユーザインターフェース）の接続は柔軟かつ簡易で動的に構築されることが望ましい。このような柔軟で動的なネットワークを簡易に構築するための技術として、アドホックネットワーク技術が注目されている。すなわち、SDO によってモデル化されたオブジェクトの管理においては、アドホックネットワーク技術の利用が有効と考えられる。

本稿では、アドホックネットワークを用いた SDO 管理ミドルウェアの機能について述べる。

以下、第 2 章では SDO の基本的なコンセプトについて述べ、第 3 章で具体的な SDO モデルの内容について述べる。第 4 章では、アドホックネットワークを利用した SDO 管理ミドルウェアについて述べる。第 5 章にて本稿をまとめる。

2. SDO モデルの概要

SDO モデルは前章で述べたように、リソース（機器またはソフトウェアコンポーネント）の論理表現であり、リソースを利用するアプリケーションからの要求に応じてリソースの情報やインターフェースを提供する。

SDO モデルは、リソースデータモデルと、このリソースデータの管理および操作のためのインターフェースからなる。SDO モデルの概要を以下に述べる。

まず、プラットフォーム非依存とするために、SDO モデルは UML⁶によって記述されている。

また、多種多様なオブジェクトを表現するために、SDO モデルは拡張可能な基本データモデル（リソースデータモデル）とインターフェースによって構成されており、これらからなる基本モデルのみを規定している。モデル化されたオブジェクトは、リソースデータモデルおよびインターフェースの任意の組み合わせによって構成され、任意の拡張を施すことができる。

以上の基本コンセプトに基づき、SDO モデルは以下のように構成されている。

- いくつかの主要データのみが属性として定義されており、それ以外の様々な属性は名前と値の組として付け加えられる。
リソースデータモデルは、オブジェクトの内容を表すデータの集合として構成される。各データは SDO のプロパティを表す属性の集合として定義される。属性は定型値あるいは名前付き値として定義される。定型値は、あらゆる実装において用いられる汎用的な属性を表現する。名前付き値は、実装（あるいはアプリケーション）依存の属性を表現する。
- 基本的なインターフェースのみを定義する。
各インターフェースは、リソースデータの管理および操作を行なうためのオペレーションの集合として定義される。

3. SDO モデル

3.1. リソースデータモデル

3.1.1. 概要

リソースデータモデルは、SDO の能力とプロパティを記述する。このリソースデータモデルは以下を構成要素として含む。

- 基本構成要素
 - SDO
 - Organizer
- Profiles.
 - Device profile
 - Service profile
 - Configuration profile
- Organization
- Status

⁶ UML は Object Management Group, Inc の登録商標。

以下に、主なリソースデータの概要を述べる。また、Figure.1 にリソースデータモデルのUML表記を示す。

3.1.2. 基本構成要素

(1) SDO

SDO はリソースをモデル化するための汎用的なプロパティを定義する。ひとつの SDO はひとつのリソースを表現する。SDO は SDO モデルのコアであり、Profile, Organization, Status 等の他のリソースデータを統合する。

(2) Organizer

Organizer は、SDO に関するオブジェクト (SDO, ユーザ, 位置情報など) を表現する基本要素である。SDO は Organizer のサブクラスとして定義され、人 (ユーザ等) や情報 (位置情報など) などの SDO 以外のオブジェクトも Organizer のサブクラスとして定義される。

3.1.3. Profiles

(1) DeviceProfile

DeviceProfile は、SDO としてオブジェクト化している機器のプロパティを定義するものであり、プロパティとして、機器タイプや製造元などの属性を持つ。

(2) ServiceProfile

ServiceProfile は、リソースによって提供されるサービスのプロパティの集合を定義するものであり、サービスの識別子やサービスの特性を表す属性を持つ。加えて、各 ServiceProfile は、その機能によってサービスを実行するために、実装に応じたインターフェースを持つ。

(3) ConfigurationProfile

SDO は設定可能なパラメータを ConfigurationProfile のプロパティとして持つ。つまり ConfigurationProfile は、例えばユーザの要求、ユーザコンテキスト、環境のコンテキストなどに応じて SDO の状態を変えるために用いられるパ

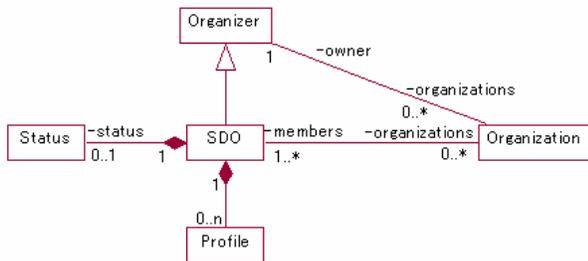


Figure. 1 UML diagram of the resource data model

ラメータの集合を格納する。

3.1.4. Organization

Organization は Organizer 間の関係を定義するものであり、ひとつの Organization は、複数の異なる SDO の間、あるいは SDO とユーザ等 SDO 以外のオブジェクトの間に定義される。Organization のプロパティは OrganizationProperty として保持される。

3.1.5. Status

Status は SDO の現在の状態を、名前と値の組からなる状態値の集合として格納する。例えば、“電源-ON/OFF”、“待機状態/活動状態”などの状態値を持つ。

3.2. インターフェース

3.2.1. 概要

SDO モデルでは、下記のインターフェースがオブジェクトを管理および操作するための基本インターフェースとして定義されている。

- SDO Interface
- Monitoring Interface
- Configuration Interface
- SDO Service
- Organization Interface

以下に、主なインターフェースの概要を述べる。また、Figure.2 にインターフェースのUML表記を示す。

3.2.2. SDO Interface

SDO Interface は、SDO モデルの持つリソースデータおよびインターフェース (Monitoring Interface, Configuration Interface, SDO Service) を管理するためのインターフェースとして定義される。

3.2.3. Configuration Interface

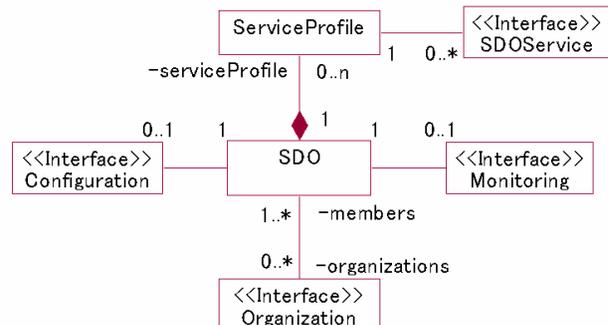


Figure. 2 UML diagram of the interfaces

Configuration Interface は、リソースデータとして定義されているデータ (DeviceProfile, ServiceProfile, ConfigurationProfile, Organization) の追加や削除を行なうオペレーションを提供する。

3.2.4. SDOService Interface

SDOService Interface は、サービス提供機能によって提供される具体的なサービスを実行するためのオペレーションを提供する。個々のサービス提供機能は ServiceProfile として表されている。このインターフェースは、サービスのタイプ毎に異なり、またサービスが同じであっても使用される相互接続技術毎に異なる。そのため、このインターフェースはプラットフォーム非依存の汎用的なモデルを持たない。

3.2.5. Monitoring Interface

Monitoring Interface は、SDO のプロパティを監視するためのインターフェースとして定義される。

3.2.6. Organization Interface

Organization Interface は Organization の属性を管理するために使用される。具体的には、Organization のオーナーやメンバーを追加/変更/削除し、あるいはオーナーやメンバー間の関係を変更するためのインターフェースを提供する。加えて、このインターフェースは OrganizationProperty の管理にも利用される。

つまり、Organization Interface は SDO および Organizer によって構成される集団の管理に利用される。

4. SDO モデルを利用したオブジェクト管理ミドルウェア

4.1 概要

前章で述べた SDO モデルに基づいて実際にリソースをオブジェクト化し、そのオブジェクト間でのインタラクションを実現するために、SDO モデルを利用したオブジェクト管理ミドルウェアを構築する。

このような管理ミドルウェアを構築するためには多様な機能の実装を考慮しなければならないが、本稿では管理ミドルウェアの基本機能として、サービスの登録および検索について述べる。

この機能は、各 SDO が提供するサービスを、ユーザ端末など他の SDO からネットワークを通じて検知できるようにし、また任意の SDO のサービス

情報を取得できるようにすることで必要なサービスを探検・発見するためのものである。

4.2 アドホックネットワークを用いた SDO 管理

4.2.1 アドホックネットワーク

アドホックネットワークは、基地局や固定局等備え付けのネットワークインフラやネットワークの集中管理機構を利用することなく、無線ネットワーク端末群によって動的に構成されたネットワークである。インフラや管理機構による制限を受ける通常の無線ネットワークに対し、アドホックネットワークでは各端末がネットワークの中継点の役割も果たすため、端末群によって自律的かつ容易に柔軟なネットワークを構成することが可能となる。

SDO では、ユーザが持ち歩く端末 (ユーザインターフェース) を始めとして、サービスを提供するリソースの構成が動的に変化することを想定しているため、リソース間のネットワークを DHCP サーバ等を用いて集中的に管理することは困難である。

そのため、備え付けられたネットワークインフラに依存せずに柔軟かつ動的になネットワークをサーバ等による集中管理不要で構成できる、アドホックネットワーク技術が SDO の管理においては適すると考えられる[9]。

そこで本稿では、アドホックネットワークを用いた SDO の管理ミドルウェアについて述べる。

4.2.2 ルーティングプロトコル

アドホックネットワークにおけるルーティングプロトコルには、プロアクティブプロトコルとリアクティブプロトコルの2種類がある。

リアクティブプロトコルは、すべてのネットワークポロジを継続的に保守しているのではなく、ルートの情報はルートが利用されたときのみ保守され、ルートが利用されなくなるときにはキャンセルされる。

プロアクティブプロトコルでは、ネットワークノード間の適切な情報を交換することでネットワークポロジは継続的に更新されており、したがって要求に応じていつでも接続先へのルートを知ることができる。

本稿で述べるミドルウェアでは、ネットワークを構成するノードが頻繁に変化する状況への親和性を考慮し、リアクティブプロトコルの1種である OLSR (Optimized Link State Routing) を用いたアドホックネットワーク[10]を利用する (詳細については 4.3.2 にて述べる)。

4.3 SDO 管理ミドルウェア

4.3.1 SDO 管理ミドルウェアの概要

SDO 管理ミドルウェアは、機器が持つ計算機リソース（ノード）上で動作し、ローカルに接続された機器およびインストールされているソフトウェアコンポーネントを SDO として管理する（Figure. 3）。

また、ネットワーク層とアプリケーション層の間に位置し、ネットワークを通してやり取りされる情報をアプリケーションに受け渡す（エラー! 参照元が見つかりません。）。これにより、SDO モデルに基づいてオブジェクト化されたネットワーク上のリソース間でのインタラクションを可能とし、異なるネットワーク基盤間での情報のやり取りを可能とする。

SDO 管理ミドルウェアは、ノードで管理する SDO のリストの生成、更新、および周囲への発信を行なうための SDORegistration と、ノードの管理する SDO の情報を取得、あるいはそれらにアクセスするための SDODiscovery の、2つのモジュールによって構成される。

SDODiscovery は、ネットワークで接続されたノード上の SDO を検索する機能を提供する。

SDODiscovery のアドホックネットワーク環境での実装を考える上での課題は、アドホックネットワークがネットワークインフラを必要としないことに起因する。つまり、固定されたネットワークや集中管理を前提としないために、ノード（およびそれが内包する SDO）がネットワークに参入したときの登録処理（以下これをレジストレーションと呼ぶ）を集中管理に頼ることが出来ない。

そこで SDORegistration は、レジストレーションをすべてのノードで分散的に行なう機能を提供する。

各ノードはネットワーク上の SDO の情報として、SDO の ID のリストをレジストリに持つ。

ここで、レジストリには以下の 2 種類がある。

- ローカルレジストリ

ノードがローカルに保持するリソース（基本的

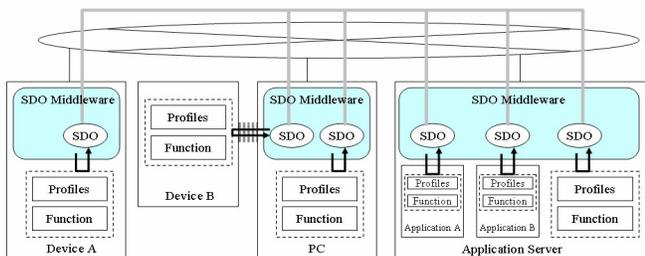


Figure. 3 Location of the SDO middleware

には、ローカルに接続された機器およびノード内に存在するソフトウェアコンポーネント)の SDO の ID リスト. このリストは SDORegistration モジュールに保持され、ネットワーク上に公開される. また、他のノードがローカルに保持するリソースを登録して、プロキシ的に利用することも可能である。

- グローバルレジストリ

ネットワーク上すべての SDO のリアルタイムな ID リスト. ただし、サービス要求に対して応答できるノードのみによってリストを生成する。

4.3.2 SDO の検索方式

ネットワークで接続されたノード上の SDO を検索する方式としては、以下の 3 つが考えられる。

- 集約管理型

サービスを提供するリソースのサービス情報を集約管理されたレジストリにすべて保持しておき、サービス要求者はそのレジストリに問い合わせる必要サービスを検索する。

- 分散・プッシュ型

各サービス提供リソースは、自身の提供可能なサービスの情報をネットワーク上のノードに周期的に発信している。

- 分散・プル型

サービス要求者が必要なときに、必要なサービスの要求をネットワーク上に発信し、該当するサービスを提供可能なノードからの応答を待

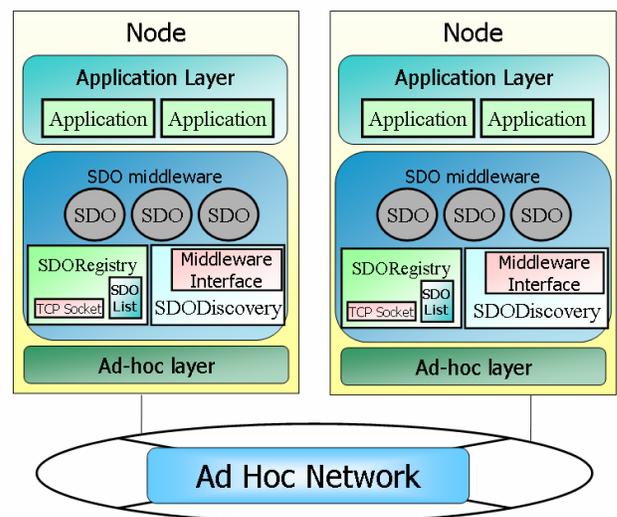


Figure. 4 Layer of the SDO middleware

つ。

頻繁にサービス要求 SDO およびサービス提供 SDO が変化するアドホックネットワーク環境においては、サービス要求 SDO が要求を出した時点で応答可能なサービス提供 SDO が適宜応答する分散・プル型方式が適すると思われる[11]。そこで SDODiscovery では、SDO 検索の方式としてこの分散・プル型を用いる。

ここで、リアクティブ・アドホックルーティングプロトコルである OLSR を利用し、ルーティングテーブルから適宜アクセス可能なノードの IP を取得できる仕組みを用いることで、サービス提供 SDO が能動的に自身のサービスを周囲に知らせることはなく、サービス要求 SDO が必要なときにサービス要求をアドホックネットワーク上に、その時点で存在するすべてのノードにユニキャストすることが可能となる。

4.3.3 SDORegistration

各ノードは SDORegistration により、ノードにローカルに存在する SDO (ノードにローカルに接続されている機器、あるいはノード上で動作するソフトウェアコンポーネント) をローカルレジストリに登録している。また、それらの SDO を他のノードのグローバルレジストリに登録する。

ここで、SDO を ID とポートのみによって管理すれば、SDO がローカルに存在するか他のノードに存在するかに依らず、ID とポートを登録することによって SDO の登録が可能となる。

これにより、ノードの計算機リソースが不十分であり、単独で SDO を管理できない場合でも、他のノードをプロキシのように利用することで対応可能となる。

また、ヘテロなネットワークが混在する環境に対しても同様に、各ネットワークとの接点となっているノードをプロキシ的に利用することによって対応可能となる。

この方式は、突き詰めればネットワーク上の SDO の登録を集約的に管理する方式となるので、アドホックネットワークに必ずしも適さない。

しかし、小型家電製品など十分な計算機リソースを持たない機器の混在、また USB やシリアル接続などによるノードへのローカル接続や電灯線ネットワーク等多様なネットワークの混在を許容するために有効である。

4.3.4 SDODiscovery

SDODiscovery は、サービス要求 SDO からの要求によって実行される SDO ミドルウェアの機能であ

り、指定された検索キーに合致するサービス提供 SDO を検索する機能を提供する。この検索キーは String で表現され、SDO のリソースデータの記述に対応する。

また、アドホックネットワーク上でのホップ数によって検索範囲を限定することもできる。これは、サービス要求 SDO からサービス提供 SDO までの物理的な距離がサービス検索の上で重要である場合、あるいはネットワーク負荷への影響を制限する場合に有効である。

5. おわりに

本稿ではまず、機器やソフトウェアコンポーネントを記述するためのオブジェクトモデルである SDO について述べた。SDO モデルは、オブジェクトのプロパティを記述するためのリソースデータモデルと、オブジェクトおよびオブジェクトが提供するサービスを管理するためのインターフェースから成る。なお、SDO モデルはオブジェクト指向技術の国際標準化団体 Object management Group において標準仕様として提案中のものである。

また、SDO モデルを利用した機器管理ミドルウェアについて、アドホックネットワーク技術の利用を前提とした場合の設計指針について述べた。SDO モデルの実装を想定する環境では、ユーザが持ち歩く端末など環境中の機器の変化が激しいと考えられるため、アドホックネットワーク技術による動的ネットワーク構成を併用することがシステム構築のためのひとつの方向性として有効と考えられる。

謝辞

本研究は、文部科学省の科学技術振興調整費「先導的研究等の推進」プログラムにより実施している「横断的科学的ユビキタス情報社会の研究」の成果の一部である。

また、本研究にご助力いただいた Carlo Cubeddu 氏に感謝いたします。

参考文献

- [1] M. Weiser, "The computer for the 21st century", Scientific America, 256(3), pp.94-104, 1991.
- [2] S. Sameshima, K. Kawano, M. Funabashi, "A trend to Super-distributed System and standardization activity in OMG", 2001 Symposium on Applications and the Internet Workshop, pp.5-8, 2001.
- [3] UPnP (Universal Plug and Play), <http://www.upnp.com>

- [4] HAVi, <http://www.havi.org/>
- [5] OSGi, <http://www.osgi.org>
- [6] ECHONET (Energy Conservation and Homecare Network), <http://www.echonet.gr.jp>
- [7] Jini, <http://www.jini.org>
- [8] 鮫嶋, 河野, 新, “環境適応サービスを狙いとした超分散オブジェクトモデルと自律プラグアンドプレイ方式”, 日本電気学会論文誌 C 編, Vol. 124, Num.1, 2004. (掲載予定)
- [9] 川崎, 松井, “アドホックネットワークを用いたユビキタスサービスの提案とその実現に向けた検討”, 情報学ワークショップ 2003, 2003.
- [10] 長船, P. Zhao, 塚田, 松井, “アドホックネットワークプロトコルの評価”, 電気情報通信学会 第 1 回 アクティブネットワーク研究会 (IEICE AN2002), Vol.1, No.1, pp.19-25, 2002.
- [11] L. Cheng, “Service advertisement and discovery in mobile ad hoc networks”, Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, in conjunction with the ACM 2002 Conference on Computer Supported Cooperative Work, 2002.