

分散環境におけるデバイス検索とその機能の連携方式

髭白 茂男[†] 北島 聡史[†] 金子 祐真[†] 小泉 寿男[†]

近年、情報家電やネットワーク家電のようなネットワーク機能を持ったデバイスの登場が見られる。これらのデバイスが連携することによって1つの機能を提供することは、利用者の利便性を高めることにつながる。本稿は、分散環境下に存在するデバイスの機能を検索し、それらを複合化し利用者に提供するエージェント方式を提案する。本方式は、モバイルエージェントを用いることでデバイスの環境に依存しない検索処理を実現する。また、本方式では、機能提供中のデバイスが機能提供不可になった場合、モバイルエージェントを用いることで、他のデバイスに移動し機能提供を継続することにより、耐故障を可能とする。提案方式のプロトタイプを構築し、本方式の評価、実現可能性を確認した。

A Method of Searching Functions of devices and connecting their functions in distributed environment and its evaluation

Shigeo Higeshiro[†] Satoshi Kitajima[†] Yushin Kaneko[†] Hisao Koizumi[†]

In recent years, consumer devices with networking capability, such as information appliances and networked electric appliances, are emerging. Offering a function by connecting these devices is expected to increase users' benefits. This paper proposes an agent system that searches the functions of the devices in distributed environment and offers integrated capability for users by connecting searched functions. By using mobile agents, this method realizes searching system that does not depend on the environment of devices. This method enables Fault Tolerant system by using a mobile agent. If it becomes impossible to provide functions with the current devices, it moves to other devices and continues offering functions by using a mobile agent. The proposed system was prototyped, and the feasibility of the system was confirmed through evaluations.

1. はじめに

分散コンピューティング上でデバイス同士が連携し1つの機能を提供することは、利用者の利便性を高めることにつながる^{1)~4)}。

デバイス管理においては、利用したいデバイス情報を得た後、デバイス間の連携を行う。デバイス間連携は、気温などのセンサー情報を取得することや、音を鳴らすことやテレビをつけるといった各デバイスが持つ機能呼び出すことである。デバイス間連携を実現するために、デバイス間連携時に必要なソフトウェアを取得し、デバイス間連携を行うモデルを考える。この際、モバイルエージェントを利用しデバイス間連携に必要なソフトウェアを組み込む。

本稿では、分散環境上におけるデバイス間連携を目的としたエージェント方式の提案を行う。本方式は、集中管理するサーバを必要としない方式をベースとしており、利用者が希望する機能を持つデバイスを検索する方式とそれらのデバイスを連携する方式から成る。また、デバイス間連携をモバイルエージェントとして実現することで、機能提供を行うデバイスに必要なソフトウェアを組み込むことができる。

2. デバイス間コミュニケーション方式と課題

サーバ管理方式の場合、サーバコンピュータの停止はサービスの停止を意味し、高信頼性が求められる。また、分散環境でのサーバ動作はオーバーヘッドを増加させる可能性がある。一方、非集中システムと言う考え方が注目されている。非集中システムは生物界に存在する管理システムを参考にした協調・連携の考え方であり、各要素間の局所的な相互作用によって様式が決定される。また、サーバ方式のような管理機能を必要としない特徴がある⁸⁾。

機能連携を目的としたミドルウェアには、J2ME、Jini、HAVi、がある^{5)~7)}。しかし、これらの方式は、デバイスを管理するサーバコンピュータを必要とし、アドホックネットワークのようなデバイスの参加、脱退が多いネットワーク環境下において、デバイス連携を実現するのは難しい。また、サーバコンピュータを発見する lookup メッセージが増えてしまうなどの課題がある。また、機能連携中のデバイスが機能提供不可能になった場合、他の同機能を持つデバイスに機能提供を継続させるなどの耐故障を実現していないため利用者から見るとサービスが中断してしまうことがある。

東京電機大学大学院 理工学研究科
情報システム工学専攻
Graduate School of Tokyo Denki University
System Engineering

3. デバイス間連携エージェント方式

3.1 機能の検索と連携の概要

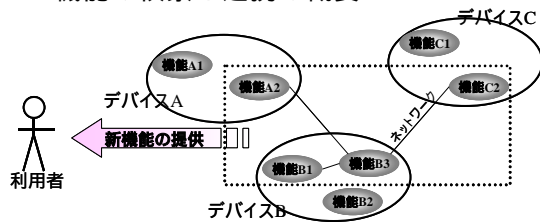


図1 機能の発見と連携

本方式は、複数のデバイスが持つ機能を複合化させ利用者に提供するものである。図1に機能連携のイメージを示す。図1では、デバイスAのA2という機能、デバイスBのB1、B3という機能、デバイスCのC3という機能を複合化させ利用者に提供している。このような機能の連携をするためには、利用者が要求する機能を持つデバイスの検索処理と検索より得られたデバイスの機能を呼び出すことが必要となる。

本方式での検索処理の流れを図2に示す。

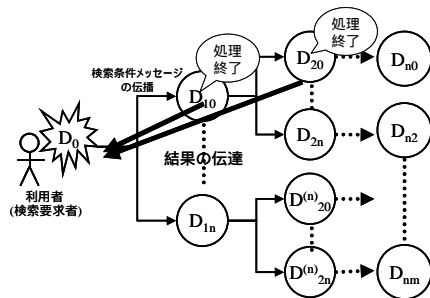


図2 機能発見の全体的な流れ

図において D_0 が検索要求発生地点である時、利用者が D_0 に検索処理を命令すると、 D_0 はまず自身の機能を検索し、該当する機能がない場合、 D_{10} から D_{1n} へ検索処理を伝播する。同様に D_{10} は D_{20} から D_{2n} に検索処理を伝播する。各デバイスで検索した結果は直接検索要求者に伝達する。

3.2 エージェントの構成と機能

本方式は、3つのエージェントから成る。

- (1) 隣接デバイス情報を提供し、検索メッセージの伝播、検索結果の転送を行うエージェント
- (2) デバイス上で検索を行うエージェント
- (3) 検索結果のデバイス群から1つを決定し、連携動作を行うエージェント

3.2.1 情報管理エージェント (IM-ag)

IM-ag は、近接デバイス情報提供機能を持っている固定型エージェントである。利用者が IM-ag に検索を命令すると検索処理の重複を避けるための検索処理識別コード、検索条件に必要な複数のキーワード、伝播デバイスの履歴を格納した検索条件メッセージを作成する。

IM-ag は、検索結果転送時、検索条件、どのデバイスから転送されたか、転送開始時刻を知識として格納する。この知識により、直接通信可能なすべてのデバイスに検索条件メッセージを送信するのではなく、検索条件を満たすデバイスがあ

ると予測されるデバイスに検索条件メッセージを送信し、検索メッセージ数を削減する。IM-ag の知識は式1で表す組の集合として表現できる。

$$\text{knowledge}_{\text{ima}} = \{\text{RcPd}, t\} \quad (1)$$

(Rc:検索条件、Pd:転送元デバイス、t:転送開始時間)

情報管理エージェントは、この知識を利用して、近接デバイス群の中より検索条件メッセージを転送するデバイスを選択する。情報管理エージェントは、検索条件メッセージが他のデバイスから転送されたとき、転送されてきた検索条件メッセージに格納されている検索条件と自己が持つ知識の R_c が一致する P_d の集合 $D_{\text{knowledge}}$ を作成する。近接デバイス群の集合を D_{closed} としたときの検索条件メッセージの転送先デバイスの集合 D_{next} は式2で与えられる。

$$\{D_{\text{next}}\} = \{D_{\text{knowledge}}\} \text{and} \{D_{\text{closed}}\} \quad (2)$$

$\{D_{\text{next}}\}$ が2つ以上の要素から成るとき、IM-ag は $\{D_{\text{next}}\}$ から、 $\text{knowledge}_{\text{ima}}$ の t が現時刻に近い Device を転送先デバイスとして選択する。また、 $\{D_{\text{next}}\}$ の要素数が0の時は、 $\{D_{\text{closed}}\}$ からランダムに選択する。

IM-ag は検索依頼者が発した検索条件メッセージの伝播と検索エージェントによる検索結果を依頼者に送信する。IM-ag は検索条件メッセージ受信時に検索条件メッセージに格納されている伝播上限回数値を1つ減らし、他のデバイスに伝播する。伝播上限回数値が0である、または伝播時間の上限値を超えている場合は、他のデバイスへの伝播は行わない。

3.2.2 検索エージェント (S-ag)

検索エージェントは検索機能を持ったモバイルエージェントである。検索に使う情報は検索メッセージにより提供される。検索エージェントは知識としてユーザ検索の履歴、シソーラス(類語)関連があるキーワードを持つ。検索エージェントのシソーラスと関連キーワードの情報は、W3Cによって開発された RDF(Resource Description Framework)形式により格納する。検索エージェントは、検索依頼先デバイスに検索に必要な背景知識をもつ検索エージェントが存在しない時、自己を複製し検索依頼先デバイスへ移動し、検索を行う。

3.2.3 デバイス間連携エージェント (CF-ag)

デバイス間連携エージェントは、デバイスが持つ機能を利用するモバイルエージェントであり、他のデバイスに存在するデバイス間連携エージェントと協調動作し、複数のデバイスの機能を用いた複合機能を利用者に提供する。デバイス連携エージェントには、機能提供機器の選択、バックアップ機能がある。

(1) 機能提供機器の選択

機器の選択動作について図3を用いて説明する。

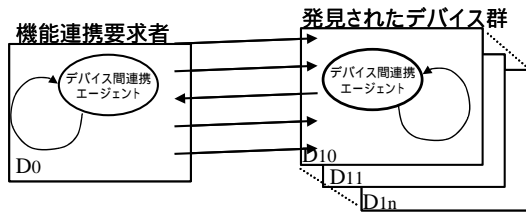


図 3 デバイス間連携エージェント方式

機能連携要求者である D_0 は、検索結果より $D_{10} \sim D_{1n}$ を取得する。本方式では、条件を満たすデバイスが複数個発見された時、契約ネットプロトコルを用いて 1 つを選択する。 D_0 の CF-ag は $D_{10} \sim D_{1n}$ に CF-ag を送信する (図中)。 D_0 の CF-ag は、 $D_{10} \sim D_{1n}$ のエージェントに対して機能連携要求メッセージを送信する (図中)。この要求を受けた CF-ag は自デバイスの状況を調査し (図中) CPU 使用量、メモリ空き容量、 D_0 との通信コストなどを機能連携要求者に返信する (図中)。機能連携要求者はあらかじめ定められたメモリ空き容量、バッテリー残量などの評価基準に基づき適切なデバイス (D_{1x} とする) を選択し (図中)、 $D_{10} \sim D_{1n}$ 全てのデバイスに報告する (図中)。機能連携要求者は、 D_{1x} に機能連携に必要なプログラムを含んだ CF-ag を送信する (図中)。

(2) デバイス連携

CF-ag は、機能提供機器を 1 つに選択した後、デバイス連携処理を行う。CF-ag は、機能連携要求者で動作するエージェントと、機能提供側で動作するエージェントから成る。利用者側で機器選択を行った後、利用者側の CF-ag は、自己を複製し、機能提供側に送信する。

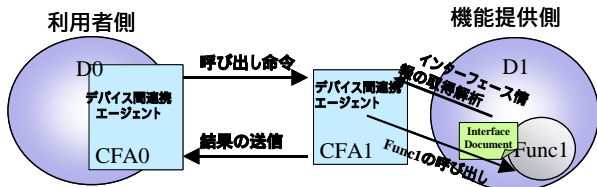


図 4 デバイス間連携エージェントによる機能呼出

利用者側のデバイス間連携エージェント (CFA0) は機能提供側のデバイス間連携エージェント (CFA1) に呼び出し命令を送信する (図中)。CFA1 は、 D_1 の機能呼び出すためのインタフェース情報を取得し (図中) 機能の呼出を行う (図中)。機能の呼出しによりえた結果を CFA0 に送信し、機能提供が終了する (図中)。以上 ~ の処理を繰り返し行い、連携動作を行う。

(3) バックアップ機能

機能提供中のデバイスが、バッテリーの消耗やプライオリティの高いタスク発生により、機能提供が不可能になった場合、デバイス間連携エージェントが同機能を持つ他のデバイスに移動することで、機能提供を切り替える。

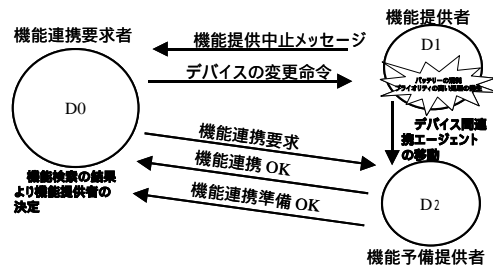


図 5 機能連携エージェントのバックアップ機能

機能提供者 D_1 において、機能提供が不可能になった場合、 D_1 のデバイス連携エージェントは機能連携要求者である D_0 に機能提供中止メッセージを送信する (図中)。 D_0 のデバイス間連携エージェントは D_1 の代わりとなる機能をもつデバイスを検索する (図中)。 D_0 のデバイス間連携エージェントは検索結果として得られたデバイスに機能連携要求 (図中) を送信し、機能連携準備 OK メッセージが得られた場合 (図中)、 D_0 から D_1 にデバイス変更命令が送信され (図中)、 D_1 のデバイス間連携エージェントは D_2 に移動し (図中) D_0 との連携作業を引き続き行う。

機能連携要求を送信し、機能連携準備 OK メッセージが得られなかった場合、機能連携は中止され、機能連携エージェントは消滅する。

3.3 エージェントによる検索処理

エージェントを用いた検索方式として、検索要求をメッセージで伝播していき検索処理を行う方式と、エージェントが自ら検索対象に移動し、検索を行う方式がある。

(1) 検索条件メッセージによる検索

検索条件メッセージを受け取ったデバイスでは、IM-ag と S-ag が連携して検索処理を行う。図 6 にエージェントの連携動作を示す。図における D_1 が検索依頼デバイスであり、 D_{21} が検索先デバイスである。

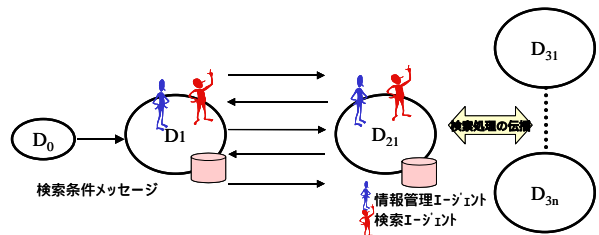


図 6 エージェントによる検索処理

D_1 の IM-ag は直接通信可能なデバイス群から検索条件メッセージの伝播先を決定する。 D_1 の IM-ag は、 D_{21} の IM-ag へ認証命令を送信し (図中) 許可された時 (図中) 検索条件メッセージの送信を行う (図中)。許可されなかった時は、 D_{21} への検索条件メッセージの送信は行わず、IM-ag が持つデバイス群の中から D_{21} 以外のデバイスを対象とする。

D_{21} が検索処理機能を実装していなかった時、 D_1 へ S-ag の移動受付メッセージが送信 (図中)

され、 D_1 においてS-agが複製され D_{21} へ移動する(図中)。S-agは D_{21} において検索処理を行う(図中)。検索条件を満たす場合、IM-agは検索依頼者の元へ検索結果と伝播デバイスの履歴を送信し、S-agは消滅する。検索条件を満たさない、かつ、伝播上限回数が正の時、 D_{21} のIM-agがもつデバイス群に検索条件メッセージを送信する。検索条件を満たさない、かつ、伝播条件回数が0の時、IM-agは検索依頼者の元へ検索失敗メッセージを送信し、S-agは消滅する。

(2) エージェントの移動による検索

検索エージェントの移動による検索は、検索対象の検索機能の有無にかかわらず、検索エージェントが移動を行うものである。

検索要求を受けたIM-agは検索条件メッセージによる検索方式の時と同様に、直接通信可能なデバイス群から検索対象を決定し、S-agに伝達する。S-agは検索対象デバイスに移動を行い、検索の許可を求める。許可されなかった場合、そのデバイスを検索対象から除外する。許可された場合、移動先のデバイス上で検索を行う。検索条件を満たす場合は、検索対象デバイス上にあるIM-agによって検索結果を送信し、S-agは消滅する。また、検索条件を満たさない、かつ、伝播条件回数が0の時は、検索失敗となりS-agは消滅する。

4. プロトタイプの構築

4.1 プロトタイプの構築環境

表 1 プロトタイプ構築ツール

OS	Windows XP
モバイルエージェントフレームワーク	Agent Space
XML Parser	JDOM
サーブレットエンジン	TOMCAT4.05
メッセージング	Apache AXIS 1.0
機能記述・格納・検索機能	SOAP UDDI
機能の呼び出し	IBM WSIF

プロトタイプの構築基盤・構築ツールを表1に示す。プロトタイプでは、検索方式を検索条件メッセージの伝播による方式のみとし、構築を行った。本研究ではJavaを用いたシステムを構築し、モバイルエージェントフレームワークとしてはAgentSpaceを用いた。エージェント間の通信はSOAPを利用し、SOAPの実装としてApache-AXISを利用した。機能のインタフェース記述にはWSDLドキュメントの記述方式を用いた。WSDLとSOAPを用いることで、機能呼び出しにおいて緩やかな結合や遅延バインディングを可能にしている。また、ディレクトリ構造を持つUDDIドキュメントの記述方式を用いることでデバイスの機能の記述を行った。UDDIの実現にはオープンソースであるSOAP-UDDIを用いた。

4.2 エージェントの構築

エージェントはAgentSpaceで提供されるAgentクラスを継承しており、同じコンピュータ上に存在する時はAgentSpaceの通信機能を用いてエージェント間通信を行っている。しかし、AgentSpaceに依存した通信機能になるのを避けるため、通信内容はXMLを用いる方式とした。XMLドキュメントへのアクセスはJavaと親和性の強いJDOM APIを用いた。以下、クラスという言葉はJavaのクラスファイルを示すものとする。

4.2.1 情報管理エージェント

(1) 検索条件XMLの作成

利用者がIM-agに検索を命令した時に作成する検索条件メッセージはXMLで表現する。

(2) 検索条件の伝播

検索条件の伝播は、伝播先のデバイスのroutingMessageというメソッドを呼び出すことで実現している。routingMessageメソッドは、引数として与えられた検索条件XMLのperformativeによって処理分岐を行う。PerformativeがSearchのとき、検索条件XMLに格納されているuniquecodeを取得し、自身の履歴変数に格納されていないかを比較する。自身の検索履歴にある場合、検索処理を中止し、nullを返す。検索履歴にない場合、検索処理を行い、uniquecodeを検索処理識別のために保存する。検索処理、検索条件キーワードを条件とし、UDDIレジストリの検索機能を用いて行う。検索の結果、条件に合うものが見つからない場合、検索条件XMLの伝播上限回数を1減らし、伝播先デバイスのroutingMessageメソッドの引数に検索条件XMLの伝播を行う。

検索の結果、条件に合う場合、検索条件XMLのパフォーマティブをanswerとし、検索依頼者の元へ検索条件XMLを送信する。

4.2.2 検索エージェント

プロトタイプで構築したS-agの知識の一部を図7に示す。図において、thesaurus属性はシソーラス情報を示し、“サウンド”と“sound”などの表記のゆれを格納しておく。relation属性は関連語句情報を示し、soundとstereoなど関連がある言葉を格納し検索時に使用する

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:NS0="http://www.itlab.k.dendai.ac.jp/" >
```

```
<rdf:Description rdf:about="sound">
  <NS0:relation rdf:resource="stereo"/>
  <NS0:relation rdf:resource="music"/>
</rdf:Description>
```

関連語句
情報

```
<rdf:Description rdf:about="サウンド">
  <NS0:thesaurus rdf:resource="sound"/>
</rdf:Description>
</rdf:RDF>
```

シソーラス
情報

```
<rules>
  <performative action="move"><to>before</to></performative>
</rules>
```

アクション
ルール

図 7 検索エージェントの知識(一部)

4.2.3 デバイス間連携エージェント

プロトタイプでは、機能連携依頼者側のデバイス間連携エージェントを CFAR クラスで実現し、CFA クラスにより制御している。

(1) デバイス間連携エージェントの動作フロー CF-ag の連携動作時のフローを図 8 に示す。

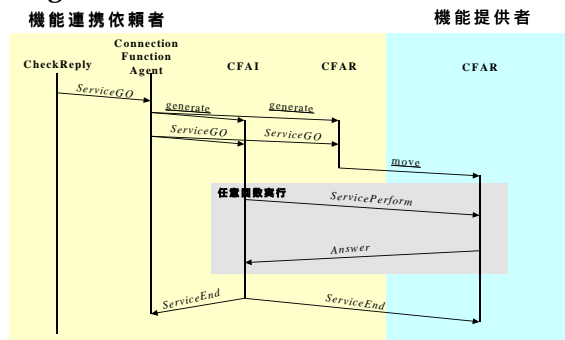


図 8 デバイス間連携エージェントの動作フロー

IM-ag は、CFA に ServiceGo メッセージを送信する。CFA は、CFAI と CFAR を生成し、それぞれに、ServiceGo メッセージを送信する。CFAI は ServiceGo メッセージを受信するとユーザにインタフェース画面を提供する。CFAR は、ServiceGo メッセージを受信すると機能提供デバイスに移動し、機能提供準備を行う。機能の実行は CFAI が CFAR に ServicePerform メッセージを送信することで行う。また、CFAR は結果を Answer メッセージとして CFAI に送信する。機能の実行は任意回数行う。機能連携を中止するとき、CFAI は ServiceEnd メッセージを CFA と CFAR に送信する。

(2) デバイス間連携エージェント (CFA)

CFA は、AgentSpace 上で動作するクラスであり、動作を開始するとポート番号 19765 でメッセージを受付ける。IM-ag から performative が ServiceGO である AgentSpaceXML を受信すると CFAI と CFAR を AgentSpace 上に生成する。生成後、CFAI と CFAR に対して performative が ServiceGO である AgentSpaceXML を送信する。

(3) 機能連携依頼者側 CFA (CFAI)

CFAI は、CFA によって生成され、AgentSpace 上で動作する。動作を開始すると、ポート番号 29765 でメッセージを受け付け、ユーザインタフェースを表示する。CFA から、performative が ServiceGO である AgentSpaceXML を受信すると CFAR が動作しているホスト名と機能連携に必要なインタフェースドキュメント (WSDL ドキュメントの URL) を自身のメモリ内に格納する。CFAR から performative が Answer である AgentSpaceXML を受信すると、AgentSpaceXML の Data 要素に含まれている情報を画面に表示する。

利用者が CFAI のユーザインタフェース画面でボタンを押すなどのアクションをすると、CFAI は CFAR に対し、performative を

ServicePerform にし、命令情報を格納した AgentSpaceXML を送信する。

(4) 機能提供側 CFA (CFAR)

CFAR は、によって生成され、AgentSpace 上で動作する。動作を開始すると、ポート番号 39765 でメッセージを受け付ける。

CFA から、performative が ServiceGO である AgentSpaceXML を受信すると AgentSpaceXML の host 情報を取得し、機能提供側デバイスに移動する。移動後、ポート番号 39765 でメッセージを受け付ける。

CFAR は、機能連携依頼者に存在する CFAI から performative が ServicePerform である AgentSpaceXML を受信すると、機能の呼び出しを行う。機能呼び出しの結果は、performative を Answer とした AgentSpaceXML を作成し、CFAI に送信する。CFAR の機能呼び出しには、WSIF (Web Services Invocation Framework) を利用し、スタブやプロキシを必要としない動的な呼び出しを実現している。

4.2.4 機能の記述

デバイスが持つ機能は UDDI ドキュメントの記述方式を用いて、記述した。UDDI は、ビジネス情報、サービス情報、バインド情報、サービス型からなるディレクトリ構造をもっている。サウンド機能を表す UDDI ドキュメントを図 9 に示す。

```
<bindingTemplate
bindingKey="6dec810b-f100-0000-0080-daf7e0e414aa"
serviceKey="0a5b6d0b-f100-0000-0080-daf7e0e414aa">
<description xml:lang="en">sound out with STEREO</description>...
<accessPoint URLType="http">http://127.0.0.1:9080/axis/SoundPlayer.jws</accessPoint>
<tModelInstanceDetails>
<ModelInstanceInfo tModelKey="uuid:30ec810b-f100-0000-0080-daf7e0e414aa">
<instanceDetails>
<instanceParms>itlab</instanceParms>
<overviewDoc>
<overviewURL>http://127.0.0.1:9080/axis/SoundPlayer.jws?wsdl...
</overviewURL>
</overviewDoc>
</instanceDetails>
</tModelInstanceInfo>
</tModelInstanceDetails>
</bindingTemplate>
```

図 9 サウンド機能の記述

機能についての記述は、bindingTemplate 属性の description 要素に行った。また、サウンドサービスの利用方法を記述した WSDL は、Tmodel 属性に記述した。

5. 評価

機能連携エージェントの評価

(1) 評価環境

評価環境を図 10 に示す。

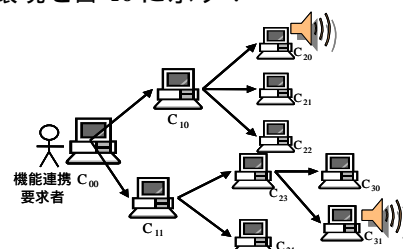


図 10 評価環境コンピュータの配置

100BASEのLANに接続された10台のPCに本システムをインストールした。各PCにはUDDIドキュメントを保存し、図中C₂₀とC₃₁だけは「SOUND出力」機能を用意し、SOUND OUT機能を記述した。C₀₀を機能連携要求者とし、サウンドソースを保存した。C₂₀またはC₃₁の発見と機能連携を目指す。情報管理の設定情報を表2に示す。

表2 情報管理エージェントと検索エージェントの設定値

コンピュータ名	情報管理エージェントの設定値	検索機能の有無
C ₀₀	C ₁₀ C ₁₁	あり
C ₁₀	C ₂₀ C ₂₁ C ₂₂	あり
C ₁₁	C ₂₃ C ₂₄	あり
C ₂₀	設定なし	なし
C ₂₁	設定なし	あり
C ₂₂	設定なし	なし
C ₂₃	C ₃₀ C ₃₁	あり
C ₂₄	設定なし	なし
C ₃₀	設定なし	なし
C ₃₁	設定なし	なし

(2) 評価結果

C₀₀において、“サウンド”をキーワードとして検索を行った。検索過程を示す。検索条件を受け取ったIM-agは、S-agに検索条件を伝え、各PCで検索処理が見られた。検索機能を持たないデバイスへはS-agの移動が行われた。S-agは、C₁₀からC₂₀とC₂₁への移動、C₁₁からC₂₄への移動、C₂₃からC₃₀とC₃₁への移動が行われた。このときS-agの移動処理には10回平均で1.7[s]がかかった。検索の結果、検索条件を満たすC₂₀とC₃₁が発見され、検索依頼者の元へ通知が行われた。C₀₀からC₂₀、C₃₁へCF-agが送信され、契約ネットプロトコルよりC₂₀が候補となった。C₀₀から、サウンドの再生命令を送信することで、C₂₀からサウンドが再生されることを確認した。また、サウンド出力を行っているC₂₀のバッテリーの減少時、CF-agのバックアップ機能によりCF-agがC₂₀からC₃₁へ移動し、サウンド機能が引き続き行われることを確認した。この間のサウンド機能の停止時間は10回平均で2.4[s]であった。停止時間のうち、デバイス間連携エージェントのシリアルライズ、デシリアルライズといった処理が1.9[s]であり、停止時間の79.2[%]を占めている。

6. まとめ

本稿では、モバイルエージェントを用いたデバイス間連携エージェント方式の提案を行った。エージェント機能を情報管理・メッセージ伝播機能、検索機能に分割することで、様々な環境に適應できる検索システムを提案した。デバイス間連携エージェントをモバイルエージェントとして設計することで、デバイスの連携に必要なソフトウェアをデバイス連携時に組み込むことを可能とした。エージェントの自律性により機能提供が

不可能になった場合には、他のデバイスに移動し機能提供を続けるバックアップ機能を提案し、耐故障を実現した。提案方式のプロトタイプをLAN上のPC環境に構築し、その動作を確認した。プロトタイプシステムにより提案方式の実現可能性を見出した。

プロトタイプによる評価から、検索エージェントの移動には平均1.7[s]の時間が生じる。検索処理効率化を図るため、検索エージェントの機能を分割し、検索に必要な最小限の機能だけをモバイルエージェントとして送信するなどといった改良が必要になると考える。また、デバイス間連携エージェントのバックアップ機能においては、平均2.4[s]の機能提供停止時間が生じた。デバイス間連携エージェントのサイズは、連携処理を複雑化するほど大きくなると考えられるため、事前に機能提供不可能になることを予測して、他のデバイスにデバイス間連携エージェントを送信しておき、機能提供不可能の決定時には、メッセージを送信するだけでバックアップ機能を実現するといった改良が必要であると考えられる。

今後はPDAなどの携帯デバイスを含めた環境での実装を行い、方式の検証を行う。また、検索により得られたデバイス群からデバイスを選択する処理への知識の適用、デバイス間連携を記述するフロー言語の検討を行う。

<参考文献>

- 1) Mark Weiser, "Hot Topics: Ubiquitous Computing" IEEE Computer, October 1993
- 2) 坂村健, "ユビキタス・コンピュータ革命", (株)角川書店, 東京, 2002
- 3) 青山友紀, "ユビキタスコンピューティング世界を実現する革新的ネットワーク技術", 情報処理学会誌 vol43, no6, pp. 611-652, June, 2002
- 4) 美濃導彦, 中村素典, "情報家電の考え方", 情報処理学会誌 vol42, no11, pp. 1049-1054, Nov, 2001
- 5) NTTデータJava研究会, "Javaがもたらす近未来のネットワーク技術Jiniって何だ?", (株)カットシステム, 東京, 1999
- 6) Jini, <http://www.jini.org/>
- 7) HAVi, <http://www.havi.org/>
- 8) 長橋和哉, 髭白茂男, 北島聡史, 金子祐真, 塩澤秀和, 小泉寿男, "分散環境におけるサーバレスコミュニケーション方式と検索システム方式の提案", Dicom2002 シンポジウム論文集, pp 33-36, July, 2002