

ユビキタスネットワーク環境の実現に向けた モバイルミドルウェアの開発

本永 公章 本城 啓史 箱守 聡 松本 隆明
株式会社 NTT データ 技術開発本部

ユビキタスネットワーク環境においては、様々なサービスがネットワーク上に遍在し、いつでもどこでも、ユーザはそれらのサービスを利用できるようになるであろう。既にブロードバンドや第三代携帯電話の普及により、いつでもどこでもサービスに接続できる環境が実現されつつあるが、さらにはいつでもどこでも、ユーザの好みやその時々状況に応じて適切なサービスを提供する仕組みを実現することが求められる。そこで本論文では、ユーザの嗜好や環境に応じて適切なサービスを提供できることを目標とした次世代モバイルミドルウェアの開発に対する取り組みを報告する。

Mobile Middleware Towards Ubiquitous Computing Environment

Kiminori Motonaga Hiroshi Honjo Satoshi Hakomori Takaaki Matsumoto
Research and Development Headquarters, NTT DATA CORPORATION

It is getting possible to utilize various services any time, anywhere due to popularization of broad band networks or third generation mobile services. To proceed the next step, towards ubiquitous computing environment, it will be also necessary to provide adaptive services in consideration of users' preferences and contexts. In this paper, we report our activity to develop the mobile middleware for realizing such next generation services.

1. はじめに

ブロードバンドや第三代携帯電話の爆発的な普及に伴い、いつでもどこでもネットワークに接続できる「ユビキタス」なネットワーク環境が現実的なものとして、ますます認識されるようになってきている。このユビキタスネットワークにおいては PC はもとより、PDA や携帯電話などのモバイルデバイス、さらには各種センサも接続されるようになる。それらセンサがコンテキスト [1] と呼ばれる、人や物の状態およびそれらを取り囲む周囲の状況を感じ、センサ情報の購読者、例えばサービスプロバイダへとセンサ情報を送ることにより、ユーザに対していつでもどこでも、ユーザの好みやその時々状況に応じた適切なサービスを提供することが可能となる。

この来るべきユビキタスネットワーク時代を見据え、NTT データ、Fraunhofer Institute FOKUS および DoCoMo Communications Laboratories Europe GmbH の 3 社で共同研究プロジェクト "Mercury" を実施し、ユビキタスネットワーク環境下に必要となるべきミドルウェアの研究開発を行った。このミドルウェアは、ネットワーク上に分散する様々なサービスを動的に見つけ出し、ユーザの嗜好や環境に応じた適切なサービスを提供すると共に、ユーザにとって不必要な操作を軽減するサービスプラットフォーム機能を提供するものである。

本論文では、第 2 章でミドルウェアの設計を行うための要求分析について述べ、第 3 章にて Mercury ミドルウェアの論理アーキテクチャと機能を解説する。第 4 章ではミドルウェアの応用シナリオと、その実装例として我々の作ったプロ

トタイプシステムについて紹介する。そして第5章ではプロトタイプの評価を行い、最後に結論と今後の方向性を述べる。

2. ミドルウェア設計のための要求分析

本プロジェクトにおいて目指すミドルウェアは、ユーザの嗜好や行動、ユーザを取り巻く環境を考慮して、ユーザにとって適切なサービスを提供するという、ユーザ中心のサービス提供を実現できるものである。

そこで、ユーザ中心のサービスとは何かを具体化するために、5つのサービスシナリオを作成し、UML[2]のユースケースを用いて分析を行った。これにより、ユーザ中心のサービスの実現に必要な、ミドルウェアが提供すべき機能群を特定した。以下はその主要なものである。

- ユーザの嗜好、機器の特性、サービスパラメータをプロファイル化し、プロファイルのデータに基づいてサービスを変更
- ユーザのコンテキストに応じて、提供するサービスを変更
- サービス毎にコミュニティを生成し、コミュニティ内のメンバ(ユーザ、サービスプロバイダ、機器等)をコーディネート
- 信頼性の低い断続的な通信環境下でも、継続的なサービスを実行
- ユーザの代わりに、エージェントが自律的に煩雑な処理を実行
- サービスプロバイダが自分のサービス広告をネットワーク上に発行し、ユーザが発行されたサービス広告を検索
- ユーザの目的に応じて、複数のサービスを組み合わせて1つのサービスを動的に構成

さらに、ドメインモデルの作成により、上記で特定された機能群を、誰が(何が)どのようにして利用するかという非機能的な要求を整理し、それに加えて、ミドルウェア、スマートカード、エージェント等の技術の研究動向を調査して、我々の要求を実現するために必要な技術を検討した。

3. Mercury ミドルウェア

本章では、先ず始めにミドルウェアの論理アーキテクチャの概要を述べ、その後、論理アーキテクチャの構成要素の解説を行う。

3.1. 論理アーキテクチャ

第2章で述べた要求分析の後、その分析結果から導かれた我々のターゲットを明確にするため

に、目指すミドルウェアの機能や構成を表現した論理アーキテクチャを構築した(図1)。このアーキテクチャは WWRF の参照モデル[4]に基づいたもので、ユーザサポート層、サービスサポート層およびネットワーク層の3層からなる。そして各層のサブシステムは、3層を貫く Dynamic Service Delivery Pattern により有機的に結合されている。

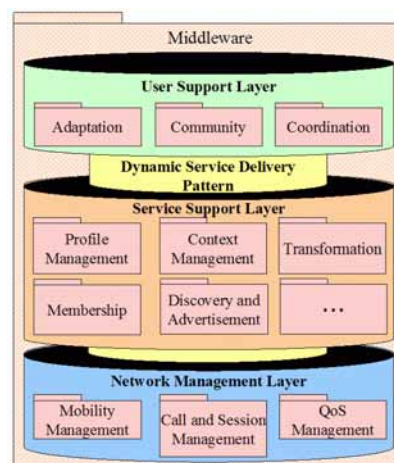


図 1 : 論理アーキテクチャ

論理アーキテクチャにおいてこのような構造を採用した理由として、ネットワーク上の既存システムの機能群を本ミドルウェアのサブシステムとして再利用するため[5]であることが挙げられる。さらにネットワーク上に分散して存在する各種サービスを動的に見つけ出し、それらを組み合わせる新たなサービスを再構成する[6]のために、3層間を有機的に結合できる仕組みが必要となるためである。

なお、本プロジェクトではネットワーク層についてはターゲットとしていないため、ここでは詳細な説明を省略するが、簡潔に説明すると、この層はルーティング、セッション管理、QoS等、ヘテロなネットワーク環境における基本的な通信制御機能を提供するサブシステムから成る。

3.2. ユーザサポート層

この層は、ユーザがシステムに対して行う操作を可能な限り削減するために設けられており、エージェントがユーザの代わりに自律的に処理を行うことでそれを実現する。以下は、この層に含まれるサブシステムである。

Adaptation : パーソナライゼーションやデータの変換等、様々なタイプの適応機能を提供する。例

えば、ユーザプロファイルに記述された嗜好情報に、あるいはコンテキストや適応ポリシーに基づいて、ユーザにとって適切であるようにサービスを適応させる。適応を行うための幾つかのテンプレートが用意されており、状況に応じて適当なテンプレートが選ばれ、そのテンプレートに記述された手順で、実際に適応を実行する。ここで言う適応は、次の3つのタイプに分類される。それらは、1)ユーザに提供するサービスの各種パラメータの変更を行う「サービス適応」、2)QoS等のネットワークパラメータの変更を行う「ネットワーク適応」、3)ミドルウェアの動作を決定付けるパラメータの変更を行う「ミドルウェア適応」である。

Community：コミュニティの生成や削除等の維持管理や、コミュニティ内のユーザ、機器、エージェントの登録・脱退の管理を行う。複数ユーザの共通目的の達成のため、コミュニティは主に短期的かつアドホックに生成・消滅され、階層化等の構造化を自己組織的に行う。

Coordination：コミュニティ毎の目標達成のためのプランを作成し、プランに沿ってコミュニティメンバー間のコーディネートを行う。例えば、資源共有のコミュニティの場合、どのメンバーがどの程度の資源を持ち、それらをどのように他のメンバーに配分するかを計画する。また、何かの障害などで状況が変化したときには、その変化に応じてプランを変更する。

3.3. サービスサポート層

この層は、ユーザにサービスを提供するために必要な基本機能を集めたものであり、ユーザサポート層のサブシステムから利用される。以下では、この層に含まれるサブシステムの中から、特に重要なものを抜粋して説明を行う。

Profile Management：各種プロファイルの登録、検索、削除、更新を行う。セキュリティの観点から、サーバ上のみならず、プログラムが動作可能なスマートカード上への実装も可能であることを前提としている。ここで想定しているプロファイルは次の3つのタイプに分類できる。それらは、1)ユーザ固有の環境や嗜好等のデータを記述したユーザプロファイル、2)サービスプロバイダが提供するサービス毎に、そのサービスを構成するための各種パラメータを記述したサービスプロファイル、3)ユーザが使用する機器、あるいはサービスプロバイダがサービスとして提供する

機器の動作を規定する設定パラメータを記述したデバイスプロファイルである。

Context Management：測定されたコンテキストを、そのコンテキストの購読者へと配信を行ったり、コンテキストプロバイダや購読者の登録・削除等の管理を行う。コンテキストの配信方法は、次の2つに分類される。それらは、1)コンテキストプロバイダから購読者にコンテキストを配信するプッシュ型、2)購読者からリクエストがあった際にコンテキストを配信するプル型である。1)のプッシュ型の配信方法にも2種類あり、1つは定期的にコンテキストを配信するタイプと、もう1つはある条件に一致した時にだけコンテキストを配信するタイプである。また、ここで扱うコンテキストの種類は、以下の4つに分類される。1)ユーザの状況を表すコンテキスト、2)時間的な状況を表すコンテキスト、3)温度、照度や交通等の物理的状況を表すコンテキスト、4)機器の処理能力や通信能力等のコンピューティング状況を表すコンテキストである。

Transformation：ユーザサポート層のAdaptationサブシステムの指示に従って、データの変換を行う。例えば、画像や映像のフォーマットやビットレートの変換、音声からテキストへというようなデータの種類の変換や、XML等の構造化文書の構造の変換を行う。

Discovery and Advertisement：サービスプロバイダが提供するサービスや、利用者が望むサービスを広告として発行したり、発行された広告を条件に基づいて検索を行う。

3.5. Dynamic Service Delivery Pattern

ユーザがネットワーク上に分散する所望のサービスを発見し、サービスの利用に関してそのプロバイダと交渉を行い、実際にサービスの提供を始めるまでの1つの流れをパターンとして定式化したものであり、以下の2つの概念から構成される。

ユーザ/プロバイダ・パラダイム：このパラダイムにおいては、全てのエンティティはあるサービスのユーザ、あるいはプロバイダとみなされる。エンティティ同士のインタラクションが発生する際に、各エンティティの責務を明確化するために必要な概念である。ユーザとプロバイダの区別は静的なものではなく、そのときの状況に応じて動的に変化するものであると捉えられる。

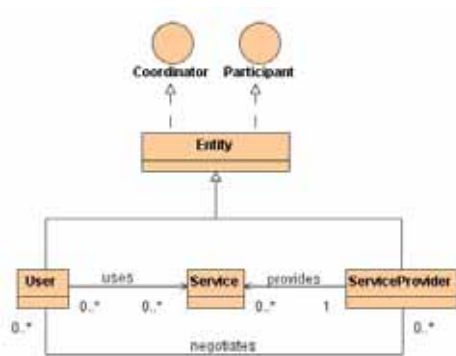


図 2：ユーザ/プロバイダ・パラダイム

ダイナミック・パラダイム：

このパラダイムにおいては、動的にサービスを提供する際の3つのフェーズを定義している。この時、全てのエンティティは、サービス提供に関する交渉のコーディネータあるいは参加者のいずれかの役割を動的に担うこととなる。

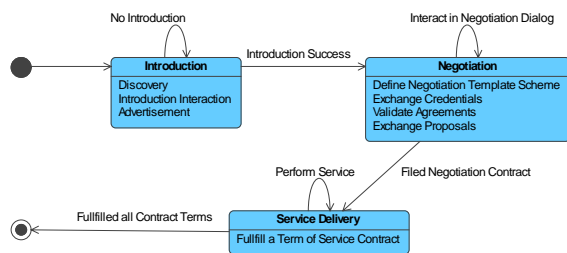


図 3：フェーズの遷移

3つのフェーズは以下の通りである。

導入フェーズ：サービスプロバイダが発行したサービスの広告をユーザが発見し、そのサービスを利用するためにコーディネータを介してサービスプロバイダを見つけ出す。そしてユーザとプロバイダの間でアクセスインターフェースの交換を行うと共に、どの交渉プロトコルで交渉を実行するかを決定することである。

交渉フェーズ：コーディネータを介して、交渉プロトコルに沿ってユーザとサービスプロバイダの間でサービスの利用に関する交渉を行う[9]。ユーザとプロバイダから提案を受け取り、提案内容のマッチングを行う。もし両社の提案内容が大きく食い違うようであれば、コーディネータは妥協案を作成し、両社にそれを送付する。ユーザとプロバイダは各々の交渉戦略に応じて妥協案に基づき再提案を行い、コーディネータは再び提案内容のマッチングを行う。

もしマッチングが成功したら、契約書を作成し、両者に同意を求めて交渉を終了させる。

サービス提供フェーズ：契約書に基づき、ユーザはサービスにアクセスするためのインターフェースをサービスプロバイダから受け取り、サービスの提供を要求する。サービスプロバイダは契約書の合意事項に則り、利用者にサービスを提供する。

4. プロタイプシステム

本プロジェクトでは、第3章で解説された機能を備えたミドルウェアと、その上で動作するアプリケーションから成るプロタイプシステムの実装を行った。本章では、まずアプリケーションの動作シナリオを紹介し、プロタイプシステムを実装するにあたり、採用したプラットフォームや技術について述べる。そしてプロタイプシステムにおける重要なコンポーネントの実装について、幾つか抜粋して解説を行う。

4.1. アプリケーションシナリオ

あるユーザが空港のラウンジに入ると、そこには個人用のビデオブースがあったため、それを利用して映画を鑑賞したいと思ったが、ブースは既に満室だったため、自分のPDAで映画を見ることにする。

PDAからラウンジのビデオ配信サービスにアクセスし、パーソナライズされたコンテンツリストから見たい映画を選択する。するとコンテンツプロバイダと価格や利用条件に関する交渉が自動的に行われ、ユーザのためにブースが予約される。その後、合意内容とユーザの嗜好や環境に合わせてパーソナライズされた映画がPDAに配信され、ユーザは映画の鑑賞を始める。

映画の鑑賞中にブースが空いたという通知が届き、ユーザは予約したブースに移動する。ユーザの移動に伴い、ブース内のディスプレイが利用可能なデバイスとして認識され、映画の配信先がPDAからブースのディスプレイに切り替えられる。そしてユーザはブースのディスプレイで映画の続きを鑑賞する。

映画の鑑賞中に、今度は搭乗時間が近づいた旨の通知が届いたため、ユーザはラウンジを出て搭乗口に向かう。ユーザの移動を察知し、再び映画の配信先がPDAに切り替わり、搭乗後もユーザは座席で映画の続きを楽しむ。

4.2. プロトタイプシステムの概要

本プロトタイプでは、図 4に示したような、前節のアプリケーションシナリオの中で起こる幾つかのイベントをミドルウェア上で実現できるように実装を行った。

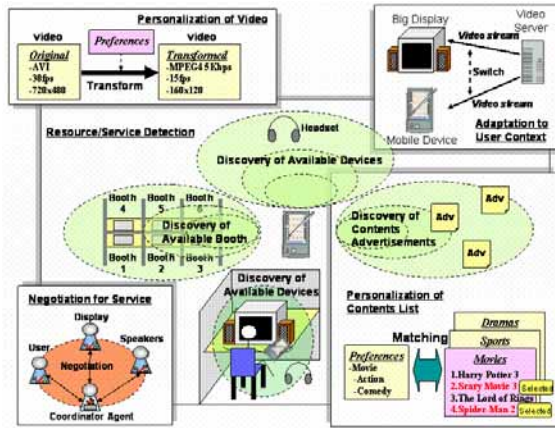


図 4：シナリオ中の主なイベント

すなわち、プロトタイプシステムの主な機能として以下の項目が挙げられる。

- ユーザの好みにあったサービスを容易に検索できる機能
- サービスプロバイダが提供するコンテンツの中から、ユーザの嗜好に合わせて自動的に選択する機能
- ユーザが選んだコンテンツの利用に関して、モバイルエージェントがサービスプロバイダとの間で自動的に交渉を行う機能
- ユーザの好みや環境に応じて、利用可能な機器を自動的に選択する機能
- ユーザの好みや環境に応じて、コンテンツを変換する機能

4.3. システム構成の概要

プロトタイプの機器構成は図 5の通りで、PDAが無線LANを通じてビデオサーバ等のPCと通信を行う。

PDAにはiPAQ h5550を用いており、CFタイプのRFIDリーダとスマートカードが利用可能なジャケットが装着できる。これにより、RFIDを利用してユーザの位置を特定することができ、またスマートカードに格納した各種プロフィールを参照することが可能となる。

iPAQのOSはWindows Mobile 2003を使用しており、Java VMとして、J2ME CDC Personal Profile 1.0をサポートしているIBM

J9がインストールされている。PCのOSはWindows XPを試用しており、Java1.4.2がインストールされている。そしてiPAQとPCの両方にJMFがインストールされている。

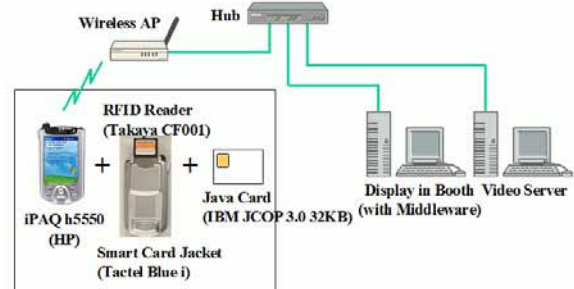


図 5：プロトタイプシステムの機器構成

4.3. プロトタイプのプラットフォーム

本プロトタイプシステムはJavaを用いて実装されており、そのプラットフォームとして、enago Mobile agent platform[7]とJXTA[8]が採用されている。以下の図 6は、プラットフォームの実装についての概要を表している。

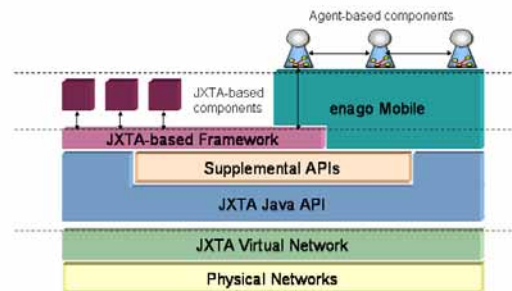


図 6：プラットフォームの実装

モバイルエージェントを利用する主な理由は、エージェントをユーザの身代わりとして働かせることで、ミドルウェアにおけるユーザサポート層の実現を図ることである。そして、JXTAを用いる主な理由は、ピア・トゥ・ピアネットワークを容易に構築できること、ピアグループを構造的に管理できること、および広告の発行・検索機能を利用できることである。

4.4. モバイルエージェントの利用

前節でモバイルエージェントをユーザの身代わりにならせることについて触れたが、それはすなわち、第 3 章で述べた Dynamic Service Delivery Pattern の下で、ユーザやサービスプロバイダ間の交渉をモバイルエージェントに自律

的に行わせることである。本プロトタイプでは、その仕組みを以下のように実装した。

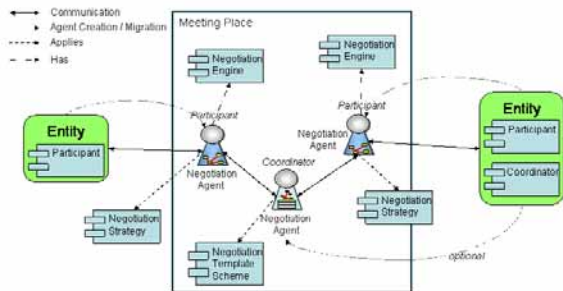


図 7 : エージェント間の交渉

Participant あるいは Coordinator インタフェースを実装したエンティティは、Negotiation Agent と呼ばれるモバイルエージェントを動的に生成し、交渉を行うためのエンジンとなる Negotiation Engine を Negotiation Agent に組み込む。そして生成されたモバイルエージェントを、ネットワーク上に複数存在するエージェントの集合場所に送信する。そこで Coordinator 役のエージェントを介して、交渉相手となる Negotiation Strategy Agent と出会い、Dynamic Service Delivery Pattern の交渉フェーズの手順に沿って交渉を始める。

交渉の間、Coordinator 役のエージェントは、交渉の進行手順が定義されている Negotiation Template Scheme に従って、交渉に関わる Negotiation Agent 達の仲介を行う。また、各 Negotiation Agent は自分達がどのように行動すべきかが記述された Negotiation Strategy に従って、互いに提案を行う。

4.5. 実装コンポーネント概要

本プロトタイプシステムの開発において、第 3 章で紹介したサブシステムのうち、ユーザサポート層の Coordination を除いて全て実装したが、ここでは特に、ユーザの嗜好や環境に応じてサービスを適応させるために必要となるコンポーネント群と、それらコンポーネント間の関連(図 8) についてのみ解説を行う。

Adaptation : Adaptation Engine が、他のコンポーネントからサービスの適応を依頼されると、まず Profile Management サブシステムを用いてサービスや機器の情報等を各種プロファイルから読み取り、その情報に基づいて適当な Adaptation Strategy を Adaptation Strategy

Repository から取り出す。得られた Adaptation Strategy には適応の手順が定義されている。Adaptation Strategy はユーザの嗜好を Profile Management サブシステムから、ユーザの状況を Context Management サブシステムから取得し、決められた手順に従って、各種パラメータの変更や、Transformation サブシステムを使ってデータの変換を行う。

Context Management : Adaptation サブシステムから依頼があったとき、Adaptation サブシステムが購読しているコンテキストの提供者を Context Manager が特定する。そして、当該 Context Provider から購読コンテキストを取得し、それを Adaptation サブシステムに返却する。Context Provider はプログラミング可能なセンサー上にも搭載され得る。

Profile Management : Adaptation サブシステムから依頼があったとき、Profile Manager がユーザプロファイル、デバイスプロファイルあるいはサービスプロファイルを検索して返却する。このコンポーネントはスマートカード上に実装される。

Transformation : Adaptation サブシステムからの指示を受け、Transformer Factory が適当な Transformer を作り出す。この時、Transformer は単体として動作するベーシックなもの(例えば、単に BMP を JPEG に変換する)の場合もあるし、ベーシックな Transformer がチェーン状に幾つか組み合わせられたより複雑なものである場合もある。

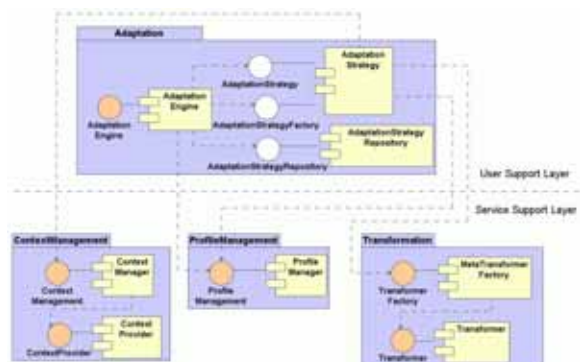


図 8 : 実装コンポーネント概要

5. 評価

本章では、プロトタイプシステムの評価方法と、その評価結果を述べる。

5.1. 評価方法

プロトタイプシステムの評価としては、そのパフォーマンスの評価を行うのが普通であるが、本プロジェクトの主要な目的の1つは、将来ユビキタスネットワーク環境が実現したときに、どのような技術を用いればユーザに対して適切なサービスを提供できるかを特定し、将来に向けた研究開発の方向性を導き出すことにある。

そこで、本プロジェクトに直接関わっておらず、また本プロジェクトで扱った技術に精通していない同僚 16 名に対してアンケートを行い、ユーザの立場からプロトタイプシステムで使われている技術の有用性を評価してもらうこととした。

図 4で説明した、本プロトタイプシステムにおける主要な機能に対して、アンケートの中で、以下の質問を行った。

- Q1. スマートカードに格納されているユーザの嗜好に応じて、アプリケーションに表示する言語の変更（英語、ドイツ語、日本語）や、ユーザの環境に応じて使用できる機器の動的な変更を行ったが、このような適応技術はユーザにとって有用であるか。
- Q2. ユーザの手間を省くために、モバイルエージェントがユーザの代わりに、サービスの利用に関して自動的にサービスプロバイダとの交渉を行ったが、このような自動交渉技術はユーザにとって有用であるか。
- Q3. 無駄な通信を削減するために、モバイルエージェントを用いて非同期通信を行ったが、このようなモバイルエージェント技術はユーザにとって有用であるか。
- Q4. 必要なときに必要なものを動的に利用できるようにするために、利用できる機器やサービスに関する広告の発行や検索を行ったが、このような広告の発行・検索技術はユーザにとって有用であるか。
- Q5. ユーザの位置を特定するために RFID を用いたが、RFID は今後様々な場面でユーザに対して有益なサービスを提供する技術となるか。

5.2. 評価結果

前節の質問に対し、被験者には以下の 5 段階での評価を行ってもらった。

- とても有用である
- 有用である

- 有用でない
- 全く有用でない
- わからない

以下の図 9はアンケートの結果を示したグラフである。

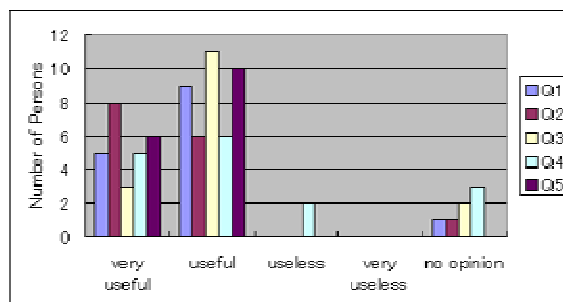


図 9：アンケート結果

全ての質問に対し、殆どの被験者がとても有用、あるいは有用であると答えている。このことは、我々のプロトタイプシステムにおいて実装を行った機能は、大抵の被験者にとって有用であると認識されたことを示すものである。すなわち、サービスの適応技術、自動交渉技術、モバイルエージェント技術、広告の発行・検索技術などを有機的に結合することでシナジー効果を発揮し、総合的に見て、本ミドルウェアはユーザに対して適切なサービスを提供していることを示している。

特に、Q2で質問した自動交渉の技術は、とても有用であると考えられており、彼らの多くはシステムとのインタラクションが少なくなることを非常に歓迎している。すなわち我々の重要な研究項目の1つとして考えてきた Dynamic Service Delivery Patternの重要性が認識されたことを意味している。

それに比べると、Q1、Q3、Q5でそれぞれ問われた適応技術、モバイルエージェント技術、RFIDに関する技術については、非常に有用とは言い切れないが、どちらかといえば有用であるというように認識されているものと捉えることができるであろう。このことは、我々のプロトタイプシステムにおいて、ユーザの満足感を得るレベルに達するまでの十分な実装が行えなかったことや、それらの有用性をアピールするための見せ方が十分でなかったことを示していると考えられる。すなわち、これらについては、もっとユーザの利便性を追及する立場からの研究が必要であると考えられる。

Q4で問われている広告の発行・検索技術は、殆どの処理がシステムのバックグラウンドで行われ、ユーザからは見えにくいものとなっていたことが、他の技術に比べて、この技術が有用でないと思われたり、特に何の意見も持たれなかった原因であると考えられる。これも前段落で述べた通り、この技術があるとこんなに便利であると思わせるような工夫が必要である。

6. 結論

本論文では、Mercury プロジェクトにおける次世代モバイルミドルウェアの開発に対する取り組みを紹介した。ピア・トゥ・ピア通信、モバイルエージェント技術、広告の発行・検索技術、サービスの適応技術および自動交渉技術など、本ミドルウェアの主要機能を構成する個々の技術については、これまでに様々な研究がなされているが、本研究ではそれらの技術を Dynamic Service Delivery Pattern を媒介として有機的に結合することにより、ユーザに適切なサービスを動的に提供できる仕組みを提案した。第5章での評価より、この仕組みの有用性を認識することができた。

今後の研究の課題は以下の通りである。将来のユビキタスネットワーク環境においては、PDAや携帯電話などのモバイルデバイスのみならず、各種センサも接続されるようになると、それらセンサから送られてくる大量のコンテキストがネットワーク上に氾濫しないように、各ネットワークノード上で効率的にセンサデータを集約し、必要なデータのみを上位ノードに上げて行く処理[10]が必要となる。さらには、幾つかのセンサデータの組み合わせから、人間が見て意味のあるような情報を抽出する仕組みが必要となるであろう。これには、データフュージョン、あるいはセンサフュージョンといった技術に対する取り組みを行うこと必要になると考えている。

謝辞

Mercury プロジェクトの遂行にあたり協力して頂いた Fraunhofer Institute FOKUS および DoCoMo Communications Laboratories Europe GmbH の関係各位に心より感謝の意を表したい。

参考文献

- [1] Chen, G., Kotz, D.: "A survey of context-aware mobile computing research", Tech. Rep. TR2000-381, Dartmouth College, Computer Science, Hanover, NH, November 2000.
- [2] Larman, C.: "Applying UML and Patterns, 2nd Ed.", Prentice Hall, ISBN 0 13 092569 1, 2002.
- [3] No Magic Inc., <http://www.magicdraw.com/>
- [4] Kellerer, W., Berndt, H., Hirschfeld, R., Prehofer, C.: "Configurable 4G system architecture", 6th WWRP meeting, London, Jun 2002
- [5] Fatoohi, R., McNab, D., and Tweten, D.: "Middleware for Building Distributed Applications infrastructure: A State-of-the-art Report on Middleware", Nasa Ames Research Center, Technical Report NAS-97-026, December 1997.
- [6] Tasic, V., Mennie, D., Pagurek, B. "On Dynamic Service Composition and Its Applicability to E-Business Software Systems - The ICARIS Experience", in Corchuelo et al. (eds.) *Advances in Business Solutions*, Catedral Publicaciones (Salamanca, Spain), ISBN: 84-96086-01-1, pp. 93-104, 2002
- [7] IKV++ Technologies AG: enago Mobile Agent Platform, http://www.ikv.de/content/Produkte/enago_mobile.htm
- [8] Project JXTA: JXTA, <http://www.jxta.org>
- [9] Bartolini, C., Preist, C., "A generic software framework for automated negotiation", Jennings, N.R., HP Laboratories Technical Report 2002-2, <http://lib.hpl.hp.com/techpubs/2002/HPL-2002-2.pdf>
- [10] Chen, G., Kotz, D.: "Context aggregation and dissemination in ubiquitous computing systems", Technical Report TR2002-420, Dept. of Computer Science, Dartmouth College, December 2001, Submitted to WMCSA 2002.