

エージェントフレームワークを用いた 車載端末向け情報提供システムの構築と評価

服部 正典^{†1} 長 健太^{†1} 大須賀 昭彦^{†1}
本位田 真一^{†2,†3} 深澤 良彰^{†4}

本稿ではユビキタス環境向けのパーソナライズエージェントフレームワークと本フレームワークによって構築した車載端末向け情報提供システムについて示す。本エージェントフレームワークは状況依存処理フレームワークとモバイルエージェントによって構成されており、ユビキタス環境向けサービスシステムの開発コストを抑えるとともに、端末とサーバシステムの連携処理をモバイルエージェントによって実行することによるレスポンスの向上を実現している。さらに、実際に車載機器向け情報提供システムを構築することによる評価を実施して、車載機器、情報家電機器などの端末と連携した多様な情報提供パターンを効率的に開発可能であることと、車載端末上で取得される位置データに基づいた情報通知処理におけるレスポンスを、走行時の通信不安定性の影響を最低限に抑えて確保することが可能であることを確認した。

Development and Evaluation of Information Provision System for In-Vehicle Terminal Based on Agent Framework

MASANORI HATTORI,^{†1} KENTA CHO,^{†1} AKIHIKO OHSUGA,^{†1}
SHINICHI HONIDEN^{†2,†3} and YOSHIAKI FUKAZAWA^{†4}

This paper proposes "Ubiquitous Personalize Agent Framework" and its application: "Information Provision System for In-Vehicle Terminal". Ubiquitous personalize agent framework is constructed by combining context reasoning framework and lightweight mobile agent platform. This framework can decrease programming cost and response latency of ubiquitous services. Information provision system for in-vehicle terminal is constructed based on the framework. We confirmed the effectiveness of the framework in terms of development cost and response speed.

1. はじめに

インターネットに代表される情報ネットワーク環境においてユーザが利用する情報機器は近年、携帯電話、情報家電、車載端末などその種類が多様化している。機器の多様化とネットワークインフラの充実によって実現が進むユビキタス環境においてユーザは様々な環境からネットワークを利用することになるが、その際

情報サービス側にはユーザのその時々利用端末や行動状況に応じた情報提供形態を持つことが期待される。すでにLBS(Location-Based Service)などにおいてユーザの環境や行動状況に関連するデータを活用した、状況依存型の情報提供サービスの実現が進んでいる。

筆者らはネットワーク上でユーザが柔軟かつ高品質な情報サービスを利用するための技術として Plan-agent¹⁾、Bee-agent²⁾ などのエージェント指向ソフトウェアの研究開発を行うとともに、ドライバ支援³⁾ や歩行者支援⁴⁾ など様々な分野への適用を通してその有効性の検証を行ってきた。それらの検証を通して筆者らは今後のユビキタス環境における状況依存型サービスの実現に関していくつかの課題を得た。

(1) 状況依存処理のレスポンス

状況依存型処理においては、GPSによる位置情報やユーザの端末操作の検出といった状況依存処理を行うトリガとなるデータの取得が車載端末や携帯電話など

†1 株式会社東芝研究開発センター
Corporate Research & Development Center, TOSHIBA Corporation

†2 国立情報学研究所
National Institute of Informatics

†3 東京大学大学院情報理工学研究所
Graduate School of Information Science and Technology, The University of Tokyo

†4 早稲田大学理工学部情報学科
Department of Computer and Information Science, Waseda University

ユーザが所持する端末上で行われる場合が多い。一方で検出したデータに関連する提供するコンテンツを生成する処理はネットワークを介して端末と接続されるサーバシステム上で実行される場合が大半である。このような処理形態は様々な端末にまたがったサービスを提供するユビキタスサービスや、その上でユーザ個人の嗜好や状況に基づいたパーソナライズ処理を必要とする場合にはさらに顕著になることが予想される。しかしこのような処理形態においては端末上とサーバ間での通信が煩雑に発生することが予想され、通信遅延がコンテンツ提供のレスポンスへ与える影響が危惧される。この問題は自動車のように通信回線が有線回線に比べ不安定かつ変動性の高い環境ではさらに深刻なものとなる。

(2) ユビキタスシステムの開発コスト

ユビキタス環境では各端末から位置情報などのセンサデータ、ユーザの端末操作履歴、音声認識などのマンマシンインタフェースから取得されたデータなどの個々の端末固有のデータが取得可能になり、その結果それらのデータを活用した各種サービスが提供可能となる。端末ならびにユーザの状況に関連するデータは今後ますます多様化し、それらを組み合わせたサービスも無数に登場することが予想される。しかし個々のサービスを特定の端末や入力データの組合せを固定的にとらえて新規に開発していくと、そのつどフルシステムの開発コストが必要となり効率的ではない。ユビキタス環境において端末とサービスは1対1に固定的に対応するものではなく、同一サービスの複数端末からの利用や新規端末を既存のサービスの提供対象に追加するといったような、端末とサービス間の柔軟な関係が必要になると考えられる。

上記のような課題に対して筆者らはユビキタス環境におけるサービスを効率的に構築するとともに、サーバと端末間の連携処理を最適化しユーザに対する情報提供のレスポンスを確保することを目的とした「ユビキタスパersonalizeエージェントフレームワーク」を開発した。本稿では本フレームワークを用いて車載端末向け情報提供サービスを構築し、自動車という通信速度の変動や切断の可能性ある環境での処理レスポンスの評価を行うとともに、車載端末以外の端末や位置情報以外のデータの新規導入にともなう開発コストおよび開発柔軟性に関する評価を行った。

以下本稿は、2章でユビキタスパersonalizeエージェントフレームワークの詳細を、3,4章では本フレームワークを用いて構築した車載端末向け情報提供システムについて示す。続く5章では本章で述べた課

題の観点から本システムの有効性に関する評価を行う。その後6章では関連研究について述べ、最後に7章で本稿のまとめと考察を述べる。

2. ユビキタスパersonalizeエージェントフレームワーク

本稿で提案するユビキタスパersonalizeエージェントフレームワークの概念図を図1に、アーキテクチャ構成図を図2に示す。本フレームワークは車載機器、情報家電や携帯電話などユビキタス環境における各種機器からユーザの行動状況に関連した様々なデータが得られるようになることを前提としており、それらのデータ群からユーザに適したコンテンツを自動的に選択し提供するという状況依存型処理を実現する。本フレームワーク中では状況依存型処理の実現のためにユーザ状況の認識処理を行うことを特徴とした「状況依存処理フレームワーク」を導入している。本フ

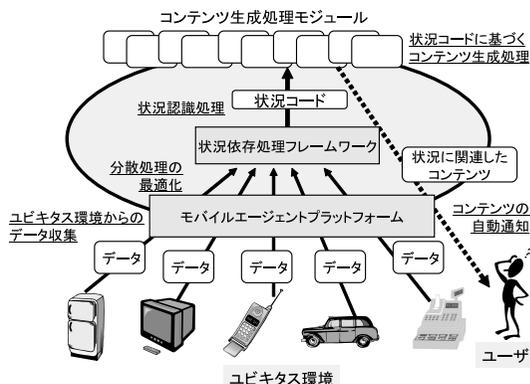


図1 ユビキタスパersonalizeエージェントフレームワーク
Fig.1 Overview of ubiquitous personalize agent.

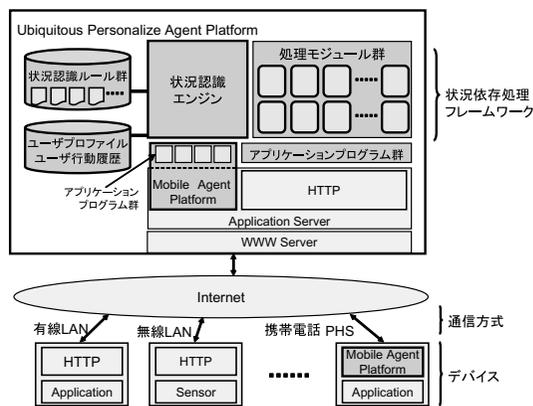


図2 ユビキタスパersonalizeエージェントフレームワークのアーキテクチャ構成

Fig.2 Architecture of ubiquitous personalize agent.

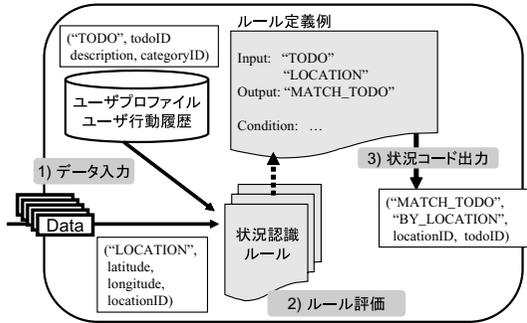


図 3 状況認識処理のメカニズム

Fig. 3 Mechanism of context reasoning.

レームワークの導入により新規の入力データや処理パターンの追加・構成に対する柔軟性を実現し開発効率の向上を図る。またサーバと各種端末との連携処理を実現する上位の通信ミドルウェアとしてモバイルエージェントプラットフォームを導入している。これにより連携処理における通信を含んだ処理効率の向上、および端末依存/非依存処理の分離による開発効率向上を図る。また状況依存処理フレームワークから端末依存処理を分離することによる開発効率の向上も期待される。

2.1 状況依存処理フレームワーク部

本フレームワークは位置や端末操作といったユーザの行動状況に関連する入力データに基づいて当該ユーザ向けの処理を選択するという一連の処理においてユーザの「状況」を表す中間データである状況コードを導入することで、一連の処理を(1)入力データに基づいたユーザ状況の認識処理(2)状況コードに基づくユーザ向け処理モジュールの動的選択の2つの段階に分離している。本フレームワークを利用する外部のクライアントプログラムは入力データを本フレームワークに与えることで、当該入力データおよびそこから予測されるユーザ状況に関連した処理モジュールを動的に獲得することが可能となる。これにより端末から取得する各種データと個々のユーザ向け処理との関連を疎結合に構成することが可能となり、新規データやユーザ向け処理の追加などに対する柔軟性を確保することが可能となる。以下に本フレームワークのメカニズムの詳細を示す。

2.1.1 状況認識処理

本フレームワークにおける状況認識処理のメカニズム、ならびに入力データ、状況コードの具体例を図3に示す。以下各手順に沿って状況認識処理の詳細を示す。

(1) 新規データ入力による状況認識処理の駆動
状況認識処理は位置データ更新などの外部クライ

```
public class TodoRule implements Rule {
//入力データ型の定義
public String[] inputTYPE
    = {"LOCATION", "TODO"};
//出力状況コードの定義
public final String CONTEXT = "MATCH_TODO";
//状況出力条件の定義
public Context resolveRule(SabioUser user,
    String[] inputs) {
//ルール判定処理
String userID = user.userID;
Location location = new Location(inputs[0]);
Todo todo = new Todo(inputs[1]);
Context context = new Context();
if (location.isRelated(todo))
    context =
        new Context(userID, CONTEXT, attributes);
return context; //状況認識結果生成
}
}
```

図 4 状況認識ルールの記述例

Fig. 4 Example of a context reasoning rule.

アントプログラムからの新規データ入力に基づいて駆動される。入力データは(‘LOCATION’, latitude, longitude, locationID)のような形式で表現され、データ型の名称(LOCATION)と関連するパラメータで構成される。

(2) 状況認識ルールの評価

本フレームワークでは状況を判定するためのルール群をあらかじめ定義し、各ルールを評価することで状況コードの出力判定を行う。ルールは、1)当該ルールが受け付ける(評価の対象とする)入力データ型の宣言、2)当該ルールが生成する状況コードの宣言、3)状況コード出力の可否を判断する条件記述から構成される。Javaによって実装されている本フレームワークにおける状況認識ルールの記述例を図4に示す。ルール定義は上記1)~3)の要素を保持する形式が規定された特定インタフェースを実装したクラスファイルとして定義される。条件の書式についてはJavaの言語仕様範囲で記述可能であるが、内部で参照できる入力データに関しては当該ルール内で明示的に宣言している入力データに制限される。本記述例では、入力データとして位置データ(LOCATION)とTODO項目(TODO)が宣言されており、定義される条件(「現在位置付近の施設分類がTODO項目の分類と関連する」)を満たした場合に「TODO通知」(MATCH_TODO)という状況コードを出力するというルール定義になっている。

(3) 状況コードの出力

全ルール定義の評価終了後、条件を満たしたルール中で宣言されている状況コードに基づいて状況認識結果を出力する。出力される結果は(‘MATCH_TODO’,

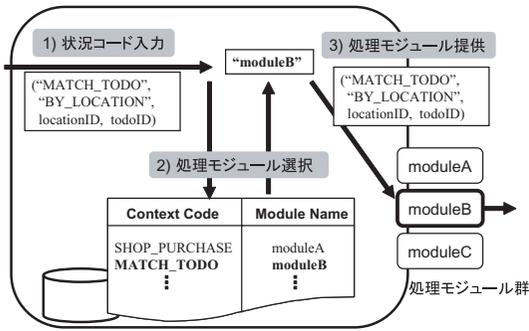


図 5 処理モジュール選択処理のメカニズム
Fig. 5 Mechanism of action module selection.

‘BY_LOCATION’, locationID, todoID) のような形式で表現され、状況コードとそれに関連するパラメータによって構成される。

2.1.2 処理モジュールの動的選択

上記処理で生成された状況認識結果に基づき処理モジュールの選択を行うことでユーザ状況に応じた処理の動的選択を実現し、状況依存処理フレームワークにアクセスしたクライアントプログラムに対して選択した処理モジュールを提供する。処理モジュール選択・提供のメカニズムおよび具体例を図 5 に示す。以下手順に沿って処理の詳細を示す。

(1) 状況コードの入力

前段の状況認識処理の結果である状況コードを入力とすることで処理モジュール選択処理が駆動される。

(2) 処理モジュールの選択

状況認識結果と処理モジュールの対応付けを行うために、本技術では図 5 に示されるような状況コードと処理モジュール名の対応表を用いる。当該対応表を用いて入力された状況認識結果中の状況コードから処理モジュール名の検索を行うことで、実際に実行する処理モジュールを動的に選択している。

(3) 処理モジュールの提供

入力データを与えて状況依存処理フレームワークに対して処理モジュールの取得要求を発行した外部のクライアントプログラムに対して、選択した処理モジュールと状況認識の結果である状況コードを提供する。クライアントプログラムは動的に獲得した処理モジュールに対して状況認識処理によって得られた結果を与えて実行を行う。処理モジュール内で行われるコンテンツ作成などの各種処理は状況認識結果に基づいて実行される。

2.2 モバイルエージェント部

本フレームワークではモバイルエージェントシステムとして各種端末上でも動作可能な軽量化を実現した知的

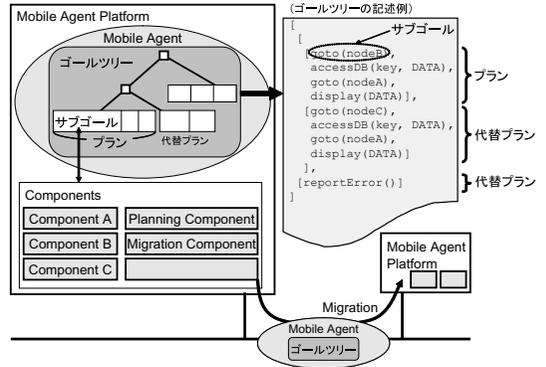


図 6 軽量のモバイルエージェント picoPlangent
Fig. 6 picoPlangent: Lightweight intelligent mobile agent platform.

軽量モバイルエージェントシステム「picoPlangent」⁵⁾を導入する。

2.2.1 picoPlangent

picoPlangent のアーキテクチャを図 6 に示す。picoPlangent は車載端末や携帯電話などの資源が乏しい計算機環境においても動作可能であると同時に、Plangent¹⁾ が持つプランニングによる知的処理も実現可能な軽量のモバイルエージェントである。本システムは移動を行う「モバイルエージェント」と移動せずに特定のエージェントプラットフォームに付属する「コンポーネント」によって構成されている。

モバイルエージェントは自身が実行するプログラムをプランと呼ばれる単位で保持する。またプランはサブゴールの列として表現される。それぞれのコンポーネントはプラン中の特定のサブゴールに対応し、対応するサブゴールを解決する役割を持つ。つまりゴール列がプログラム、プラン中の各サブゴールがエージェントから見た場合のコマンド、コマンドの実装がコマンドの型 (サブゴールの形式) で対応付けられた各コンポーネント、ととらえることができる。さらにモバイルエージェントは複数のプランを木構造の形式 (ゴールツリー) として管理し、エージェントが特定のプランの実行に失敗した場合にゴールツリー中の代替プランを実行プランに切り替えて実行を継続することが可能である。ゴールツリーの実際の記述例を図 6 中に示す。ゴールツリーは各コンポーネントを起動するコマンドとなるサブゴールから成る階層的なリスト構造として表現される。

コンポーネントの導入によって特定の実行環境に依存する処理をモバイルエージェントの外部に配置することでモバイルエージェントの軽量化が実現される。またエージェント移動などの基本機能も含めた多くの

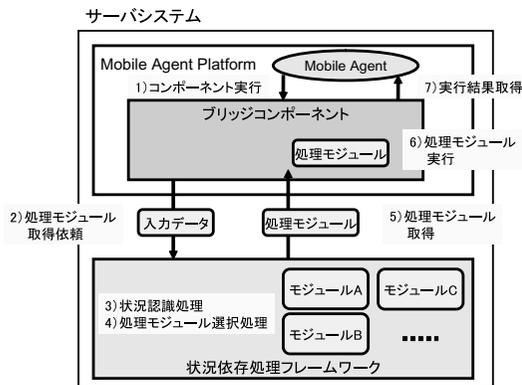


図7 ブリッジコンポーネントによるモバイルエージェントフレームワークと状況依存処理フレームワークとの連携

Fig.7 Connection between mobile agent framework and context reasoning framework via bridge component.

処理をコンポーネントとして独立させプラットフォーム上に配置する構成を採用しているため、プラットフォーム単体の軽量化とともにコンポーネントの配置数を調整することによる個々の実行環境への柔軟な対応も可能となる。本アーキテクチャに基づき各端末の計算機資源に合わせたコンポーネントの最適配置を行うことで、各端末の計算資源を有効に活用した分散処理が実現可能となる。またコンポーネントを端末固有の機能への依存度によって分離することで端末非依存のコンポーネントを他の端末上で再利用することも可能となる。

2.2.2 コピキタスパーソナライズエージェントフレームワークへの適用

1章でも述べたように、コピキタス環境における状況依存型処理ではサーバ/端末の双方にまたがった処理が必要な場合が多く、応答速度や計算効率を確保することが重要な課題となる。そこで本稿ではサーバ上で動作する状況依存処理フレームワークとモバイルエージェントプラットフォームフォーム picoPlangent とを組み合わせることにより、サーバで実行される処理の状況依存処理の一部を端末上に委譲することで処理の分散配置を実現し応答速度の向上を図る。

モバイルエージェントフレームワークと状況依存処理フレームワークの連携メカニズムを図7に示す。両者の連携はモバイルエージェントプラットフォーム内にブリッジコンポーネントを導入することで実現される。ブリッジコンポーネントはモバイルエージェント側からは通常のコンポーネントとして機能するが、コンポーネント内部の処理は状況依存処理フレームワークから動的に獲得する。つまりブリッジコンポーネントは状況依存処理フレームワークに対して入力データ

を与え処理モジュールを取得するクライアントとして機能する。その結果ブリッジコンポーネントは状況依存処理フレームワークから獲得した処理モジュールを実行した結果をモバイルエージェントの一部として端末側に移送し、端末上での実行に活用することが可能となる。

3. 車載端末向け情報提供システムの設計

本章では2章で述べたエージェントフレームワークを用いた車載端末向け情報提供システムの設計について示す。本システムの全体構成を図8に示す。本システムは車載端末、情報家電、デスクトップPC、PDAの各端末とサーバシステムで構成される。ユーザが各端末を利用する際に発生するTODO項目、スケジュール項目、Webページのブックマーク、買物メモ情報などの入力データは端末上のデータ取得デモンもしくはWWWサーバのアクセス履歴などからサーバシステムに自動的に収集・蓄積される。コピキタスパーソナライズエージェントフレームワークは車載端末の利用時にその走行位置などを入力データとして、そのときのユーザに適したコンテンツを動的に生成しユーザに対して通知する。たとえば走行中に郵便局付近を通過する可能性がある際に過去にユーザが登録した「小包を出す」というTODO項目をその施設情報とともにユーザに通知する、といった動作を行う。

上記システムを実現するために、車載端末の位置や時刻に基づいて情報家電やPC、PDAから収集した情報から関連するコンテンツを動的に生成する枠組みに状況依存処理フレームワークを適用する。また車載端末とサーバシステム間の連携処理部分にモバイルエージェントを適用することで、低速かつ速度変動性の高い通信路におけるコンテンツ通知レスポンスを確保する。

3.1 状況依存処理フレームワーク部

実証システムにおいて各種端末から収集する各種入力データを表1に定義する。これらのデータは状況認識処理を駆動するトリガ入力として利用されるとともにその一部はユーザに通知されるコンテンツの素材として利用される。

次に本システムにおいて定義する主要な状況コードと状況認識ルールの一覧を表2に示す。個々のルールは状況認識処理時に与えられた入力データに基づき評価され、ルールの判定条件を評価しそれが満たされる場合に当該ルールで定義している状況コードが出力される。表2に定義されるルールの一例として、たとえば「TODO通知」(MATCH_TODO)という状況コードは

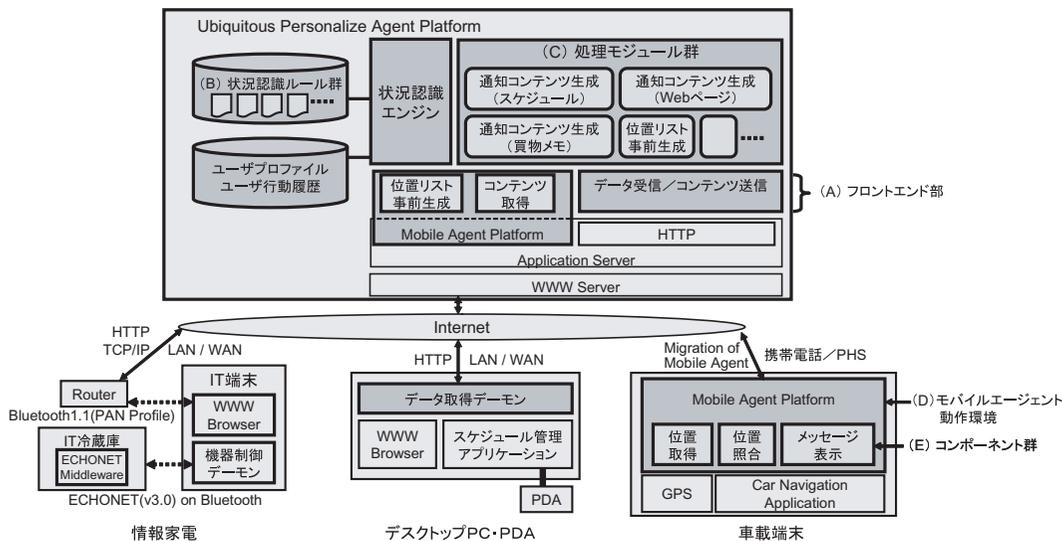


図 8 車載端末向け情報提供システムの構成

Fig. 8 Architecture of information provision system for in-vehicle terminal.

表 1 各端末から収集される入力データの定義

Table 1 Definitions of input data.

データタイプ	取得端末	内容
TODO	PC/PDA	TODO 情報
SCHEDULE	PC/PDA	スケジュール情報
WEB_MARK	PC	Web ページブックマーク
SHOPPING_MEMO	情報家電	買物メモ情報
LOCATION	車載端末	GPS による位置データ
ACTION_LOG	全端末	ユーザの端末操作履歴
TIMER_EVENT	サーバ内部	タイマイベント
REQ_LOCLIST	車載端末	位置リスト取得依頼

表 2 状況認識ルールの定義

Table 2 Definitions of context reasoning rules.

状況コード (内容)	入力データ型	状況コード出力の判定条件
MATCH_TODO (TODO 通知)	LOCATION TODO	現在位置近辺の施設分類が TODO 項目の分類と関連
MATCH_SCHD (スケジュール通知)	LOCATION SCHEDULE	現在位置近辺の施設分類が スケジュール項目の分類と関連
MATCH_SCHD (スケジュール通知)	TIMER_EVENT SCHEDULE	現在日時がスケジュール項目の設定時刻と一致
MATCH_WEBM (ブックマーク通知)	LOCATION WEB_MARK	現在位置近辺にブックマークした施設が存在
MATCH_SHOPMEMO (買物メモ通知)	LOCATION SHOPPING_MEMO	現在位置近辺の店舗分類が 買物メモ項目の品種と関連
CREATE_LOCLIST (位置リスト事前生成)	REQ_LOCLIST	(無条件)

表 3 処理モジュールの定義

Table 3 Definitions of action modules.

状況コード	処理モジュール名 (クラス名)	処理内容
MATCH_TODO	TodoMsgCreator	TODO 通知コンテンツの生成
MATCH_SCHD	SchdMsgCreator	スケジュール通知コンテンツの生成
MATCH_WEBM	WebmMsgCreator	ブックマーク通知コンテンツの生成
MATCH_SHOPMEMO	ShopmMsgCreator	買物メモ通知コンテンツの生成
CREATE_LOCLIST	LocListCreator	通知候補となる位置リストの事前生成

車載端末で取得される位置データ (LOCATION) と登録されている TODO 項目 (TODO) を入力とする状況認識ルールの評価において現在位置と TODO 項目の関連に関する判定処理が実行され、判定条件が満たされた際に出力される。

表 2 に示した状況コードに対応する処理モジュールの対応表を表 3 に示す。本対応表は生成された状況コードに基づき処理モジュールを動的に選択する際に利用される。実際のシステムではこれ以外に端末から受信したデータ種別に応じたメタデータの抽出処理/データベース登録処理なども処理モジュールの形で実装する。

3.2 位置に基づくコンテンツ自動通知処理の実現

本システムでは車載端末とサーバシステム間で実現される車載端末の位置データに基づいたコンテンツ自動通知機能を 2.2.2 項で示したモバイルエージェント

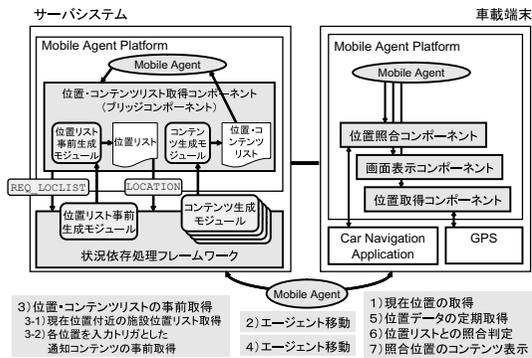


図 9 位置に基づいたコンテンツ自動通知機能の構成

Fig. 9 Construction of location-based content notification function.

プラットフォームと状況依存処理フレームワークとの間の連携メカニズムを用いて実現することで、低速かつ速度変動性の高い通信回線における応答速度（コンテンツ通知レスポンス）の向上を図る。

3.2.1 各機能の配置

本機能に関連する車載端末およびサーバシステムの機能配置を図 9 に示す。車載端末上には GPS を利用した位置取得コンポーネントや画面表示コンポーネントといった端末固有の機能を利用するコンポーネントを配置する。それに加えて、取得した現在位置データをサーバに送信せずに端末上で評価するための位置照会コンポーネントを車載端末上に配置する。位置照会コンポーネントは特定のコンテンツと対応付けられている位置データ（コンテンツに関連する施設の所在位置）と位置取得コンポーネントから取得される現在位置データとの距離を算出し、その距離が特定の値以内（例：車載端末では 500 m 以内と設計）であった際に位置照会（当該施設付近にいる）と判定する。位置照会の際には画面表示コンポーネントによって該当する位置データと対応付けられているコンテンツが表示される。

上述したような端末上でのコンテンツ自動通知をサーバとの通信を一定時間行わずに実現するためには、照会対象となる位置データとそれに対応する当該ユーザ向けのコンテンツから成るリストを事前取得しておく必要がある。本システムではモバイルエージェントによってサーバ上の状況認識フレームワークと端末上のコンポーネントとを連携させることで上述のリスト（位置・コンテンツリスト）の事前取得を行い、サーバとの通信回数を抑えた処理を実現する。

図 9 に示すように、サーバ上のモバイルエージェントプラットフォームには状況依存処理フレームワー

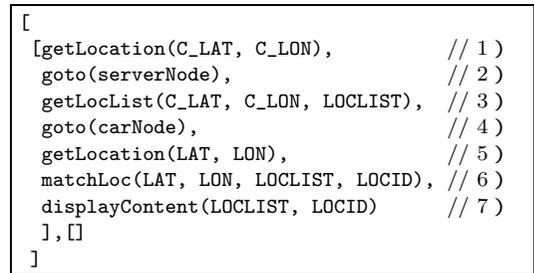


図 10 モバイルエージェントが保持するプラン

Fig. 10 Plan kept by mobile agent.

クをアクセスするためのブリッジコンポーネントを配置し状況依存処理フレームワークから動的に処理モジュールを獲得する。

3.2.2 モバイルエージェントの動作

前項で示した機能配置に基づいて位置による自動通知機能を実現するために、モバイルエージェントが実行する手続き（ゴールツリー）を図 10 に示す。本ゴールツリーは 1 つのプラン（サブゴールの列）から構成されており、図 6 の例で示したような代替プランは含んでいない。以下本プランの実行順に沿って動作の詳細を示す。

本プランを実行するモバイルエージェントは端末上で図 10 に示されるゴールツリーを与えられ生成される。生成されたモバイルエージェントは、1) 端末上の位置取得コンポーネントにアクセスして現在位置を取得する。次にモバイルエージェントは、2) サーバシステム上に移動し、3) サーバ上の位置・コンテンツリスト取得コンポーネントに現在位置データを与えることで端末に運ぶ位置・コンテンツリストの取得要求を行う。

モバイルエージェントから起動されたコンポーネントはブリッジコンポーネントとして機能し、3-1) 入力データ（REQ_LOCLIST）を状況依存処理フレームワークに与えて処理モジュールの取得依頼を行う。状況依存処理フレームワークは表 2 で定義されている当該入力データに対応するルールを評価し状況コード（CREATE_LOCLIST）を出力し、次に表 3 で対応付けられている位置リスト事前生成モジュール（LocListCreator）を選択し、ブリッジコンポーネントに提供する。位置リスト事前生成モジュールの提供を受けたブリッジコンポーネントは当該モジュールに現在位置データを与えて実行することで、データベースから検索された現在位置付近に存在する各種施設の位置データから成るリストを取得する。

次にブリッジコンポーネントは、3-2) 得られた位置

データリスト中の各項目（位置データ）を仮に車載端末が当該位置付近にいたと仮定してピックアップし、当該位置データを仮の入力データ（LOCATION）として状況依存処理フレームワークに与え、コンテンツ生成のための処理モジュールの取得依頼を行う。仮の位置データを入力として受け取った状況依存処理フレームワークは表2で定義されている位置データを入力とするルール群の評価を行う。判定条件を満たしたルールが存在する場合には当該ルールに定義されている出力状況コードを生成し、表3で当該状況コードごとに対応付けられているコンテンツ生成処理モジュールを選択、ブリッジコンポーネントに提供する。コンテンツ生成処理モジュールの提供を受けた場合、ブリッジコンポーネントはピックアップした位置データを与えて当該モジュールを実行することで、車載端末が実際に当該位置データ付近に移動した際に表示されるべきコンテンツを事前取得する。取得された位置データとコンテンツの対は位置・コンテンツリストに格納される。位置リストからの仮入力データのピックアップによるコンテンツ事前取得処理を、位置・コンテンツリストが一定のサイズ（例：30項目）を満たすまで繰り返した後に、モバイルエージェントに対して当該リストを提供する。

ブリッジコンポーネントから位置・コンテンツリストを取得したモバイルエージェントは、4) 当該リストを保持したまま車載端末上に移動し、5) 現在位置データの取得と、6) 取得した現在位置データと位置・コンテンツリスト中の位置データとの照合処理を定期的に行う。位置照合処理において現在位置が位置・コンテンツリストのいずれかの項目に照合した場合には、7) 該当項目のコンテンツを通知メッセージとして表示する。

上記のモバイルエージェントの処理手順中で、5)、6)の繰り返し実行される部分に関しては車載端末上で完全に閉じて実行されており、位置・コンテンツリストの再更新を行わない限りサーバとの通信処理はいっさい生じない。なおリストの再更新処理は定期的（例：10分周期）にモバイルエージェントを1)の手順から再実行すると実現される。このようにコンポーネントの分散配置とモバイルエージェントによる連携処理を適用することで、サーバとの間で発生する通信回数を最小化し、その結果として即時性の高い位置に基づくコンテンツ通知処理が実現可能となる。

4. システムの実装と動作

4.1 サーバ部の実装

サーバシステムはアプリケーションサーバ（iPlanet Application Server）上に各端末との通信を行うフロントエンドのアプリケーション部を Java サブレットおよびモバイルエージェントプラットフォーム上のコンポーネントとして実装することで構築した。状況依存処理フレームワークは Java のフレームワークとして実装した。状況依存処理はフロントエンド部から共通に利用される。

状況依存処理フレームワーク上には表1、表2で設計した個々の入力データ型、出力状況コードをクラスとして実装した。個々の状況認識ルールは図4で示したように特定インタフェースを実装したクラスとして定義し、表2で定義した各状況認識ルールを実装した。個々の処理モジュールも処理モジュールの形式を規定するインタフェースを実装するクラスとして実装され、表3で設計したモジュールに加えて、位置候補リストの生成処理モジュールなどを実装した。状況コードと処理モジュール名の対応表、および位置照合処理に必要な各種施設データ（施設名称、施設分類、緯度経度情報）はデータベース上で管理する。

サーバ上のモバイルエージェントプラットフォーム（picoPlangent）は Java のサブレットとして実装したものを利用した。サーバ上には図9で示したように状況認識フレームワークとの連携を行うブリッジコンポーネントを配置し、端末上での位置照合・コンテンツ通知処理の実行のために必要なデータの取得を行う。

4.2 端末部の実装

デスクトップ PC 上では Web ブラウザのブックマークおよび PDA（PalmOS 搭載機）と同期した TODO/スケジュールデータを自動収集するためのデーモンプログラムを Java で実装した。また、該当 Web ページ内の住所/郵便番号記述を基に緯度経度情報を抽出する処理と、TODO/スケジュール受信サブレットでは記述内容中の単語に対応する施設分類を抽出する処理を処理モジュールとして実装した。これにより状況認識ルール内で現在地付近の施設分類との対応判定を行うために必要なメタデータの抽出を実現している。

情報家電部には東芝が 2002 年に発売した情報家電機器「フェミニティ」シリーズ⁶⁾を利用した。本機器は Bluetooth によってワイヤレスでの家電ネットワークコントロールを実現しており、家庭内端末として家電コントロール機能などの役割を持つ IT ホーム端末と、本端末との間で Bluetooth で通信を行うことが可



図 11 情報家電機器 (左から IT レンジ, IT 冷蔵庫, IT ホーム端末)

Fig. 11 Intelligent home appliances.

能な家電製品群によって構成される (図 11). 本システムにおいては各機器が持つ在庫管理機能やレシピ検索サービスと連動した買物メモ登録機能を IT ホーム端末上の Web ベースのサービスとして実装した. 登録される買物メモ項目には分類情報が付加される.

車載端末部はノート PC (Windows2000) 上にカーナビゲーションアプリケーション, モバイルエージェントプラットフォームを Java により実装することで実現した. モバイルエージェントプラットフォーム上には GPS を利用した位置取得処理, 候補位置リストとの照合処理, 画面上へのコンテンツ表示処理のそれぞれをコンポーネントとして実装した. サーバとの通信には無線 LAN, 携帯電話や PHS などの無線通信環境を利用することを想定する.

4.3 システムの動作

図 12 は情報家電 (IT ホーム端末) 上に表示される買物メモ管理サービスの画面で, 買物メモ情報の登録/削除/閲覧機能が提供される.

図 13 は車載端末上の画面例である. 通常時はカーナビゲーションシステムとしての機能を提供し, エージェントによるコンテンツ通知の際には地図画面上に通知コンテンツがオーバーラップ表示されると同時に, 通知内容に該当する地図上の地点にアイコンが表示される. 図 14 に車載端末上で通知される各種通知コンテンツを示す. コンテンツは施設情報, およびそれに関連するユーザデータ (TODO/スケジュール/Web ページ/買物メモ) から構成されている.

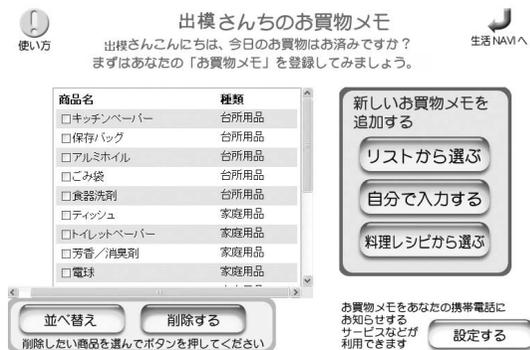


図 12 情報家電端末の画面

Fig. 12 Screenshot of shopping assistance service.



図 13 車載端末上の画面

Fig. 13 Screenshot of in-vehicle terminal.

5. 評価

本章では構築した車載端末向け情報提供システムに基づき, 本稿で提案したユビキタスパersonライズエージェントフレームワークの有効性に関する評価を 1 章で示した 2 つの課題の観点に基づいて行う.

5.1 状況依存処理のレスポンスに関する評価

本節では 1 章で示した状況依存処理のレスポンスに関する課題の観点から, 本稿で導入したモバイルエージェントプラットフォームの有効性に関する評価を行う. 評価は 3.2 節に示したモバイルエージェントを適用して実現した位置に基づくコンテンツ通知処理の応答時間を測定し従来手法と比較することで行う.

本評価の比較対象として用いるサーバ処理型の処理構成を図 15 に示す. 本構成は図 9 に示したモバイルエージェントによる処理構成と比較すると, モバイルエージェントによる方式では車載端末上で実行されていた位置データの照合処理がサーバ上で実行されることが大きな違いとなる. サーバ処理型の方式では車載

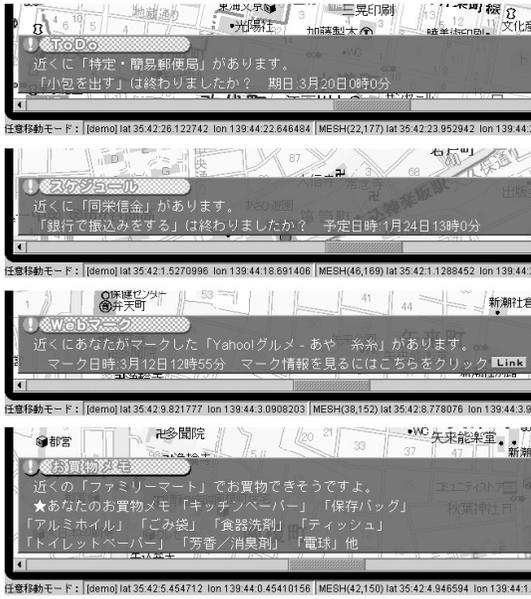


図 14 車載端末上のコンテンツ通知画面（上から TODO, スケジュール, ブックマーク, 買物メモ）

Fig. 14 Screenshots of notification messages.

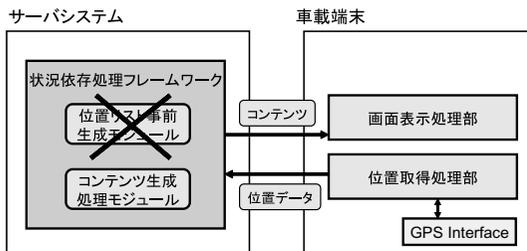


図 15 比較評価対象：サーバ処理型の処理構成

Fig. 15 Diagram of the server-centric architecture.

端末上で GPS により取得された現在位置データが逐次サーバ上に送付され、状況依存処理フレームワークへの入力データとして直接処理される。現在位置データと通知候補の施設位置リストとの照合処理は 2.1.1 項で示したような流れで状況依存処理フレームワーク内部で実行されるため、3.2.2 項で示した位置リストの事前生成モジュールについてはいっさい利用しない。

通知レスポンスの評価のため、モバイルエージェント型とサーバ処理型の双方の処理構成で車載端末上での GPS による位置取得完了後から同端末上で通知メッセージが表示されるまでの応答時間を測定した。測定は自動車内に車載端末用システムをインストールしたノート PC を設置して実施し、車載端末用 PC とサーバとの接続には PIAFS2.1 (ベストエフォート型、最大通信速度 64 kbps) 準拠の PHS を用いた。測定は

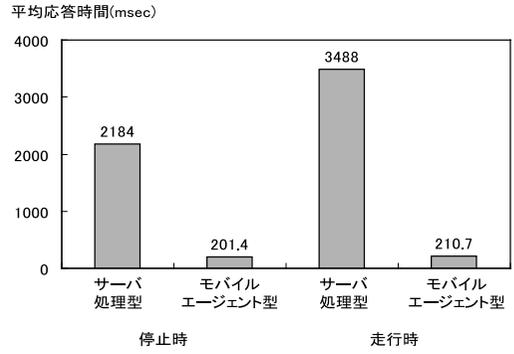


図 16 コンテンツ通知レスポンスの比較

Fig. 16 Comparison of message notification response.

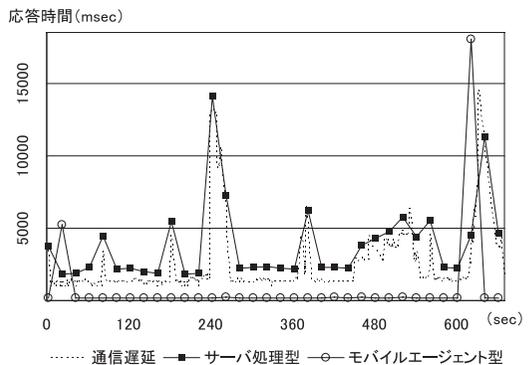


図 17 走行時におけるコンテンツ通知レスポンスの変動

Fig. 17 Measurement of message notification response while moving.

常時停止した状態と走行時に分けて実施し、走行時に予想される通信速度の変動が与える影響についても調査した。所要時間の測定周期 (GPS データの検出周期) を 20 秒に 1 回と定め、停止時・走行時ともに 5 時間分の測定データを収集して分析を行った。またモバイルエージェントがサーバから車載端末へと運ぶ位置・コンテンツリストの最大件数は 30 件とし、同リストの再更新周期は 10 分と設定した。

図 16 に測定した応答時間の平均値を示す。停止時・走行時ともにモバイルエージェント型の応答時間がサーバ処理型に対して大幅に短いことが確認できる。これはサーバ処理型では位置照合処理のためにサーバとの通信が発生し、その通信による遅延が大きく影響したためと考えられる。また停止時と走行時を比較した場合には、走行時の平均応答時間が停止時を上回っていることも確認できる。これは走行時の通信環境が停止時と比較して安定しておらず、通信遅延の増加として影響を与えたと考察できる。

図 17 に走行時の応答時間測定値の時系列グラフを

示す．本グラフでは通信速度の変動との関連を確認するためにサーバへの ICMP (Internet Control Message Protocol) エコー要求の応答時間の測定結果も併記している (図 17 点線部分, ICMP パケットサイズ 2KB で測定). グラフからはサーバ処理型の応答時間が走行時の通信速度変動と連動していることが確認できる．逆にモバイルエージェント型の応答時間に関しては通信速度の影響を受けずに安定していることが確認できる．ただし, 図 17 のモバイルエージェント型の応答時間データの中にはその平均を大幅に上回る 5 秒以上のデータが存在している (0 秒および 600 秒付近の 2 カ所). これは今回の評価実験ではモバイルエージェントが保持する位置・コンテンツリストの再更新周期を 10 分ごととしており, その際にモバイルエージェントの移動が発生したことが関係している．そのため通信遅延が応答時間に影響したと考えられる．

5.2 開発コストに関する評価

本項では 1 章で示した開発コストに関する課題の観点から, 本稿で導入した状況依存処理フレームワークの有効性に関する評価を行う．具体的には 3, 4 章で設計・構築したシステムに対して新規端末および新規入力データの導入を行い, そこで発生する開発コストおよび開発容易性についての評価を行う．

5.2.1 新規端末の導入

本項では新規のサービス対象端末として携帯電話端末を導入する際に必要となる開発コストについて評価する．追加する携帯電話端末は GPS および Java 実行機能を持つ端末で, 車載端末上で実現した端末位置データに基づくメッセージ通知と同等の機能を実現する．

表 4 に実際に行った新規開発項目および各項目における新規追加コードの割合を示す．新規に導入する端末側の開発は基本的にはすべて新規開発となるが, 位置照合コンポーネントに関しては車載端末向けに開発したものを再利用することが可能であった．その結果新規コードの割合は 44.8% に抑えることができた．これは本稿で導入したモバイルエージェントプラットフォームでは, 実現する各機能をコンポーネントの単位で独立して構築するアーキテクチャとなっていたことに起因している．これにより端末非依存のコンポーネントを再利用することができており, 開発効率向上の効果が確認できる．

一方, サーバ側で新規開発を実施したのは携帯電話とサーバ間でのデータ送受信のためのサーバ側のフロントエンド部分のみにとどまり, サーバ部全体に対する新規追加コードの割合も 5.48% にとどまった．

表 4 携帯電話導入時の開発項目

Table 4 Development items for adding mobile phone.

開発箇所 (サーバ/端末)	新規コード割合 (新規ステップ数/ 全ステップ数)	内容
フロントエンド (サーバ, 図 8 (A))	12.7% (398/3129)	携帯電話とのデータ送受信部のみ拡張
状況認識ルール (サーバ, 図 8 (B))	0% (0/387)	既存ルール群を再利用
処理モジュール (サーバ, 図 8 (C))	0% (0/3752)	既存処理モジュール群を再利用
サーバ部全体	5.48% (398/7268)	
モバイルエージェント動作環境 (端末, 図 8 (D))	100% (350/350)	携帯電話用のプラットフォームを導入
コンポーネント (端末, 図 8 (E))	100% (10/10)	携帯電話上での位置取得機能を新規開発
コンポーネント (端末, 図 8 (E))	100% (180/180)	携帯電話上での画面表示機能を新規開発
コンポーネント (端末, 図 8 (E))	0% (0/665)	既存の位置照合コンポーネントを再利用
端末部全体	44.8% (540/1205)	

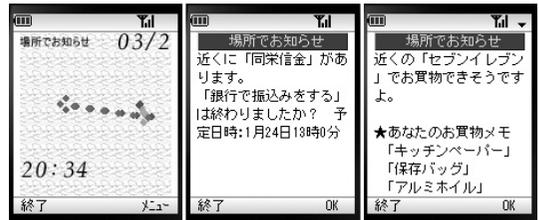


図 18 携帯電話上の画面 (左から待機画面, TODO 通知画面, 買物メモ通知画面)

Fig. 18 Screenshots of messages on mobile phone.

特に位置データに基づいて各種コンテンツ通知を実現するという状況依存型処理を実現する部分については, 状況依存処理フレームワーク上にすでに構築されている既存のルールおよび処理モジュールをそのまま再利用することができており (新規コード割合 0%), 端末依存の処理を状況依存型処理から分離したアーキテクチャを導入したことによる開発効率向上の効果が確認できる．

本追加開発により実現した携帯電話上のコンテンツ通知アプリケーションの画面を図 18 に示す．車載端末上で表示されるものと同様, 端末位置と連動したコンテンツ通知が携帯電話上でも実現されている．

5.2.2 新規入力データの導入

本項では車載端末上の音声認識機能を利用してユーザの発話単語に関連するメッセージの通知を行う新規処理の導入を行い, そこで生じた開発コストについて

表 5 音声入力対応時の開発項目

Table 5 Development items for adding voice input.

開発箇所 (サーバ/端末)	新規コード割合 (新規ステップ数/ 全ステップ数)	内容
フロントエンド (サーバ, 図 8 (A))	3.75% (122/3251)	発話単語の受信インタ フェース部のみ拡張
状況認識ルール (サーバ, 図 8 (B))	29.3% (160/547)	発話連動型の情報通知 ルール (4 件) を新規追 加
処理モジュール (サーバ, 図 8 (C))	5.99% (239/3991)	既存のコンテンツ生成 モジュール (4 件) を拡 張
サーバ部全体	6.69% (521/7789)	
モバイルエージェント 動作環境 (端末, 図 8 (D))	0% (0/687)	既存のプラットフォー ムを再利用
コンポーネント (端末, 図 8 (E))	100% (273/273)	外部音声認識エンジ ンアクセス部を新規追加
コンポーネント (端末, 図 8 (E))	0% (0/855)	既存の画面表示・位置取 得/照合機能を再利用
端末部全体	15.0% (273/1815)	

評価する。

表 5 に実際に行った新規開発項目および新規追加コードの割合を示す。サーバ側の新規開発項目は発話単語の受信インタフェースの拡張、発話単語に基づいた情報通知を判定するための新規状況認識ルールの追加、発話連動型通知コンテンツ文面追加のための既存コンポーネントの拡張などで、サーバ部全体に対する新規追加コードの割合は 6.69%にとどまった。特に発話単語に関連した新しい処理パターンを追加する部分に関しては、新規ルールの追加（新規 4 件を追加）と関連する既存モジュールに対する拡張（全 26 件中 4 件のみ拡張）という形で実施でき、既開発部分への影響を最小限にとどめかつ独立に追加開発を行うことができた。これは状況依存フレームワークにおいて状況依存型処理のための開発単位を独立かつ疎結合にしたことに起因しており、状況依存処理フレームワーク導入による開発効率の向上（新規ユーザデータの追加に対する柔軟性）の効果が確認できる。

一方、端末側における新規コードの割合は 15.0%であったが、実際には発話単語を取得するための機能を新規コンポーネントとして 1 件追加したのみにとどまった。他の部分に対する新規コードの割合はすべて 0%となっており、既存部分については影響もなく完全に再利用することが可能であった。

本追加開発により実現した車載端末上でのユーザの発話単語に基づく通知コンテンツの例を図 19 に示す。



図 19 音声入力に基づくコンテンツ通知画面

Fig. 19 Screenshot of a message based on voice input.

既存のメッセージ生成モジュールに追加した音声連動通知用のコンテンツテンプレートによって、発話単語（本例では「食事」）と連動したコンテンツ通知が実現されている。

6. 関連研究

本章では関連技術について述べる。ユビキタスコンピューティングの研究分野では状況依存型処理を実現する技術への取り組みがさかんである。たとえば文献 7) では状況依存型のサービスを柔軟に構成する環境が提案されており、新規サービス構築時の再利用性が実現されている。ただし同文献では本稿で提案したようなユーザの明示的要求なしにユーザの状況を自動認識し処理を駆動するという方式は提案されていない。

状況依存型処理の中でも位置情報の活用に関する取り組みはさかんである。文献 8) では位置情報を用いたスケジュール通知システムが提案されている。同システムではコンテンツの通知方式が CTI もしくはメールに限られているが、本稿で示したシステムでは様々な端末に応じた通知処理を同一フレームワーク上に実現可能な柔軟性を備えている。また文献 9) では位置依存情報提供システムにモバイルオブジェクトの概念を導入し分散処理の実行効率を向上する手法が提案されている。ただしその目的は端末処理の軽減が主体であり、本稿で提案したような端末上での処理を活用したレスポンスの向上については検討されていない。

文献 10) などでも指摘されているように、ITS の研究分野では自動車からのネットワーク接続性の確保が重要な研究課題となっている。筆者らは過去にモバイルエージェントを適用したドライバ情報支援システムを構築³⁾している。本システムは車載端末からの情報検索処理をモバイルエージェントによって実現することで、通信回線切断時における処理継続（disconnected operation）を実現した。ただし上記研究ではレスポンスに対する考慮はなされておらず、情報通知速度の向上が大きな課題として残ったことは本稿における目的の 1 つとなっている。類似のアプローチとして文献 11) などがあげられるが、そこでは協調型エージェントが導入されておりモバイルエージェン

トを導入した本稿のアプローチとは異なる。

上述のドライバ情報支援システムでは通信回線の切断が発生することを前提として、通信レイヤより上位のレイヤでその影響を低減することを目的としていたが、インターネット自動車プロジェクト¹²⁾では自動車からのネットワークコネクティビティを安定的に提供することを目的とした研究が行われている¹³⁾。本プロジェクトでは無線 LAN, PDC など複数の無線通信媒体を通信状況に応じて動的に切り替えることで連続的に安定した IP 通信レイヤの提供を実現している。ただし複数の通信媒体を切り替えて利用するためその通信速度は状況に応じて変動し、アプリケーションのレスポンスに対する影響は残る。本稿で提案した方式では 5.1 節で確認されたように通信速度が変動する状況でもその影響を最小限にとどめ安定したレスポンスが実現可能であり、両者が相補的に機能することでより効果的に働くと考えられる。

7. おわりに

本稿では状況依存型処理におけるレスポンス確保とユビキタスシステムにおける開発効率向上を目的としたユビキタスパersonalizaエージェントフレームワークを提案し、本フレームワークによって構築した車載端末向け情報提供システムに基づいた評価を行った。本技術は状況依存処理フレームワークとモバイルエージェントプラットフォームによって構成されており、その評価結果からは走行中の通信回線における不安定性の影響を低減可能であることと、様々な端末/端末取得データを活用して実現されるユビキタス/状況依存型サービスを効率的に開発可能であることが確認され、提案した手法の基本的な有効性が確認された。

ユビキタス環境の普及にともない、ユーザの利用端末、取得可能なセンサ情報、通信環境などはさらに増加・多様化し、ユーザの状況にフィットした様々な情報提供サービスを低コストかつ効果的に提供することが期待されるなか、本稿で示した課題はより重要なものとなることが予想される。今後はモバイルエージェント自身のサイズや移動周期が与える影響を考慮した処理レスポンスに関するさらなる評価や、異なる応用システムの開発を通じた本フレームワークの開発効率に関する評価などを進めたい。加えて本フレームワークの拡張として検討を進めている学習型の状況認識メカニズムやユーザの状況や動作環境の状態に基づいたコンポーネントの自動配置機能の実現などにも取り組む。

参考文献

- 1) Ohsuga, A., Nagai, Y., Irie, Y., Hattori, M. and Honiden, S.: PLANGENT: An approach to making mobile agents intelligent, *IEEE Internet Computing*, Vol.1, No.4, pp.50-57 (1997).
- 2) 川村隆浩, 田原康之, 長谷川哲夫, 大須賀昭彦, 本位田真一: Bee-gent: 移動型仲介エージェントによる既存システムの柔軟な活用を目的としたマルチエージェントフレームワーク, 信学論, Vol.J82-D-1, No.9, pp.1165-1179 (1999).
- 3) Hattori, M., Kase, N., Ohsuga, A. and Honiden, S.: Agent-based drivers' information assistance system, *New Generation Computing*, Vol.17, No.4, pp.359-367 (1999).
- 4) 池谷直紀, 加瀬直樹, 大須賀昭彦: エージェント技術を適用したヒューマンナビゲーションシステム, 情報処理学会研究報告, Vol.99, No.ITS-3, pp.97-104 (1999).
- 5) 長 健太, 林 久志, 大須賀明彦: picoPlangent: 携帯機器向け知的移動エージェント, エージェント合同シンポジウム (JAWS2002), pp.297-304, 信学会 (Nov. 2002).
- 6) 東芝ネットワーク家電「フェミニティ」.
<http://feminity.toshiba.co.jp/feminity/>
- 7) 板生知子, 松尾真人: 適応型ネットワークサービス環境 DANSE, 信学論, Vol.J82-B, No.5, pp.730-739 (1999).
- 8) 中西泰人, 辻 貴孝, 大山 実, 箱崎勝也: Context Aware Messaging Service: 位置情報とスケジュール情報を用いたコミュニケーションシステムの構築および運用実験, 情報処理学会論文誌, Vol.42, No.07, pp.1847-1857 (2001).
- 9) 寺西裕一, 種茂文之, 梅本佳宏, 寺中勝美: 移動体計算機環境における位置依存情報提供システムの設計と実現, 情報処理学会論文誌, Vol.39, No.4, pp.1077-1087 (1998).
- 10) 松下 温, 屋代智之: ITS の通信基盤の展望と課題, 信学論, Vol.J82-B, No.11, pp.1950-1957 (1999).
- 11) 西山 智, 中尾康二, 小花貞夫: ITS 通信におけるエージェントを用いたメッセージ委譲サービスの提案, 情報処理学会研究報告, Vol.2000, No.ITS-001 (2000).
- 12) インターネット自動車プロジェクト.
<http://www.sfc.wide.ad.jp/InternetCAR/>
- 13) 植原啓介, 湧川隆次, 佐藤雅明, 渡辺恭人, 砂原秀樹, 寺岡文男, 村井 純: 自動車情報化のためのインターネットを用いた通信システムの構築, 情報処理学会論文誌, Vol.42, No.2, pp.286-296 (2001).

(平成 15 年 4 月 1 日受付)

(平成 15 年 9 月 5 日採録)



服部 正典(正会員)

1972年生。1994年九州大学工学部情報工学科卒業。1996年同大学大学院工学研究科情報工学専攻修士課程修了。同年(株)東芝入社。現在、同社研究開発センター知識メディアラボラトリー所属。知的エージェント、パーソナルエージェント技術の研究開発、ならびにユビキタス環境における応用システムの研究に従事。



長 健太(正会員)

1972年生。1995年早稲田大学理工学部情報学科卒業。1997年同大学大学院理工学研究科修士課程修了。同年(株)東芝入社。現在、同社研究開発センター知識メディアラボラトリー所属。知的移動エージェントの研究開発に従事。日本ソフトウェア科学会会員。



大須賀昭彦(正会員)

1958年生。1981年上智大学理工学部数学科卒業。同年(株)東芝入社。1985年~1989年(財)新世代コンピュータ技術開発機構に出向。現在(株)東芝研究開発センター知識メディアラボラトリー主任研究員。工学博士(早稲田大学)。2002年より電気通信大学大学院客員助教授ならびに大阪大学大学院非常勤講師兼任。ソフトウェアのためのフォーマルメソッド、エージェント指向技術等の研究に従事。1986年度情報処理学会論文賞受賞。IEEE CS, 日本ソフトウェア科学会各会員。



本位田真一(正会員)

1958年生。1976年早稲田大学理工学部電気工学科卒業。1978年同大学大学院理工学研究科電気工学専攻修士課程修了。(株)東芝を経て2000年より文部科学省国立情報学研究所教授。現在に至る。2001年より東京大学大学院情報理工学系研究科教授を併任。工学博士(早稲田大学)。エージェント技術、オブジェクト指向技術、ソフトウェア工学の研究に従事。IEEE, ACM, 日本ソフトウェア科学会等各会員。



深澤 良彰(正会員)

1953年生。1976年早稲田大学理工学部電気工学科卒業。1983年同大学大学院博士課程修了。同年相模工業大学工学部情報工学科専任講師。1987年早稲田大学理工学部助教授。1992年同教授。工学博士。ソフトウェア工学、コンピュータアーキテクチャ等の研究に従事。電子情報通信学会, 日本ソフトウェア科学会, IEEE, ACM 各会員。