

タスクオントロジーを活用したエージェントシステム開発基盤の構築

齊藤 拓路[†] 澤本 潤^{††} 小泉 寿男[†] 辻 秀一^{†††}

[†]東京電機大学 理工学研究科 情報システム工学専攻

^{††}岩手県立大学 ソフトウェア情報学部

^{†††}東海大学 情報理工学部 情報メディア学科

自律性と協調性を持ち、自由競争や組織的動作のような社会的行動を可能とするソフトウェアソリューションとしてエージェント技術が注目されており、研究開発が進められている。しかし、現在のエージェントの研究の多くは特定の領域におけるアプリケーションの研究開発であるため、それら開発資源の一般化が進んでおらず、既存の研究のフレームワークを他の分野に適用しにくくなっている。また、エージェントシステムの実用化の例が少ないこともあり、エージェントの実用的な開発手法論が十分に検討されていない。このため、開発期間やコスト面での課題があり、エージェントシステムの普及のボトルネックの一つとなっている。本研究では、マルチエージェントシステムにも適用可能な問題解決型フレームワークを提案する。提案手法においては、開発を支援する技術として、タスクオントロジーを活用したエージェントモデルとタスクモデルのテンプレート作成と使用法を示す。そして、本手法に基づいたアプリケーションの開発を行い、適用性について評価する。

An Agent System Development Framework Using Task Ontology

Takuro Saito[†] Jun Sawamoto^{††} Hisao Koizumi[†] Hidekazu Tsuji^{†††}

[†]Department of Computers and Systems Engineering Graduate School of Science and Engineering
Tokyo Denki University

^{††}Faculty of Software and Information Science Iwate Prefectural University

^{†††}Department of Information Media Technology School of Information Technology and Electronics
Tokai University

The agent technology is expected as a software solution that enables a social action like a free competition and organizational approach by autonomy and cooperation, and its research and development are proceeding. However, the generalization of those development resources doesn't advance, because many of present researches of the agent are researches and developments of the application in a specific area. The framework of an existing research is not applied easily to other fields. Moreover, the examples of the agent system applied to practical services are few, and the way of practical agent system development has not been examined enough. Therefore, those facts cause the lengthening at the development period and the expansion of the cost. It is the bottleneck of the spread of the agent system. This research describes problem solving framework that can be applied to multi-agent system development. Then, we present usage of templates of the agent model and the task model as the development support technique. And we develop the application based on this proposed process, and evaluate its applicability.

1. はじめに

近年、コンピュータやデジタルネットワークが社会に浸透し、ユビキタス環境が整いつつある。それにともない、ユビキタス環境に対応した動的なサービスを提供することが求められている。サービス内容としては、ユーザの好みや位置、時間を含むコンテキストの変化に動的に適応したものである。このようなサービスを実現する仕組みとしてセマンティックWebやエージェント技術などの適用が考えられている。^{[1][2]}

しかし、現状でのエージェントシステム構築技術においては以下のような問題点を挙げるができる^{[3][4]}。

(a) エージェントシステム研究の一般性がなく、アプリケー

ション特化であり、技術の他ドメインへの転用が困難

- (b) 設計開発方法論の整備が不十分であり、応用ごとにエージェントモデルや処理系をアドホックに選択している
- (c) デザインパターンの利用は依然単純なタスクに限定されている
- (d) マルチエージェントシステム開発環境において自立的なエージェント間のインタラクションの複雑さをコーディネートすることは困難である

このような問題点に対しての先行研究として、リポジトリ型エージェントフレームワークを利用したインタラクティブな開発方法の研究や位置情報サービスの構築に関する研究が

ある。[45]

本稿では、問題点の中で特に構築方法論に係わる(b)および(c)に対する解決策を提案する。問題点(b)に対しては、基本となるマルチエージェントによる問題解決フレームワークと、それらを用いたシステム構築手法を示す。問題解決フレームワークは、各エージェント間の問題解決手順に関する処理と、各エージェントでの内部（ローカル）処理を分け、前者をエージェントレベル、後者をタスクレベルとして統合を行うことによりシステム構築を行うことを提案する。問題点(c)に対しては、デザインパターン（テンプレート）の利用強化を行うことを考える。本フレームワークにおいて、エージェントモデルに基づくテンプレートの利用を導入する。さらに、タスクオントロジー[46]に基づいてタスクレベルの問題解決知識の抽出と整理を行い、オントロジー階層に従ったタスクモデルのテンプレートを活用する。エージェントシステム開発のプロセスに対応しエージェントモデルとタスクモデルの統合を行うことにより構築を支援する方式を示す。本稿では、さらにこのシステム構築基盤をもとに、遠隔商品管理システム等の開発に適用し、その有効性の評価を行う。

2. エージェントシステム開発基盤

2.1 エージェントシステムの設計開発

エージェントシステムは、種々のエージェントによって構成され、要求された情報処理(解決すべき問題)を実行・解決するソフトウェアシステムであると捉えることができる。こうしたエージェントシステムの構築において、設計開発目標となるのは、様々な機能や用途を持ったエージェント(テンプレート単位)と、それらが構成する問題解決のためのエージェント組織である。

エージェントシステムの設計開発過程では、従来のソフトウェアシステムの場合と同様に、設計目標を規定する仕様、例えば、設計目標の全体的構造、エージェントとして実現される機能、エージェントタイプ・粒度・制御構造・知識、エージェント協調方式、エージェント間通信方式などが決定され、目標とする問題解決を遂行するエージェントシステムが実現される[7]。

エージェントシステム設計におけるアプローチとしては以下の2つがある。本提案手法においては、トップダウン的な問題分析・解決を行うが、再利用性向上のためテンプレート単位ではボトムアップ的な方法を用いている。

(1) トップダウン的な問題解決アプローチ

設計者が与えられた問題を解決するためのエージェントシステムを実現するために、設計を進める。

(2) ボトムアップ的な問題解決アプローチ

予め用意されたエージェント群の自律性に任せて、問題解決に適したエージェントを探し出し、エージェントシステムを実現する。

2.2 エージェントレベルとタスクレベルの分離

マルチエージェントによる問題解決フレームワークにおいて、各エージェント間の問題解決手順やプロトコルに関する処理と、各エージェントでの内部（ローカル）処理を分け、前者をエージェントレベル、後者をタスクレベルとして効率よく統合を行うことによりシステム構築を行う。

エージェントレベルとタスクレベルの基本的なプロセス分離の状況を図1に示す。ここで、エージェントレベルでは、マルチエージェント間の協調機能として、FIPA エージェント通信言語（ACL）による相互作用プロトコル[47]、例えば契約ネットワークプロトコル、ファシリテータプロトコル、黑板アーキテクチャによるプロトコル等を提供する。タスクレベルでは、データベースアクセス、検索、ローカルなデータ処理等のモジュールの集まりとなる。

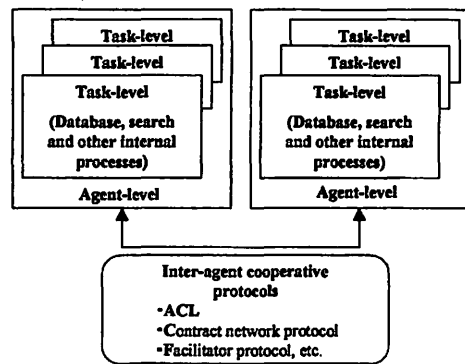


図1 エージェントレベルとタスクレベルの分離

2.3 問題解決基本モデル

本研究のシステム基盤におけるエージェントレベルとタスクレベルの分離方法として、図2に示す問題解決基本モデル[48]を用いる。エージェントシステム構築には、この問題解決基本モデルを考慮して作成する。

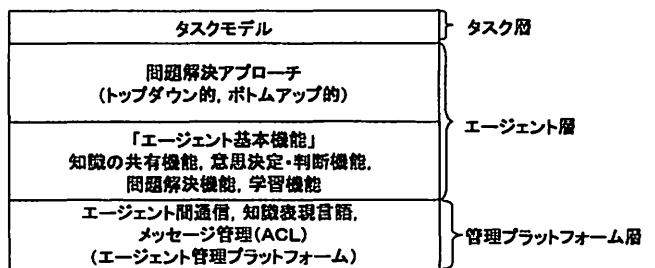


図2 問題解決基本モデル

問題解決基本モデルは、エージェント機能を階層構造に分割したモデルであり、それぞれ下から管理プラットフォーム層、エージェント層、タスク層の3階層構成とする。

(1) 管理プラットフォーム層

管理プラットフォーム層では、FIPAが提案する管理プラットフォーム機能であり、エージェント管理、ディレクトリ、エージェント通信路などの機能により構成されている。

(2) エージェント層

エージェント層のエージェント基本機能としては、共通的な

機能の抽出を行い、エージェント通信機能、知識の共有化機能、意思決定・判断機構を共通基盤ライブラリとして持つものとする。意思決定機構は、内部機能として、状況に関する情報知識、実行目標、プランニング機能などを持っている。エージェント層は共通的なエージェント基本機能の部分と問題解決モデルに対応した部分の2階層で構成される。

(3) タスク層

タスク層では、タスクオントロジーを考慮したタスクモデルに基づく問題解決構造を提案する。さらに、このタスク層ではエージェント技術の適用対象ごとにことなる機能が定義されていく。例えば情報検索を行うエージェントシステムであった場合、ユーザの性格などに合わせた情報を検索するための機能や、ユーザにとって不要な情報を削除するための機能といったものなどが定義されていく。

3. エージェントシステム開発方法

3.1 システム構成とアーキテクチャ

本稿では、図3のようなエージェントシステム構成例に基づく開発を支援する。



図3 エージェントシステムの構成

本提案手法で用いるエージェントとして、ユーザ側に常駐してユーザの代わりにサービス要求・取得を行うユーザエージェント、サービスエリアに常駐してエージェント間の中継役や商品情報の読みとりを行うファシリテータエージェント、サービスを提供するセンターに常駐してサービスを連携・検索することで適切なサービス内容を決定するブローカエージェントから成る。以下で各エージェントの詳細を述べる。

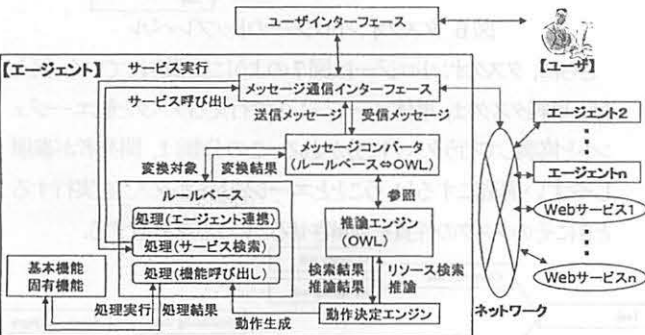


図4 エージェントアーキテクチャ

① ユーザエージェント(UA)

ユーザプロフィールの獲得、行動履歴によるユーザ要求の補完等の役割を担う。また、ユーザと共に移動する対象に付加された情報やユーザ情報を管理し、サービスを利用するユーザの特徴に合わせて、エージェントをカスタマイズできると同時に、システムにおけるユーザインターフェース機能を持つ。

② ファシリテータエージェント(FA)

サービス提供場所にエージェントを用いることで、設置する場所ごとにエージェントを実装する。このエージェントの性質を変えることで、設置場所ごとにシステム設計ができることから、ロケーションサービスを実現する。主な機能として、ユーザ要求を満足するサービス提供可能なエージェントの検索を行う。

③ ブローカエージェント(BA)

サービスセンタに常駐し、「履歴DB」などから取得した情報を基に個々のユーザに対応した処理や手続きを行い、サービス内容などを臨機応変に設定し、ユーザに提供する。具体的には、過去のサービス履歴や商品情報などから、様々なサービスを連携して、最適なサービスを提供する機能を持つ。

また、上記3種のエージェントのアーキテクチャとして、図4の内部構成を用意する。このアーキテクチャは、5つの機能モジュール、「通信モジュール」、「推論モジュール」、「制御モジュール」、「外部インタフェースモジュール」、「内部管理モジュール」によって構成されている。エージェントは、ルールベースに基づき自身の行動・処理を決定する。すなわち、エージェントが自分自身の知識を用いて問題解決処理を実行するための機構であり、また、このルールベースに新たな機能を定義することで、エージェントに新しい機能を追加したり、様々な動作や処理を行わせたりするエージェント構築が可能になる。

3.2 システム開発手順

本提案の開発方法の手順を図5に示し、開発方法論の枠組みを説明する。

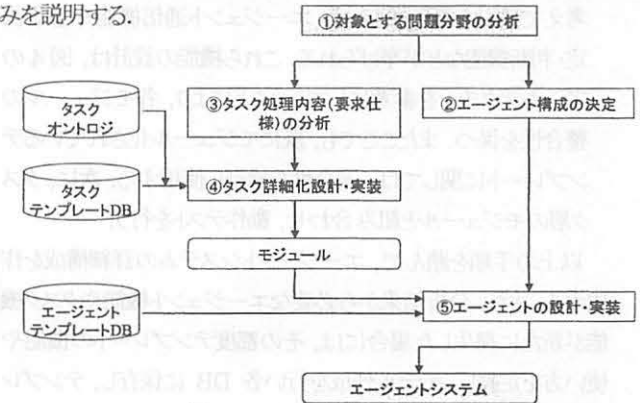


図5 システム開発手順

① 対象とする問題分野の分析

UML(ユースケース図,シーケンス図,クラス図)と問題解決基本モデルによる分析を行うことにより、エージェントシステム概念を把握し、さらに、システムを構成する各エージェント機能を把握する。この後、エージェント層の機能に関しては②、タスク層の機能に関しては③の工程へ移る。

② エージェント構成の決定

①の分析結果を基に、エージェントの構成を決定する。エージェントの構成とは、2.4章において前述したような、ユーザ

エージェント、ファシリテータエージェント、ブローカエージェントをどのように配置、連携させるかである。

③タスク処理内容の分析

タスクの設計モデルを得るために、問題解決基本モデルのタスク層に当たる機能に対して、さらに分析を行う。具体的には、①の分析では、タスク層とエージェント層へ振り分ける程度までしか詳細化していないため、ここで要求仕様の分析を行い、ユースケースを詳細にする。次の工程で、タスクオントロジーを参照するため、要求されたシステムの機能構造の概要を把握しておくことも必要である。

④タスク詳細化設計・実装

タスクオントロジーとタスクテンプレート DB を参照しながら、タスクの詳細設計と実装を行う。タスクオントロジーに記述されている問題解決の機能群と③での分析結果を照らし合わせ、再利用できるテンプレートやモジュールを取り出し、各機能が連携できるようにテンプレートはモジュールに具体化し、既にモジュール化されているものは調整を加える。この工程で、タスク層の機能を実現するモジュールについては完成させる。なお、タスク層のモジュールはエージェント層からの呼び出しで動作し、処理結果を返して終了する。

⑤エージェントの設計・実装

②と④の結果を基にエージェントテンプレートDBの構成を参照しながら、エージェントの設計・実装を行う。まず、タスクレベルに依存しない粒度のテンプレートを組み合わせる。具体的には、エージェントが必要とする基本的な機能を①や④の分析結果から洗い出し、それら機能を1つのモジュールとして考えて設計を行う。例えば、エージェント通信機能や意思決定・判断機能などが挙げられる。これら機能の設計は、図4のアーキテクチャを参考にして行うことにより、各モジュールの整合性を保つ。またここでも、既にモジュール化されているテンプレートに関しては、適合性を検証し使用する。次に、タスク層のモジュールと組み合わせ、動作テストを行う。

以上の手順を踏んで、エージェントシステムの詳細構成を作成する。また、分析結果から必要なエージェント機能やタスク機能が新たに発生した場合には、その都度テンプレートの機能や使い方を定義し、モデル作成を行い各 DB に保存し、テンプレートの情報を蓄える。

問題解決フレームワークは、エージェントの機能をアプリケーションレベルで依存しない粒度に小さく分けたエージェントテンプレートの蓄積・管理機構を備えることが大きな特徴である。エージェントシステムの設計開発の過程では、テンプレートDBに蓄積された資産の効果的な活用を支援・推進することにより、設計開発作業を効率化できる。

本稿で提案する設計開発方法論は、上述した問題解決フレームワークの特徴を活かしたトップダウン+ボトムアップ型設計開発を支援する方法論であり、既存のトップダウン型のエージェン

トシステム設計開発方法論と比較し、以下の特徴を持つ。

- (1) 再利用性:テンプレートDBに蓄積された開発済みエージェント機能などの資産の利用や再利用による設計開発効率の向上に繋がる
- (2) インタラクティブ性:設計者とエージェントの相互補完的は役割分担と協働による設計開発の柔軟性向上や円滑化を図る
- (3) システム拡張性:動的にエージェント機能を増やすことは、新たな処理や動作をエージェントに追加することになるので、エージェントシステムの拡張に繋がる

3.3 テンプレートデータベースの構成

3.3.1 タスクオントロジーとタスクテンプレート DB

タスクオントロジーは、実装レベルに関係なく問題解決のための知識（機能）を抽出し、体系化したものである。タスクオントロジーは、ロール、タスク行為、ステータス等の属性から構成される。

本手法では、タスクオントロジーに問題解決に必要な知識、つまり、どのようなテンプレート（機能）をどのような順で使用することによって処理を行うかを記述し、タスクテンプレート DB にテンプレートを格納しておくことにより、タスクに必要な知識と使用するテンプレートを同時に抽出できる。

図6にタスクオントロジーのトップレベルを示す。タスクオントロジーは、タスクとドメインから成り、さらにタスクは、動作の最小単位とする単純タスクと様々な単純タスクの組み合わせから成る複合タスクがある。ドメインは、事柄、実体、ロールから成り、それぞれでタスクにおいて作用するモノや対象と成るモノを定義する。

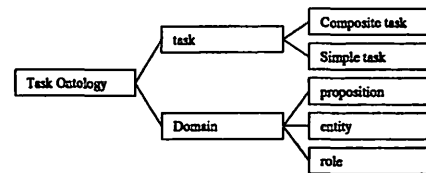


図6 タスクオントロジーのトップレベル

さらに、タスクオントロジーは図7のように拡張されていく。例えば、単純タスクは、単体エージェントで行えるタスクと他エージェントと協調して行うタスクに分かれる。この分類は、開発者が参照しやすい構造にすることとエージェントがタスクを実行するときにそのタスクの性質を理解させるという点を考慮する。

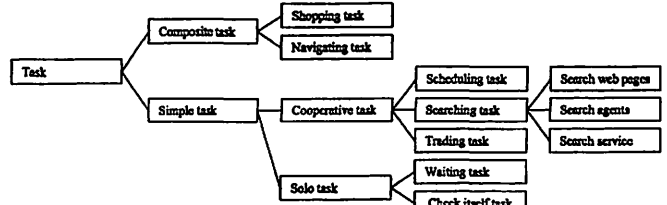


図7 タスクオントロジー下位レベルの例

上記のように様々なタスクを性質に着目して分類したが、それだけではテンプレートを参照するのに適してはいない。そのため、図8のスケジューリングタスクの例のように、その機能がどの

ような処理によって構成されているのか記述する必要がある。

以上のようにタスクオンロジーを構築することによって、タスクを構成する要素がわかるだけでなく、そのタスクが他のタスクとどのように関わっているのかが分かり、組み合わせる際の整合性が確認できる。

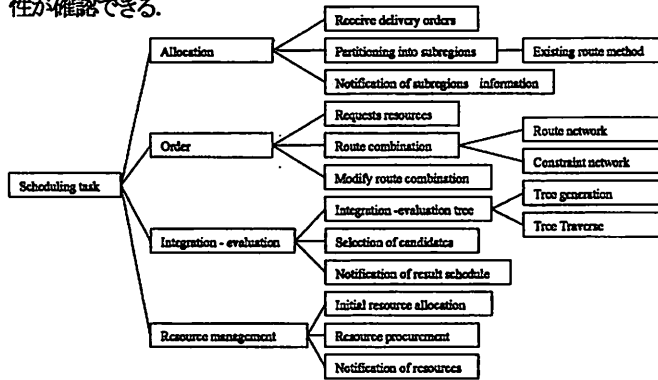


図8 スケジューリングタスクの構成例

3.3.2 エージェントテンプレートDB

エージェントモデルに基づくテンプレートについて説明する。エージェントテンプレートは、タスクに依存しない、モジュール構成の枠組み、推論エンジン、他のエージェントとの相互作用といった機能を提供する。テンプレートの設計段階において、再利用性を考慮する。エージェントシステム構築の容易性はエージェントモデルの設定に左右されると言える。典型的なエージェントテンプレート階層は、図9のように、本手法のエージェント構成とエージェントアーキテクチャに基づいている。テンプレートは、FIPA-OS^[10]とJADE^[11]を参照しながら構成し、テンプレートの要素はそれらのクラスライブラリから誘導されている。

図9のテンプレートDBに基づいて、エージェントを作成する場合、まずユーザ、ファシリテータ、ブローカなどのエージェントタイプを選択し、その後、そのエージェントタイプを構成するテンプレートを選択する。例えばユーザエージェントを作成する場合、図の黒枠で囲まれている「ユーザインタフェース」、「エージェント通信機能」、「意思決定・判断機能」を選択することで、基本的なユーザエージェントを作成することができる。

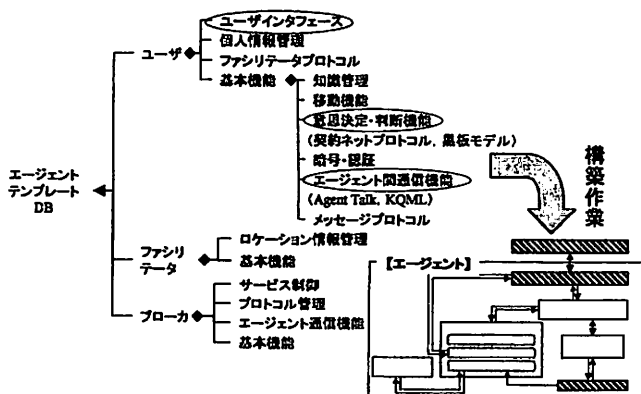


図9 エージェントテンプレートの構造

エージェントテンプレートは、エージェントの構造(クラス構造)

とJavaコードスケルトンにより構成される。表1に示す情報をテンプレートに付加することにより、テンプレート検索に利用する。

表1 エージェントテンプレートの付加情報

付加情報	説明
名前	テンプレートの名称
分類	コンテキストと機能に基づく分類情報
目的	テンプレートの利用目的, 作成意図
動機	テンプレートのエージェントが問題解決を行う設計問題とその解決シナリオ
適用可能性	テンプレート適用の事前条件
構造	上下階層のクラス構造
関連テンプレート	本テンプレートに関連して適用すべきテンプレートとその役割

テンプレートの選択・抽出範囲の目安としては、表1のテンプレート付加情報にある関連テンプレートを用いる。必要機能の関連テンプレートたどっていき、繰り返し(現在参照している関連テンプレートが指すテンプレートの関連テンプレートが現在のものである場合)が出てくるまで選択することによって、関連テンプレートをひとまとまりに抽出できる。次に、抽出したテンプレートの適用可能性を参照することで整合性のチェックが可能となる。ただし、この段階で整合性の問題が示されなくても、後の工程のテストで問題が出てくる可能性がある。その場合は、新たに適用可能性を追加する必要がある。

4. 遠隔商品管理システムの構築と適用評価

4.1 システム概要

動的なサービス連携を実現するシステム基盤の仕組みとしてセマンティックWebやWebサービス等の技術の適用を考える。具体的には、遠隔で商品を管理するシステムである。商品を管理するとは、保守・サービス提供や商品のトレーサビリティのことを示す。ここでは、車両の修理や故障についての保守システムを考える。遠隔商品管理システムを問題解決フレームワークに適用して構築する。

4.2 システム開発

図5の開発手順に従い、開発を行う。

(1)対象とする問題分野の分析

本システムの要求項目を大まかに挙げると、自動車の修理・故障に関する保守システムであること、メンテナンス工場に行かず車両情報を把握することを可能とすること、駐車場やガソリンスタンドの施設にシステムの中継基地を置くことが可能であることの3つである。これらの要求は、問題解決基本モデルの分類によるとすべてタスクレベルであるため、エージェント層に関してはこれら要求を達成できる構成で決定する。

(2)エージェント構成の決定

(1)の分析とエージェントシステムの構成図(図 3)を参照し、エージェント構成を決定する。点検のために、各車両に車両情報を取得するユーザエージェントを配置する。メンテナンス向上のためサービスセンタには、ユーザエージェントからの要求をもとにサービス(部品、修理)を提供するブローカとして、カーカンパニーエージェントを配置し、上記 2 つのエージェント中継するファシリテータとして、ステーションエージェントを駐車場の端末に配置する。図 10 にシステム構成を示す。

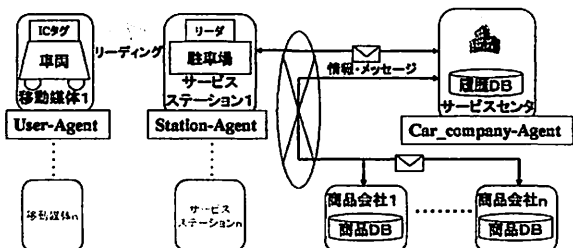


図 10 遠隔商品管理システムの構成

(3)タスク処理内容の分析

以下のようにタスク処理内容を詳細化する。

- ① 移動媒体(車両): 車両で起こる事象は受動的な部分であり、サービスステーションを認識しない限り動作することはないため、常にサービスステーションとの連携のために待機状態になっている。
- ② サービスステーション(駐車場): サービスステーションで起こる事象は、車両に取り付けられたICタグをサービスステーションのセンサが認識して、情報を読み取ることである。取得情報をもとにサービス要求を提案し、サービスステーションと連携してサービスを提供する。実際のシステムでは、「どこが故障している」や「〇〇部品の交換を提案する」などのメッセージを車両のインターフェースに投げかける。これらサービスステーションは、どこでも場所を選ばず、常駐設置や臨時設置など臨機応変にすることが可能である。
- ③ サービスセンタ: サービスセンタで起こる事象は、サービスステーションで商品情報とサービス要求を取得し、その情報をもとに適切なサービス検索と連携を行うことである。また、顧客の履歴情報から顧客に合わせた商品調達やサービス提供を行うものである。
- ④ 商品会社: 商品会社で起こる事象としては、サービスセンタからの商品問い合わせに対する返答を行う。この部分は、各商品会社によって内容が大きく異なることになる。

(4)タスク詳細化設計・実装

ここ以降は、現在構築中である。ここでは、(1)~(3)の分析結果を用いて、タスクオントロジーとタスクテンプレート DB を参照し、テンプレートを再利用、もしくは再利用しやすいような設計の調整を加え、実装する。例えば、メンテナンスのタスクオントロジーが既にあればそれを利用して設計できるが、無い場合でもサービス授受関係のある他のタスクオントロジーに変更を加えることで利用可能となる。

(5)エージェントの設計・実装

エージェントテンプレート DB を参照し、エージェントを設計・実装する。タスク層モジュールと組み合わせ、システムを完成させる。

以上のようにシステムを構築し、本提案手法の評価を行う。

5. まとめと今後の予定

エージェントシステム開発システムの問題点である、設計開発方法論の整備が不十分であることや、デザインパターンの利用が単純なタスクに限定されていることなどの解決を目的とした開発基盤として、エージェントシステムの設計・開発手法と、タスクオントロジーを用いたテンプレート運用方法について提案した。現在、この基盤をもとに遠隔商品管理システムの開発を行っており、本手法の有効性の評価をテンプレートの再活用の観点から行っていく。

今後の課題として、まず本手法の客観的な評価を行う上で適用事例を増やす必要がある点が挙げられる。次に、テンプレート DB の構造に関する評価も行う必要があるという点である。前者は本稿で挙げた適用事例の開発後に行っていく、後者は評価方法の検討後に本手法の評価に織り込んでいく。

<参考文献>

- [1]西山 智, 山田 満, 越塚 登, 坂村 健, "ユビキタスサービスのためのエージェントプラットフォームの提案" 情報処理学会研究報告, DPS-120, 2004
- [2]小倉 弘敬, 村上 佐枝子, 佐藤 宏之, 小島 富彦, 清水 昇, 細見 格, "セマンティック Web の応用システム", 情処会誌 43 巻 7 号, pp.743-750, 2004.
- [3]澤本 潤, 辻 秀一, 小泉 寿男, "マルチエージェントによるサービス支援システム構築の検討", DICOMO2006, pp.53-56, 2006.
- [4]打矢隆弘, 前村貴秀, 菅原研次, 木下哲男, "エージェントシステムのインタラクティブ開発環境", 電子情報通信学会, 2005
- [5]澤本 潤, 樋口 博, 岩橋 努, 臼井 澄夫, 辻 秀一, 小泉 寿男, "位置情報サービス(Location Based Services, LBS)向けマルチエージェントシステム構築基盤の提案", DICOMO2004, pp.41-44, 2004.
- [6]津口 理一郎著, "知の科学 オントロジー工学", オーム社, 2005
- [7]木下 哲男編著, "エージェントシステムの作り方", 電子情報通信学会, 2001
- [8]FIPA (Foundation for Intelligent Physical agents): <http://www.fipa.org/>.
- [9]川村 潤, 田那部 洋平, 澤本 潤, 小泉 寿男, 辻 秀一, "遠隔商品管理向けマルチエージェントシステムの構築", DICOMO2005, pp.405-408, 2005.
- [10]FIPA-OS (FIPA-Open Source): <http://sourceforge.net/projects/fipa-os/>
- [11]JADE (Java Agent DEvelopment Framework): <http://jade.cselt.it/>