

# マルチエージェントによるチーム航行支援

安藤 英幸 広野 康平 大和 裕幸

## 1 はじめに

ビジネスや製造のプロセスモデルを整理し、情報や業務の引継ぎの自動化を目指したワークフロー管理システムが注目されている。

ワークフロー管理システムの中心となるのは、プロセスモデルとそれを実行するワークフローエンジンである。一般にプロセスモデルは、コミュニケーション、作業の対象、ユーザーのアクティビティといった視点から有向ネットワークにより表現される。またワークフローエンジンは、プロセスのテンプレートからのインスタンスの作成、タスクの割り当て、実行とモニタリング、例外処理を行う。

これらの枠組みは、定型的にプロセスモデルを記述できる場合には十分である。しかし、とりまく環境の変化に対応してプロセスの更新が必要であったり、プロセスを外部環境との関連で表現するのが適している場合には、プロセスモデルの表現の工夫と、持続的にプランニング、スケジューリング、実行のモニタリングを行うワークフローエンジンが必要となってくる。

本研究では、船舶のブリッジ業務に動的なワークフロー管理システムを実装する事を目的としている。プ

リッジの業務は、周囲の船舶や自船の位置といった状況に応じて、実行する業務を適応させる必要がある。このために必要となるプロセス表現の整理とワークフローエンジンの実装を行う。

## 2 背景

船上業務の種類は、例えば見張り、船位確認といったレベルの業務をユニットとして約 600~700 ユニット程度であると言われる [1]。それぞれの業務は、Hutchins も述べているように構造の良くわかった問題で、あらかじめ作業手順を定型化する事ができる [2]。しかし、それら定型的な業務も、自船位置と航路の関係、他船の位置と速度、ワッチレベル (watch level) と呼ばれる危険度の指標、他の作業、作業の記録といった多様なコンテキストに応じて、行うべき業務、優先順位、処理に要する時間、繰り返し業務に関する時間間隔等が変更する。

これらの船上業務についての詳細は SMS(Safety Management System) マニュアルという印刷物の形で各運航会社毎にまとめられ、船上で利用されている。しかし、作業の状況への依存性が高い事からマニュアルが複雑化したり、マニュアルの見直しに必要な実作業の履歴の入手が現状では困難であることから、ワークフロー管理システムに SMS マニュアルの情報を移植し、作業の見逃し、対応の遅れの防止、さらには、作業履歴の分析による基のプロセスモデルの改善が期待される。

Team Navigation Support with Multi-Agents System  
Hideyuki Ando, 東京大学 新領域創成科学研究科, Graduate School of Frontier Sciences, University of Tokyo  
Kouhei Hirono, (株) 日本海洋科学, Japan Marine Science Inc.

Hiroyuki Yamato, 東京大学 新領域創成科学研究科, Graduate School of Frontier Sciences, University of Tokyo

### 3 エージェントとアーキテクチャ

#### 3.1 エージェントの構成

ブリッジ業務の動的なワークフロー管理システムは、プロセス、外的状況、作業者にに関する情報をネットワーク上で交換し、それらに基づいて適切な業務のプランニング、スケジューリング、実行モニタリングを持続的に行う必要がある。これらの実装にマルチエージェントを採用することとし、その実装のベースに、Jin らがワークフローによる協調設計支援のために開発した ActiveProcess 環境を利用する [3]。ActiveProcess の各エージェントは、ネームサーバーを介して KQML メッセージを交換する。コンテンツ言語としては、順序なしリストを表現できる独自のフォーマットを利用している。本研究では、航海士の支援を行うユーザーエージェント、船長につきチーム全体の業務プロセスを管理するマネージャーエージェント、船舶運航に関する情報をもつセンサーデータサーバーのラッパーとして働くリソースエージェントを配置した (図 1)。

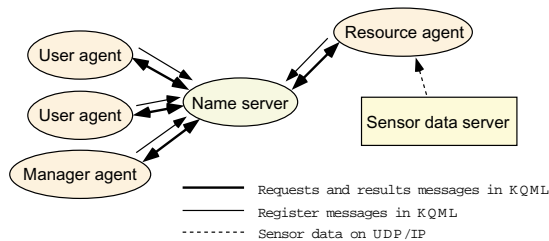


図 1 Multi agents configuration

#### 3.2 多層型アーキテクチャ

ActiveProcess では、自律的な振る舞い、推論、エージェント間の協調といった知的な機構に関しては、ドメインに合わせて開発を行う必要がある。本システムでは、エージェントに求められるタスクの多様性から、即応、熟考、協調の 3 つの機能を独立したレイヤーとして実現し、協調ロボットの実装を良く整理した InterRAP アーキテクチャの概念を基本的に採用することとした [4]。ただし、熟考、協調層の推

論部分へのエキスパートシステム JESS (Java Expert System Shell) [5] の利用、各レイヤーのコンポーネントの構成の変更といった改良を加えた。

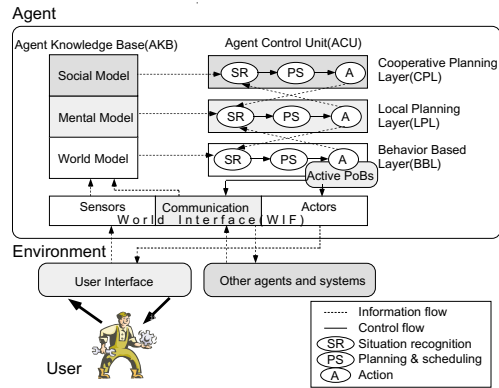


図 2 Agent architecture

エージェントの内部構成を図 2 に表す。全体は World Interface (WIF)、Agent Knowledge Base (AKB)、Agent Control Unit (ACU) によって構成される。WIF はエージェントと外部環境とのインターフェースで、ユーザーインターフェース、他のエージェントとの通信、センサーデータの取得に利用する。エージェントの持つ周囲の環境に関する知識、作業者の行うワークフロー、他エージェントに関する知識は World model, Mental model, Social model の 3 層に分けて AKB に置かれる。また、AKB はそれらの知識の登録、取り出しのためのインターフェースを持つ。ACU では AKB の知識を基に即応 (BBL)、熟考 (LPL)、協調 (CPL) のそれぞれの層が独立して状況の認識、プランニング、アクションを行う。ACU 内のレイヤー間の通信には KQML を使用する。即応層では、Pattern of Behavior (PoB) と呼ばれるエージェントの振る舞いのパターンを記述したモジュールの Active/Inactive の切り替えを行う。PoB には反射的な対応を行うための reactive PoB と他の層からの委託により Active な状態になる procedure PoB がある。

エージェントは全体として、AKB の知識をベースに即応、熟考、協調の各層がプランニング、スケジュー

リングを行い、PoB の実行リスト (Active PoBs) の切り替えを判断している事になる。

### 3.3 実装

システムは WindowsNT/2K 上で開発を行った。エージェントの開発には Java 言語を利用したが、他のアプリケーションとの連携を容易にするため Visual J++ を用いて COM(Component Object Model) の DLL(Dynamic Link Library) として実装した。また、ユーザーインターフェースの実装は Visual Basic を用いた。

## 4 プロセスマネージメント

### 4.1 プロセスモデル

Wilkins らは、現実問題へのプランニング技術の適用を考える場合、ドメイン知識の利用が不可欠である事を指摘している [6]。ドメイン知識を利用したプランニングの代表として HTN(Hierarchical Task Network) がある。HTN ではツリー構造によってプロセスを記述し、ゴールに合わせて上位概念を選択し、下位のツリーを展開し、最終的にプリミティブな操作で構成されるプランを得る。本研究でも、記述が可能であることから、階層構造というドメイン知識を用いたプロセス記述を採用し、特に Jin らが ActiveProcess で利用している Workflow, Activity, Work Element, Operation の 4 階層によるプロセスモデルに従う事とした。プロセスモデルのツリーの各階層の役割を、表 1 に示す。

また、動的な環境の変化に対応して適切なプランニングを行うために、全体のツリーのうちどのサブツリーを駆動するかという知識をエキスパートシステム JESS のルールという形で記述する。(図 3)。

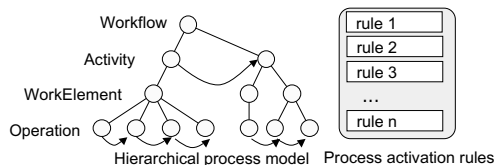


図 3 Process model

Workflow	チーム全体の作業プロセス
Activity	1 人の作業者に割り当てる作業の最小単位
Work Element	Operation の集合
Operation	作業の最小単位

表 1 Role of each layer

### 4.2 プランニング、スケジューリング

ユーザーの作業のプランニング、スケジューリングは、図 4 の手順で行われる。外部環境の変化にマッチするルールが発火し、実行すべきサブツリーが選択される。選択されたサブツリーは、順序の制約を満たすように展開され、一連のオペレーションにより構成されるプランとなる。これに、それぞれの作業時間を適応してスケジュールが構成される。この作業は継続的に繰り返される。また、現時点で、サブツリーの選択のために用意したルールの例を以下に示す。

- 周囲に船が接近してきた場合に起動
- ワッチレベルに応じた起動
- デフォルトで起動

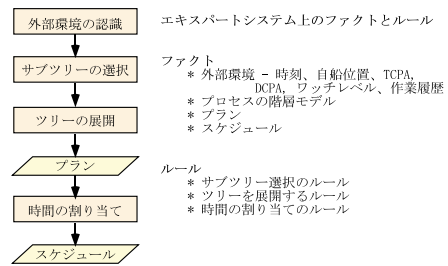


図 4 Planning and scheduling

### 4.3 実行とモニタリング

熟考層でスケジューリングが行われた作業は、ユーザーインターフェースを通して、ユーザーに対して提示され、その終了までをモニタリングされる。各作業のモニタリングは、図 5 に示す状態遷移規則に基づいて行われる。エージェントとユーザーインターフェー

スの間で、図 5 の矢印の上に表示されたフラグを交換し、状態を遷移する。作業のモニタリングは、即応層のレベルの作業モニタリング用 PoB が担当する。開始時には熟考層から Active にするメッセージが即応層に送られ、ユーザーインターフェースを介してユーザーに提示される。作業終了時には、ユーザーからの終了の通知を受けて、熟考層が業務の終了を認識し、先のモニタリング用 PoB を Inactive にするように即応層にメッセージを送る。

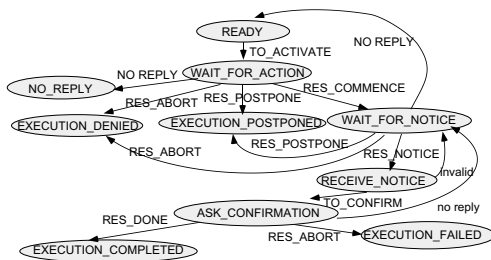


図 5 State transition for operation monitoring

## 5 検証

開発したエージェントの動作を確認するため、船舶のデスクトップ型の操船シミュレータ（仕様を表 2 に示す）に接続した。自船、他船に関する情報は、レーダー装置を代替するセンサーデータサーバからネットワーク上のリソースエージェントに対して送られる。シミュレーションの海域は、伊豆大島東方で、自船は洲崎沖変針点向け航行中とした。シミュレーションは、ワッチレベル 1（当直航海士 1 名体制）として開始した。当直航海士は船長から「見張り」、「船位測定」、「避航操船」の 3 つの Activity を与えられている。船長は船橋にいない状態であるが、「ワッチレベルの切り替え」の Activity が常時割り当てられている。この状態で、以下の点についてシステムの動作を確認した。

1. プロセスツリーの展開
2. 作業のモニタリング
3. 動的な環境でのサブツリーの選択

自船データ	1 秒間隔で生成
他船データ	3 秒間隔で生成（一般的なレーダーの周期が 3 秒であるため）。
自船の運動計算	舵角とエンジンテレグラフ値の入力より、応答遅れを考慮して船体運動計算する。
他船の運動計算	予め設定されたコースをトレースさせて運動を再現。変針動作は一定の角速度で回頭させている。
表示	電子海図上に自船位置と他船位置、それぞれの速度ベクトルを重ねて表示。ECDIS と同等

表 2 Specification of the simulator

### 5.1 プロセスツリーの展開

航海士エージェントは、「見張り」Activity 以下の階層構造を展開し、目視による監視、レーダーによる監視といった Operation を順次促し、次に「船位測定」Activity 以下の階層構造を展開し、下位の Operation の実行を提示し、当直航海士の作業遂行支援を行う事を確認した。

### 5.2 作業のモニタリング

航海士エージェントは、状態遷移規則に基づき、モニタリング用の PoB を通して、ユーザーインターフェースとの間でメッセージの送受信を行い、選択された Operation が最終的に、正常終了、異常終了、延期、応答なしのいずれかの状態になるまでモニタリングを行った。また、応答が許容時間内にない場合には、入力催促を行った。

### 5.3 動的な環境でのサブツリーの選択

「ワッチレベル切り替え」Activity が起動される条件を、この検証では“ TCPA（最接近までの時間）がプラスで、DCPA（最接近時の距離）が 1 マイル以下となる船舶が、同時に、10 マイル以内に 2 隻以上現れたとき ”とした。船長エージェントは、リソースエージェントから TCPA, DCPA 情報を定期的に

取得し、この条件を満たす状態に変化した直後に、船長に対してワッチレベルの切り替え Operation を提示し、船長はワッチレベルを変更した。航海士エージェントはワッチレベルが変更された事を受け、「避航操船」Activity の担当を外し、代わって船長エージェントが「避航操船」Activity を担当するツリーに加えた。

## 6 まとめ

船上のブリッジ作業支援というドメインにおける動的ワークフロー管理システムの設計を行うにあたり、プロセスモデルとして、階層構造で表現されるプロセスモデルと、駆動すべきサブツリーを選択する規則を記述したエキスパートシステムのルールを併用した。また、それらを実行するためのワークフローエンジンのアーキテクチャとして、マルチエージェントを採用し、特に、多様なタスクを並行してこなすために、InterRAP の多層アーキテクチャを基に、エキスパートシステムを推論エンジンとして使用するなどの拡張を行った。

これらにより実装した動的ワークフロー管理システムが、環境を認識し、条件に適應するサブツリーを選択し、それを展開し、作業者のスケジュールを作成し、その実行をモニタリングする事を、船舶の操船シミュレーターを利用した実験において確認した。

実用に向けては、まだ非常に多くの課題がある。現時点では、プロセスモデルの準備とルールの数が十分ではなく、この整備が必須となる。また、エージェントの認識できる情報は大きく制限され、例えば気象や海象といった情報を扱うことができない。これらを含めて、どのように人間とエージェントが協調的に作業を行うかを考えていく必要がある。

ワークフローエンジンの機能としては、作業スケジュールの作成に関して、iterative repair の導入等による、より無理のないスケジュールの更新手法の検討、作業者間の負荷分散、ユーザーによるスケジュール変更を許すためのインターフェースが必要と考えている。

## 7 謝辞

本研究の遂行にあたり、(株)日本海洋科学の東京大学名誉教授小山健夫先生から多数のご助言を頂きました。また、東京大学助教授の白山晋先生、増田宏先生、海上安全技術研究所の伊藤博子氏には研究の遂行にご協力いただきました。システム開発にあたり、南カリフォルニア大学の Prof. Yan Jin からは ActiveProcess のソースコードの提供と多数のご助言を頂きました。ここに感謝いたします。なお、本研究は平成 12 年度文部省科学研究費補助金による基礎基盤 (A)(2)-1230817 の補助を受けました。

## 参考文献

- [1] 日本造船協会第 238 研究部会. 新しいフリートサポートシステムの研究、平成 10 年度報告書. 社団法人 日本造船研究協会, 1999.
- [2] E. Hutchins. The technology of team navigation, 1990.
- [3] Yan Jin, Li Zhao, and Arun Raghunath. Activeprocess: A process-driven and agent-based approach to supporting collaborative engineering. In *Proc. of ASME Design Engineering Technical Conference, Las Vegas, Nevada*, 1999.
- [4] Jörg P. Müller. *The design of intelligent agents: a layered approach*, Vol. 1177. Springer-Verlag Inc., New York, NY, USA, 1996.
- [5] Jess. <http://herzberg.ca.sandia.gov/jess/>.
- [6] D. Wilkins and M. desJardins. A call for knowledge-based planning. *AI Magazine*, Vol. 22, pp. 99–115, 2001.