

エージェントフレームワーク DASH における スナップショット管理機構の構築

岸上 友樹† 打矢 隆弘‡

† 名古屋工業大学 工学部 情報工学科

〒 466-8555 愛知県名古屋市昭和区御器所町

内匠 逸‡ 木下 哲男§

‡ 名古屋工業大学 大学院 工学研究科

〒 466-8555 愛知県名古屋市昭和区御器所町

§ 東北大学工学部 電気通信研究所

〒 980-8577 宮城県仙台市青葉区片平 2-1-1

1 はじめに

インターネット等における広域分散処理の発展に伴い、ユーザによるサービス要求が多様化している。そこで、サービス要求に柔軟に対応できるシステムの設計手法としてエージェントシステム(以降 AS)が注目されている。しかし、AS はサービスに応じて動的に組織構成され、サービスの提供終了後に消滅するため、再利用時は初めから構成し直す必要がある。

そこで本研究では、エージェントの実際の動作基盤となるフレームワークの一つである DASH を対象として、AS 情報をデータベースによって保存・管理し、AS を保存時の状態からの再利用を可能にするスナップショット管理機構を提案する。

2 エージェントフレームワーク DASH

DASH[1] は、ユーザが作成したエージェントを保持するサーバであるリポジトリと、実際に AS が動作する環境であるワークスペースで構成されている。ユーザのサービス要求に応じて、リポジトリで AS を構成してワークスペースに生成し、ワークスペース上で AS が動作することでユーザにサービスの提供を行う。エージェントの動作は if-then 型のルールによって実行される。エージェントが予め持っている知識情報や、他のエージェントなどから送られてくるメッセージが、ルールの条件部と一致した際に、そのルールのアクション部に記述された命令が実行される。全てのエージェントは命令を実行している推論状態と、実行できるルールがなくなりメッセージ等を受け取るまで動作を停止している待ち状態のいずれかの状態となっている。

Construction of Snapshot Management System for Agent Framework DASH

†Yuki KISHIGAMI ‡Takahiro UCHIYA †Ichi TAKUMI §Tetsuo KINOSHITA

†School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan

‡Graduate School of Engineering, Nagoya Institute of Technology Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan

§Research Institute of Electrical Communication, Tohoku University, 2-1-1 Katahira, Aoba-ku, Sendai, Miyagi, 980-8577, Japan

3 関連研究

関連研究 [2] では AS の永続性を向上する手法として、帰還型フレームワークが提案されている。この手法ではワークスペースに構成された AS を、サービス終了後にリポジトリに移動(帰還)させる。サービスの再利用時はリポジトリに帰還した AS をワークスペースに生成することで、以前のサービスを継続利用できる。しかし、関連研究には以下のような問題がある。

(P1) リポジトリの再起動等によって AS 情報が消滅

(P2) 障害等で帰還以前に消滅した AS は復元不可能

4 スナップショット管理機構の提案

(P1), (P2) の解決手法としてスナップショット管理機構(図 1)を提案する。管理機構は下記 (F1), (F2) の機能を有する。

(F1) サービス提供終了時の AS の構成・状態を保存

AS はサービスが終了した後で消滅するため、AS を消滅させる前に AS を構成する各エージェントの持っている知識や状態をスナップショットとして取得する。取得したスナップショットはデータベースに保存して管理する。保存したエージェントを再利用する際は、データベースからスナップショットを読み出し、ワークスペースに保存時の状態でエージェントを復元する。

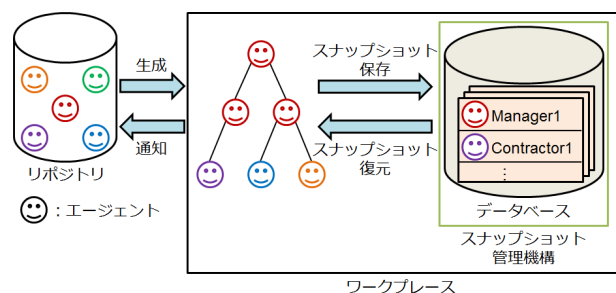


図 1: スナップショット管理機構

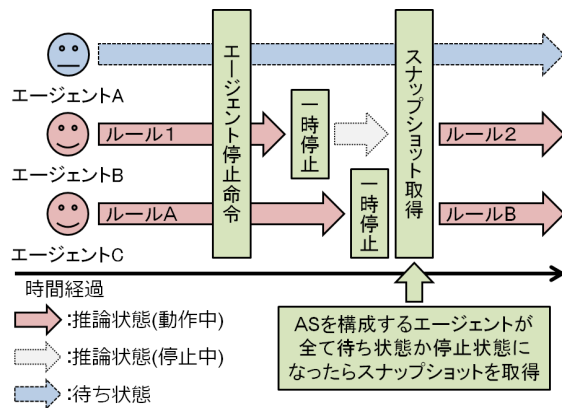


図 2: スナップショット取得のタイミング

(F2) サービス提供中の AS の構成・状態を保存

一定時間毎にスナップショットを取得してデータベースに保存することで、障害発生等によって突然 AS が消滅した時でも AS を復元できるようにする。

一般的に、エージェントが推論状態の時は、知識情報が逐一変更される。そのため、AS のいずれかのエージェントがルールを実行している間はスナップショットを取得できない。よって、全てのエージェントが待ち状態になってからスナップショットの取得を行う必要があるが、そのような状態になることは稀である。そこで、推論状態のエージェントは、実行中のルールを読み終えて次のルールが実行される前に動作を一時停止させて、スナップショットの取得を行う (図 2)。

5 実験と評価

実装した機構が正しく動作するかの検証と動作に要する時間の計測実験を行った。

5.1 動作検証

検証概要

ワークスペースで AS を動作させてその保存を行う。推論状態のエージェントと待ち状態のエージェントの両方を含んだ AS を用意し、最大 5 つのワークスペースにエージェントを分散して配置し、スナップショットの保存ができるかを検証する。また、保存したエージェントが保存時の状態で復元できるかを検証する。

結果・考察

多くのエージェントが保存時の状態で復元されることを確認した。しかし、データベースや Web サーバなど、エージェント以外のシステムと通信をしている最中のエージェントは、復元した後には通信が再開されないなど、一部のエージェントは復元後に正常に動作

しないことも確認した。これは、通信を開始する命令を含んだルールが実行済みとなっているため、復元時にそのルールが再び実行されないからと考えられる。

5.2 AS の保存・復元に要する時間の計測実験

実験概要

自動的なスナップショットの取得を行う際は、推論状態のエージェントを一時的に停止する必要がある。そのため、スナップショットの取得の度に AS の動作に遅延が発生する。そこで、エージェントの数に応じたスナップショット取得による AS の遅延時間の計測を行った。また、AS を復元するのに要する時間も計測した。計測はそれぞれ 10 回行い、その平均を求めた。

表 1: 各行程に要する時間

エージェント数	保存 (秒)	復元 (秒)
1	0.163	0.113
10	0.815	1.173
20	1.568	1.799
20(全て待ち状態)	1.015	1.715
20(全て推論状態)	1.867	1.754

結果・考察

実験の結果を表 1 に示す。AS 保存時は、保存を行うエージェントが待ち状態か推論状態かによって、遅延時間の差が大きいことが確認できた。よって、AS に占める推論状態のエージェントの割合が大きいほど遅延時間が長くなる。復元に要する時間は、推論状態か待ち状態かには因らずほぼ同じであり、復元を行うエージェントの数に最も大きく影響される。

6 まとめ

本研究では、サービス終了時及び提供中にデータベースへのスナップショットの保存を行い、スナップショットから AS の復元を行える機構を開発した。これにより、エージェントの継続利用において問題点があった従来の DASH に、より高い永続性を持たせた。今後は、保存可能なエージェントの種類の増加と保存に要する時間の短縮を目指し、提案機構の改善を進めてゆく。

参考文献

- [1] “DASH マニュアル”
<http://uchiya.web.nitech.ac.jp/idea/html/>
- [2] 伊藤 翔太 他, “ユビキタス環境用帰還型エージェントフレームワークの設計”, 情報科学技術フォーラム講演論文集 9(2), pp.369-372, 2010.