

構造化メッセージを用いたプロセス構造の動的な再編成*

2 N - 4

井上 創造 岩井原 瑞穂

九州大学 大学院システム情報科学研究科 情報工学専攻

1 はじめに

協調作業においては、参加者が必要に応じて議論のプロセスを生成し、そのプロセスを実行する。また、参加者は議論の内容が有効であるかを知る必要がある。しかし、議論の内容が有効かどうかは全体のプロセスの状態によって動的に決定または変更される。このようなプロセスには、非定型的な作業を不得手とする現在のワークフロー管理システムでは適応できない。本稿では、利用者間でやり取りされる電子メールなどの蓄積型メッセージを構造化してプロセス制御を行ない、その構造を実行時に動的に構造を再構成する手法を検討する。本手法は実行時に生成されるメッセージを中心としたモデルに基づくため、議論を始めとする、人間が行なう予測困難な作業の進行に対し柔軟なプロセス制御を行うことが可能になる。

2 プロセスとメッセージのモデル化

電子メールなど、利用者間のコミュニケーションに用いられる蓄積型通信による情報をメッセージと呼ぶ。また、意味的なまとまりのある作業をプロセスと呼ぶ。我々のモデルでは、プロセスをメッセージの集合により表現する。しかし、一般に、プロセスは作業を開始する前に定義されるのに対し、メッセージは作業の実行中または実行後に生成され、メッセージだけでは実行されるべきプロセスを記述できない。またメッセージは、プロセス間の交渉といった、従来の木構造のプロセスモデル [2] では表せない構造を持つ。このようなことから、メッセージによりプロセス構造を表す新たな枠組が必要となる。そこで我々は、より柔軟な構造を持つことができる、メッセージグループ（以下 m-group） [1] を用い、プロセス制御を行なう手法を検討する。

2.1 協調作業の例

図1では、協調作業において参加者の議論や発言の進行に応じてプロセスが発生する例を示している。図1(a)では承認された計画 (plan) に沿って作業 (task) が進行しているが、(b) 計画に問題 (problem) が生じる。そこで (c) problem プロセスを起動する。この中で、(d) 対策の提案 (proposal) がなされ proposal プロセスが起動されるが、(e) それに対し反対意見 (objection) とともに新たな提案 (proposal) がなされ、別の proposal プロセスが起動する。

2.2 メッセージ形式

メッセージにより協調作業の進行状況、扱われるデータを記録する。メッセージは本体（テキスト／音声／画像）や日付といった電子メールと同様の情報のほかに、作業の進行の状態の集合（メッセージ状態集合）、他のメッセージへのリンク (m リンク)、設計データ、ドキュメントなどへのリンク集合（ハンドル集合）の情報を持つ。

メッセージ状態は作業の状況を示すものであり、「提案」「意見」「賛成」「結論」など、構造化議論モデル (gIBIS) [3] で用いられているものがあげられる。

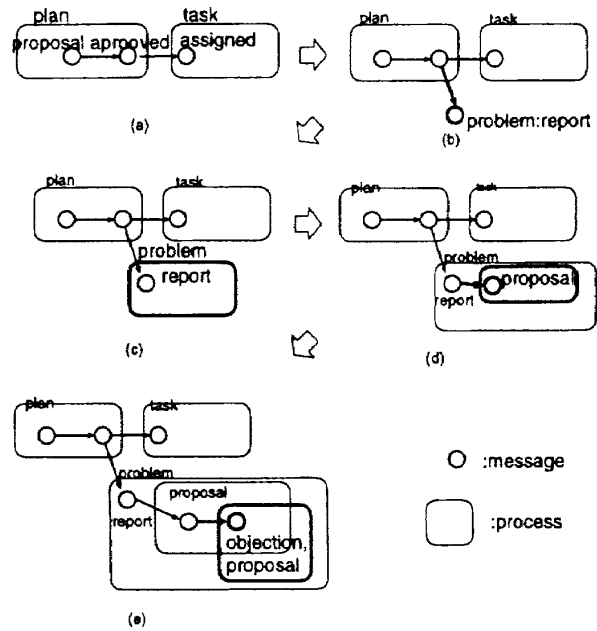


図1: 議論におけるプロセスの例

2.3 メッセージグループ

定義 1 m-group S は、 $MG(S, C)$ で宣言される。 C は空を許す集合 $\{S_1, S_2, \dots, S_n\}$ である。ここで $S_i (1 \leq i \leq n)$ はメッセージまたは既に宣言された m-group である。 S_i を S の子 group と呼び、子 group 集合を $CG(S) = C$ とする。 S を S_i の親 group と呼ぶ。宣言された m-group 全体の集合を MG で表す。

m-group は非巡回有向グラフ (以下 DAG) 構造を持つことができるものとする。

2.4 メッセージグループのトランザクション状態

各 m-group S はその実行状態を表すため、トランザクション状態集合 $\{active, aborted, committed\}$ の要素をトランザクション状態 $tState(S)$ として持つ。トランザクション状態は以下のイベントに応じて変化する。

- $t(S)$: (active) m-group S が生成された。
- $a(S)$: (aborted) m-group S が abort した。 m-group S の示す内容は無効であるとみなされる。このイベントはシステム、ユーザ、またはエージェントにより与えられる。
- $c(S)$: (committed) m-group S が commit した。 m-group S の示す内容は有効であるとみなされる。このイベントも $a(S)$ と同様に与えられる。

これらのイベントは、発生した時点でイベント列 E に加えられる。

2.5 トランザクション従属性

ここではトランザクション間の従属性を記述するために [4] で用いられている以下の記号 \prec, \rightarrow を用いる。

* "Dynamic Reconfiguration of Process Structures using Structured Messages," S. Inoue and M. Iwaihara, Kyushu University

定義 2

- $e_1 < e_2$: イベント e_1 とイベント e_2 の両方が起きるなら, e_1 が先に起こる. ただし, e_1 と e_2 の起きる可能性に関しては何も限定しない.
- $e_1 \rightarrow e_2$: イベント e_1 が起きるなら, イベント e_2 も起きる. これは e_1 と e_2 の起きる順序に関しては何も規定しない.

我々はこれまでに, $<$, \rightarrow および論理記号 \wedge , \vee , \Rightarrow を用い, m-group に対しトランザクション従属性を割り当て演算を定義し, その演算の性質に応じて m-group の子の数を制限する機能について検討している. また, トランザクション従属性間の競合に関して, MG の充足可能性を議論しており, MG の構造にある制限を与えると充足可能性を多項式時間で判定できることを示している [1].

2.6 メッセージグループによるプロセス制御

メッセージは実行時に生成され, 今後のプロセスの動きを表すことはできないため, メッセージが生成された際に属すべき m-group を参加者が決定するのを支援する機能が必要である. ここでは, 以下の方針で m-group に加える.

1. メッセージの持つ, メッセージ状態, m リンク, ハンドルといった情報と, m-group の持つトランザクション従属性をもとに, メッセージが子に加わる m-group の候補をシステムが計算する.
2. 参加者が, メッセージが子に加わる m-group を指定する.

3 メッセージグループの動的な再構成

この節では, m-group を実行時に動的に変更する手法を議論する. 作業時にプロセス構造の動的再編成が要求される状況として, 以下のようなものが考えられる.

- 議論の分岐により, 1つの m-group を複数の m-group に分割する.
- 議論の題目を一旦細分化するため, 子の m-group を複数生成する.
- 子の m-group の終了を待たずに親 group を終らせるため, 子の m-group を親から独立させる.
- 内容が似通った別個の m-group を結合し, 1つの m-group とする.
- ある m-group の議論の内容が別の m-group の議論の部分的な議論である場合, m-group を別の m-group の子に加える.

以下では, m-group の構造を変更する演算を定義する.

定義 3 MG をメッセージグループ集合とする.

- $add(S', S)$: $S, S' \in MG$ かつ $S \neq S'$ に対して, S を $MG(S, CG(S) \cup \{S'\})$ とする.
- $remove(S', S)$: $S, S' \in MG$ かつ $S' \in CG(S)$ に対して, S を $MG(S, CG(S) - \{S'\})$ とする.
- $create(S)$: $S \notin MG$ に対して, $MG(S, \emptyset)$ とし, MG を $MG \cup \{S\}$ とする.

- $delete(S)$: $S \in MG$ に対して, MG を $MG - \{S\}$ とする.

これらの演算を組み合わせることで用いることにより, MG を再構築することが可能となる.

メッセージグループ集合を不用意に変更していくと, 以下のような問題が生じる. (a) メッセージグループ集合 MG が DAG 構造を保持しなくなる, (b) トランザクション従属性を与える演算で限定される子 group の個数が守られない. (c) MG が充足不能となってしまう可能性がある. (a) に対しては, m-group の変更により有向閉路が生じないようにする必要がある. (b) に対しては, 子 group の個数が制限された m-group には演算 add および $remove$ を適用することを禁止する必要がある. (c) に対しては, 変更の際に充足可能性を判定する必要がある. 以下では, m-group に対する演算を行なうための条件を定義する.

定義 4 MG をメッセージグループ集合とし, 既に起こったイベント列を \mathcal{E} とする.

- $add(S', S)$: $DS(S')$ を $\{S'\} \cup CG(S') \cup CG(DS(S'))$ とした時, $S \notin DS(S')$, かつ子グループの数が制限されていない, かつ演算を適用した後の MG が \mathcal{E} について充足可能である.
- $remove(S', S)$: 子グループの数が制限されていない, かつ演算を適用した後の MG が \mathcal{E} について充足可能である.
- $delete(S)$: $CG(S) = \emptyset$, かつ演算を適用した後の MG が \mathcal{E} について充足可能である.

条件を満たさない場合は, (a) 演算を取りやめる, (b) メッセージグループ集合の構造を変更し条件を満たすようにする, (c) トランザクション従属性を変更し条件を満たすようにするといった措置を行なうことが考えられる.

4 おわりに

本稿では, メッセージを用いたプロセス制御について検討し, プロセス構造の変更を行う演算を検討した.

我々の協調作業支援モデルはメッセージの管理に基づく. m-group は現在の作業の状態と過去の履歴, 実行されるべきプロセスを表現する. m-group のトランザクションに関する制約を利用者が実行時に与えていくことにより, 動的に変更可能なプロセス管理を行なう.

参考文献

- [1] 井上創造, 岩井原 瑞穂, “高機能トランザクションモデルを用いたメッセージ管理,” 情報処理学会研究報告 DBS-116-82, 1998 年 7 月.
- [2] L. Liu and C. Pu, “Methodological Restructuring of Complex Workflow Activities,” *Proc. 14th Int. Conf. Data Engineering*, pp. 342-350, California, 1998.
- [3] J. Conklin and M. L. Begeman, “gIBIS: A Hypertext Tool for Exploratory Policy Discussion,” *ACM Trans. Office Information Systems*, Vol. 6, No. 4, pp. 303-331, Oct 1988.
- [4] P. Attie, M. P. Singh, A. Sheth, and M. Rusinkiewicz, “Specifying and Enforcing Intertask Dependencies,” *Proc. the 19th VLDB*, pp. 134-145, 1993.