

Title	イベントベースアーキテクチャにおける事象モニタリング機構
Author(s)	柏瀬, 秀行
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1231
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

修士論文

イベントベースアーキテクチャにおける 事象モニタリング機構

指導教官 落水浩一郎 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

柏瀬秀行

1999年2月15日

要旨

分散コンピューティング環境において、共同作業を円滑に進めていくうえで共同作業の状態を把握することは、極めて重要な事柄である。

本研究では、イベントベースアーキテクチャの利用によって、各作業者のツール操作イベントの観測を行ない、またコミュニケーション手段であるメールの送信、返信等の事象を観測する。そして、共同作業が共有の生産物の状態を中心とした状態遷移図に従って、作業が進められていると仮定し、観測された事象によりその状態を予測し遷移させる事象モニタリング機構の設計が可能であるかどうかを考察する。

本研究では、タイトルやメニュー、メールの中身を規定した構造化メールやタスクの構造を細かく明示的に規定し、作業者が作業の進行状況をシステムに知らせるといった構造的アプローチをできる限り用いないことにした。

目次

1	はじめに	1
1.1	研究の背景	1
1.2	研究の目的	2
1.3	関連研究	3
1.4	本論文の構成	4
2	共同作業モデル	5
2.1	共同作業の形態	5
2.2	状態遷移図	6
2.3	クラス図作成の共同作業の流れ	8
3	イベントベースアーキテクチャと事象モニタリング	11
3.1	イベントベースアーキテクチャ	11
3.2	イベントベースアーキテクチャの実現方法	12
3.3	事象モニタリング機構	13
3.3.1	観測される事象	13
4	事象を基にした状態遷移モデル	16
4.1	ツールの説明	16
4.2	記法の定義	18
4.2.1	状態を表わす記号	18
4.2.2	入力、出力イベントの定義	18
4.2.3	イベントの定義	19
4.3	状態遷移モデルの定義	19
5	実験	25

5.1	予備実験	25
5.1.1	予備実験の目的	25
5.1.2	予備実験の設定	25
5.1.3	予備実験の結果と考察	27
5.2	本実験	28
5.2.1	本実験の目的	28
5.2.2	本実験の設定	28
5.2.3	本実験の結果と考察	30
5.3	作成したツールのモニタリング機能の確認	30
5.3.1	「調整中」に意見交換のメールが入る場合	31
5.3.2	「安定状態」の扱い	31
5.3.3	「修正の可能性」から「作成中または変更中」への遷移	32
5.3.4	例外事象の処理	32
6	事象モニタリング機構の開発	34
6.1	システムの概要	34
6.2	メール解析機構	36
6.3	事象モニタリング機構	37
6.3.1	自動遷移機構	37
6.3.2	状態のチェック機構	37
6.3.3	状態決定アルゴリズムの概要	39
6.4	事象駆動の欠点の克服	40
6.4.1	事象モニタリング機構の性能	41
7	議論	42
7.1	実験の考察	42
7.2	状態遷移図モデルの修正案	43
7.3	モニタリング精度の向上の方針	44
7.3.1	描画イベントの観測	46
7.3.2	メールの内容の解析	46
8	おわりに	47
8.1	まとめ	47
8.2	今後の課題	47

A 予備実験のデータ	50
B 本実験のデータ	53

目 次

2.1	分散同期型	6
2.2	図の状態をもとにした状態遷移図	7
2.3	共同作業の流れ	10
3.1	イベントバスの概念図	12
3.2	分散複製型の共有アプリケーション	13
3.3	事象モニタリング機構の概要	14
4.1	共有ホワイトボード (1つの画面に3つ窓を開いた所)	17
4.2	メディエータ	17
4.3	表と状態遷移図の対応	24
5.1	予備実験で用いたダブルカウンタのクラス図	26
5.2	自作のツール上での再現	29
5.3	本実験で議論したオブジェクト図	29
6.1	システムの概要	35
6.2	最新メールの自動取得	36
6.3	メール解析機構	37
6.4	自動遷移機構	38
6.5	状態チェック機構	39
6.6	状態決定アルゴリズムの概要	40
7.1	改良した状態遷移図	45
7.2	「調整中」におけるメールの処理	45

表 目 次

4.1	作成中または変更中	20
4.2	修正の可能性	21
4.3	調整中	22
4.4	安定状態	23
7.1	修正した状態遷移モデル	43

第 1 章

はじめに

1.1 研究の背景

本研究室で開発中の「自在」環境のインフラストラクチャとしてのイベントベースアーキテクチャの利用を考え、検討を行なってきた。「自在」とは、ネットワークを介した共同作業の支援環境であり、ソフトウェア分散開発の支援を主な目的としているものである [1]。

イベントベースアーキテクチャの基本的な考え方は、非同期に発生したイベントがそれに関係のあることがあらかじめ宣言されているコンポーネントに通知されるという形式で、コンポーネント間のインタラクションを支援するものである。その特徴として、コンポーネント同士は直接イベントのやりとりを行わず、イベントバスを通してイベントのやりとりやデータの通知が行なわれることがあげられる [2]。

近年、イベントベースアーキテクチャは、分散オブジェクト技術 (CORBA, Java + Java RMI など) によってコンポーネントウェアを構成する手段として注目されている [3]。

分散オブジェクト技術により、コンポーネントのインタフェースを定め、コンポーネント同士を分散オブジェクト (イベントバス) を通して、イベントに興味があるもの同士を結びつける。接続されたコンポーネントは、Implicit Invocation により非同期にイベントの通知を行なう。分散オブジェクト技術は、コンポーネントの接続手段として成功を収めつつある。

一方、コンポーネントをバスを介して接続することの利点は、コンポーネント間のイベントが必ずイベントバスを通ることにより、イベントの流れを観測できるようになる点にある。また、そのことにより、ワークフローといった共同作業のための手順を組み込むことができる。

本研究では、分散複製型のコラボレーション・ツールである共有ホワイトボードを Java ネイティブな分散オブジェクト技術である Java RMI を用いて接続し、メディエータを介して Implicit Invocation によりイベントをマルチキャストすることにイベントベースアーキテクチャを実現した。

また、Java RMI を介して行なわれるイベントのやりとりをログとして記録し、コミュニケーション手段であるメールの送信、返信の事象を合わせて時間軸上のイベントとして観測する。

そして、共同作業の状態を把握させるための共有の生産物である図の状態を中心とした共同作業の進行を表す状態遷移図を仮定し、観測された事象により状態遷移させる事象モニタリング機構の設計が可能であるかどうかを議論する。

共同作業の進行状況をシステムが得る手段として、メールのタイトルやメールの構造を制約してその内容を得る構造化メールや、ワークフローにみられるように細かくその作業内容を規定して、作業の進行状況をユーザの役割として明示的にシステムに知らせる方法がある。しかしながら、そのような方法は、作業への負担が増すばかりが、しばしばその制約により、多くの例外事象が発生する事がある。

本研究では、構造化メールや細かく作業内容を規定して、作業の進行状況をシステムに知らせるという手段を用いずに、コラボレーション・ツールを操作することによって発生するイベントやメールの送信、返信などの事象を観測することにより共同作業の進行状況を状態遷移図上の状態にあてはめ、システムが自動的に遷移させられるような自動遷移機構の開発が可能であるかどうか考察する。

1.2 研究の目的

本研究の目的は、イベントベースアーキテクチャの利用により、分散共同作業における分散複製型のコラボレーション・ツールの操作イベントとコミュニケーション手段であるメールの送信、返信の事象を観測することにより、共同作業が、ある状態遷移図に従って進行していると仮定して、システム内部に保存された同一の状態遷移図をイベント列により遷移させるような方法を、簡単な実験を通して提案することである。

また、システムに作業の進行状況を明示的に知らせることをしない方針で事象のモニタリングを行なうことにした。そのため、メールに関しても、構造化メール的な手法として、システムに作業の進行状況を知らせる目的でタイトルの限定や仕事の内容を細かく規定しメニューを設け、明示的に仕事の進行状況をシステムに知らせるといった方法は行なわないようにした。

1.3 関連研究

本研究における関連研究として以下のものをあげる。

電子メールによるグループワークコーディネーション

電子メールを用いてグループウェアのコーディネーションを支援するシステムとしては、*TheCoordinatorTM*(Action Technology)が最も有名である。ユーザは個々のメッセージがどのメッセージに対するどのタイプの応答であるかを明示的にメニュー選択により宣言し、システムは構造化電子メール機能を用いて会話の(状態遷移)モデルに基づき「会話」の流れを追跡する。ユーザに懸案事項を思い出させることによりグループワークのコーディネーションを支援する。*The Coordinator*がユーザに課す構造的制約が強すぎるという批判と、その構造ゆえに有効であるとの反応に分かれた [5]。

ワークフローによるコラボレイティブ・ワークの支援

共同作業における、個々の作業者の仕事(処理)を「アクティビティ」と呼び、担当者が実行する具体的な作業を複数のアクティビティをまとめて「ワーク・アイテム」と呼ぶ。処理すべきワーク・アイテムを担当者ごとに一覧にしたものを「ワーク・リスト」と呼ぶ。アクティビティが終了するごとにワークフロー・エンジンはプロセス定義データの内容に従って、次に処理すべきアクティビティを担当者のワーク・リストに、実行すべきワーク・アイテムとして追加する。

担当者が作業を終了させたら、ワーク・リストからワーク・アイテムを削除する。業務の担当者や監督がワークフローの進行状況を知りたいときは、「モニタリング・ツール」と呼ぶソフトウェア・モジュールを使う。具体的な機能と使用法は製品によって異なる [6]。

分散オブジェクト技術におけるワークフロー標準化を行なうための提案も始まった [7]。

以上のように、モデルに基づくグループウェアでは、構造化電子メールやハイパーテキストなどをそのベースとして利用することにより、多様なグループワークの構造をツールの中に持ち込んだ。そしてコンピュータは構造化されたメッセージやリンク情報に対して処理(フィルタリングや検索)をおこなうことにより、ユーザを支援した。しかし構造化

することによりツールはある仕事に特化され、その汎用性は低下するというジレンマを持つ [8]。

本研究では、以上のメールやツール操作による共同作業支援の仕組みを参考にしつつ、構造化メールの使用やワーク・アイテムの手動による削除など、作業者が明示的に作業の完了をシステムに知らせる目的で行なう付加作業をできるかぎり排除する方針とする。

その理由は、作業への明示的な付加作業に対する負担の軽減を行なうと共に、必ずしも明示的に作業の進行を知らせることが共同作業を行なう上で多くの例外事象の発生によりうまく機能しない場合があるからである。

ここで述べる、明示的とは、作業の進行状況をユーザが直接リストから選ぶ等の方法で知らせることであり、非明示的とは、ユーザのツール操作、メールの送信、返信等の作業の事象を観測することにより間接的にその作業の進行状況を推測することである。

本研究では、後者の非明示的な方法により共同作業の状態をモニタリングすることにした。

1.4 本論文の構成

2章では、本研究において仮定して用いた、共有生産物の状態を中心として共同作業の進行状態を表した状態遷移図について、本研究で用いた共同作業のモデルについて例をあげながら説明する。

3章では、イベントベースアーキテクチャを利用した分散複製型のコラボレーション・ツールについて説明し、ツール操作イベントとメール送信、返信の事象による事象モニタリング機構について述べる。

4章では、モニタリングした事象をもとに状態遷移の方法を表を用いながら説明する。

5章では、実際に共有ホワイトボードとメールを用いた会議を行ない、モニタリングツールを利用しその有効性を確かめる。

6章では、5章で得られたイベントログをもとに開発した事象モニタリング機構のプロトタイプについて説明する。

7章では、議論として、5章の実験もとにして、実際に観測された事象から、状態を表すイベント列および状態遷移を起こすイベントについてのモデルを提案する。

最後に8章で本論文のまとめと今後の課題を述べる。

第 2 章

共同作業モデル

本章では、本研究において仮定して用いた、共有の生産物の状態を中心として共同作業の進行状態を表わした状態遷移図について、共同作業のモデルについて例をあげながら作業の流れを順に追って説明する。

2.1 共同作業の形態

本研究においてとりあげる共同作業の形態は、ネットワークを介して行なわれる分散共同作業である。特に、共同でソフトウェア開発を行なうといったコラボレイティブ・ワーク [14] を考えている。

具体的には、オブジェクト指向モデリングにおけるクラス図などの図の共同作成をリーダーを中心として行なうことを考える。

リーダーは会議の議事進行役としてメールを使って会議の進行を行なったり、議題となっている図に対する責任者として他の作業者の意見を取りまとめて図の変更を行なうものとする。

但し、最終的には、他の作業者との確認を行なってからひとまずの終了状態となる「安定状態」とする。

自分の描いた図や他の人の描いた図が、対面でホワイトボードを用いて描いているのと同じように反映される共有ホワイトボードのツールと一般的な電子メールを用いて作業を行なう。

作業形態としては、作業者同士は異なった部屋、決められた同じ時間に共同作業を行ない、相手から送信されてきたメールに対して、明らかに返答の必要がない場合を除いて、即座に返事のメールを送信することとした分散同期型で行なうものとする (図 2.1)。

時間 空間	同期 (リアルタイム型)	非同期 (蓄積型)
同室 対面型	共有ホワイトボード + テレコミュニケーション (同室)	
遠隔 分散型 (同室でも 利用可)	共有ホワイトボード + メール(即答型) (遠隔)	共有ホワイトボード + メール(蓄積型) (遠隔)

図 2.1: 分散同期型

2.2 状態遷移図

共同作業における作業状態の遷移を共有生産物であるクラス図の状態をもとに状態遷移図モデルとして表わしたのが図 2.2である [1]。

共同で作成が行なわれているクラス図の作成の状態を、「作成中または変更中」、「修正の可能性」、「調整中」、「安定状態」と4つの状態に分けて定義し、クラス図を作成するといった共有の生産物(ここではクラス図にあたる)を中心とした共同作業の流れを表わしている。

以下に4つの作業状態をリーダーを中心としたクラス図の共同作成にあてはめて説明する。

注意事項として、共有ホワイトボードで扱っている図全体をひとつの状態としてとらえていて、個々の図の状態に対応するものではない。

また、誰が描いた図かわかるように作業者ごとに色を変えて作業しているものとする

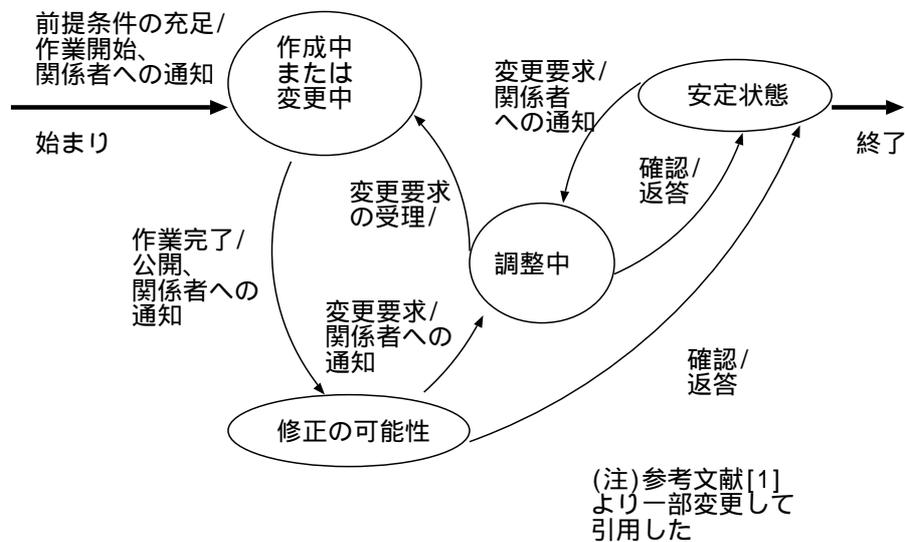


図 2.2: 図の状態をもとにした状態遷移図

- 「作成中または変更中」の状態

リーダーが共同作業を行なう他の作業者と議論するための原案となる図のドラフトを作成している状態を「作成中」の状態とした。また、議論した後で、再度、クラス図のドラフトを作り直している状態を「変更中」の状態とした。

- 「修正の可能性」の状態

リーダーが提案したクラス図について、他の作業者がクラス図の修正の可能性について考え、図の変更や気になる箇所について図に色を変えて書き込みを行なっている状態を「修正の可能性」の状態とした。

- 「調整中」の状態

「修正の可能性」の状態をへて、リーダーも含め参加者全員でクラス図の変更の書き込みや意見交換をメールを通して行なっている状態を「調整中」の状態とした。

- 「安定状態」の状態

リーダーからこれ以上の変更が現時点でないことの確認をメールによって促された後、メールによる返答が行なわれ、ひとまず図の作成が合意を得て、確認がとれた状態を、「安定状態」にある状態とした。

2.3 クラス図作成の共同作業の流れ

一例として、4つの作業状態を中心に具体的にどのように共同でクラス図の作成が進行していくのかについて説明する。文頭の番号は、図 2.3の番号に対応している。

1. まず、議論するクラス図の問題が決まったら、全員が共有ホワイトボードのツールを立ち上げる。
2. 次に、リーダーは議論の原案となるクラス図のドラフトの作成を共有ホワイトボードを用いて行なう。この状態を「作成中」の状態とする。ドラフトを作成している間は、他の作業者にも見える。(リーダーは、例えば赤色を使って描画する。)
3. リーダーによるクラス図のドラフトが完成したならば、他の作業者にドラフトの完成をメールで全員に通知する。つまり、リーダーが他の作業者に図の「公開」を伝えるのメールを送る。
4. リーダー以外の作業者に修正の可能性がある箇所について図に丸を付けるなどして記入してもらおう。この状態を「修正の可能性」の状態とする。(他の作業者はそれぞれ色を変えて描画する。例えば、作業者 B は青、作業者 C は黄色など。以下、作業者ごとに色分けして描画する。)
5. 図の変更要求を伝えるメールが他の作業者からリーダーに送られれば、「調整中」の状態へ移る。
6. クラス図の描き換え及びメールの交換を行ないながら、クラス図の変更案について議論する。この状態を「調整中」の状態とする。
7. 「調整中」をへて、リーダーにドラフトの描き直しのメールが送られ、図の描き直しの変更の要求を受理するメールの返事が送られれば「作成中または変更中」の状態へ移ってドラフトの変更を行なう。
8. リーダーから参加者全員にメールで今の時点でこれ以上の図の変更がないことが確認されれば、「安定状態」へ移る。

9. 参加者全員の図に対する変更が現時点でこれ以上ないことの合意の確認メールを得て、再び図の変更要求のメールが出されるまで状態を、「安定状態」としてクラス図の変更を行わない状態とする。
10. 参加者全員がツールを終了させたならば、終了とする。
11. また、「修正の可能性」の状態をへて図の変更なしの確認がメールで行なわれれば、いきなり「安定状態」へ遷移するものとする。
12. 再び図の変更要求が参加者の誰かからメールで出されたならば「安定状態」にあったクラス図を「調整中」の状態として参加者全員でクラス図の変更を行なう。

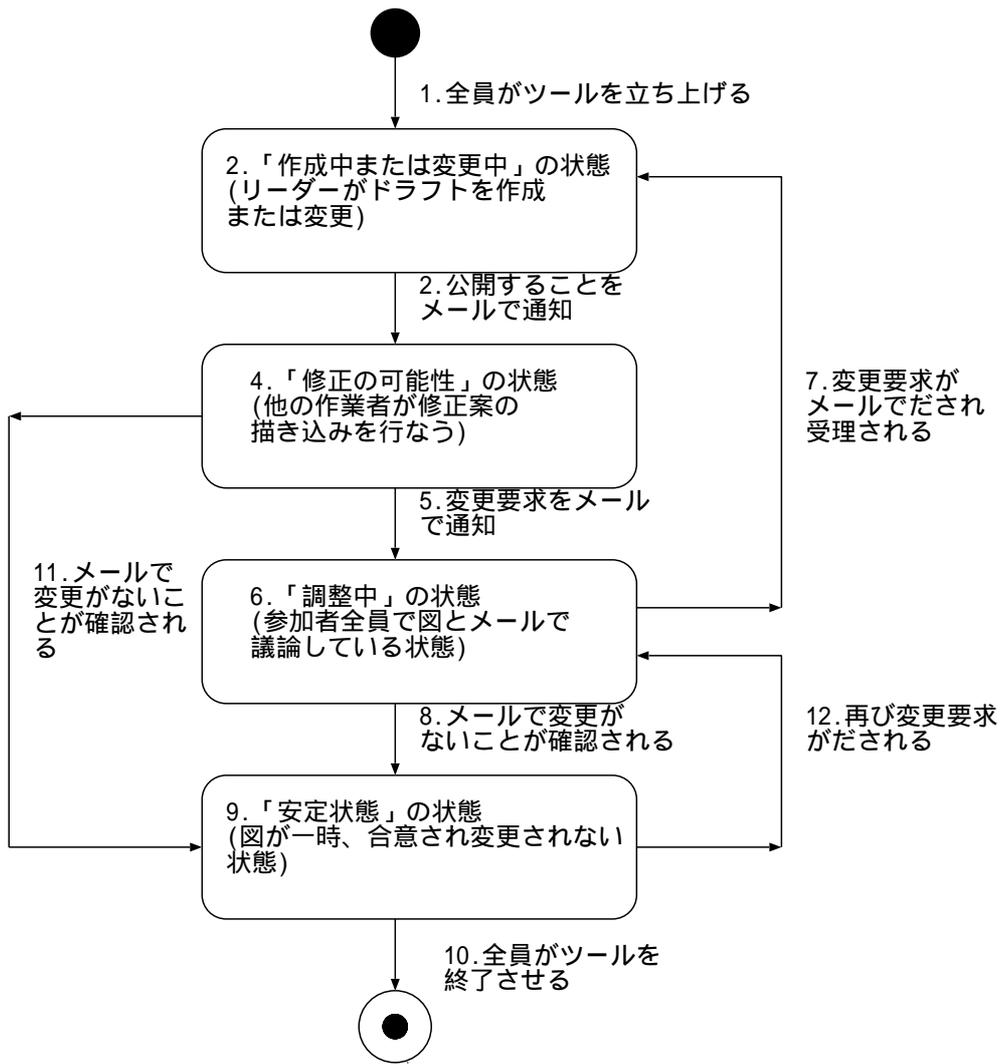


図 2.3: 共同作業の流れ

第 3 章

イベントベースアーキテクチャと事象モニタリング

本章では、イベントベースアーキテクチャを用いたコラボレーション・ツールである共有ホワイトボードより発生するイベント及びメールの送信等の事象を用いて行なう事象モニタリングについて述べる。

3.1 イベントベースアーキテクチャ

イベントベースアーキテクチャの基本的な考え方は、非同期に発生したイベントがそれに関係のあることが宣言されているコンポーネントに通知されるという形式で、コンポーネント間のインタラクションを支援するものである [3]。その特徴として、コンポーネント同士はお互いに直接イベントのやり取りを行わず、イベントバスを通してイベントのやり取りやデータの通知が行われることがあげられる。そのため、極めて柔軟で、効果的なイベントのやりとりが行なえ、コンポーネントのバスへの接続が容易になる。

イベントベースアーキテクチャの特徴

1. インプリシット・インボケーション

コンポーネント同士は、直接イベントのやり取りを行わない (バスを介して行なう)。

2. マルチキャスト

イベントは、あらかじめイベントに興味があることが宣言されているコンポーネントに対して送られる。

3.2 イベントベースアーキテクチャの実現方法

近年、イベントベースアーキテクチャは、分散オブジェクト技術 (CORBA, Java + Java RMI など) によってコンポーネントウェアを構成する手段として注目されている。

分散オブジェクト技術により、コンポーネントのインタフェースを定め、コンポーネント同士をイベントバス (図 3.1) を通して、イベントに興味があるもの同士を結びつける [11] [12]。接続されたコンポーネントは、インプリシット・インボケーションにより非同期にイベントの通知を行なう。分散オブジェクト技術は、コンポーネントの接続手段として成功を収めつつある。

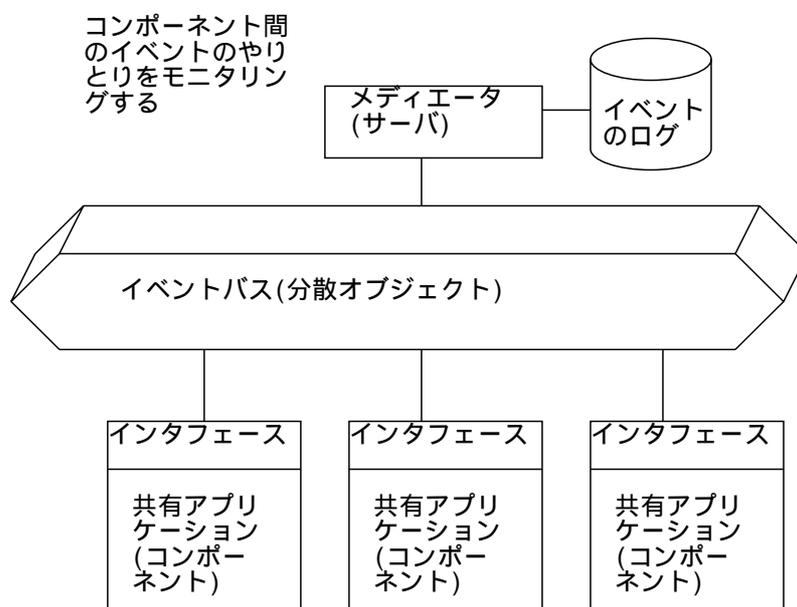


図 3.1: イベントバスの概念図

一方、コンポーネントをバスを介して接続することの利点は、コンポーネント間のイベントが必ずイベントバスを通ることにより、イベントの流れを観測できる点にある。

イベントバスのモニタリング機構を開発することにより分散複製型の共有アプリケーション (図 3.3) 間のイベントの様子を観測し、作業者の状態を知るのに役立つことができるはずである。

分散複製型の共有アプリケーションの特徴は、各ユーザからのイベントのみがアプリケーション間でやりとりされることにありネットワークトラフィックが小さい。一方、基本的に非同期にイベントの通知が行なわれるため、同期制御が複雑になる欠点もある。(本

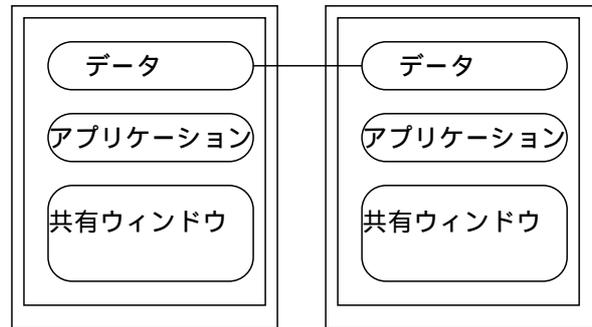


図 3.2: 分散複製型の共有アプリケーション

研究において、製作した共有ホワイトボードも作業員間のホワイトボードに同じ図が反映されるのに時間差が生じる欠点が生じた。)

3.3 事象モニタリング機構

イベントベースアーキテクチャによるコンポーネントの接続を行なう。

各ユーザのイベントバスを通して行なわれるツール操作の事象を作業状況の把握に必要な作業履歴を記録する。本研究では、分散複製型のコラボレーション・ツールとして共有ホワイトボードの実装を Java アプレットと Java RMI を用いて行なった。

本アプリケーションの特徴は、各作業員が所有するローカルで完結したホワイトボードへの描き込みイベント (描画イベント) が、あらかじめそのイベントに興味があることが宣言されている他の作業員が保有する共有ホワイトボードへマルチキャストで送られ、同じ図が反映される点である。

3.3.1 観測される事象

共有ホワイトボードの描画イベントに属して観測される事象は以下の通りである。

作業時間 ; 日付、時:分:秒 単位でいつ作業を行なったかがわかる。

作業員のホスト名 ; 作業員のホスト名によって誰が作業を行なったかがわかる。

作業イベント ; 呼び出したメソッド名によって作業イベントがわかる。

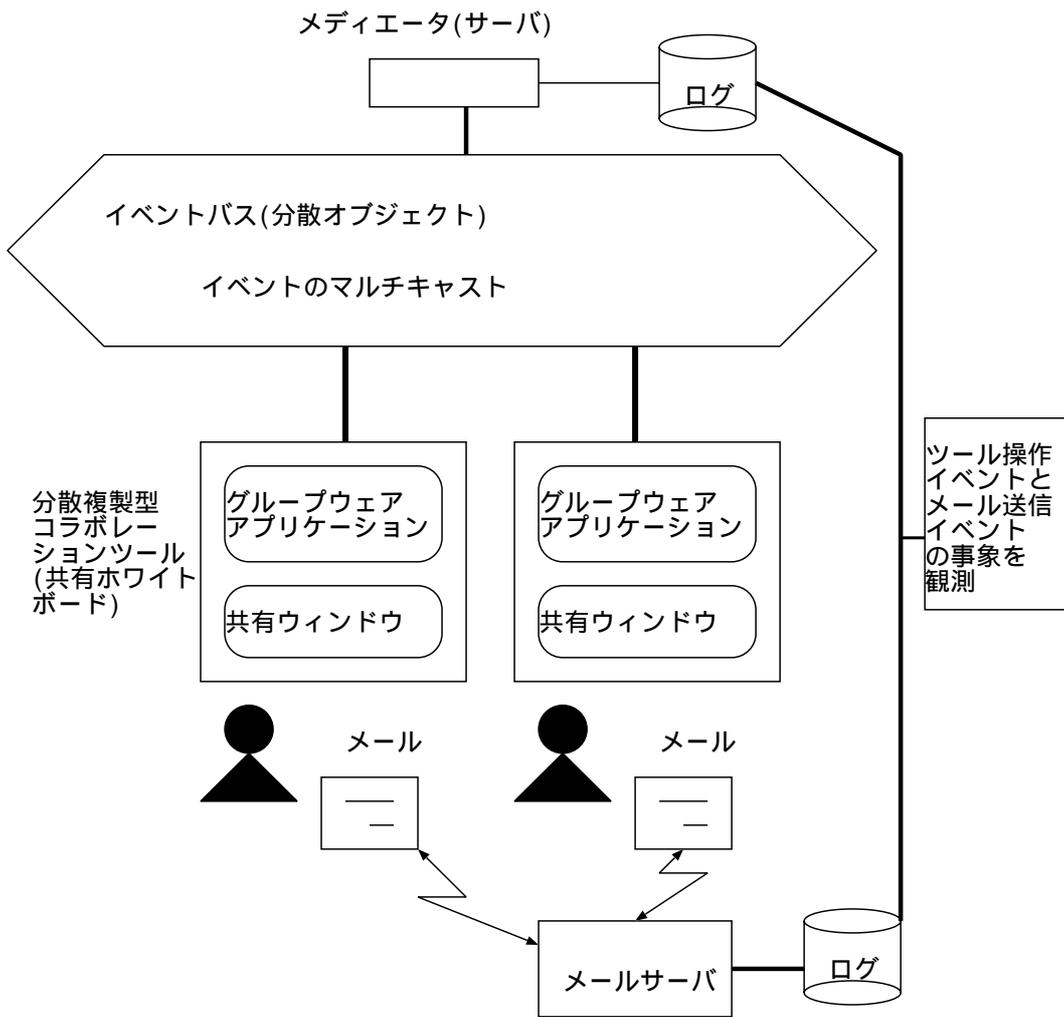


図 3.3: 事象モニタリング機構の概要

メールの送信、返信イベントに属して観測される事象は以下の通りである。

送信者、受け取り人；誰が誰にメールを送ったかがわかる。

送信時間；メールを送信した時間。即時的な使い方をしているのでメールの受け取り時間とほぼ一致するものとする。

ツールの操作イベント (描画イベント) およびメールの送信、返信の事象はともに事象の発生時間を属性として持っているので、時間軸上にイベント列を並べ、イベント列により状態や状態遷移の入力、出力となるイベントを推測することができると思われる。

第 4 章

事象を基にした状態遷移モデル

本章では、ツール操作のイベント、メールの送信、返信等の事象を観測することにより、この章より以前の章で提示した図の状態を中心とした共同作業の状態を表す状態遷移図を遷移させる方法についてモデル化する。

4.1 ツールの説明

イベントベースアーキテクチャを用いた分散複製型のコラボレーション・ツールとして共有ホワイトボード (図 4.1) を Java 言語により製作し、Java ネイティブな分散オブジェクトである Java RMI を用いて接続した。

複数の作業者がネットワークを介して共同で図の作成が行なえるようになっている。つまり、各ホワイトボードユーザからのマウスを用いた描画イベントがメディエータを介して他の作業者のホワイトボードへマルチキャストされ他の作業者のホワイトボードに図が反映される。

今回は、イベントベースアーキテクチャを実現する手段として、Java + Java RMI を利用した。描画キャンパスとマウスによるイベント、及びデータは各クライアントのコンポーネントが保持し、Java RMI を介して、メディエータ (図 4.2) に登録されたクライアントに対し、イベントが Implicit Invocation により通知される。

クライアント (共有ホワイトボード) は、描画イベントの発生をメディエータに通知する。メディエータは、受け取った描画イベントをすべての登録クライアントにブロードキャストする。

描画イベントのログは、このメディエータから記録した。

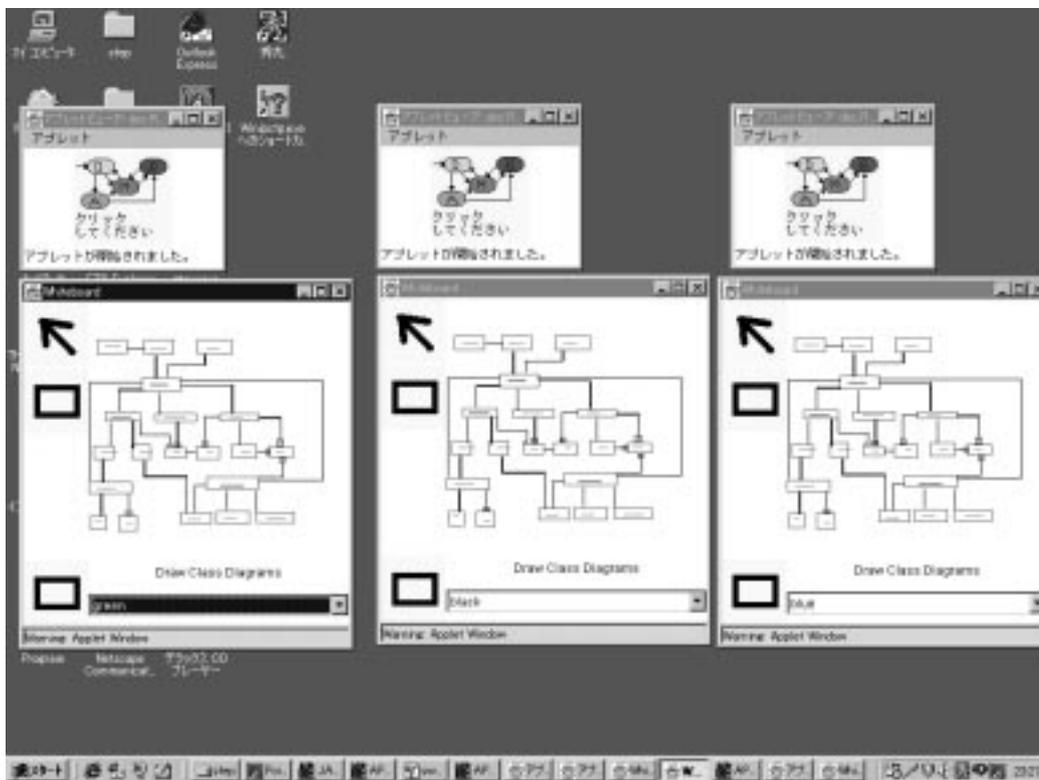


図 4.1: 共有ホワイトボード (1つの画面に3つ窓を開いた所)

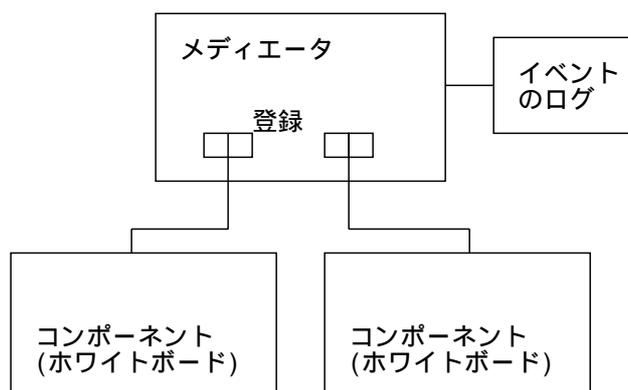


図 4.2: メディエータ

以下にログの一例を示す (冗長なデータは削除してある)。

```
Mon Feb 01 23:13:06 1999:sekun2.jaist.ac.jp addElement(java.lang.Object)
Mon Feb 01 23:13:09 1999:sekun.jaist.ac.jp getUpdate(int)
Mon Feb 01 23:13:09 1999:sekun2.jaist.ac.jp addElement(java.lang.Object)
Mon Feb 01 23:13:11 1999:sekun2.jaist.ac.jp getUpdate(int)
Mon Feb 01 23:13:12 1999:sekun.jaist.ac.jp addElement(java.lang.Object)
Mon Feb 01 23:13:13 1999:sekun2.jaist.ac.jp addElement(java.lang.Object)
```

ログの例 ; addElement は図の描き込み、getUpdate は図の移動の意味である。

4.2 記法の定義

以下に状態遷移モデルの状態と状態への入力、出力と観測される事象の関係を表 4.1,4.2,4.3,4.4 を用いて説明する。

4.2.1 状態を表わす記号

「作成中または変更中」、「修正の可能性」、「調整中」、「安定状態」の 4 つの状態を以下の記号で表す。また、矢印 () は、状態の遷移を表わしている。

1. 作成中または変更中 : 記号 D (Draft)
2. 修正の可能性 : 記号 P (Possibility)
3. 調整中 : 記号 C (Coordinating)
4. 安定状態 : 記号 S (Stable)

4.2.2 入力、出力イベントの定義

状態遷移のトリガとなるイベントの記述は以下の記法で定義する。正規表現 [17] を基本とし (1,2)、新たに記法を 2 つ定義する (3,4)。

E_1 と E_2 は単独のイベントを表している。 $Draw_{ABC}$ は、作業員 A,B,C の描画イベントを順序なしにランダムに 1 回ずつ含むイベント列を表している。

単に隣接したイベントは、順に発生することを表している。

1. $E_1|E_2$ は、選択を表す。
2. E_1^+ は、イベント E_1 が 1 回以上続く。
3. $E_1 \wedge E_2$ は、順序なしで隣接したイベント E_1 と E_2 が発生することを表している。
4. $Draw_{ABC}^+$ は、作業者 A,B,C の描画イベントがそれぞれ 1 回以上隣接してランダムに続く描画イベントのイベント列を表している。
(例)ABAABCCBABCCABCB... の作業者の描画イベントが続く。

4.2.3 イベントの定義

イベントの定義は以下のように定義する。

1. S_A は、作業者 A がツールを起動させるイベントを発生させた。
2. E_A は、作業者 A がツールを終了させるイベントを発生させた。
3. $Draw_A$ は、作業者 A がツールを使用して、描画イベントを発生させた。
4. M_{AB} は、作業者 A が作業者 B に対してメールの送信イベントを発生させた。
5. R_{AB} は、作業者 A が作業者 B に対して返信メールの送信イベントを発生した。つまり、リターンメールを送信した。
6. ϕ は、描画イベントやメールの送信イベントが観測されない状態を表す。

4.3 状態遷移モデルの定義

次の表 4.1,4.2,4.3,4.4により、それぞれの状態を定義するイベント列および、その状態への入力イベント、出力イベントを表すモデルを定義する。

作成中または変更中			
状態	定義	イベント列のパターン	遷移
状態	リーダーによる ドラフトの作成、 変更が行なわれ ている状態。	$Draw_A^+$ (リーダー A の描画イベント が続く)	
入力 1 (1)	作業者全員のツ ツールが立ち上 げられる。 (作業開始から 作成中または変 更中の状態へ)	$S_A \wedge S_B \wedge S_C$	作業開始 → D
入力 2 (2)	ドラフトの変更 リーダーが受理 する。 (調整中から作 成中または変更 中の状態へ)	$(M_{BA}R_{AB}) (M_{CA}R_{AC}) $ $((M_{BA}R_{AB}) \wedge (M_{CA}R_{AC}))$	C → D
出力 (3)	ドラフトの完成 が関係者全員に メールで通知さ れる。 (作成中または 変更中の状態か ら修正の可能性 の状態へ)	$M_{AB} \wedge M_{AC}$	D → P

表 4.1: 作成中または変更中の状態と遷移の方法

修正の可能性			
状態	定義	イベント列のパターン	遷移
状態	リーダー以外の作業員により修正が行なわれている状態。	$(Draw_{BC})^+$ (作業員 B, C の描画イベントがランダムに出現する)	
入力 (3)'	ドラフトの完成が関係者全員にメールで通知される。 (作成中または変更中の状態から修正の可能性の状態へ)	$M_{AB} \wedge M_{AC}$	D → P
出力 1 (4)	変更を要求するメールがリーダーに送られる。 (修正の可能性から調整中の状態へ)	$M_{BA} M_{CA} (M_{BA} \wedge M_{CA})$	P → C
出力 2 (5)	リーダー A から各作業員に確認のメールが送られ返事がくる。 (修正の可能性から安定状態へ)	$(M_{AB} R_{BA}) \wedge (M_{AC} R_{CA})$	P → S

表 4.2: 修正の可能性の状態と遷移の方法

調整中			
状態	定義	イベント列のパターン	遷移
状態	各作業者が図の変更を頻繁に行なっている間の状態。	$(Draw_{ABC})^+$ (作業者 A,B,C の描画イベントがランダムに出現する)	
入力 1 (4)'	変更を要求するメールがリーダーに送られる。 (修正の可能性から調整中の状態へ)	$M_{BA} M_{CA} $ $(M_{BA} \wedge M_{CA})$	P → C
入力 2 (7)	変更を要求するメールが他の作業員へメールで送られる。 (安定状態から調整中の状態へ)	$(M_{AB} \wedge M_{AC}) $ $(M_{BA} \wedge M_{BC}) $ $(M_{CA} \wedge M_{CB})$	S → C
出力 1 (2)'	ドラフトの変更リーダーが受理する。 (調整中から作成中または変更中の状態へ)	$(M_{BA}R_{AB}) (M_{CA}R_{AC}) $ $((M_{BA}R_{AB}) \wedge (M_{CA}R_{AC}))$	C → D
出力 2 (6)	リーダー A から各作業員に確認のメールが送られ返事がくる。 (調整中から安定状態へ)	$(M_{AB}R_{BA}) (M_{AC}R_{CA})$	C → S

表 4.3: 調整中の状態と遷移の方法

安定状態			
状態	定義	イベント列のパターン	遷移
状態	図の変更がなされない状態。	ϕ (イベントが観測されない)	
入力 1 (5)'	リーダー A から各作業者に確認のメールが送られ返事がくる。 (修正の可能性から安定状態へ)	$M_{AB}R_{BA} \wedge M_{AC}R_{CA}$	$P \rightarrow S$
入力 2 (6)'	リーダー A から各作業者に確認のメールが送られ返事がくる。 (調整中から安定状態へ)	$M_{AB}R_{BA} \wedge M_{AC}R_{CA}$	$C \rightarrow S$
出力 1 (7)'	変更を要求するメールが他の作業員へメールで送られる。 (安定状態から調整中の状態へ)	$(M_{AB} \wedge M_{AC}) $ $(M_{BA} \wedge M_{BC}) $ $(M_{CA} \wedge M_{CB})$	$S \rightarrow C$
出力 2 (8)	ツールを全員終了する。 (安定状態から終了へ)	$E_A \wedge E_B \wedge E_C$	$S \rightarrow$ 終了

表 4.4: 安定状態と遷移の方法

わかりやすいように、表と状態遷移図との対応を表した図を以下に示す。表 4.1,4.2,4.3,4.4 の状態の入力、出力における番号が図に一致している。

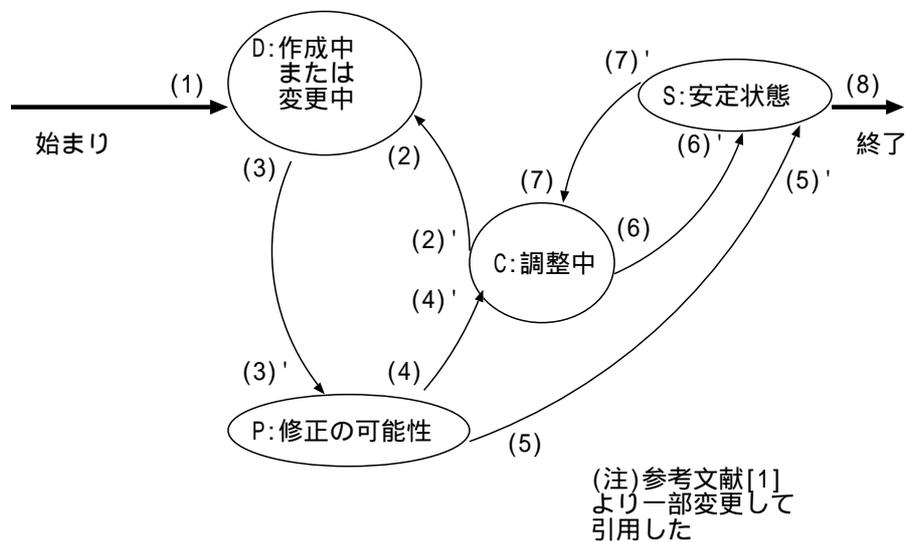


図 4.3: 表と状態遷移図の対応

第 5 章

実験

本章では、予備実験と予備実験をもとにして実験方法を少し変更して行なった本実験の 2 つの実験を通して、本研究で提案するイベントベースアーキテクチャを用いたツール操作イベント及びメール送信、返信イベント等の事象を観測することにより状態遷移図で定めた状態を遷移させ、特定することができるかどうかを検討する。

5.1 予備実験

5.1.1 予備実験の目的

共有ホワイトボードとメールを用いた会議がどのように進行するのか見当をつける。実際に状態遷移図で定めた通りに作業の進行が行なわれるかを確かめる。予備実験では、できるだけ制約を加えないで、共有ホワイトボードとメールによる作業によって事象が観測されるかを探った。

5.1.2 予備実験の設定

共有ホワイトボードとメールを用いて分散同期の形態で時間を決めて共同で図を用いた会議を行なう。

図に対する責任者であり、議事進行役となるリーダーを中心として、リーダーを含めて 3 人でオブジェクト指向モデリングにおけるクラス図の作成を行なう。

返信メールは、確認、変更要求の受理として用い、普段は通常のメール送信をしてもらうようにした。

リーダーと他の作業員 (2 人) は、それぞれ色を変えて図に描き込むことにより、誰の描き込みかわかるようにした。

実験では、自作の事象モニタリング機構を備えた共有ホワイトボードは反応性、操作性が低かったため、事象モニタリング機構を持たない単なる共有ホワイトボードを用いて実験を行ない、作業の様子をビデオに記録しておいて、自作のツール上で再現した。

議題 ダブルカウンタ [15] のクラス図 (図 5.1) をリーダーを中心として、3 人でクラス図の改良を行なう。

被験者 被験者である作業員は、3 人ともオブジェクト指向モデリングについて学んだことのある本学の学生である。デザインパターン (ここでは Observer パターンのみを扱った) についても学んでいる。共有ホワイトボードとメールによる会議にについて慣れてもらうためにあらかじめ 2 人組で練習してもらった。

作業の始まり 議論する問題についてはあらかじめ簡単に知らせてある。全員がツールを立ち上げた後、リーダーが未完成のデザインパターンを用いたダブルカウンタのクラス図を描き込むことにより作業がはじまる。

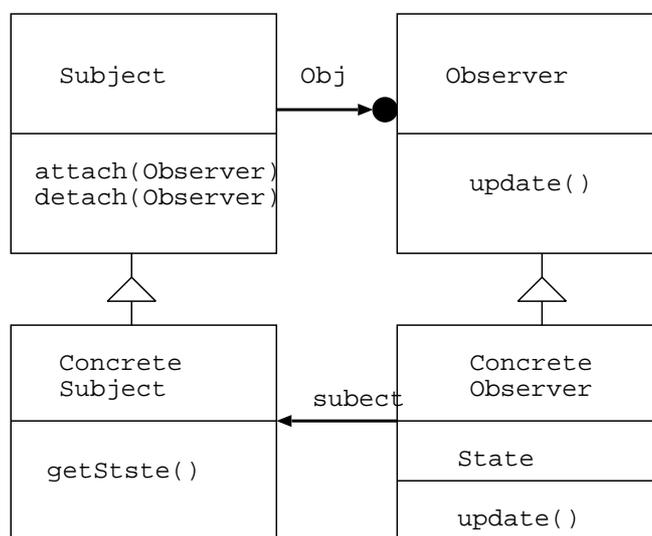


図 5.1: 予備実験で用いたダブルカウンタのクラス図

5.1.3 予備実験の結果と考察

結果 デザインパターンという比較的難しい話題であったためか略記の問い合わせ、確認のメールが多く発生し、図に描き込む作業がほとんど行なわれず図を中心とした会議にならなかった。

会議としても成功した方とはいえないものとなった。

予備実験を通して得られた事柄は以下の通りである。

- 作業時間は約 50 分。流れたメールは 46 通。観測された描画イベント数は 52 回。
- メールとツールの両方を操作しているため、どちらかを見落としてしまう場合もあった。
- 「修正の可能性」の状態は短いながらもその状態としての役割を果たしていることがわかった。
- リーダーは議事進行役としてメールによる状態遷移を促す他、第 3 者的な立場から意見を述べたり同意を求めたりすることがある。このことは、特に「修正の可能性」の状態についてあてはまる。
- 難解な話題であったためか考え込む場面が多く見られた。メールで活発な議論を促す場面が増えてしまった。このことによりイベントが一定時間観測されないといったタイムアウトによる安定状態への遷移は妥当ではないと思われる。
- 図に対して描き込みを行なってはメールを送るという作業が行なわれることがある。

考察 図を中心とした図に描き込み多く行なわれる会議を期待していたがメールによる会話中心でほとんど図に対する描き込みがなされない会議となってしまった。

共有ホワイトボードとメールを用いた制約のない自由なほとんどモデルを持たない会議としての事象が観測されたかもしれないが、図に対する短いコメントと「報告」、「変更要求」といった長めの議事進行に関わるコミュニケーションを同等に扱うことに不自然さを感じた。

図を中心としたある程度モデルを持った会議とするため、本実験では以下のように改良する。

- 図を中心とするため、比較的作業員全員が親しみやすい話題に変更する (描画イベント中心の会議を試みる)。

- 図上に、図に対して気になっていることを示すために、短いコメントを付けることを許す (会議の進行を快適なものにするため)。
- 個別に修正箇所、提案を述べるのではなく、複数箇所まとめて提案してもらうようにした。

5.2 本実験

5.2.1 本実験の目的

図を中心とした会議として、図への描き込みが多く発生しやすいように議論する問題を作業員全員に親しみやすい問題に変更した。図へのコメントを入れてもらうことにより図を中心とした議論できるようにした。描画イベントとメールの送信、返信イベント等により状態遷移図モデルを遷移させられるかを調べる。実験を通して、作成したツールのモニタリングの有効性を確かめる。

本実験では、図を中心とした会議として、ある程度状態遷移図にそって会議が進行するようにする。

5.2.2 本実験の設定

共有ホワイトボードとメールを用いて分散同期の形態で時間を決めて共同で図を用いた会議を行なう。

図に対する責任者であり、議事進行役となるリーダーを中心として、リーダーを含めて3人でオブジェクト指向モデリングにおけるクラス図の作成を行なう。

返信メールは、確認、変更要求の受理として用い、普段は通常のメール送信をしてもらうようにした。

リーダーと他の作業員(2人)は、それぞれ色を変えて図に描き込むことにより、誰の描き込みかわかるようにした。また、リーダーが他の作業員と同じ立場で意見するときはさらに色を変えて描き込んでもらうようにした。

本実験でも、自作の事象モニタリング機構を備えた共有ホワイトボードは反応性、操作性が低かったため、事象モニタリング機構を持たない単なる共有ホワイトボードを用いて実験を行ない、作業の行なわれている画面をビデオに記録しておいて、自作のツール上ビデオに撮影した状態を同じように再現し、ログを作ることを行なった (図 5.2)。

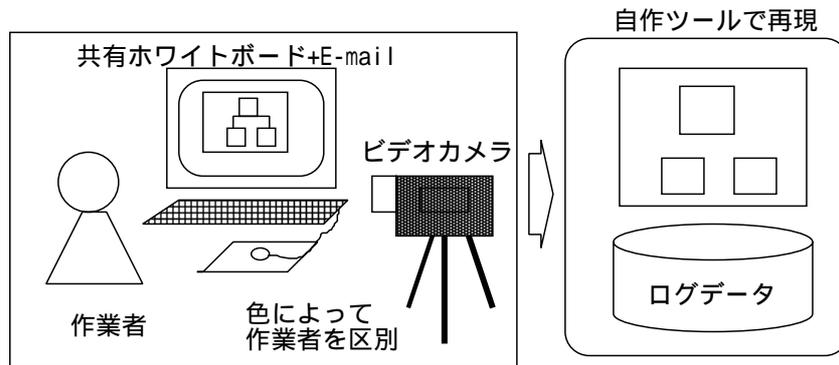


図 5.2: 自作のツール上での再現

議題 図 5.3におけるオブジェクト図は、自動車の構造を部分的に表したものである。関連のいくつかを集約に変化させることでオブジェクト図の改良を図る [16] ことをリーダーを中心とした3人で共有ホワイトボードとメールを用い分散同期の形態で行なった。

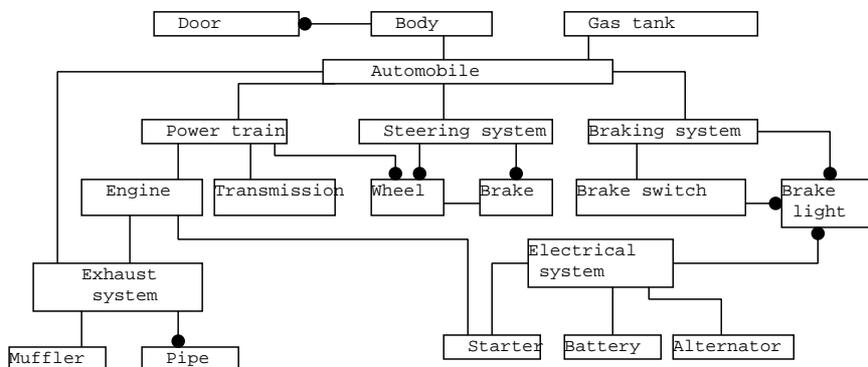


図 5.3: 本実験で議論したオブジェクト図

設定 リーダーと他の作業者(2人)は、それぞれ色を変えて図に描き込むことにより、誰が描き込んだかわかるようにした。また、リーダーが他の作業者と同じ立場で意見する時はさらに色を変えて図への描き込みをしてもらうようにした。

図上にコメントをいれることにより、図を中心としたコラボレーションを促進する。これはメールの使用を束縛するものではなく、図を使った会議を円滑に行なうために行なう。コメントは主に、図に対して気になっていることに対する注意を喚起す

るために用いる。

リターンメールは確認、変更要求の受理といった特別の状態を用い、通常は、普通のメール送信としてもらうことにした。

5.2.3 本実験の結果と考察

図を中心とした建設的で比較的円滑な会議が行なえた。以下に観測結果の概要を述べる。

- 作業時間は約 70 分。流れたメールは 17 通。観測された描画イベント数 143 回。図を中心として会議でメールは頻繁に流れなかった。
- 「調整中」の状態において、図へコメントと付けて意見交換を行なう他、メールによるコミュニケーションも一般的に行なわれる。短い意見はコメントによって行なわれ、比較的抽象的で長い意見はメールが使われる傾向がある。そのため、メールの送信頻度は描画イベントの数に対して少なかった。
- メールとツールの 2 つを操作しているため、行き違いが生じ、「変更中」の状態に遷移して図に描き込む権限がリーダーのみにしかない状態になっても、他の作業者が誤って、図を記入してしまう状態が生じてしまった。
- 「安定状態」を単に誰からも描画イベントが発生せず、変更要求のメール送信イベントが発生しない状態としたが、それだけでは、「安定状態」の役割が十分でないものと思われる。また、実験では、「安定状態」から「調整中」への遷移がうまく観測できなかった。
- 「修正の可能性」でいろいろ図に描き込むことで複数の意見を出してもらうようにしたが、「調整中」として議論する前に、リーダーが図を整理する場合は生じた。リーダーから他の作業員へ変更することをメールを使って認めてもらったうえ「作成中または変更中」の状態として整理すると解釈して作業を進めた。

5.3 作成したツールのモニタリング機能の確認

実験において発生した様々なイベント発生事例やまた、考えられる事例について以下に考察する。

5.3.1 「調整中」に意見交換のメールが入る場合

「調整中」では、描画イベントの間に「質問」、「意見」等のメールが頻繁ではないが混入することが実験により確かめられた。

作業員	システムが観測したイベント	システムが誤って判断した状態	実際の状態
B	描画イベント	C	C
A	描画イベント	C	C
C	描画イベント	C	C
C	メール	-	C
A	描画イベント	D	C
A	描画イベント	D	C
C	描画イベント	エラー	C

5.3.2 「安定状態」の扱い

実験では、「安定状態」と「調整中」を行き来する遷移が上手に観測することができなかった。

作業員	システムが観測したイベント	システムが誤って判断した状態	実際の状態
B	描画イベント	C	C
A	描画イベント	C	C
C	描画イベント	C	C
A	メール	-	-
B	返信メール	-	-
C	返信メール	-	-
			S
C	メール	エラー	
B	描画イベント		C
C	描画イベント		C

「安定状態」については、図を参加者全員の確認を得て、ひとまず完成に至った状態としたが、「安定状態」におけるイベント列としては、何も観測されない状態としただけでは、実験においてははっきりと状態として区別させるに至らなかった。

「安定状態」としてその状態の持つ意味をはっきりと区別するためには、「安定状態」へ遷移した後、図の状態がバージョン番号をつけられて自動的に保存されるなど、作業者に負荷をかけない方法で、状態を区別する必要がある。

5.3.3 「修正の可能性」から「作成中または変更中」への遷移

「修正の可能性」の状態を図の複数箇所の変更を指摘する場として用いたが、「調整中」の状態に遷移して図の変更について議論する前に、「作成中または変更中」の状態へ戻る遷移が実験により観測された。これは、リーダーにより図の整理を行なう状態への遷移である。再度、「修正の可能性」の状態をへて「調整中」の状態で議論が行なわれる。

作業者	システムが観測したイベント	システムが誤って判断した状態	実際の状態
A	描画イベント	D	D
A	描画イベント	D	D
A	描画イベント	D	D
A	メール	-	-
B	描画イベント	P	P
C	描画イベント	P	P
B	描画イベント	P	P
A	メール		
A	描画イベント	C(エラー)	D
A	描画イベント	C	D

5.3.4 例外事象の処理

例外事象の混入

議事進行役であるリーダーを中心として、進行通りに会議が進行しない例外事象の発生が実験においても観測された。

例えば、他の作業者がツールまたはメールの操作のどちらかに集中していたため、「作成中または変更中」の状態としてリーダー以外図の変更を行なえない状態に遷移したのに

関わらず他の作業者が図の変更を行ってしまうなどの行き違いによる例外事象の発生が観測された。

「作成中または変更中」の状態において紛れ込んだ他の作業者の描画イベントは、前後のイベント列から例外として除去できる可能性がある。

作業者	システムが観測したイベント	システムが誤って判断した状態	実際の状態
A	描画イベント	D	D
A	描画イベント	D	D
A	描画イベント	D	D
B	描画イベント	エラー	D
A	描画イベント		D
A	描画イベント		D
A	描画イベント		D

前後にリーダー A の描画イベントが続き、例外事象の B の描画イベントの後、再び、A の描画イベントが続く場合など。

しかしながら、事例に基づく状態の推定では多くの例外事象に対応するのは難しいと思われる。特に、連続して混入する例外事象の検知は難しい。

遷移モデルの例外

また、リーダーを中心として、モデルにそった会議が行なわれることを期待しているが、モデルにそった指示どおりのメールのやりとりが行なわれない場合も考えられる。

その場合は、システムは誤って状態遷移を判断してしまう場合が考えられる。常にイベント列から正しいと推測できる状態をチェックし、状態を修正できる機能が必要である。

第 6 章

事象モニタリング機構の開発

本章では、5章の実験を通して得られた事例をもとに開発した、事象モニタリング機構のプロトタイプについて述べる。

6.1 システムの概要

実験を通して得られた事象をもとに、新たに図 6.1に示した、メール解析機構、及び自動遷移機構を備えた事象モニタリング機構のプロトタイプの開発を行なった。

電子メールの情報は、移植性を高めるために、メールサーバからではなく、ユーザのメールボックスから得るようにした。また、メール解析機構は自動的にメールボックスから最新のメールを取得し、解析した後、イベントとしてログファイルに記録される。

一方、ツール操作は、メディエータを介して、ツール操作のイベントとして同じログファイルに記録される。

事象モニタリング機構は、得られたログファイル(イベントリスト)を解析することにより、システムが持っている状態遷移図に照らし合わせて、共同作業の状態を予測し、自動的に遷移する。

尚、メール解析機構および事象モニタリング機構は、Perlにより開発した。

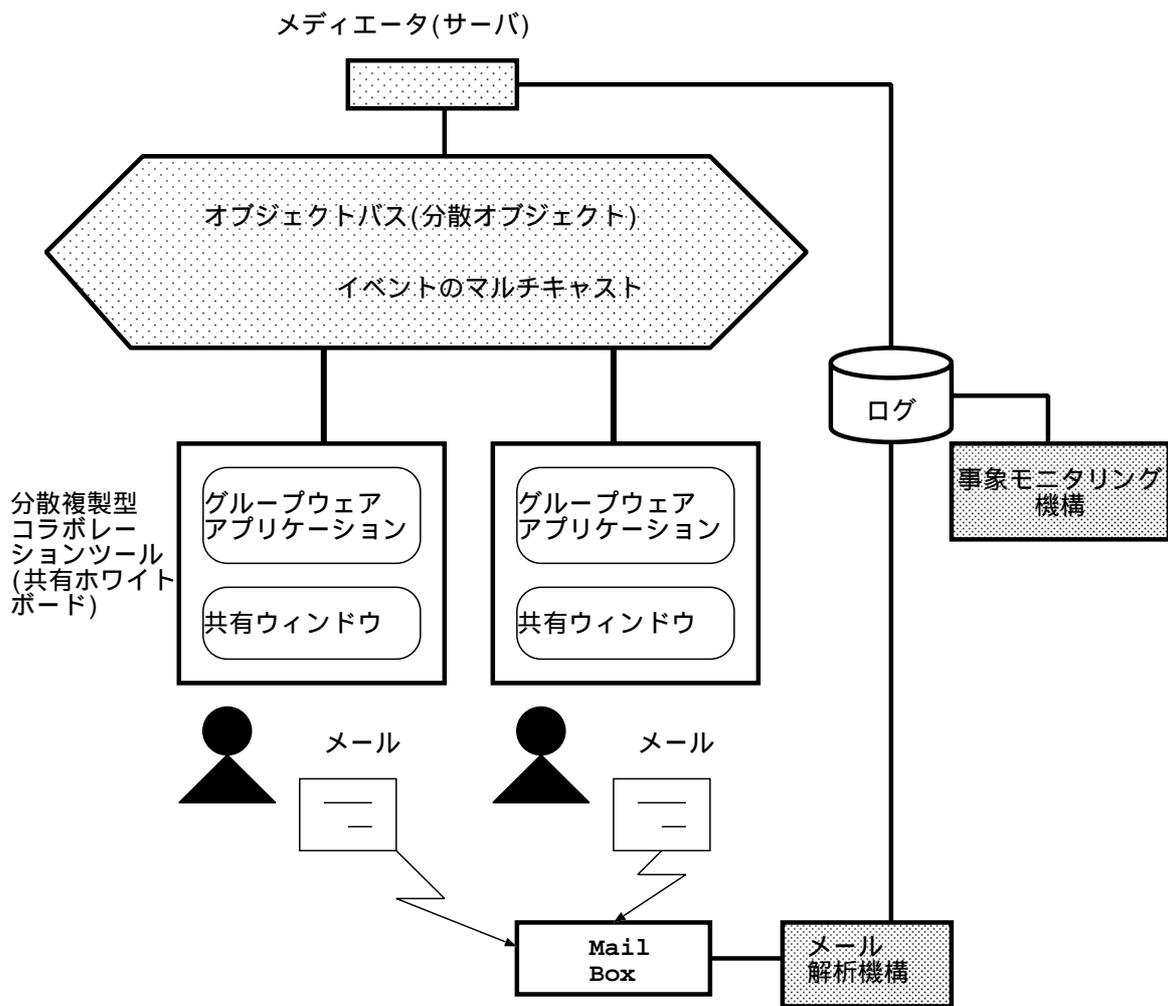


図 6.1: システムの概要

6.2 メール解析機構

電子メールは、Mail Box(各作業者の定義したフォルダー) より情報を取り出す。Mail Box に保存された最新のメールをシステムが読み込み、情報の解析を行なう (図 6.2)。

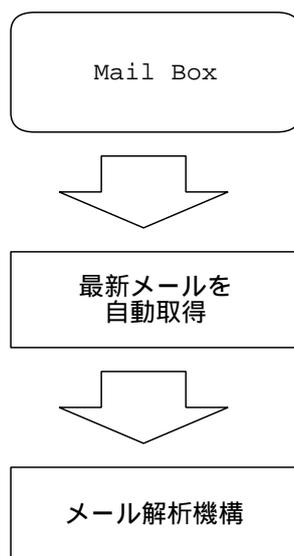


図 6.2: 最新メールの自動取得

Mail Box から得られた最新のメールより、電子メールのヘッダー解析と内容解析をプログラムが自動的に行なう (図 6.3)。

電子メールのヘッダー部分より、電子メールが送信、または返信されたことをイベントとして記録する。その際、電子メールの送信、返信イベントのイベント属性として、日付、時間、送信者、受信者、タイトルを同時に記録する。

また、オプションとして、メールの内容から「公開」、「変更要求」、「確認」などのキーワードや文字列を検出し、記録できるようにした。メールの内容に含まれるキーワードを用いることにより、状態遷移のトリガとななりうるメールを区別するのに役立つものと思われるが、はっきりとした相関関係がまだ得られていないので、オプションとしてスイッチを用いて、ユーザが選択できるように設計した。

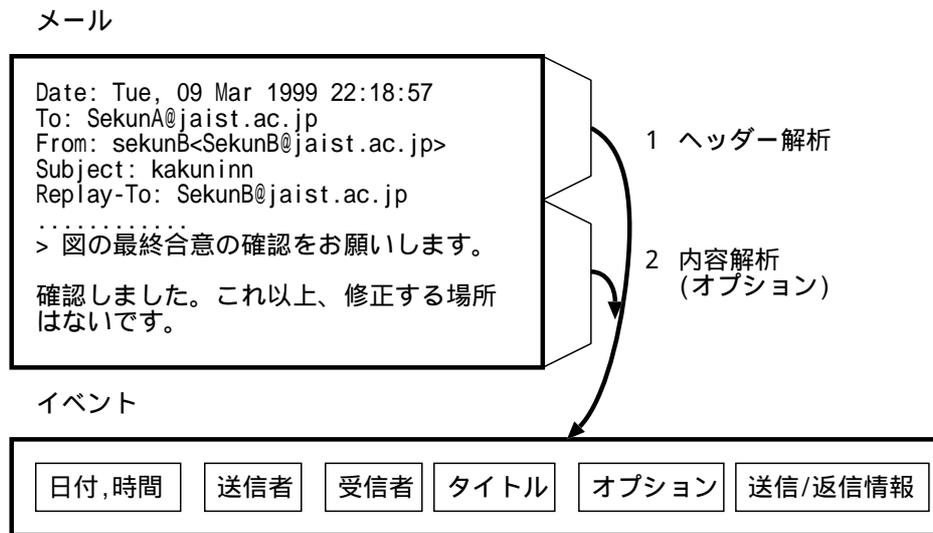


図 6.3: メール解析機構

6.3 事象モニタリング機構

6.3.1 自動遷移機構

自動遷移機構は、第 4 章において説明した状態遷移モデルを内部に持っている。また、同じく第 4 章において定義した事象系列パターンに従って、入力、出力の条件がそろえば自動的に状態遷移モデルに従って遷移させる。

時間の経過にそって順に記録されるイベントリストに対して、遷移のトリガとなりうるメールの送信、返信イベントが、イベント属性をチェックして、第 4 章で定義した入力、出力の条件に一致したとき、遷移を起こす。

自動遷移機構は、遷移する条件がそろえば、遷移してしまうので、次にくる状態が正しい状態であるかの判断はできない。

図 6.4 に示すように、調整中のあと、確認のメールがくれば、安定状態に遷移してしまうが、次にきた状態が安定状態であるかはわからない。

6.3.2 状態のチェック機構

状態チェック機構は、イベントリストがある程度できあがってから状態を判断する。メールの送信、返信イベントを区切りに、その間で発生した、描画イベントの事象系列パター

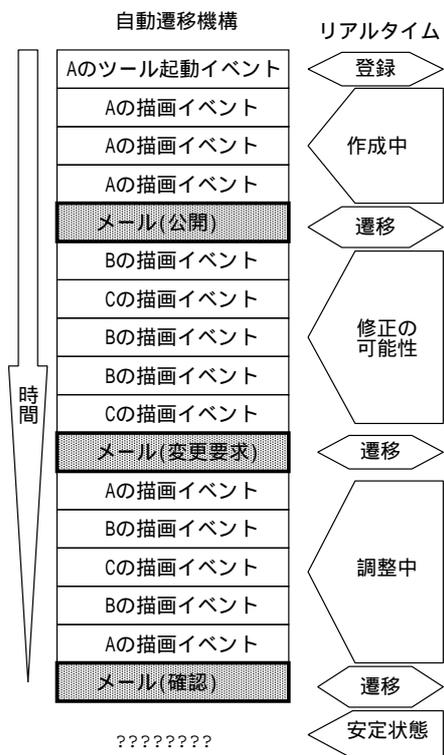


図 6.4: 自動遷移機構

ンを調べ、状態を判断する。

状態の判断は、第4章で定義した、事象系列パターンに従う。状態チェック機構は、メール送信イベントが、遷移の対象であるかなどを調べたりはしない。

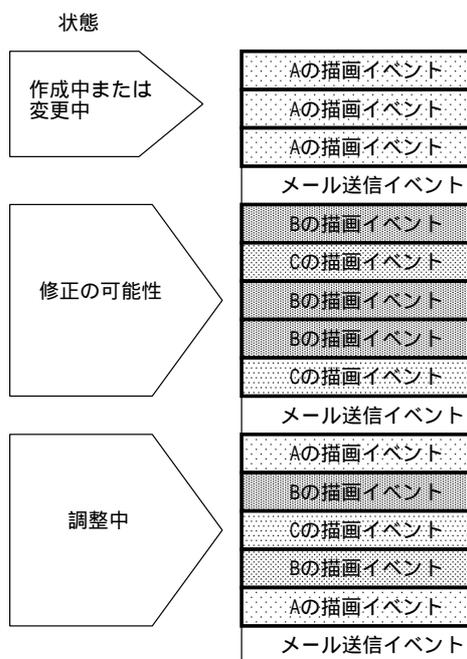


図 6.5: 状態チェック機構

6.3.3 状態決定アルゴリズムの概要

事象モニタリング機構における状態決定アルゴリズムについて図6.6を用いて説明する。

自動遷移機構は、時間の経過に従って得られる事象により条件がそろえば、自動的に遷移を行なう。しかし、実験で確認されたように、質問や注意など遷移の対象とならないメールの送信、返信の事象が条件に一致して間違っただ遷移を起こす場合もある。

状態チェック機構は、自動遷移が行われたあとに、事象系列パターンを調べて状態を判断し、後から間違っただ遷移を起こしている場合、正しい状態に戻す。

図6.6では時事刻々と得られる事象に対して、条件がそろい、「作成中」「修正の可能性」「調整中」と遷移しているが「修正の可能性」から「調整中」への遷移の際、質問を意図して送信されたメールに反応して間違っただ遷移を起こした場合を示している。状態チェック機構は後から得られた事象系列パターンを調べ、自動遷移機構によって得られた

状態が正しいかを判断し矛盾を起こしていた場合、元の状態に戻す。図 6.6では、「修正の可能性」の状態でも矛盾を起こしているため正しい状態として「修正の可能性」の状態に戻す。

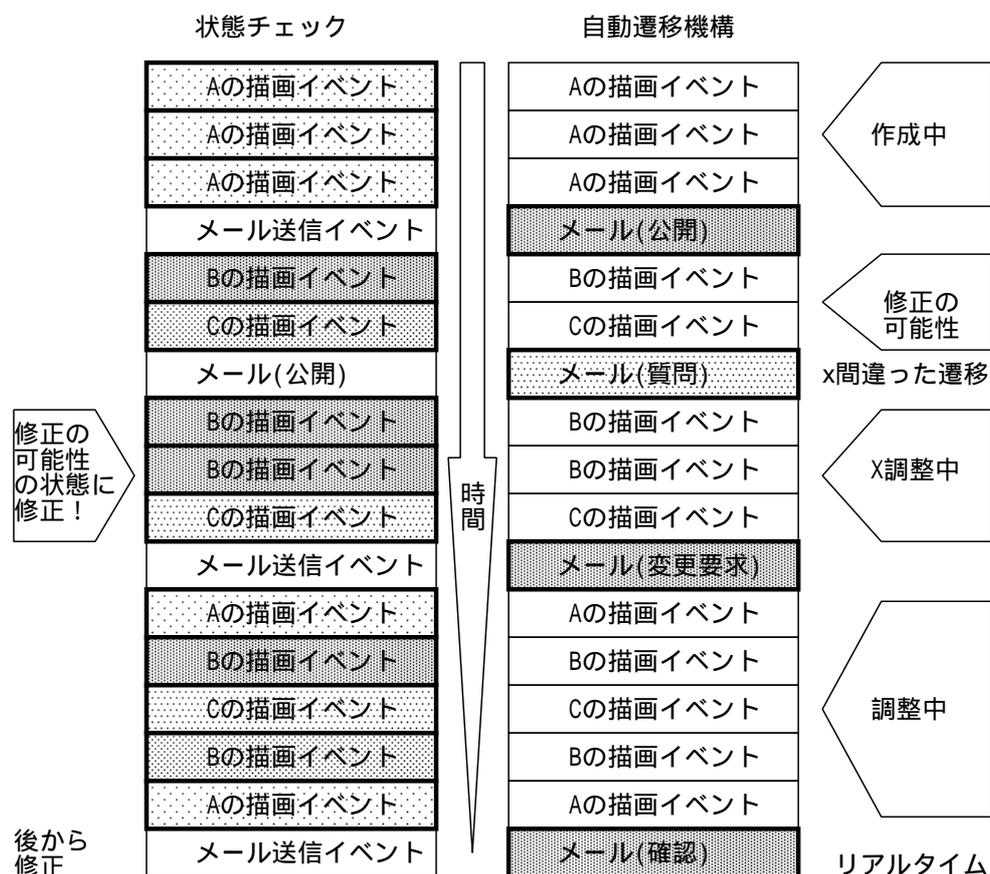


図 6.6: 状態決定アルゴリズムの概要

6.4 事象駆動の欠点の克服

事象駆動型制御の欠点は誤った情報の混入や必要な情報の欠落に対して弱いことである [18]。システムは、定義された事象の出現に対して有効に機能するが、定義されていない事象の出現や、定義されている事象が発生しない場合、駆動させていくのが困難である。

6.4.1 事象モニタリング機構の性能

本研究で開発した事象モニタリング機構のプロトタイプのパフォーマンスについて以下に述べる。

- 対応済みの機能
 - 遷移の対象とならないメールの送信、返信の事象を区別する。遷移の対象とならないメールの送信、返信の事象とは、実験でみられたように、質問や質問に対する回答、注意を促すためのメールや議論を促すためのメールである。
 - 状態を確認することにより間違った遷移を起こした場合は、ロールバックする。
 - メールに含まれる、キーワード（公開、変更要求、確認など）や文字列をもとに遷移のトリガとなりうるメールの送信、返信イベントを区別する。
- 未対応の機能
 - 「作成中はまたは変更中」の状態において他の作業者が誤って描画してしまうといった場合の例外事象（描画イベント）の処理。
 - 遷移の対象となるメールの送信、返信イベントの欠落に対する処理。

第 7 章

議論

本章では、議論として、5章の実験で得られた事例や、6章で開発した事象モニタリング機構のプロトタイプの開発をもとに、描画イベント列で状態を定め、実際の共同作業で起こり得るイベント列より共同作業の状態と状態遷移の可能性に対応できるように議論する。

7.1 実験の考察

実験の結果に基づき以下のような考察が得られた。

- 「調整中」の状態において、図に対するコメントにより意見交換が行なわれ、頻度は高くないがメールによる意見交換も行なわれる。発生頻度の高くない「質問」、「返答」などの「調整中」の状態の中で生じるメールの送信、返信イベントを処理する必要がある。
- 状態遷移モデルで対応しきれないパターンとして「修正の可能性」から「作成中または変更中」への遷移がみられた。
- 単にイベントを時系列上で観測するだけでは、「安定状態」を観測するには不十分であることがわかった。
- 「修正の可能性」の状態は短いながらもリーダーの提示した図に対して意見を描き込む場として十分に機能していることがわかった。また、この間はメール送信、返信の事象は観測されない。

以下に問題点を整理しつつ、今後の課題を議論する。

	作成中 または 変更中	修正の 可能性	調整中	安定状態
作成中 または 変更中	D	1	—	—
修正の 可能性	7	P	2	3
調整中	4	—	C	5
安定状態	—	—	6	S

表 7.1: 修正した状態遷移モデル

7.2 状態遷移図モデルの修正案

表 7.1は、縦軸をソース、横軸をシンクとして状態遷移図モデルを表わしたものである。以下に、4つの状態におけるイベント列を判別する条件をあげる。

作成中または変更中 $Draw_A^+$

修正の可能性 $Draw_{BCD}^+$

(リーダー A が第 3 者の作業者 D として図に描き込む場合を含む)

調整中 $Draw_{ABC}^+$

(単に描画イベントの接続ではなく、間にコミュニケーション手段としてメール送信イベントが含まれる。)

安定状態 ϕ

(「安定状態」へ入る毎に図はセーブされ、バージョン番号が付けられるものとする。変更要求のメール送信イベントが発生し、「調整中」の状態をへて、再び安定状態になった時は、新たにセーブされ、バージョン番号が付けられる。)

以下に状態の遷移を促すイベント列の条件をあげる。基本的に遷移を起こす前の状態と遷移のトリガとなるメールの送信、返信イベントにより遷移させる。

しかしながら、メールの送信は、「調整中」の状態において意見交換に使われる場合もあるので、その場合は、メールの送信を状態の区切りとして次の状態を先読みすることにより遷移させるかどうかシステムに判断させる。

1. $Draw_A^+(M_{AB} \wedge M_{AC})$
(リーダーの描画イベントが続き、リーダーの公開のメールが送信される。)
2. $Draw_{BCD}^+(M_{BA}|M_{CA}|(M_{BA} \wedge M_{CA}))$
(他の作業員として描画イベントが続き、変更要求のメールが送信される。)
3. $Draw_{BCD}^+((M_{AB}R_{BA}) \wedge (M_{CA}R_{AC}))$
(参加者全員の描画イベントが入り乱れて続き、メールの送信イベントを区切りとして、先読みしたイベント列の状態が前の状態と異なっていれば遷移する。)
4. $Draw_{ABC}^+((M_{BA}R_{AB})|(M_{CA}R_{AC})|((M_{BA}R_{AB}) \wedge (M_{CA}R_{AC}))|(M_{AB} \wedge M_{AC}))$

(ただし、遷移のトリガとならない「質問」、「解答」等のメールが間に入る時の処理については後述する。)

5. $Draw_{ABC}^+((M_{AB}R_{BA}) \wedge (M_{AC}R_{CA}))$
6. $\phi((M_{AB} \wedge M_{AC})|(M_{BA} \wedge M_{BC})|(M_{CA} \wedge M_{CB}))$
7. $Draw_{BCD}^+(M_{AB} \wedge M_{AC})$
(新たに「修正の可能性」から「変更中」への遷移を設ける。リーダーが他の作業員に変更することをメールで通知することによって遷移する (図 7.1)。

「調整中」におけるメールの処理 (図 7.2) を以下に説明する。

メールの後、いくつかイベント列を読み込むかによって精度が変わってくる。また、偶然にもリーダーの描画イベントが続き、「作成中または変更中」の状態と判断してしまう場合が考えられる。

7.3 モニタリング精度の向上の方針

本研究では、観測されるイベントの粒度を比較的粗く設定したが、さらに詳しく作業員のツール操作で行なわれた事象を観測する事やメールの内容を解析する方法などにより構造化メールを用いる事なしに精度を向上させることは可能であると思われる。以下に実験で得られたデータをもとに議論する。

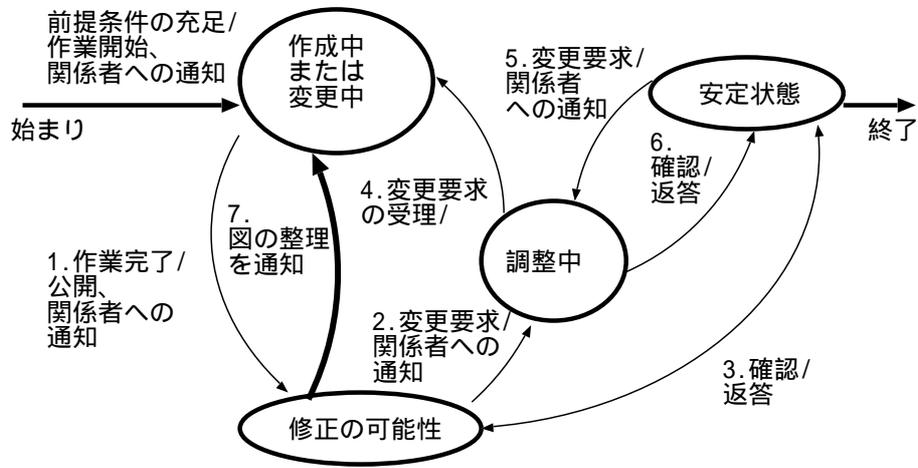


図 7.1: 改良した状態遷移図

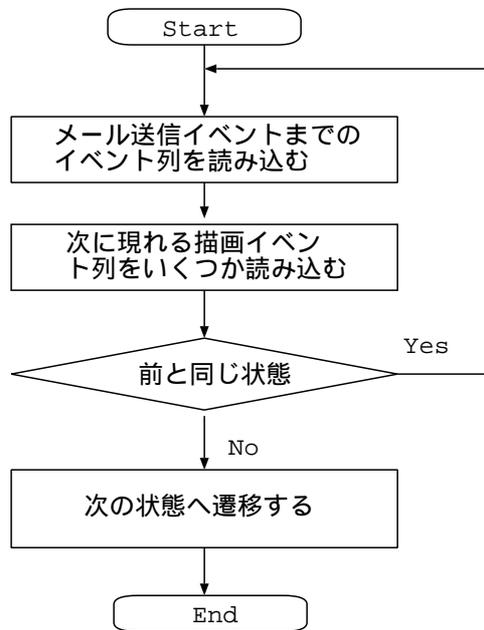


図 7.2: 「調整中」におけるメールの処理

7.3.1 描画イベントの観測

描画イベントとして観測されたイベントをさらに詳しく分類してみた。

今回使用された描画イベントは、「四角」、「線」、「三角」、「矢印」、「コメント」、「丸」のみである。

さらに詳しく位置関係や、会議の様子を再現できるまで精度をあげて観測することができれば、モニタリングの精度を向上させられることが期待できると思われる。

7.3.2 メールの内容の解析

今回の実験データをもとに、メールの内容を見る事によって分類してみた。

おおよそ「報告」（「公開」）、「質問」、「返答」、「議論を促す」、「修正案」、「修正の完了」、「確認」、「確認の返事」、「提案」、「変更要求」、「変更要求の受理」に分けられた。

メールの内容が解析できるようになれば、モニタリングの精度を向上させられることが期待できると思われる。

特に、「調整中」の状態では、「質問」、「返答」、「報告」、「議論を促す」などのメールが送られる。状態の遷移のトリガとなる得る「変更要求」、「変更要求の受理」、「確認」、「確認の返事」を区別できればモニタリングの精度を向上できると思われる。

そのためには、「変更要求」、「確認」、「質問」などのメールが交換されるときどのような内容や言葉を含む場合が多いか分析する必要があるように思われる。

事象モニタリング機構のプロトタイプの開発では、簡単にキーワードによる検出を行なった。

第 8 章

おわりに

8.1 まとめ

本研究では、まず最初に本研究で扱うネットワークを介した共同作業のモデルについて説明し、メールの送信、返信イベントおよび共有ホワイトボードのツールより発生するイベントを観測することにより共同作業の状態遷移モデルを遷移させる方法を定義した。また、実際の共同作業を通して得られた事象のモニタリングをもとに、事象モニタリング機構のプロトタイプを開発し、一部の機能を備えた自動遷移機構を開発した。

その後、事象モニタリングの実験やプロトタイプの開発を通して、問題点を整理しつつ今後の課題を議論した。また、モニタリング精度の向上の方針について描画イベントの観測、メール内容の解析について述べた。

8.2 今後の課題

今後、イベントベースアーキテクチャにおける事象モニタリング機構として発展させるため、残された課題として以下の事柄が考えられる。

状態遷移モデルの向上 仮の共同作業を表す状態遷移モデルとして、4つの状態しか持たない比較的簡単なモデルを使用した。しかしながら、より共同作業の把握に役立つ状態遷移モデルを追求する必要があると思われる。

イベント解析の定式化, 精度の向上 ほとんど作業ルールを定めないまま、ツール操作イベント、メール操作イベントの事象を観測し、状態を遷移させるには、多くの例外

に対応する必要がある。そのためには、なんらかのイベント列の定式化及び精度の向上が望まれる。特に事象の欠落に関して対応する必要があるものと思われる。

自動遷移機構の開発 イベント解析の定式化、精度の向上が測れれば、イベント列をプログラミング言語により処理させることによりシステムによる自動遷移機構の開発が行なえるものと思われる。本研究では、一部の機能を実現したプロトタイプの開発を行ったが、より様々な例外事象に対応できる自動遷移機構の開発が望まれる。

CORBA 上でのシステム開発 今回、実験的かつ比較的簡単に実装の行なえる分散オブジェクトの利用ということで、イベントベースアーキテクチャの実現のために、Java 言語にネイティブな分散オブジェクトである、Java RMI を用いた。しかしながら、本研究室で開発中の「自在」のコンセプトに応じたより実用に耐える環境での使用できるようにするためには、CORBA で実装した方が良いと思われる。

謝辞

最後に、本研究を行なうにあたり、終始御指導賜りました北陸先端科学技術大学院大学情報科学研究科の落水浩一郎教授に心より感謝申し上げます。

本研究を進めるにあたり、他方面から御助言を賜りました海谷治彦助手に深く感謝申し上げます。

また、論文審査にあたって適切な御助言、御意見を賜りました北陸先端科学技術大学院大学の篠田陽一助教授、権藤克彦助教授に深く感謝申し上げます。

そして、本研究に関して多くの有意義な助言を頂きました博士課程の藤枝和宏氏、堀雅和氏、村越広亨氏、ならびに落水研究室、篠田研究室の皆様方に感謝申し上げます。

高瀬泰宏氏、林崎浩典氏には、実験に参加して頂き、有意義な助言を頂きました。深く感謝致します。

付録 A

予備実験のデータ

作業者	イベントの種類	時間	観測イベント	状態
A	四角	21:05:30	描画イベント	D
		中略		
A	コメント	21:11:18	描画イベント	D
A	コメント	21:11:35	描画イベント	D
A	コメント	21:11:45	描画イベント	D
A	コメント	21:11:55	描画イベント	D
A	コメント	21:11:59	描画イベント	D
A	公開	21:12:00	メール	-
B	丸	21:14:55	描画イベント	P
C	丸	21:17:32	描画イベント	P
B	質問	21:18:00	メール	-
B	丸	21:18:34	描画イベント	P
C	質問	21:18:54	メール	
A	修正	21:19:00	メール	

作業者	イベントの種類	時間	観測イベント	状態
A	確認	21:21:00	メール	C
A	確認	21:27:00	メール	C
B	質問	21:32:00	メール	C
B	報告	21:37:00	メール	C
A	公開	22:08:00	メール	C
B	丸	22:08:32	描画イベント	C
C	丸	22:08:43	描画イベント	C
B	質問	22:09:43	メール	C
C	質問	22:11:00	メール	C
A	返答	22:11:00	メール	C
A	促す	22:18:00	メール	C
A	返答	22:22:00	メール	C
A	促す	22:23:00	メール	C
B	質問	22:23:00	メール	C
A	返答	22:24:00	メール	C
B	丸	22:25:34	描画イベント	C
C	丸	22:26:32	描画イベント	C
A	修正	22:27:00	メール	
C	提案	22:28:00	メール	
A	四角	22:18:12	描画イベント	D
A	四角	22:18:14	描画イベント	D
A	四角	22:19:32	描画イベント	D
A	線	22:20:22	描画イベント	D
A	線	20:20:44	描画イベント	D
A	線	20:21:57	描画イベント	D
A	線	20:24:13	描画イベント	D
A	線	20:25:43	描画イベント	D
A	三角	20:26:14	描画イベント	D
A	丸	20:27:14	描画イベント	D
A	丸	20:28:43	描画イベント	D
A	注意	22:29:00	メール	D
A	コメント	22:29:34	描画イベント	D
A	コメント	22:30:44	描画イベント	D
A	コメント	22:31:13	描画イベント	D
A	コメント	22:31:32	描画イベント	D
A	公開	22:32:00	メール	

作業者	イベントの種類	時間	観測イベント	状態
B	再送	22:32:00	メール	P
A	注意	22:34:00	メール	P
B	再送	22:34:00	メール	P
A	返答	22:36:00	メール	P
B	丸	22:36:12	描画イベント	P
B	コメント	22:36:32	描画イベント	P
C	丸	22:36:43	描画イベント	P
C	コメント	22:36:55	描画イベント	P
C	意志	22:36:59	メール	P
B	報告	22:37:00	メール	P
C	質問	22:38:00	メール	P
B	質問	22:40:00	メール	P
C	意志	22:40:00	メール	P
B	丸	22:41:12	描画イベント	P
C	丸	22:41:32	描画イベント	P
A	返答	22:42:00	メール	
A	丸	22:42:12	描画イベント	C
B	意志	22:42:32	メール	C
A	修正	22:43:00	メール	C
A	質問	22:46:00	メール	C
C	確認	22:46:00	メール	C
A	返答	22:48:00	メール	C
A	質問	22:49:00	メール	C
B	提案	22:49:00	メール	C
A	注意	22:51:00	メール	C
B	報告	22:51:00	メール	C
A	注意	22:53:00	メール	C
B	確認	22:53:00	メール	
A	終了	22:55:00	メール	
C	了承 Re	22:57:00	メール	
B	了承 Re	22:58:00	メール	
				S

付録 B

本実験のデータ

作業者	イベントの種類	時間	観測イベント	状態
A	案内	20:26:00	メール	-
A	開始	20:45:00	メール	-
A	四角	20:45:37	描画イベント	D
A	四角	20:45:39	描画イベント	D
A	四角	20:45:57	描画イベント	D
A	四角	20:47:29	描画イベント	D
		中略		
A	線	20:50:05	描画イベント	D
A	線	20:50:13	描画イベント	D
A	線	20:50:15	描画イベント	D
A	線	20:50:17	描画イベント	D
A	線	20:50:30	描画イベント	D
A	線	20:50:42	描画イベント	D
A	線	20:50:45	描画イベント	D
A	線	20:50:50	描画イベント	D
A	線	20:51:12	描画イベント	D

作業者	イベントの種類	時間	観測イベント	状態
A	線	20:51:15	描画イベント	D
A	線	20:51:19	描画イベント	D
A	線	20:51:30	描画イベント	D
		中略		
A	丸	20:52:55	描画イベント	D
A	丸	20:52:57	描画イベント	D
A	丸	20:52:58	描画イベント	D
A	丸	20:52:59	描画イベント	D
A	コメント	20:53:02	描画イベント	D
A	コメント	20:53:12	描画イベント	D
A	コメント	20:53:14	描画イベント	D
A	コメント	20:53:17	描画イベント	D
A	コメント	20:53:24	描画イベント	D
A	コメント	20:53:58	描画イベント	D
A	コメント	20:54:03	描画イベント	D
A	コメント	20:54:14	描画イベント	D
A	コメント	20:54:03	描画イベント	D
A	コメント	20:54:14	描画イベント	D
A	コメント	20:54:17	描画イベント	D
A	コメント	20:54:19	描画イベント	D
A	コメント	20:54:23	描画イベント	D
A	コメント	20:54:32	描画イベント	D
A	コメント	20:54:42	描画イベント	D
A	コメント	20:54:44	描画イベント	D
A	コメント	20:54:47	描画イベント	D
A	コメント	20:54:50	描画イベント	D

作業者	イベントの種類	時間	観測イベント	状態
A	コメント	20:54:51	描画イベント	D
		中略		
A	公開	20:55:00	メール	-
B	丸	20:55:03	描画イベント	P
C	丸	20:55:32	描画イベント	P
B	丸	20:56:21	描画イベント	P
C	丸	20:56:48	描画イベント	P
C	コメント	20:57:02	描画イベント	P
B	丸	20:57:24	描画イベント	P
B	コメント	20:57:32	描画イベント	P
C	丸	20:58:00	描画イベント	P
B	丸	20:58:12	描画イベント	P
B	コメント	20:58:32	描画イベント	P
C	丸	20:59:13	描画イベント	P
B	丸	20:59:17	描画イベント	P
C	丸	20:59:45	描画イベント	P
C	コメント	20:59:55	描画イベント	P
C	修正完了	21:02:00	メール	-
B	修正完了	21:07:00	メール	-
A	消去	21:07:31	描画イベント	D
A	消去	21:08:12	描画イベント	D
		中略		
A	移動	21:09:56	描画イベント	D
A	移動	21:09:58	描画イベント	D
A	注意	21:10:45	メール	D
A	注意	21:11:16	メール	D
A	消去	21:13:22	描画イベント	D
A	消去	21:13:43	描画イベント	D
A	再公開	21:14:00	メール	-
B	丸	21:14:45	描画イベント	P

作業者	イベントの種類	時間	観測イベント	状態
C	丸	21:14:55	描画イベント	P
B	丸	21:15:02	描画イベント	P
B	コメント	21:15:30	描画イベント	P
C	丸	21:15:35	描画イベント	P
C	コメント	21:16:33	描画イベント	P
B	丸	21:16:39	描画イベント	P
B	コメント	21:17:18	描画イベント	P
C	丸	21:17:34	描画イベント	P
C	コメント	21:17:48	描画イベント	P
B	丸	21:17:56	描画イベント	P
B	丸	21:18:44	描画イベント	P
C	丸	21:18:59	描画イベント	P
B	丸	21:19:03	描画イベント	P
B	丸	21:19:16	描画イベント	P
B	コメント	21:19:18	描画イベント	P
C	コメント	21:19:34	描画イベント	P
B	確認	21:20:00	メール	-
C	確認	21:22:00	メール	-
A	コメント	21:23:13	描画イベント	C
B	丸	21:23:45	描画イベント	C
C	丸	21:23:55	描画イベント	C
A	丸	21:24:01	描画イベント	C
A	コメント	21:24:05	描画イベント	C
C	質問	21:24:10	メール	C
B	丸	21:24:16	描画イベント	C
B	コメント	21:24:35	描画イベント	C

作業者	イベントの種類	時間	観測イベント	状態
C	丸	21:24:44	描画イベント	C
C	丸	21:25:13	描画イベント	C
A	丸	21:25:33	描画イベント	C
B	コメント	21:25:35	描画イベント	C
B	コメント	21:25:37	描画イベント	C
C	丸	21:25:41	描画イベント	C
A	返答	21:25:10	メール	C
C	丸	21:25:45	描画イベント	C
B	コメント	21:25:52	描画イベント	C
C	丸	21:25:53	描画イベント	C
A	丸	21:25:55	描画イベント	C
B	丸	21:25:57	描画イベント	C
C	コメント	21:26:00	描画イベント	C
A	丸	21:27:03	描画イベント	C
B	丸	21:27:16	描画イベント	C
C	丸	21:27:33	描画イベント	C
C	丸	21:28:13	描画イベント	C
B	コメント	21:28:32	描画イベント	C
B	コメント	21:28:34	描画イベント	C
B	丸	21:28:44	描画イベント	C
A	丸	21:29:04	描画イベント	C
A	コメント	21:29:32	描画イベント	C
A	コメント	21:29:34	描画イベント	C
A	丸	21:29:44	描画イベント	C
A	丸	21:29:55	描画イベント	C
A	確認	21:30:00	メール	-
B	確認 Re	21:34:00	メール	-
C	確認 Re	21:34:00	メール	-
A	終了	21:35:00	メール	-

参考文献

- [1] 落水浩一郎, 漸増的ソフトウェア設計・実現のためのプロセスモデル日本ソフトウェア科学会第14回大会論文集
- [2] Mary Shaw, David Garlan, “SOFTWARE ARCHTECTURE”, PRENTICE HALL,1996.
- [3] G.Cugola, E.Di Nitto, A.Fuggetta, “Exploiting an event-based infrastructure to develop complex distributed systems” IEEE Software,pp261-270,1998.
- [4] Daniel J.Barrett, Lori A.Clarke, Reri L.Tarr, Allexander E. wise, “A Framework for Event-Based Software Integration” ACM Transactions on Software Engineering and Methodology, Vol 5, No. 4, Oct.,PP378-421,1996.
- [5] 石井裕著, CSCW とグループウェア, オーム社,1994.
- [6] 速水治夫, (1) ワークフロー入門, 39 巻 11 号, 情報処理,1998 年 11 月.
- [7] 速水治夫, 坂口俊昭, 渋谷亮一, (2) ワークフロー製品の標準化, 39 巻 12 号, 情報処理,1998 年 12 月.
- [8] 石井裕著, グループウェアのデザイン, 共立出版,1994.
- [9] Gianpaolo Cugola, “Tolerating Deviations in Process Support Systems Via Flexible Enactment of Process Models” Dipartimento di Elettronica e Informazione,1997.
- [10] Jim Farley 著, 小俣裕一監訳, 豊福剛訳, JAVA 分散コンピューティング, O'REILLY,1998.
- [11] Robert Orfali,Dan Harkey,Jeri Edwards 共著, 分散オブジェクトサバイバルガイド (上), JUSTSYSTEM,1998.

- [12] Robert Orfali, Dan Harkey, Jeri Edwards 共著, 分散オブジェクトサバイバルガイド (下), JUSTSYSTEM, 1998.
- [13] 坂下善彦, 井手口哲夫, 滝沢誠, 水野忠則著, 分散システム入門, 近代科学社, 1993.
- [14] 松下温, 岡田謙一編著, コラボレーションとコミュニケーション, 共立出版, 1995.
- [15] 鈴木大輔, 自律オブジェクト群の制御を目的とするデザインパターンの拡張, 修士論文, 北陸先端科学技術大学院大学, 1998.
- [16] J. ランボー, M. ブラハ, W. プラメラニ, F. エディ, W. ローレンセン著, オブジェクト指向方法論 OMT トッパン, 1991.
- [17] 斎藤靖, 小山祐司, 前田薫, 布施有人共著, 新 Perl の国へようこそ, サイエンス社, 1996.
- [18] 小林重信著, 知識工学, 昭晃堂, 1987.