社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
SS 93-42 (1994-01)

プロセス記述による ソフトウェア開発における連絡支援の試み

玉井昌朗† 飯田元† 井上克郎† 鳥居宏次‡

†大阪大学 基礎工学部
‡奈良先端科学技術大学院大学 情報科学研究科

†〒560 大阪府豊中市待兼山町 1-1
‡〒630-01 奈良県生駒市高山町 8916-5

あらまし

近年のソフトウェアの大規模化により，ソフトウェア開発は複数人による協調作業や分散開発を強いられることが多い．この場合，開発者は相互にコミュニケーションを行ないながら開発を進める．ソフトウェア開発に特化したコミュニケーション支援を行なえば，開発効率を高めることが期待できる．

本稿では，開発時における要求や質問などに対象を特定化した連絡のモデルを提案する．また，このモデルに基づいて連絡作業の一部をエージェントを用いて自動化する方法について述べる．

和文キーワード プロセスモデル，連絡モデル，エージェント，オブジェクト，構造化メッセージ

A Communication Support Model with Process Description for Software Development

Masaaki Tamai† Hajimu Iida† Katsuro Inoue† Koji Torii‡

†Faculty Engineering Science, Osaka University
‡Graduate School of Information Science, Nara Institute of Science and Technology

†1-1 Machikaneyama, Toyonaka, Osaka 560, Japan
‡8916-5 Takayama, Ikoma, Nara 630-01, Japan

Abstract

This paper proposes a communication model which is specific to software development activities, such as a request of a work or a question on a document. In this model, objects such as products, processes and workers request a work to other object and replies a result of the work. This model is activated by agents that partially automate communication activities instead of the real developers.

英文 key words process model, communication model, agent, object, structured message

1 はじめに

近年のソフトウェアの大規模化により，ソフトウェア開発は複数人による作業を強いられるようになった．このような場合，作業者間のコミュニケーションに要するコストを無視できなくなる．したがって，協調作業においてコミュニケーションを支援することは，作業の効率を高めるために重要である．

本稿では，ソフトウェア開発に特化したコミュニケーションの支援を目的として，分散環境を考慮した開発時における要求や質問などに対象を特定化し，連絡を行なうためのモデルを提案する．また，このモデルに沿った連絡を行なうための仕組みとしてエージェントを用いた開発支援システムについて述べる．

以下，2節では，関連研究について，3節ではソフトウェア開発における連絡の対象を述べ連絡モデルを提案し，4節でエージェントを用いた連絡支援システムの仕組みについて，5節で連絡の実行例について述べる．

2 関連研究

オフィス等における一般的なコミュニケーションの支援を目的とした研究は多くなされてきた．Information Lens[4] や ObjectLens[3] では．メッセージを構造化し，ルールベースでのメッセージフィルタリングを可能にしている．また Coordinator では対話の流れに注目し，会話の状態遷移モデル [7] に基づいてメッセージのやりとりを行なう．さらに PilotMail[2] では，この状態遷移モデルに基づいた会話の支援に加え，発信者がメッセージに手続きを付加して受信者に送信することで，受信者の行動を発信者が管理できる．これらのグループウェアは，いわば汎用のコミュニケーションの支援を目的としたもので，ソフトウェア開発に対する支援効果は十分とはいえない．

一方，ソフトウェア開発における利用を目的としたものでは，D^2[5] やめだか [6] などがある．D^2では，エージェントを用いてメッセージのフィルタリングやルーティングを行なうことで，作成プロダクトの収集支援を行なっている．また，めだかでは，フィルタリングに加え回覧機能や期限管理，発信者が着信や開封の履歴を参照できるトレース機能などにより，電子メールの欠点を補っている．これらのアプローチは，ソフトウェア開発における連絡手段として電子メールを用いた際に発生する問題点に対する回答ではあるが，ソフトウェア開発に特化したものではない．

3 ソフトウェア開発における連絡モデル

一般にコミュニケーションとは人間同士のやりとりを指す．しかし分野をソフトウェア開発作業に限定した場合，生成物やドキュメントに対する問い合わせや仕事の依頼，進捗の問い合わせなど，話題の対象によってコミュニケーションを分類することが可能となる．そこで，まずソフトウェア開発作業を表すモデルを検討し，そのモデルに基づいたコミュニケーション支援を考える．

3.1 プロセスモデル

ここでは，開発作業とは作業者があるプロダクトを元にあるプロダクトを生成することの連接であると考える．したがって，ここではソフトウェア開発プロセスをプロダクト，プロセス，要員の3つの要素によって記述する．プロセスは複数のプロダクトを入力とし複数のプロセスを出力する．また，一つのプロセスには少なくとも一人の作業者が割り当てられる．つまり，本稿ではソフトウェアプロセスを以下のようにモデル化する．プロセス P は { 関連プロダクト D, 内部プロセス S，開発要員 M，入出力関係 R，割当 A} の5字組で表される．さらに，D は { 入力 i, 出力 o, 中間 t} に，S は { 基本作業 a, プロセス p} に分類され，階層的に定義される．また，R とは D と S の要素間の関係，A とは M と S の要素間の関係である．

図 1は，要求定義書から設計書とレビュー結果を出力するプロセスの例である．図の楕円はプロダクトを，四角はプロセスを表している．また，開発者 M_0はこのプロセス全体を，M_1は設計および再設計プロセスを，M_2はレビューを担当していることを表す．また，要求仕様は設計プロセスの入力プロダクト，設計書は設計プロセスの出力プロダクトである．

3.2 連絡の種類と対象

次に，このようにモデル化されたソフトウェア開発プロセスで行なわれるコミュニケーションについて検討する．

ソフトウェア開発は，複数人による協調作業であるため，必然的に問い合わせや意思決定などといった連絡がなされる．経路の種類としては問い合わせ，意思決定，作

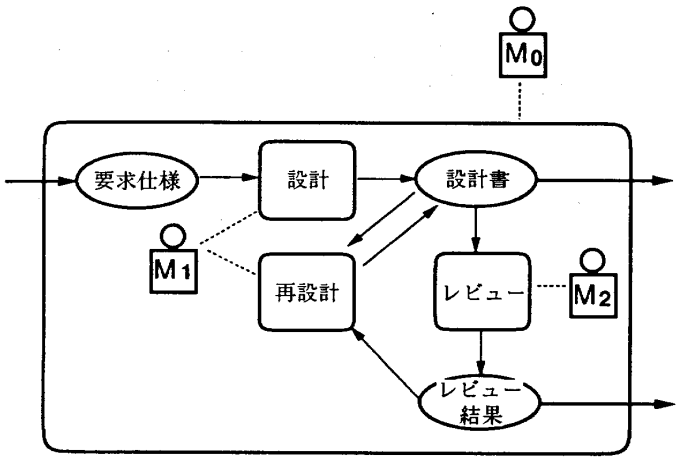

図 1: 開発プロセスの例

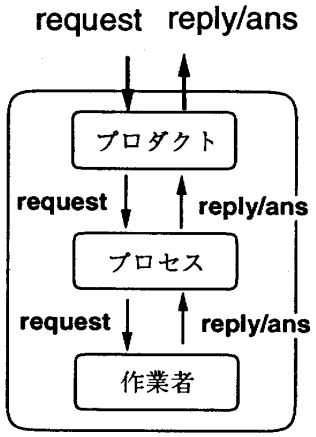

図 2: 連絡モデル

業の依頼や指示，作業に付随した連絡事項などが考えられる．

次にコミュニケーションの対象について述べる．今日の複数人による協調作業では，作業者がチーム単位で地域的に分散していることが少なくない．そのため，各作業者にとっては，自分の担当しているプロセスやその入出力プロダクトは何か，同じチームの作業者は誰かといったこと以外の情報を得ることは難しいと考えられる．そこで，連絡をとる対象には自分の担当しているプロセスの入出力プロダクトやプロセス自身の構成要素，他の作業者などに限定する．つまり，連絡の相手として，人間のみではなく，プロセスやプロダクトも許すものとする．外部の要素に対する連絡はこれらの要素が「取り継ぎ」を行なうものと考える．

3.3 連絡モデル

ここでは，以上の連絡の種類と対象に基づいてソフトウェア開発における連絡をモデル化する．

このモデルでは，ソフトウェア開発プロセスの構成要素に対応して，上から順にプロダクト層，プロセス層，作業者層という階層構造を持つと考える．

プロダクト層は成果物に関する情報を持つ．例えばその成果物を作成した作業者や現在の所有者，そのプロダクトを作成したプロセスなどである．プロダクト層は，この層の持つ情報で解決できる要求に対しては適当な処理を行ない返答を返す．解決できない要求に対しては，プロセス層に対して要求を出し，その結果を受け取り返答する．

プロセス層は開発プロセスに関する情報を持つ．例えば，入力プロダクトや出力プロダクト，担当者などである．プロセス層も同様にこれらの情報で解決できるのであれば所定の処理を行なった後返答する．解決できない場合は作業者層に対して要求を出しその結果を受け取って返答する．

作業者層は作業者に関する情報を持つ．例えば作業者のフルネームやメールアドレス，上司などである．作業者層の持つ情報で自動的に解決できない質問や要求は，作業者本人に転送してその返答を取り継ぐ．

ひとつの要求を処理するための連絡は，ある作業者がこれら３層のいずれかに対して要求メッセージを送信することで開始される．この要求は，送信されたメッセージに応じてこれらの層の間で作業の依頼と実際の作業，返答を繰り返すことで達成され，最後に，送信した層から要求者に返答が返ってくることでその連絡は終了する．

4 連絡の仕組み

本稿では，前節で述べたモデルに基づいた連絡を実現するために，エージェントの概念を用いる．作業者が従来行なっていた仕事の一部をエージェント自体が持つ情報を利用して一部自動化がはかれる．また，作業者が特に興味のある情報，ない情報の種類をエージェントに指示することで，メッセージのフィルタリングや整理が容易に行なわれる．

4.1 エージェントとは

ここではエージェントを人間の代わりに仕事（サービス）を行なってくれる仕組みと定義する．ここでいうサービスとは，3 章で述べた，要求や質問の処理を指す．さらに，エージェントはサービスを実行する際に他のエージェントと協力したり，仕事を他のエージェントに依頼するといった特徴を持つ [5]．この特徴は，連絡モデルにおける各階層が他の階層に要求を出してその結果を返答する動作に相当する．

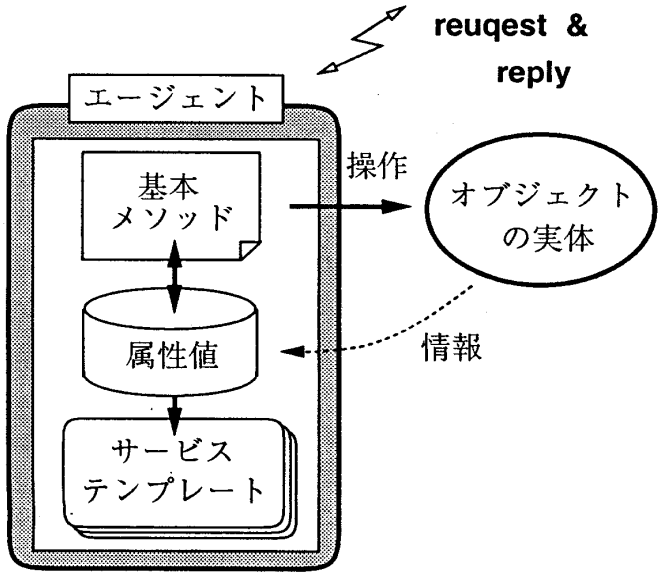

図 3: エージェントの概念図

各エージェントはそれぞれ一つのオブジェクト（プロダクト，プロセス，作業者）に付随して存在し，サービスの実行のために属性，基本メソッド，サービステンプレートを持つ．属性はエージェントが受け持つオブジェクトに関する基本知識の集合である．エージェントはこの情報に基づいてサービスの実行や要求の取り継ぎを行なう．基本メソッドはエージェントが実オブジェクトに対して行なうことのできる基本操作の集合で，そのオブジェクト内で処理が完了できるものである．これに対し，サービスとは，他のエージェントとの協調によって達成されるものである．したがってサービスは，その状態遷移規則を記述したテンプレートに従って実行される．サービステンプレートはサービスの種類によって異なるため，種類ごとにあらかじめ記述しておく必要がある．

4.2 各エージェントの特徴

エージェント持つ属性やメソッド，行なうサービスはエージェントの種類によって以下のように異なる．

プロダクトエージェント

属性： プロダクトを管理するための情報．例えば，プロダクトの所有者，作成者，パス，状態，作成されるプロセスなど．

基本メソッド： プロダクトに対する基本操作を提供する．例えば初期化や編集，表示，印刷など．プロダクトがソースファイルの場合はコンパイルや文法チェッカといった操作もメソッドに加えられる．このようにプロダクトの種類によって用意されるべきメソッドは異なってくる．

サービス： プロダクトの作成や変更要求，プロダクトに関する質問など．

プロセスエージェント

属性： プロセスとプロダクト，作業者間の関係を表現する情報（入力プロダクトや出力プロダクト，プロセスを担当している作業者など）を保持している．

基本メソッド： 作業の実行や状態など，作業に対する手続きを提供する．

サービス： プロセスが定義している作業手順の実行や作業の進捗報告，プロセスに関する問い合わせなど．

作業者エージェント

属性： その作業者の個人情報（例えば，開発者名，メールアドレス，所属，直接の上司，現在担当しているプロセスなど）

基本メソッド： 開発者に対して連絡する手段や作業者の状態の問い合わせを行なう方法．例えば，開発者への通知や質問の送信，現在の仕事の一覧など．

サービス： 作業要求や回答依頼，連絡の通知など．

4.3 エージェントの動作

エージェントは，新たな要求が到着すると，メッセージの種類に応じてサービステンプレートからサービスインスタンスを生成する．このインスタンスは個々の要求

に対応して生成され，各要求の状態を保持するために用いられる．具体的な動作は以下のようになる．

1. エージェントにメッセージが到着する．
2. メッセージにリクエスト ID(サービスを区別するための識別子) が付加されていない場合，作成する．
3. リクエスト ID に対応したサービスインスタンスが存在しない場合，メッセージの種類に応じてサービステンプレートからサービスインスタンスを生成する．サービスインスタンスはメッセージ内に記述されているリクエスト ID によって識別される．
4. サービスインスタンスに保持されている状態を参照し，サービステンプレートにしたがってメッセージを送信する．送信が新たな要求である場合はリクエスト ID を生成し，メッセージに付加する．要求に対する返答である場合は，その要求のリクエスト ID をメッセージに付加し返答する．
5. サービスが終了するとインスタンスは消去される．

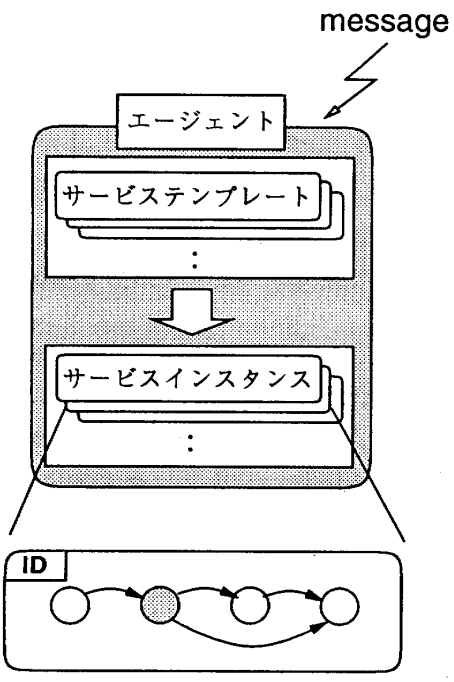

図 4: エージェントの動作図

4.4 メッセージのやりとり

エージェントは他のエージェントやその実体である人間とメッセージのやりとりを行なったり，開発ツールなどの外部システムの起動を行なったりする．したがって，メッセージのやりとりには以下の 3 通りが考えられる．

- 作業者 ↔ エージェント間
- エージェント ↔ エージェント間
- エージェント ↔ 外部システム間

作業者 ↔ エージェント間のメッセージのやりとりには電子メールを用いる．ここでは，エージェントがメッセージの種類や内容を理解できるような機構が必要であることから，メッセージは構造化されている必要がある．ここでいう構造化されたメッセージとは，Object Lens などで用いられているように，いくつかのフィールドに分割して，型付けを行なったものである．エージェントはメッセージの型によってサービスを特定し，フィールド値を参照することで人間の要求を処理し，返答を行なう．エージェント間のメッセージのやりとり，およびエージェントと外部システムのやりとりにはソケットやシグナル等の専用の機構を用いる．

5 記述・動作例

本節では，図 1のプロセスを用いて動作例を記述する．例題のプロセスにおいて，設計書の作成を行なう場合を考える．

同じ「設計書を作成する」という作業でも，次の 3 通りの仕方で達成することができる．

- 設計プロセスに対し，作業の開始を要求する．
- 設計書に対し，作成要求を行なう．
- 作業者に対し，設計を行なうよう要求する．

我々のモデルでは，上のどの要求形式をも許容する．以上の要求に対し，各エージェントの動作を順に説明する．

設計プロセスに対する要求の処理例

ここでは，設計プロセスに作業の開始要求が到着した場合について説明する．設計プロセスに対し開始要求を行なうと，メッセージは設計担当者に対して送信され作業依頼の回答を求める．回答の結果，作業を行なう場合

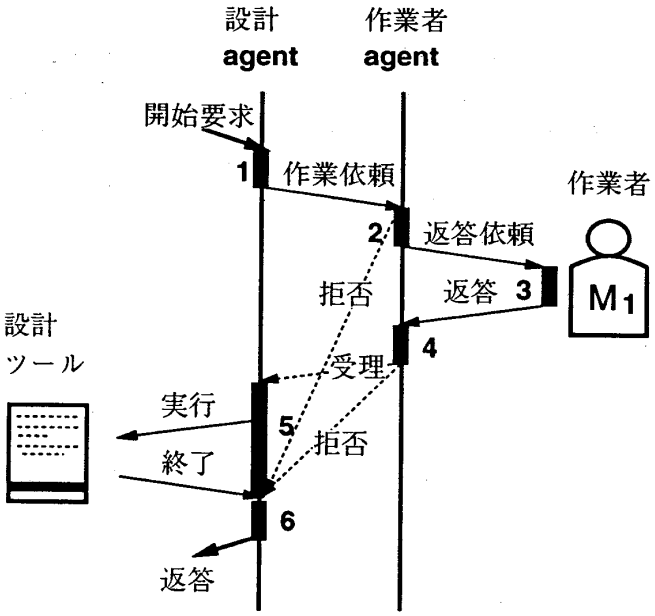

図 5: 設計エージェントおよび作業者エージェントのダイヤグラム

は設計ツールを起動し，作業の終了と作業結果が報告される．また，設計担当者が多忙な場合は，自動的に作業の拒否を要求者に返答する．これらの処理は以下の手順で行なわれる．

1. 設計プロセスエージェントに作業の開始要求が到着すると，まずサービスタイプ（ここではプロセスの開始要求）に応じてサービスインスタンスが生成される．また，エージェントの属性であるプロセスの作業者を参照して，作業者は M_1 であることがわかる．設計プロセスエージェントは，サービステンプレートの記述に基づき新たにリクエスト ID を作成し，作業者 M_1 のエージェントに対し作業の依頼をする．

2. 作業者 M_1 のエージェントは，まず属性の一つである担当中プロセスを参照し，現在 M_1 が担当しているプロセスの多重度を調べる．多重度がある一定の閾値を越える場合は依頼に対して拒否の返答をする．閾値を越えない場合は作業依頼に対する返答を自動的には拒否できないので，リクエスト ID を作成し，承諾/拒否の判断を作業者 M_1 に依頼する．この依頼は電子メールにて行なわれる．

3. M_1 は，到着した電子メールに対して作業を受理するのか拒否するのかを書き込み，返信する．

4. 作業者 M_1 のエージェントに M_1 の実体からの返答が返ってきたら，その内容を設計プロセスエージェントに返答する．

5. 返答が受理の場合，設計ツールを起動し，作業者に対し作業を誘導する．

6. ツールの実行が終ると，作業終了と共に作業の成功／失敗が返ってくる．設計プロセスエージェントは作業の要求者に対して作業の成功／失敗を返答として返す．また，5において返答が拒否の場合，拒否されたことを返す．

設計書に対する要求の処理例

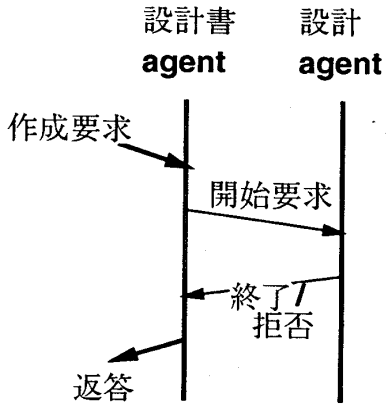

図 6: 設計書エージェントのダイヤグラム

次に設計書エージェントに対し，作成要求が来た場合を考える．設計エージェントに作成要求が到着すると，まずサービスインスタンスが生成される．次に，設計書エージェントの属性値: “作成プロセス名” から設計書は設計プロセスによって作成されることがわかる．そこで，設計書エージェントは設計プロセスに対し作業の開始要求を行なう．開始要求を受けた設計エージェントは，先の例と同様の処理を行ない，作業を開始しようとする．作業が行なわれた場合は作業終了と作業の成功／失敗が，作業が行なわれなかった場合は要求が拒否されたことが設計書エージェントに返される．設計書エージェントは設計エージェントから返答を受け取ると依頼者に返答する．

作業者に対する要求の処理例

最後に，作業者に対して作業開始の要求が来た場合を考える．作業者エージェントに作業の開始要求(図 7)が到着すると，まずサービスインスタンスが生成される．作業者は複数のプロセスを担当しているが，作業を行なうプロセスはメッセージ内のあるフィールド（作業プロセスフィールド）によって指示される．そこで，作業者M_1のエージェントは設計プロセスに対し開始要求を送る．設計エージェントは開始要求を受け取った後，前述した動作をする．そして作業が行なわれた場合は作業終了と作業の成功/失敗を，作業が拒否された場合は要求の拒否を作業者エージェントに返す．作業者M_1のエージェントは設計エージェントから結果を受け取ると，依頼者に対し返答する．

メッセージタイプ：作業開始要求
主題：作業者は期限までに以下の作業を行なうこと．
作業者：M_1
作業プロセス：設計
期限：1994 年 1 月 20 日
備考：...

図 7: 作業開始の要求の例

6 まとめ

本稿ではソフトウェア開発に特化した連絡モデルを提案し，モデルに沿った連絡を行なう手法について述べた．プロダクト，プロセス，作業者の各オブジェクトを連絡の対象とすることで協調作業時や分散環境下における知識の共有度を緩和させている．またエージェントを用いたことで，情報のフィルタリングやルーティング，作業の一部自動化，誘導などが可能となる．

現在，3節から 5節で述べた手法に基づいたシステムを作成している．また，プロセス記述の表現やより細かい対話構造の例の実現などは今後の課題である．

参考文献

[1] 青山 幹雄: 分散開発環境:新しい開発環境像を求めて, 情報処理, Vol.33, No.1, pp.2-13, (1992).

[2] 市村哲, 松下温: 発言と行動の管理に基づいた協調作業支援電子メール PilotMail, 情報処理学会論文誌, Vol.33, No.7, pp.955-963 (1992).

[3] Lai, K. and Malone, T.W.: Object Lens: A "Spreadsheet" for Cooperative Work, Proceedings of CSCW'88, pp.115-124 (1988).

[4] Malone, T.W. et al.: Semistructured Messages Are Surprisingly Useful for Computer-Supported Coordination, ACM Transactions on Office Information Systems, Vol.5, No.2, pp.115-131 (1987).

[5] 松尾朗, 貫井春美, 中村英夫: 電子メールにおけるエージェントシステム, 情報処理学会第 43 会全国大会予稿集, 2J-13, (1991).

[6] 垂水浩幸, 吉府研治: めだか：ソフトウェア開発向き電子メール基盤, 情報処理学会第 45 回全国大会予稿集, 1U-3, (1992).

[7] Winograd, T. and Flores, F.: Understanding Computers and Cognition, Addison-Wesley Publishing Company Inc. (1986).