

1 J-6 プロセスモデルに基づく分散開発支援システムの試作

¹大阪大学基礎工学部 ²日立西部ソフトウェア(株) ³(株)日立製作所

玉井昌朗¹ 岡田世志彦¹ 飯田元¹ 井上克郎¹ 鳥居宏次¹ 永岡渡² 梅本肇² 津田道夫³

1. まえがき

ソフトウェア開発の大規模化に伴って、複数地域で1つのシステムを分散して開発することが行なわれるようになるにつれ、それに付随した様々な問題が生じている。我々はこれまでに、ソフトウェア分散開発の実情、手法などを多くの人々に容易に伝えたり、開発の支援の枠組を得ることを目的として、ソフトウェア分散開発プロセスのモデル化を行なっている(1)(2)。このモデルは、分散開発における各地域の開発チーム内での作業に対するモデルと、各チーム間のコミュニケーションに対するモデルから構成されている。

本稿では、後者の各チーム間のコミュニケーションモデルに基づいた、分散開発支援のためのメールシステムについて述べる。

2. コミュニケーションのモデル化

実際の分散開発のプロジェクトに対する調査から、コミュニケーションに関して、以下のような問題が生じていることが分かった。これらは、遠隔地同士での開発のために会話等の簡便なコミュニケーション手段がとれないことから生じると考えられる。

- ・連絡事項が完了しないまま、放置される
- ・関連した連絡事項が分からなくなる
- ・連絡の経路が曖昧である
- ・チーム内で非公式な情報が数多く飛び交っている
- ・記録が不完全である
- ・チーム同士で体制が異なる
- ・チーム間の連絡を促進するような基盤技術がまだ十分でない

これらの問題点の改善に役立つよう、チーム間のコミュニケーションのモデル化を行なった。このモデルでは、コミュニケーションの経路を明確にし、チーム間の連絡を一括して行なうために、他チームとの連絡窓口(インタフェース)を設定する。インタフェース部分で、具体的に行なわれる仕事や、実際に伝達されるプロダクトを表1,2に示す。

2.1 用件に基づくモデル

チーム間での連絡の基本構成単位をメッセージと呼び、図1のように特定の目的を持った一連のメッセージの系列を用件と呼ぶこととする。本モデルでは、用件に対する処理という観点から、すべてのコミュニケーション活動を表現する。

例えば、何らかの要求として送られたメッセージ1に対して、受け取りの確認としてメッセージ2が、さらに、最終的な回答としてメッセージ3が返される。また、1と2の間では、受け取りの確認を催促するメッセージがあり、2と3の間には、仕事の催促や、仕事の途中経過の報告がある場合がある。これら一連のメッセージが一つの用件である。

用件は、その特性を表すための「属性」と、また、実行の進行を示す「状態」とを持つ。

それぞれの用件は図2に示すような状態遷移を持つ。このとき、ある用件に関連して派生する用件もあるため、用件は階層構造を持つ。このとき、もとの問題から派生した用件(子の用件)全てが完了すると、もとの用件(親の用件)も完了する。用件の属性としては表3のようなものが考えられる。

用件の生成や伝達先の決定、スケジューリングなどの処理は、このような用件の状態や属性に基づいて決定することができる。

表1 処理内容の例

用件の分析
用件に対する回答の作成
チーム内部での調査・作業
用件に関連する部署の決定
上位チームへの調査・作業依頼
下位チームへの調査・作業依頼
途中経過報告の作成
など

表2 伝達されるプロダクトの例

質問票
回答報告書
ソースコード
仕様書
ノウハウ
スケジュール
タスク割り当て
テストパッケージ
など

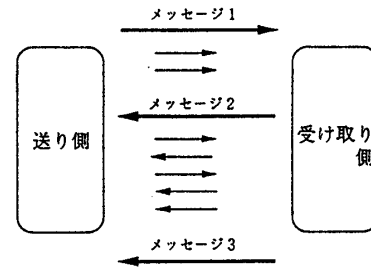


図1 用件を表すメッセージの集合

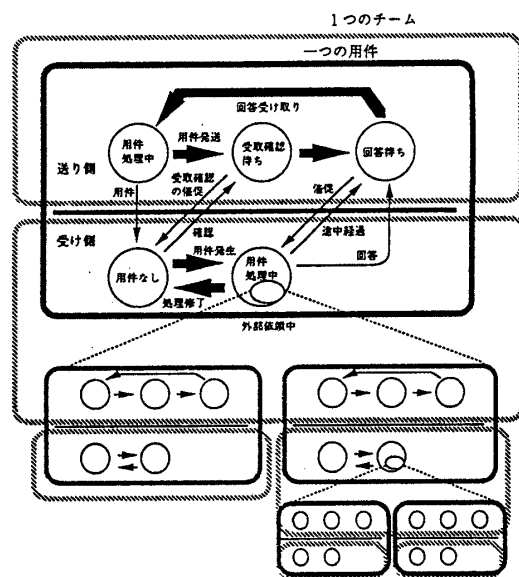


図2 用件に対する状態遷移

An E-mail System for Supporting Interaction among Distributed Software Development Teams
 Masaki TAMAI¹, Yoshihiko OKADA¹, Hajimu IIDA¹, Kazuro INOUE¹,
 Koji TORII¹, Wataru NAGAOKA², Hajimu UMEMOTO², and Michio TSUDA³
¹Faculty of Engineering Science, Osaka University.
²Hitachi Seibu Software Co., Ltd. ³Hitachi, Ltd.

type	要求・質問(返答を必要とするもの)や配布・通知(返答を必要としないもの)
content	ソース, ドキュメント, スケジュールなど
check	次のチームに無条件で渡すかどうか
from	差出人・差し出しチームなど
to	宛先
limit	期限, 緊急度など
importance	重要度
parent	親の用件
child	子の用件

表3 用件の持つ属性

3. メールシステムの概要

このモデルを基に, 分散開発を支援するためのメールシステムを試作した。このメールシステムでは, 階層化された複数チームにおいて(プロジェクトを分割して, モジュールを分担して)作業を行なう場合の連絡支援を前提としたものである。

3.1 システムの構成

図3のように, 各チームにコミュニケーションサーバを置き, チーム内外との連絡の窓口とする。コミュニケーションサーバは, 先に述べた用件の状態を基にメッセージの系列を管理することで, 用件の管理や処理を行なう。このために, サーバは, 組織テーブル, プロダクトテーブル, 処理ルールを持つ。

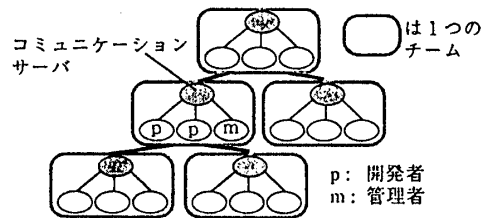


図3 システムの構成

(1) 組織テーブル

組織テーブルは, 各チームの構成を定義したもので, 親チーム, 子チーム, 役割, 自分のチームのメンバが書かれ, 用件の宛先を指定するのに用いられる。このテーブルを用いることにより, 部署による指定や機能/役による指定, グループ化されたメンバに対してなども用件を送ることなどが可能となる。

(2) プロダクトテーブル

プロダクトテーブルは, 各チームの扱うプロダクト(モジュール)を定義するもので, これを用いて, 関連のある成果物についての連絡の宛先を決定する。例えば, あるモジュールの仕様を変更する時などに, その関連モジュールがどのものかをプロダクトテーブルから判断し, その担当者の宛先を指定する。

(3) 処理ルール

処理ルールは各チームごとなどで作成される用件に対して, その発信元や種類などに応じて一定の処理方法を定めたものである。例えば, ルールとして以下のようなものが考えられる。

共通チームからの連絡 他のチームで用いられる共通部品を作成するチームからの連絡は全ての関連チームへ送る。

テストエラーの場合の発信 あるモジュールをテストしていて, エラーが起こった場合そのモジュールの担当者に対して連絡する。

版の更新 ある版を更新した場合に, 関連部署へ通知する。

これらのルールは, 用件の属性と, 要求の場合の操作, 配布の場合の操作, ブロードキャストの場合の操作などのプリミティブな操作を用いて定義される。

- ・プリミティブな操作には次のようなものがある。
- ・配布/要求それぞれに対するメッセージ集合のフロー操作
- ・宛先までの経路決定
- ・ブロードキャストの操作
- ・チェック要求機能の On, Off 時の操作
- ・期限/スケジューリングについての用件の操作
- ・チーム優先度に対する操作

処理ルールは, そのグループ内だけのローカルなもの, 全グループに適用されるものなどを定めることができる。また, ルール自体を一つの用件として配布することができる。

3.2 システムの動作

開発者はメニューの誘導によって作業を行なう。まず, 図6のように処理中の用件に関するメニューが表示される。そして選択された用件に対して行なうことができる作業のメニューが図7のように示される(実行可能な作業は, 用件の状態や前提条件からシステムが判断するので, これらの項目全てが常に表示されるわけではない)。

組織テーブル, プロダクトテーブル, 処理ルールは各チームの管理者が登録する。この場合も, 管理者はメニューによって, 必要な作業を選択することが可能である。管理者のメニューには, 図8のような項目がある。

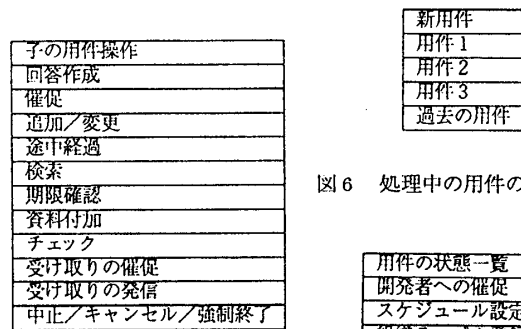


図7 用件に対する操作のメニュー

図6 処理中の用件のメニュー

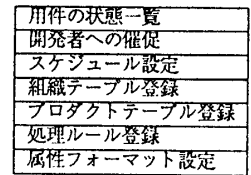


図8 管理作業のメニュー

4. あとがき

本システムでは, メール機能自体はUNIXのメールシステムを用いており, コミュニケーションサーバやXウィンドウ上で動作するインタフェースプログラムをそれに追加することで上記の機能を実現している。このシステムを使用することで, 用件の完了を明確にすることや, 連絡の経路を容易に管理すること, 用件の記録を残し, 関連するものを追跡することなどが行なえる。今後, 運用を通じた評価を行ない, 問題点の調査や改良などを加えていきたいと考えている。更に開発作業用のツールを連動させることも検討したい。

文 献

- (1) 岡田 他: 「Kellner のソフトウェアプロセス問題」の記述の試み, 情報処理学会第43回全国大会, 7J-1, (1991-10).
- (2) 岡田 他: 「ソフトウェアの分散開発のモデル化の試み」, 情報処理学会ソフトウェア工学研究会82-1, (1991-12-3).