

分散並行開発の管理支援環境

6 S - 2

青山 幹雄 小池 隆康
富士通株式会社

1. まえがき

ソフトウェア開発期間の短縮と急速なソフトウェアの成長に対応するために、ソフトウェアの分散並行開発が始まっている[1]。分散並行開発においては、特に、物理的に離れた開発拠点にまたがる開発管理が極めて困難、かつ重要な問題である。本稿では、分散並行開発の進捗と品質の管理方法、ならびに、ホストとワークステーション(W S)との連携により管理を支援する分散支援環境とその適用事例について報告する。

2. 分散並行開発の問題とアプローチ

2.1 大規模分散並行開発の問題点

大規模ソフトウェアの分散並行開発が出現した背景には次のような理由が挙げられる。

- (1)ワークステーションとネットワークの発達
- (2)開発組織の分散化：国内・国際分散開発
- (3)提供機能の多様化：地域に適應した製品開発
- (4)開発ライフサイクルの短縮

このような分散開発環境下でタイムリに開発管理を行うためには、次のような問題を解決する必要がある。

- (1)分散開発拠点間で統一した管理基準の策定
- (2)分散した管理情報の収集
- (3)各開発拠点・グループへのタイムリなフィードバック

特に、試験工程においては、進捗と品質の正確でタイムリな把握が納期と品質を確保するために必須である。従来は、メインフレームをホストとするリレーショナルDBをTSSにより利用する環境上で集中管理していた[2]。しかし、分散並行開発の進展により開発拠点内での分散管理と開発拠点間での情報共有を支援する新しい管理支援環境の実現が要求された。

2.2 解決へのアプローチ

次の4つのアプローチに基づき、分散環境における試験の管理支援環境TOPICS(Test Outcome Progress Information Control System)を開発した。

- (1)進捗・品質管理基準の統一
- (2)分散DBによる分散環境上でのデータ集計の自動化
- (3)グラフ・表などビジュアルな管理情報表現と出力自動化
- (4)結果のフィードバック機構の確立

A Management Support Environment for
Distributed Concurrent Development
Mikio AOYAMA and Takayasu KOIKE
Fujitsu Limited

3. 分散試験支援環境: TOPICS

TOPICSは、ホストとWS間の垂直分散とWS相互の水平分散により、分散開発拠点間での円滑な情報収集と管理や情報交換を支援している。

3.1 ホスト-WS垂直分散アーキテクチャ

図-1に、ホストとWS連携による進捗・品質管理情報の分散管理アーキテクチャを示す。各開発拠点内でも、さらに、グループ単位に管理DBを分散できる。各グループは進捗や品質情報をこのDBへ入力する。プロジェクトで共有する必要のある情報は、ネットワーク経由で分散DBからホストDBへアップロードされる。ホストでは、グループ毎あるいはプロジェクト全体の進捗・品質を集計し、保存する。各グループへフィードバックすべき情報はネットワーク経由で開発拠点の分散DBへダウンロードされる。ホストDBと分散拠点のDBは同一構造のRDBにより管理されている。

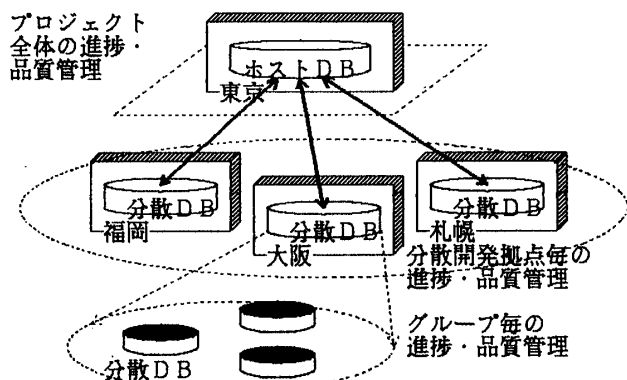


図-1 TOPICS垂直分散アーキテクチャ

3.2 WS間水平分散アーキテクチャ

図-2に示すように、各WS間でホストまたはWS間ネットワークを経由して管理情報の交換が可能である。

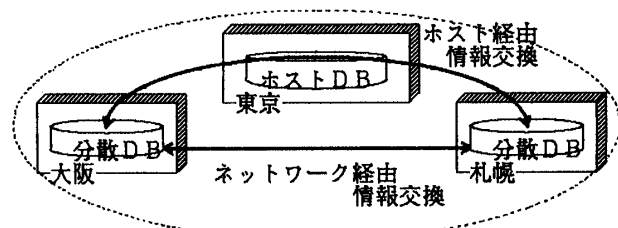


図-2 TOPICS水平分散アーキテクチャ

4. TOPICSの管理支援機能

開発管理を支援するには、マクロ管理機能とマイクロ管理機能の支援が必要である。例えば、システムやサブシステムなどを単位に、マクロに品質を把握する必要がある。一方、試験工程で発見した個々の問題の解析、処置状況も把握する必要がある。TOPICSは、進捗・品質管理においてそれぞれマクロ管理とマイクロ管理を支援している。また、進捗と品質を関係づけて管理する方法も支援している。

4.1 進捗管理

(1)マクロ進捗管理機能

システムやサブシステム単位での試験項目消化の予定と実績の推移をグラフ表示する。グラフには、過去の試験統計に基づく標準消化曲線も併せて表示する。

(2)マイクロ進捗管理機能

試験項目は、対象とするシステムや機能などの階層構造に対応して大項目、中項目、小項目などのように階層化されている。各階層毎に、その項目数と消化状況一覧表などを出力する。

4.2 品質管理

(1)マクロ品質管理機能

各グループが入力したバグデータに基づき、グループやシステム毎に、バグの発生、処置の推移をグラフ表示。グラフには、過去のバグ統計に基づく標準バグ抽出曲線も併せて表示する。

(2)マイクロ品質管理機能

問題処理票DBに基づき、バグ一件毎の内容、解析、処置状況の把握を支援する。例えば、発生問題一覧表、未処置問題一覧表などを出力する。ソースライブラリの版数管理システムと問題処理票DBの連携により修正済/未修正問題一覧表なども出力する[2]。

4.3 進捗管理と品質管理の融合

バグ発生は試験項目の進捗と密接な関係があるため、進捗と品質とを関係づけて把握する必要がある。このため、PB曲線[3]などを支援している。

5. 適用事例

本分散支援環境を、大規模通信ソフトウェアの結合試験とシステム試験に適用している。図-3に、運用プロセスの例を示す。各開発拠点、あるいは、開発拠点内の各グループ毎のローカルな管理とシステムやプロジェクト全体のグローバルな管理を行っている。それぞれの管理形態において、マクロな管理とマイクロな管理を使い分けている。

管理支援環境を運用する鍵は、無理のない情報収集にある。各グループは、『試験日報』の形で、試験結果を随時TOPICSに入力することにより、自然な形で進捗や品質情報を収集している。また、グループや開発拠点間で共有すべき情報は、自動的にホストDBへ転送される。

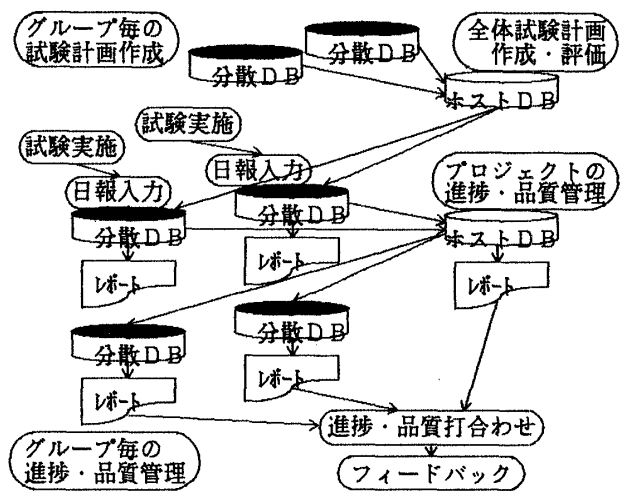


図-3 適用事例

6. 評価

TOPICSの適用により、次のような効果を得ている。

- (1)リアルタイムな進捗・品質管理の実現による納期と品質の確保：最新の正確な情報に基づく管理。
- (2)各分散拠点での情報ギャップの解消：同一情報に基づく進捗・品質管理や打合わせ。
- (3)標準化：分散拠点間で同一の基準に基づく管理。
- (4)計画に基づく試験の実施と適切なフィードバック：ダイナミックな試験プロセスの管理。
- (5)複合的管理：マイクロ管理とマクロ管理、あるいは、ローカル管理とグローバル管理の融合。
- (6)計画と予測：実績データによる後続試験の計画と予測。

7. 今後の課題

今後、適用範囲の拡大と他開発管理機能との連携の拡大のため、次のような機能拡張を行っている。

- (1)モジュール単体試験など前工程の支援。
- (2)開発プロセス管理環境との融合。

8. まとめ

分散並行開発の試験管理支援環境を報告した。このような環境は、今後ますます重要になると思われる。

謝辞：TOPICSの開発、適用にご支援いただいた当社通信ソフトウェア部門の多数の方々へ感謝致します。

9. 参考文献

- [1] 青山 "CASEと分散並行開発" 情報処理学会38回全国大会, No. 6L-5, 1989年3月, pp. 1189-1190.
- [2] 中沢, 青山 "交換ソフトウェアの障害管理" 昭和61年度電子通信学会全国大会, No.143, 1986年9月.
- [3] 安部 他 "総合デバッグ時のPB曲線の性質について" 情報処理学会論文誌, Vol.20, No.4, pp.306-313.