6L-5

ＣＡＳＥと分散並行開発

青　山　幹　雄
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、ソフトウェアの並行開発が始まっている[1, 2]。また、ソフトウェア開発組織の拡大とエンジニアリングワークステーションの発達に伴い、開発組織が分散される傾向にある。ＣＡＳＥ(Computer-Aided Software Engineering) はソフトウェア開発の生産性を飛躍的に向上させうる環境として注目されているが[3] 、現在のＣＡＳＥはこのような新しい開発パラダイムを支援していない。本稿では、ＣＡＳＥと分散並行開発について考察し今後のＣＡＳＥ像への一アプローチを述べる。

2．ソフトウェアの分散並行開発

2.1 なぜ分散開発か？

分散開発環境の出現には次のような理由が挙げられる。

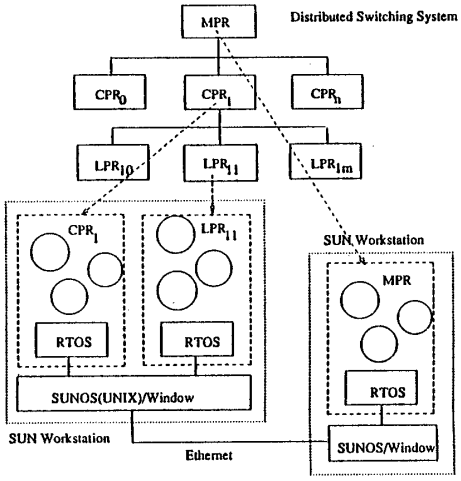
(1)ワークステーションとネットワークの発達
(2)開発組織の分散化：国内・国際分散開発
(3)提供機能の多様化：地域に適応した製品開発

これらの理由に加えて、分散処理アーキテクチャの自然なシミュレーション環境として分散開発環境を利用するアプローチがある。筆者らは、分散処理アーキテクチャに基づく通信ソフトウェアを検証する環境として、ワークステーションとＬＡＮを結合した分散開発環境上に分散処理システムのシミュレータを開発した [4]。図－１にシミュレータのアーキテクチャを示す。分散環境の提供する種々のツールを活用することにより、(1)短期間に、(2)分散処理システムの現実的なシュミレータの構築が可能となった。

2.2 なぜ並行開発か？

ソフトウェア並行開発の背景には次の２つがある [1]。

(1)ソフトウェア開発期間の短縮：ハードウェアは、パイプライン制御、並列処理により処理能力が飛躍的に向上したこの手法をソフトウェア開発に活かし、開発期間を短縮しタイムリな製品提供を行う。
(2)急速なソフトウェアの成長：ソフトウェアシステムへの要求は依然高く、筆者らが開発に関与したあるシステムでは、数年前に比べその成長速度は約３倍である。この急速な成長に対応するために、開発速度の向上が必須である。



図－１　分散シミュレータ

2.3 分散並行開発と分散処理システム

ソフトウェア分散並行開発は分散処理システムと多くの類似点がある。分散処理システムの挙動が従来の集中処理システムに比べ格段に複雑であるように、ソフトウェア分散並行開発を効率良く実現するためには分散処理システムと同様の問題を解決しなければならない。特に環境全体の資源管理は重要な課題である。例えば、同一のソースプログラムを複数のグループが同時に更新する問題は分散処理システムにおける資源アクセス制御と同一の問題である。

3．ＣＡＳＥと分散並行開発

3.1 分散並行開発におけるＣＡＳＥの問題

分散並行開発では、プロジェクト管理が複雑かつ重要な問題である。しかし、現在のＣＡＳＥではプロジェクト管理機能は弱い。また、現在のＣＡＳＥのほとんどが、パーソナルコンピュータ、または、メインフレーム上で動作するため、分散開発を支援するアーキテクチャを採用しているものは少ない。実際のソフトウェア並行開発の経験から次のような問題が分散並行開発を行う鍵となる。

(1)分散した開発グループ間・拠点間での開発プロセス管理
　①作業の割当て・進捗管理・標準化・教育等
　②コミュニケーション・協調・意思決定支援等
(2)分散されたソフトウェア開発資源・製品の管理
　①構成管理・資源管理等。特に、複数の開発グループに

CASE and Distributed Concurrent Development
Mikio AOYAMA
Fujitsu Limited

よる資源アクセスの問題は重要である。

②分散されたツールの連携：同種・異種ツール間の連携

3.2 分散並行開発の管理

分散並行開発を管理するためには、開発プロセス管理と資源・製品管理の両面からのアプローチが必要である。

(1)開発プロセス管理

開発プロセス管理の根本問題の一つは、開発プロセスを記述する手法が未熟である点にあろう。先駆例として、DeMarco[5]とOsterweil[6]が知られている。分散並行開発のモデルは、プロセスモデルに対する一般的要件に加えて、次のような要件を満たす必要がある。

①並行実行する開発プロセス間の同期・協調・資源アクセスの記述

②分散されたプロセスと資源の記述

③開発プロセスの構造や挙動の分析と評価

この要件を満たすため、Data Flow Diagram(DFD)による記述性とペトリネット(PN)の並行処理記述機能と解析性のよさとを統合したモデルを提案する[2]。図－2にアーキテクチャを示す。開発プロセスをDFDにより記述し、ペトリネットを拡張したEMPN(Extended Modified Petri Net)モデル[7]により分析・評価する。EMPNは次のような特徴を備えている。

①図形表現とテキスト表現を持つ

②オブジェクト指向に基づく階層的記述可能

③実行可能(Executable)

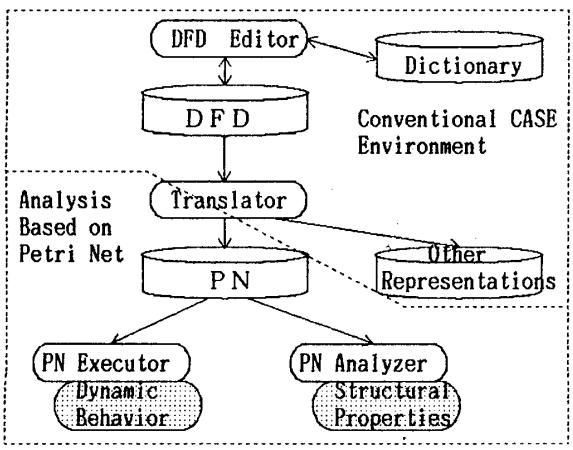
④形式的で理論的解析可能

⑤制御(プロセス)フローとデータフローの統合記述が可能

⑥時間（タイミング）記述可能

本手法では、ほとんどのＣＡＳＥが支援するDFDによりプロセスが記述できるのでＣＡＳＥとの親和性が高く、付加プロセサとして比較的容易に実現できる。

また、本手法は、分散並行開発プロセスのみならず、分散処理システムの設計にも適用できる。



図－2　プロセスの記述と分析

(2)資源・製品管理

開発環境は、運用環境とツール環境から成る2階層構造でモデル化できる。分散開発環境の発達と共に、運用環境が分散された資源を支援する環境が出現している。現在のＣＡＳＥでは、固有の単一データベース・データディクショナリ（ＤＢ・ＤＤ）を利用している。しかし、分散開発環境では、ＤＢ・ＤＤも分散される可能性がある。今後、分散環境上で運用環境が提供している機能を活用し分散された資源・製品を管理するための適切な機能分担を見出すことが、柔軟な環境構築の鍵となろう。また、並行開発を支援するため、資源のアクセス制御、版数管理、構成管理変更に伴う影響範囲の評価等[8, 9]を、開発プロセス管理と統合して管理する必要がある。

4．まとめ

今後のＣＡＳＥは、設計支援と管理支援を円滑に統合する必要がある。また、ソフトウェア開発の国際化に伴い、分散平行開発は重要な課題となろう[10]。分散平行開発は分散処理システムと多くのアナロジがあるので、分散処理システムで培われた技術を活かし、分散並行開発を支援するＣＡＳＥ環境の構築を検討する必要があろう。

謝辞：UICPBXの開発に共に日夜情熱を傾けたUniversity of Illinois のC. K. Chang 他多数の学生諸君に感謝します。また、本研究の遂行にあたり支援頂いた、当社通信ソフトウェア部門の関係各位に感謝します。

5．参考文献

[1] M. Aoyama, "Concurrent Development of Software Systems: A New Development Paradigm," ACM Software Eng. Notes, Vol. 12, No. 3, Jul. 1987, pp. 20-24.
[2] M. Aoyama, "CASE and Concurrent Development," IEEE 2nd Int'l Workshop on Computer-Aided Software Engineering, Jul. 1988, Boston, pp. 28-3 - 28-5.
[3] E. Chikofsky et al., "CASE: Reliability Engineering for Information Systems," IEEE Software, Vol. 5, No. 2, Mar. 1988, pp. 11-16.
[4] C. Chang, et al., "UICPBX: A Distributed Simulator of Switching Systems," SCS 1988 Summer Simulation Conference, Jul. 1988, Seattle, pp. 352-357.
[5] T. DeMarco "Controlling Software Projects," Yordon Press, New York, 1982.
[6] L. Osterweil, "Software Processes are Software Too," 9th ICSE, Mar. 1987, Monterey, pp. 2-13.
[7] C. Chang, et al., "A Specification Language for Real-Time Distributed Systems," IEEE Int'l Conf. Computer Languages, Oct. 1988, Miami, pp. 258-265.
[8] M. Aoyama et al., "An Integrated Software Maintenance Environment," IEEE Conf. on Software Maintenance, Oct. 1988, Phoenix, pp. 40-44.
[9] 宮崎　他，「モジュール影響評価システム REMIE」信学会通信部門全国大会, Sep. 1986, 東京, No.143.
[10] K. Nakamura, "A Practical Approach to Tele-Working Using Distributed Software Technology," Pacific Telecom. Conf., Jan. 1989, Hawaii.