

解説



分散開発環境

4. 分散開発環境の事例と今後の展望†

坂下 善彦†

1. はじめに

コンピュータ環境の分散化が著しく、かつその利用形態も多岐にわたっている。ソフトウェアの開発もこの分散化された環境で行われる。しかし「分散環境」という利用資源、利用者が物理的・地理的に分散された環境にある場合の開発手法、及び関連する研究課題については、その課題の存在を意識しながらも、これらに向けた研究開発が体系的に進んでいる状況でもない。

本特集では、分散開発に関連する諸課題を整理抽出し、その課題解決に向けたアプローチ、モデルを解説し、それらの研究開発を支える基盤技術とソフトウェア工学の技術をこれまでみてきた。本部では、これらの分散開発の事例と今後の展望に焦点を当てた解説を行う。

2. 分散開発環境の事例

分散開発環境の構築の事例を、その構築概念の特徴とアプローチを中心に紹介する。

2.1 初期の分散開発環境

分散型コンピュータシステム概念が生まれるころに、分散システムの構築に必要とされる新たな要素の形成や、プログラムの製作・開発の手法が提案された。これらの中で、その後の研究に大きな影響を与えたシステムをあげる。

(1) Spice-Project

カーネギメロン大学が分散型のコンピュータシステムの研究用環境として構築したシステムである。オペレーティングシステムに独自のものを開発し、プロセスの分散化を目指したカーネルを作成し、さらに分散型のファイルシステムを開発

している¹⁾。

1980年ごろにプロジェクトが始まり、先進的なアプローチであったが、当時では優れたワークステーションであった PERQ は、その後相対的な性能の低下のため現在では利用されていないようである。その後の分散型のコンピュータシステム環境の研究開発に大きな波紋を起こした。

(2) Andrew-system

パーソナルコンピューティングの機能と、それまでのタイムシェアリングの機能を融合させる目的をもっており、柔軟性に富み、優れたユーザインタフェースや通信機能を備え、情報の共有ができることを狙っている。VICE (Vast Integrated Communications Environment) と呼ぶ通信と資源を統合して提供する集合システム (ファイルシステム) を介し、OS の UNIX とエディタの EMACS を基本構成要素とする個々のワークステーション: VIRTUE が接続される。この VICE-VIRTUE インタフェースが、プログラミング・インタフェースになる。その後の改良はこの部分の上位互換を保持しながら行われた。また、ユーザプログラム以外はこの VICE の側で実行されるので、ユーザの都合によりワークステーションの側のソフトウェア及びハードウェアを変更することに対する影響はない。

Andrew システムの構成要素は、ワークステーション、OS; Berkeley UNIX, ネットワーク, VICE ファイルシステム, そして VIRTUE アプリケーション支援からなる。このシステムでは、その後の分散型システムでの課題となったあらゆる技術的なことが、研究開発されている²⁾。優れたユーザインタフェースを提供するために、独自のウィンドウ管理システムを備えている。

図-1 に Andrew のシステム構成を示す。VICE はユーザが交信を行う基幹となる通信及びコンピュータ資源を表す。そして VIRTUE は VICE

† Examples of Distributed Software Development Environment and Future Directions by Yoshihiko SAKASHITA (Mitsubishi Electric Corporation, Computer & Information Systems Laboratory). Internet: sakasita@isl.melco.co.jp

† 三菱電機(株)情報電子研究所

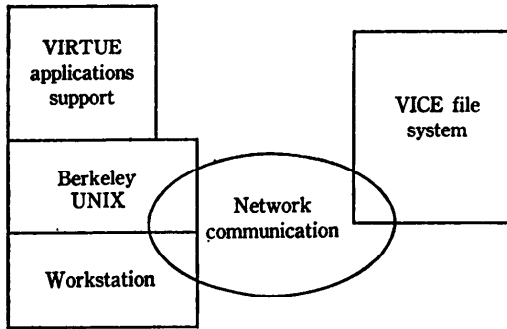


図-1 ANDREW のシステム構成 (文献 2) より引用)

に結合される高度に洗練化されたユーザインタフェースを備えたワークステーションを示す。ファイルシステムとユーザインタフェースとの間にはまったくハードウェア依存性がないことを特長としている。

(3) ソフトウェア構成管理ツール: DSEE

大規模なソフトウェアシステムの開発管理と膨大な量の生産物の管理を支援するものである。このシステムは、並行して行われる開発作業、新たにプログラミングを行いドキュメントを作成しながら既存の版の保守を行う作業、プログラムを構築するのに要する時間の削減、そしてロードモジュールを作成するためのツールと個々のプログラムのどの版を使うのかを支援する³⁾。その主なサブシステムは以下のものである。

- 履歴管理
- 上流工程から下流工程までの計画と作業の管理
- たとえば、一部が変更されることにより影響を受ける部分の担当者への警告の類の監視
- 構成管理
- 必須、オプションのものなどの出荷管理

図-2 に DSEE におけるモジュールの展開例を示す。図中、太実線が主流で、実線が分岐を表し、波線が結合された状況を示している。

2.2 プロセスプログラムによる分散開発環境

主として大規模ソフトウェアの開発手法の分野から発生してきたプロセスプログラムの概念であるが、そ

れが分散化された開発環境においても、より高度にシステムの表現を行うことを目的とした新しい開発の手法が出現している。

(1) Arcadia

大規模ソフトウェアの開発を支援するためのソフトウェアシステムの構造とプロセスプログラミングの実験環境となるシステムである。このシステムは、環境アーキテクチャの原則の発見と開発、及び新しいソフトウェアツールの創造を目的にしている。

ソフトウェア開発環境に関する研究は、特定のツール、プログラミング言語、データベース、OS に集中した段階、ソフトウェア開発環境の概念の再定義の段階、そして環境のアーキテクチャの概念を描写する段階、と進んできている。

Arcadia⁴⁾ の、環境アーキテクチャの構成要素としては、オブジェクト管理、ユーザインタフェース、そしてツール起動管理がある。Arcadia は環境アーキテクチャとソフトウェア開発支援ツールの二つの領域における問題点を明らかにするプロジェクトであり、ソフトウェア開発と保守のための自動支援を行う上での基本問題に焦点を合わせている。

既存のツールの断片とそれらが扱う抽象データ型についての全情報を中央のオブジェクト内に保持し、型構造とツール集合を変更することを、

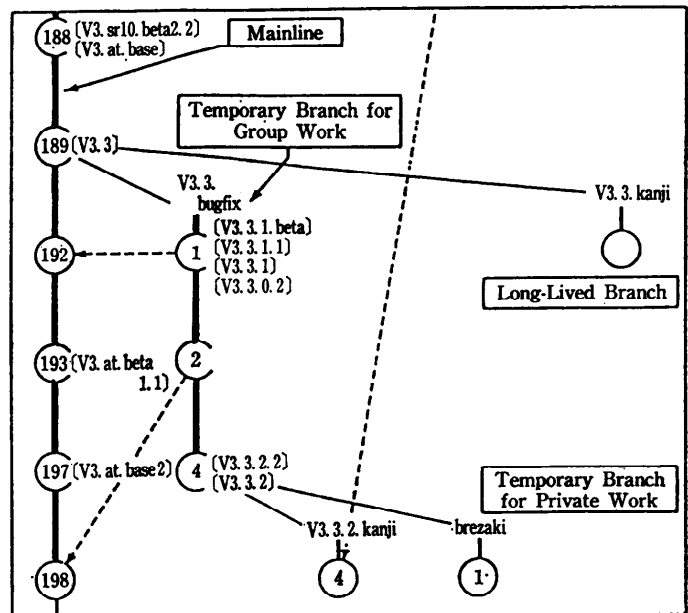


図-2 DSEE におけるモジュールの展開例 (文献 3) より引用)

Arcadia は支援する。新しい型を組み込むためのツールが必要で、そのツールは、型導出グラフ内で、新しいノードと入力するオブジェクトの型を表すノードとを接続するエッジの集合で表されている。

(2) 抽象化された表現モデル

プロセスプログラミングの概念は多く提案されているが、本モデルは、ソフトウェアシステム表現を文字及び図式表現により高度に抽象化されたレベルで記述し、開発プロセスを表現している⁶⁾。ここでは、今後の開発プロセスに関する新しいパラダイムを提供している。

ソフトウェア開発プロセスにおける基本オブジェクトには、アクティビティ・オブジェクト、ソフトウェア・オブジェクト、環境・オブジェクト、そしてシナリオ・オブジェクトがある。そのクラス・オブジェクトとしてジェネリック・オブジェクトの概念を導入している。プロセスプログラミングの概念に基づき、オブジェクト指向型及び木構造型のプロセス記述言語、さらにペトリネットに基づく振舞いを記述する言語を提案している。

ソフトウェア開発プロセスのみではなく、プロセス設計のプロセスをも、ソフトウェア開発環境は支援すべきとしている。

2.3 グループウェアによる分散開発環境

ソフトウェア開発における分担作業の結合段階では、種々の調整・協調の作業が発生する。分散化された環境では、複数の人が互いに相談したり助け合いながら、全体として一つの処理を実行するような支援環境が有効である。ここでは、個人あるいはグループの間での協調を支援する機構をグループウェアと呼び、この機構に基づく開発環境の例をあげる。

(1) 並列協調処理モデル: Cellula

協調しながら全体として一つの処理を実行する計算モデルを協調処理モデルと呼んでいる。各人は自分に関係することを原則としてすべて自分で処理する自律的存在であり、管理者や指揮者のような中央集権的な存在ではない。したがって、このモデルでは、実行主体の自律性と分散性の視点からみると、オブジェクト指向や分散処理との親和性が高い⁶⁾。

高度知識処理システムの基盤と考えられる Blackboard Model では、共通の場としての Blackboard と、それぞれの実行主体が存在する。ここで、通信機能を考えた場合、情報の衝突や不測の書換えの回避が困難、及び情報が唯一の場に存在するため情報の動的検索でのオーバーヘッドが大きい、という難点がある。

実行主体であるプロセスと通信媒体としての場を一体化し、この単位を階層的に構成する。通信形式は、通信命令の種別ではなく通信情報の属性で指定する。このような統一的な機構で、セルの階層とタプルの属性の組合せで、図-3 に示すようなさまざまな通信形態を実現している。

(2) ICICLE (Intelligent Code Inspection Environment in a C Language Environment)

C言語で記述されたソフトウェアの、複数人で行うコードレベルでの検査をサポートすることを目的としている。司会者、コードを書いた人、コードを説明する人、記録・整理する人、そして議論・コメントする人、などの役割を参加者はもつ。定型的な誤りは ICICLE が検出するため、設計仕様が実現されているかの検討にのみ利用者は集中できる。X-window 上に構築されており、制御エリア、コードエリア、そして注釈エリアからなり、ライン番号の横にあるシンボルがコメントの存在を示している。そのシンボルをマウスでクリックすると別のウィンドウが現れ、コメントの参照、コメントの記述ができる⁷⁾。

(3) IBIS による設計過程情報の獲得

大規模で複雑なコンピュータシステムの開発過程で、「何がどうして」という理由を把握する必要性が認識されつつある。開発過程で何度も開発方針を変えるわけにはいかない。開発者による理由付けや計画が理解されないまま、そのコードを

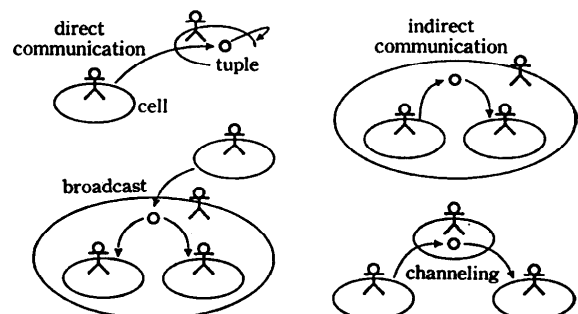


図-3 さまざまな通信形態 (文献 6) より引用)

変更するわけにはいかない。したがって、できるだけ早期に誤りや誤解を感知することを十分に行わねばならない。

大きなプロジェクトでは、膨大な量の情報を共有せねばならないこと、そしてそれらが構造化されていないために、判断の材料にするための明確化や他の部門でも利用できるようにする苦勞のコストはきわめて大きい。このために、これらの情報をデータベース化し効率良く検索せねばならない。

本プロジェクトでは、共同作業支援：CSCWの技術を利用して、設計の決定過程の会話や情報の流れを把握するために、情報の獲得に焦点を当てている。この目的のために IBIS: Issue-Based Information System を利用して、実際の開発フィールドからの情報の獲得に成功している⁹⁾。

(4) ソフトウェア開発における協調作業支援
ソフトウェア開発において、メンバが互いに相談したり助け合いながら、全体として処理を実行する協調作業支援の手法である。ソフトウェアの成果物の構成要素と、それらの関係情報をオブジェクト指向型のデータベースに格納する。仕様、記述内容などの変更などにもとない、関連する成果物の内容を変更する際に、その製作者間で協議の必要が生じる可能性がある。その場合の協調作業を支援するコミュニケーションモデルを備えている特長がある^{9),10)}。この支援ツールの構成を図-4に示す。

(5) ハイパ機構による情報の共有

分散開発環境での、メンバ同士の継ぎ目のないコミュニケーション、情報の共有構造そしてグループでの実行・試験が重要である。ハイパテキストを基盤とし、開発過程を計画・設計、作成、統合の三つに分割し、その各段階でのコミュニケーションの形態を定め、同期・非同期通信を継ぎ目なく実行するシステムにおける情報の共有を実現している。

ハイパテキスト・データモデルにより、ソフトウェアモジュール、ドキュメント類の分散管理を行い、さらに、ハイパリンクによりそれらを接続し、全体として一つのソフトウェアシステムとして統一的に管理している^{11),12)}。このシステムにおける情報構造とモジュールの統合の様子を図-5に示す。

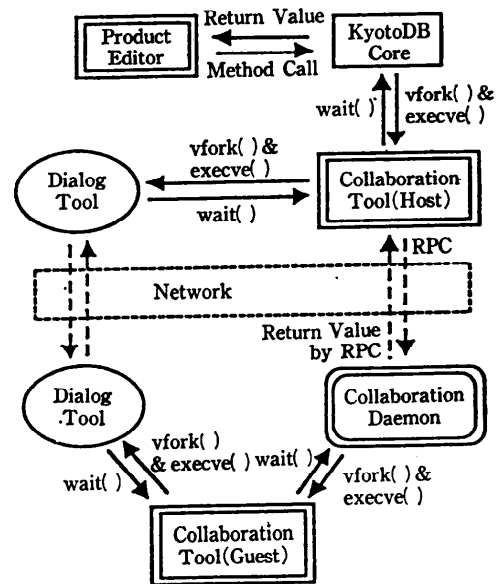


図-4 協調活動支援ツールの構成 (文献 9) より引用)

3. 今後の動向

ソフトウェア開発環境を構築する方法には数多くのものが考えられてきている¹³⁾。UNIXに代表されるように、使いやすいオペレーティングシステム上に、ソフトウェアツールを整備しておき、かつツール間をやり取りするデータは全てファイルとして扱うという開発環境である。

共通のデータ構造とデータベースを設け、ツール間の関係を協調させる統合的環境を目指す開発環境である。最近では、ソフトウェア開発のプロセスをモデル化し、それに対応する開発環境を提供するプロセス指向型の開発環境が盛んに研究されている。

3.1 技術動向

3.1.1 分散コンピューティングの動向

LAN, WAN に代表されるネットワークの普及は目ざましい。より広いレベルの機能・性能や特徴をネットワークを介して利用することを利用者は期待している。そのために、異機種のコピュータシステムがネットワークに結合された環境において、互いにその環境にある資源を利用し合う相互運用の実現が求められている。

ソフトウェア開発者は全てのシステムとその資源が広い範囲にわたって、利用できる環境を必要とする。すなわち、情報を初めとする環境資源が

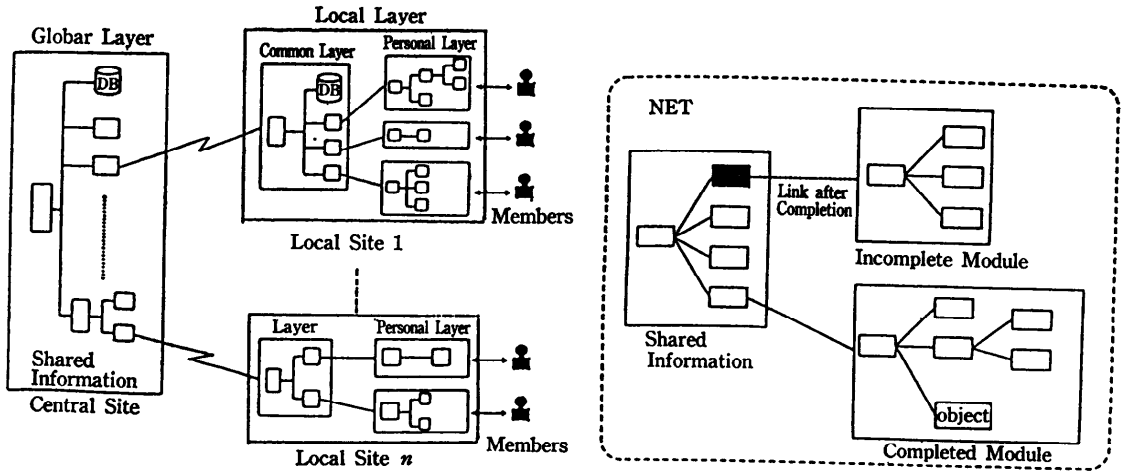


図-5 情報構造とモジュールの統合 (文献 12) より引用)

どこにあるかは独立に操作できることが、基本的な環境である。これらの情報は、利用者が必要とするものの所在場所を利用者が意識せずに、自身のワークステーションの利用可能な情報と同様に利用できなければならない。

ここでは、分散コンピューティング環境を基にした動向¹⁴⁾を紹介する。基本的な分散サービスとして、

- リモートプロシジャコール
- ネーミング、セキュリティサービス
- タイムサービス

があげられ、データ共有サービスとして、

- 分散ファイルシステム
- ディスクレスサポート

などが、あげられる。

(1) リモートプロシジャコール

アプリケーションが用いる個々のプロシジャをネットワーク上のどこかにある他のコンピュータで実行させる、というきわめて簡単な原則である。プロシジャコールモデルをリモートシステム上のプロシジャを直接呼び出すことにより、容易に分散アプリケーションを開発できる。

(2) ネーミング

分散環境全体の中で単一の名前を付けるモデルを提供し、データ、ファイル、機器、などの資源を名前で識別し、それらの資源がネットワーク上のどこに実際に存在するかを意識せずにアクセスできる。これは、機密保護の機能すなわちセキュリティの機能と深く関係する。

(3) タイムサービス

アプリケーションは、その動作をスケジュールし、イベントの順番・接続時間を決定するために単一の時間を参照する。分散アプリケーションのおおのの部分では個別のコンピュータシステム上の時計から時間を得る。このために、コンピュータネットワーク内でシステムクロックを調整し、お互いに密接にし、個々の分散アプリケーションに対して正確な時間を提供する。

(4) スレッドサービス

プログラマは分散環境のコンピュータ能力と複数存在する並行性を利用することを当然ながら期待している。すなわち、アプリケーションは複数の処理を同時に実行するために、一つのスレッドがリモートプロシジャを実行している間にもう一つのスレッドは利用者の入力を処理することができる。

(5) セキュリティサービス

分散サービスとデータ共有に関係したセキュリティサービスを提供するために、認証と利用権と利用者のアカウント管理の三つのサービスを備えるネットワークを提供する。

(6) 分散ファイルシステム

個々のワークステーションのファイルシステムを結合し、一貫したインタフェースを提供することにより、この分散ファイルシステムを用いて、ローカルアクセスと同じようにファイルにアクセスする。

3.1.2 グループウェアなどのソフトウェア 工学的アプローチの動向

コンピュータによる共同作業支援の、ソフトウェア開発への適用は、その開発工程の一部にすぎない。ソフトウェア開発の上流から下流にいたるライフサイクル全体を対象とした体系的な研究はこれからである。しかし、すでにみてきたように、電子メールを利用した非同期型の会議システム、共有ウィンドウを用いた同期型の会議システム、そしてかなり制約は強いが協議・協調の場面がある議論の制御を支援するシステム、などが部分的に適用されている。さらには、この共同作業の重要な焦点である協議・協調をモデル化する積極的な研究が行われている¹⁵⁾。

特に、複数人で行われる大規模なソフトウェア開発におけるこの性質と機能の抽出とその機構の構造化は大きな課題であり、これからの研究が期待される場所でもある。他方、ソフトウェア開発環境の研究は、環境アーキテクチャの概念描写の時期にある。この環境アーキテクチャでは、プロセス、オブジェクト、インタフェースを扱っている。プロセス指向型のアプローチが今後増える傾向にある中で、CASE 環境との連携を保ちながら体系的な対処が必要になる。

3.1.3 分散並行開発

開発拠点の分散化、地方における人材の活用、などの現実面への対処の必要性から、企業にとってもこの課題は急を要する課題でもある。個々のソフトウェア開発において、複数の開発担当者がそれぞれの部分を担当し、さらに物理的に分散した環境で、内容の一貫性を保持させて成果物を作成させるには、作業の枠組みを具体的なものにし、予想されるさまざまな規則、指示、協調、などの作業をその枠組みに従ったものとさせる方式が必要になる。

物理的、地理的に分離・分散された開発の拠点において、時間的に並行してソフトウェア開発を行う状況を分散並行開発と呼んでいる¹⁶⁾。そこでこの課題は以下のものがある。

- 開発拠点での一貫した開発基準の策定と運用
- 分散した開発資源と情報の管理
- 開発プロセスへのタイムリなフィードバック

3.2 今後の課題

ソフトウェア開発環境を論ずる場合、その定義は必ずしも一定ではないと思われる。ソフトウェア開発環境の整備には二つの目標がある¹⁷⁾。一つは、ソフトウェア開発の技術的支援であり、もう一つは、開発プロジェクトの管理の支援である。

この環境を構築するためのシステムの形態は、コンピュータシステムの形態の発展と相まって、大型集中方式から、ワークステーションを中心としたネットワーク型の分散方式へと、変化してきている。さらに、そこで処理の対象とする情報は、従来の文字のみによるものに加えて、画像・図形・音声などのマルチメディアと呼ばれる形態の情報が今後増大する。

3.2.1 大規模（広域）分散開発環境の

運用・管理

分散開発環境における運用・管理の情報は、対象システムや開発組織に依存するもの、及び部品あるいは同一の構造を備える成果物などに依存するものがある。前者は、グループごとの情報と開発拠点やプロジェクト全体の情報などが相当し、システムや組織の階層構造を反映した階層的情報構造を成す。後者は、同一水準にある開発拠点やグループ間に分散する共通の同一構造をもつ情報構造である。このような性質に対応するために、垂直分散型構造を支援する Client-Server モデルを適用し、ワークステーション間での情報交換を支援する水平分散型モデルを適用したシステム例がある¹⁸⁾。

3.2.2 ソフトウェア開発技術・形態への インパクト

ソフトウェアの開発・製作における、要求定義に関する重要性は大きい¹⁹⁾。この開発・製作活動をソフトウェアのライフサイクルとの関係で実施するための方法及び環境が整備されなければならない。他方、現在までに実現されているグループウェアそのものは、必ずしも成功はしていない。主な理由は、適用する組織のメンバそれぞれに効果や利益をもたらすものではなかったからである²⁰⁾。

しかし、ソフトウェア開発の現場では、メンバと必要に応じて対話しながらプロジェクトの遂行していく形態が重要な部分を占める。それがゆえに、共同作業の形態を取り得る職場として、違和

感はないと考えるからである。この場合、分散化が進めば進むほどコミュニケーションギャップは大きくなるので、このコミュニケーションを支援する仕組みが重要になる²¹⁾。

3.2.3 既存の開発支援環境の分散開発環境へのマイグレーション

主たる投資が人材であるソフトウェア産業においては、人的資源の点からソフトウェア生産拠点の地方分散化は今後ますます活発になる、との指摘がある¹²⁾。そこでは、上流工程は都市圏で下流工程は地方で、という分割統治方式の役割分担が当面続くが、高速ネットワークの実現とワークステーションの浸透により、既存のソフトウェア開発過程が分散協調型に変化していくと、予想している。さらに、その開発体制も中央の緩い調整の機関と、それぞれの特色を有する地方の開発拠点といった組織の分化が進む、としている。

ソフトウェアの分散開発の場合は、分散する対象の粒度や構造などに、より多くの選択肢があり、かつ流動的である。したがって、複雑な要因が多く、単純な分散化とはいけなくなる。ここにおいても、最大の課題は、人間同士のコミュニケーション手段がどれだけ開発環境にとけ込んだ形態で用意されているかにある。

4. おわりに

分散開発環境における課題は、ワークステーションに代表される情報処理技術とネットワークなどの通信技術を活用し、時間と物理的距離の制約を解き放つことにある。

同様な課題を、最近話題の多い共同作業支援あるいはグループウェアと呼ばれる分野でも抱えている。多くの開発者が参加し複数のグループに分かれて、かつ開発するソフトウェアのライフサイクル全体にわたり関与し支援する環境の構築が急がれている。

ソフトウェア開発環境の整備の目的には、開発技術そのものの支援と、そして本論で主として対象にした開発技術を実行させる全体的な機構が有効に動作するための支援がある。前者に関しては、新しい手法・パラダイムが数多く提案され実験されている。また、後者に関しては、コンピュータのアーキテクチャ及び分散処理技術の発展とあいまって、多くの研究開発が行われている。

ソフトウェア工学は、ソフトウェアの信頼性、生産性の向上に大きく貢献してきた。今後は、開発技術と環境構築技術など、まだまだ研究が推進され、体系的にさまざまな要素、技術が創造・構築されていくことになろう。

謝辞 富士通(株)の青山幹雄氏、日本電信電話(株)の長野宏宣氏、そして日本電気(株)の垂水浩幸氏に本特集の構想、技術面からの検討、そして資料の提供などにいろいろお世話になった。また、本論の執筆に当たり、京都大学の松本教授・鎌坂助教授、九州大学の牛島教授・荒木助教授、そして福岡大学の宇津宮教授から事例の紹介及び関連資料を快く提供していただいた。特に、今後の研究課題や動向については、宇津宮教授から丁寧かつ貴重な助言をいただいた。ここに深謝する。

参考文献

- 1) Computer Science Research Review, Computer Science Department, Carnegie-Mellon University (1982).
- 2) Morris, J. H. et al.: ANDREW: A Distributed Personal Computing Environment, Comm. ACM, Vol. 29, No. 3, pp. 184-201 (1986).
- 3) DSEE: A Software Configuration Management Tool: HEWLETT-PACKARD JOURNAL JUNE (1991).
- 4) Richard, N. Tayler. et al.: Arcadia: A Software Development Environment Research Project, 2nd International Conference on Ada Application and Environments, pp. 137-149, Apr. (1990).
- 5) Aoyama, M.: Distributed Concurrent Development of Software Systems: An Object-Oriented Process Model, Proc. IEEE COMPSAC '90 Nov. (1990).
- 6) 吉田紀彦, 檜崎修二: 場と一体化したプロセスの概念に基づく並列協調処理モデル Cellula, 情報処理学会論文誌, Vol. 31, No. 7, (July 1990).
- 7) Brothers, L. et al.: ICICLE: Groupware For Code Inspection, CSCW '90 Proceedings, ACM, pp. 169-181 (1990).
- 8) K. C. Burgess Yakemoviec and E. Jeffrey Conklin: Report on a Development Project Use of an Issue-Based Information System, CSCW '90 Proceedings, ACM, pp. 105-118 (1990).
- 9) 山下 薫, 沢田篤史, 鎌坂恒夫, 松本吉弘: CASE環境における協調活動支援ツールの試作, 日本ソフトウェア科学会, 第7回ソフトウェア研究会「分散環境におけるシステムと応用一般」資料 (1990).
- 10) 山下 薫, 鎌坂恒夫, 松本吉弘: ソフトウェアエンジニアリングデータベース KyotoDB におけるユーザインタフェースと協調活動支援機能の実現, 情報処理学会第41回全国大会, 6H-2 (1990).

- 11) 凍田和美, 宇津宮孝一, 吉田和幸: 統合コミュニケーションを用いたグループプログラミング支援の試み, 情報処理学会研究報告ソフトウェア工学, 7-6 (1991).
- 12) Utsumiya, K. et al.: A Hypertext System for Supporting Distributed Software Development, Proc. JWCC '91, pp. 137-144 (1991).
- 13) 斎藤信男: ソフトウェア開発環境, 情報処理, Vol. 28, No. 7, pp. 887-897 (July 1987).
- 14) 分散コンピューティング環境: Computer Today, 1991/7, No. 44, pp. 17-45 (1991).
- 15) Matsumoto, Y. and Ajisaka, T.: A Data Model in the Software Project Database Kyoto DB, Advances in Software Science and Technology, Vol. 2, pp. 103-121, JSSST (Japan Society for Software Science and Technology), Academic Press (1990).
- 16) 青山幹雄, 小池隆康: 大規模ソフトウェア分散並行開発の品質管理, 1991年2月, ENGINEERS (1991).
- 17) Aoyama, M.: Concurrent Development of Software Systems: A New Development Paradigm, ACM Software Eng. Notes, Vol. 12, No. 3, pp. 20-24, (July 1987).
- 18) 大野 豊: ソフトウェア工学の背景と展望, 情報処理, Vol. 28, No. 7, pp. 845-852 (July 1987).
- 19) 松本吉弘: ソフトウェアに対する要求の形成, 情報処理, Vol. 28, No. 7, pp. 853-861 (July 1987).
- 20) 石井 裕: コミュニケーションからコラボレーションへ, 電子情報通信学会技術研究報告, HC-91-15, pp. 31-38 (1991).
- 21) 仲谷美江, 西田正吾: 協同作業におけるコミュニケーション支援に向けての一考察, 電子情報通信学会技術研究報告, HC-90-22, pp. 51-57 (1991).
(平成3年10月8日受付)



板下 善彦 (正会員)

1947年生. 1971年東北大学工学部応用物理学科を卒業. 同年, 三菱電機(株)に入社. 自動制御システムの開発に従事. その後, 情報処理システムの分野, 日本語処理・文書処理・ユーザインタフェースの研究を行う. 現在は, 分散処理システムの構築方法論に興味を持つ. 電子通信学会会員.

