

ツール部品によるグループ開発支援環境構築基盤の構築

辻 新太郎 齊藤 裕介 上田 賀一

茨城大学 工学部 情報工学科

〒 316 茨城県日立市中成沢町 4-12-1

グループ開発では、開発者間でコミュニケーションを取りながらの協調作業が必要になり、これらの機能を開発支援ツールに組み込むことが有効である。また、開発では多くの情報を扱う必要があるので、開発に関わる情報をデータベースで集中的に管理できることが望まれる。本研究ではデータベースを核とする協調開発のための基盤を設計し、この基盤上でグループ開発支援ツールを提供することによって、これを利用したグループ開発支援環境を構築するための枠組を与える。また、この基盤を実際にグループによるドメイン分析のプロセスの支援に適用し、この基盤によるグループ開発の支援が有効であることを示す。

A Platform for Constructing Cooperative Development Environment composed of CSCW Tools

Shintaro Tsuji Yuusuke Saitoh Yoshikazu Ueda

Ibaraki University

4-12-1 Naka-Narusawa, Hitachi, Ibaraki, 316 Japan

In cooperative works, the developers should have their interactive communications to progress. It is efficient that the tools used in the development include functions to support cooperative works. On the other hand, many informations are handled in a development project. So it is required to manage all of the informations about the development and to store them in a database. In this study, a platform to support cooperative works is designed. And the platform has introduced a database as a core. CSCW tools are provided on this platform, as parts to construct an environment for cooperative works. As an example for this platform, we construct a supporting environment for a domain analysis and modeling process achieved by a group.

1 はじめに

開発するソフトウェアシステムが大規模になると、1人の開発者がすべての作業を行うなどということとはなくなり、複数の開発者がグループで作業を進めていくことになる。

開発者が複数になると、開発者間でコミュニケーションをとりながらの協調作業を進めていくことになるので、これらを支援するための環境が必要になってくる。

現在、開発者間のコミュニケーションの手段として、電子メールやUNIXのtalkのような対話システムなどが存在するが、これらはグループ開発に間接的には役立つが、直接的に開発作業に組み込まれるものではない。もし開発に用いるツールそのものにグループ開発のための機能を持たせることができれば、協調作業の効率はより改善されるはずである。

また、開発を進めて行くには多くの情報を扱わなければならないが、開発に参加しているメンバがこれらのデータをそれぞれ個人で管理していたのでは、データの受け渡しや再利用における効率が悪い。よって、開発に関わる情報をデータベースで集中的に管理することが望まれる。

本研究では、データベースを核とする協調開発のための基盤を設計し、この基盤上でグループ開発支援ツールを提供することによって、これを利用したグループ開発支援環境のための枠組を与える。この基盤は、データ管理、データベース管理、コミュニケーション管理、ツール管理、プロセス管理のための5つの層から構成されるものである。この基盤上では、開発者間のコミュニケーションのための機能を持ったツールを部品として提供し、これらを組み合わせることによって、協調開発の支援環境を構築することができる。また、作業を進める過程をデータベースに記録しておくことにより、意思決定の過程を支援することも試みている。

本稿では、協調開発基盤の開発について述べ、実際にグループによるドメイン分析のプロセスに適用した例についても示す。

2 グループ開発支援環境構築基盤の概要

2.1 グループ開発に望まれるもの

グループ開発の中で、開発者間のコミュニケーションは重要な意味を持ち、これを支援することによって、開発者間で意思の疎通を行うことや、協力して作業を行うことがより便利になる。

また、複数の開発者による開発では、目的は共通であっても、その過程では各個人がいろいろな考えを持ち、また影響を与え合いながら作業を進めることになるはずである。そのために、意思決定の過程を蓄積しておき、それを自分で見直したり、他の開発者に伝えたりすることによって、レビューや開発過程の再利用にも役立てることができるとは思われる。

2.2 本研究の目的

本研究では、開発者間のコミュニケーションのための機能や意思決定を支援するための機能を持ったツールを部品として提供し、それを組み合わせて開発支援環境を構築するための基盤を構築する。

また、これらのツールによって支援する開発のプロセスについて、ナビゲーションを行うための機構も用意する。

そして、この基盤では、ツールで扱うデータ、ツールに関する情報、開発の履歴など開発に関わるすべての情報を管理するために、データベースを核とした構成を考える。データベースでは、プロダクトの管理にとどまらず、プロセスの管理、さらにはコミュニケーションの管理についても行えるようにすることを目標とした。

3 グループ開発支援環境構築基盤の設計

本基盤では、データの管理、データベースの管理、コミュニケーションの管理、ツールの管理、開発プロセスの管理の機能をそれぞれの機能を持った層として考え、階層構造を持つアーキテクチャを設計した。これらの層をそれぞれ、「物理層」、「概

念層」,「コミュニケーション層」,「タスクツール層」,「プロセス層」と呼ぶ。

3.1 階層構造アーキテクチャ

以下で各層について下の層から順に説明する。

3.1.1 物理層

データベース内の物理的な記憶領域を管理する層。データ管理の役割を果たす。これらはデータベース管理システムの機能として実現されているので、ユーザが扱う必要はない。

3.1.2 概念層

上位の層が扱う概念的なデータを管理する層。データベース管理の役割を果たす。そのための機能は、データベース管理システムによって提供されるものであり、上位の層からはデータベース管理システムにアクセスすることによってこれらのデータを扱うことになる。

3.1.3 コミュニケーション層

開発者間のメッセージのやりとりや、開発者がデータベースへ格納する情報の仲介を管理する層。コミュニケーション管理の役割を果たす。上位の層のツール間のコミュニケーションは、すべてこの層を経由することになる。また、ツールがデータベースにアクセスする時も、この層を経由することになる。そのために、作業中のデータやデータベースに格納されているデータに関するアクセス権は、この層で管理される。

3.1.4 タスクツール層

グループ開発を行うユーザを支援するためのツールを提供する層。ツール管理の役割を果たす。ツール間での対話は、コミュニケーション層を経由して行われ、それによって協調作業が行われる。ツールは、作業そのものを支援する外に、開発の履歴を記録したり、特別な変更が行われた時にユーザにコメントを付加させたりすることにより、意思決定の過程についても支援を行う。

3.1.5 プロセス層

開発を行うための作業の流れを管理する層。プロセス管理の役割を果たす。タスクツール層で提供されるツールを集約し、プロセスの流れと結びつけることによって、開発プロセスのナビゲートを行うことができる。

3.2 本アーキテクチャの特徴

グループ開発で扱う情報は、データベースで集中的に管理され、ユーザ間のコミュニケーションは、コミュニケーション層を経由して行われる機構になっている。

マルチプラットフォームでの開発に利用しようとする場合には、コミュニケーション層より上の2層だけを各プラットフォーム用に用意するだけで済む。また、データベースを他のものに切り替えるような場合にも、その影響の波及範囲はコミュニケーション層までに限定できる。

3.3 意思決定の支援

開発者の意思決定の支援を行うために、開発に携わるそれぞれの開発者の作業の過程を記録することにする。しかし、作業の過程をすべて記録したのでは、データが膨大になってその保存がたいへんだけでなく、後でその履歴を確認することも困難である。また逆に、作業の開始時点と終了時点を記録しただけでは、途中の作業や変更の意味が後で分かりづらくなってしまう。

そこで、本研究では、作業の途中での重要な場面や大きな変更を加えた場合には、その変更の記録とともに、変更を行ったユーザが明示的にコメントを付加して保存しておくことにより、後になってからも、その変更の意味を確認できたり、そのユーザの意思決定の過程の履歴を追跡することができるようにする。

このために、タスクツール層で提供されるツールにおいて、特別な変更が行われた時にユーザがコメントを入力し、ツールが扱うデータが保存される時に、同時にこれらの情報もデータベースに保存されるというしくみを採用している。

3.4 ツール機能の再利用

ある作業を支援するために用意されたツールを、そのまま流用したり、あるいはカスタマイズしたりすることによって、他の作業を支援するためにも利用できることがある。

例えば、OMTやCoad&Yourdon手法による分析では、問題や要求の記述からモデル作成のために用語を抽出する作業がある。これらには、同じ用語抽出の支援ツールを流用できるはずである。また、これらの方法論には、分析の結果を特定の表記法でモデル化する作業がある。OMTでは、オブジェクトモデル、動的モデル、機能モデルの3つが作成され、Coad&Yourdon手法では、オブジェクト図、オブジェクト状態図、サービスチャートの3つが作成される。これらは、それぞれ異なった表記法を持つため、共通の作画ツールによってすべてを支援することは有効ではない。しかし、作画ツールが持つ機能には、共通するものも多いはずである。そこで、汎用的な作画ツールのための機能をまとめたものを用意しておき、それをカスタマイズして、専用のモデル記述ツールを定義して利用することを考えた。これらを行うためには、オブジェクト指向の継承の概念を用いることが適していると考え、ツールをクラス階層を用いて実現することにする。

このように、開発支援ツールを再利用することにより、個々の方法論ごとにそれに対応するための支援ツールを用意する必要はなくなる。また、ユーザが開発手法を他のものに切りかえても、同じような使い勝手のツールを利用できるという利点もある。

3.5 開発プロセスの支援

開発プロセスには、OMT、Coad&Yourdon手法などのような開発方法論の大規模なプロセスから、プログラミング(プログラム作成→コンパイル→リンク)のような小規模なプロセスまで考えられる。本研究で支援するプロセスの規模に関しては、特に規定せず、そのプロセスを進める上での作業を支援するツールさえ用意されていれば、どのような開発プロセスであっても支援できる。また、開発プロセスの定義を変えることによって、プロセスのカスタマイズを行うことも可能とする。

4 グループ開発支援環境構築基盤の実現

前節で設計した階層構造に従って実現した基盤の概略を示すと、図1ようになる。ここでは、それぞれの部分の実現について説明する。

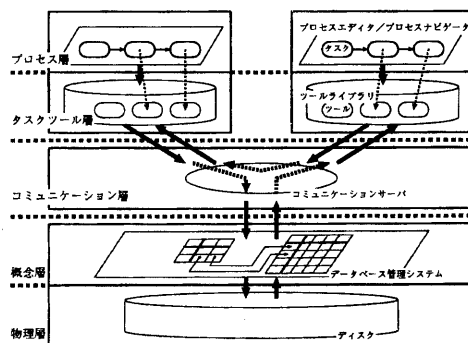


図1: システム構成

4.1 データベース

開発で扱う情報を格納するデータベースとしては、MATISSE[9]を利用した。MATISSEの特徴は、オブジェクト指向データベースであることである。格納する情報をオブジェクトとしてそのまま格納することができるので、柔軟に情報を扱うことが可能である。

4.2 コミュニケーションサーバ

コミュニケーション層でのコミュニケーション管理機構として、ツール間のコミュニケーションの管理を行うためのコミュニケーションサーバを用意した。コミュニケーションサーバはツールとデータベースとの間の仲介の役割も果たしている。つまり、コミュニケーションサーバは、ツール間のコミュニケーションのためのサーバであると同時に、データベースのクライアントになっている。ツールとの通信にはTCPを用いたプロトコルを使用し、データベースとの通信にはMATISSEのクライアントライブラリを用いることにした。

4.3 ツールライブラリ

タスクツール層で提供されるツールは、C++のクラスライブラリとして用意される。ツールのGUI部分を記述するためには、ネットワークプログラミングのための低レベルな部分のカスタマイズが可能で、しかもC++からの利用に適したものが必要である。そのために、筆者らが以前に構築したC++用のGUIライブラリ[7]を拡張し、ネットワーク部分のカスタマイズをできるように再構成して利用した。

ツールライブラリでは、ツールのGUI部分のイベント管理とコミュニケーションサーバとの通信の管理の役割を果たす「ツールマネージャクラス」と、すべてのツールのルートクラスにあたる「タスクツールルートクラス」という2つの基本クラスをはじめに用意しており、新たに提供されるツールは、この基本クラスを用いて記述される。

4.4 プロセスエディタ

開発に用いるプロセスの作業手順を定義し、そのプロセスの各作業(タスク)を支援するのに役立つツールとの関係を定義することによって、そのプロセスのナビゲータを構成するためのエディタを用意した。このエディタによって出力されたデータを用いてコンパイルを行うと、定義されたプロセスのためのナビゲータが構成される。

4.5 プロセスナビゲータ

プロセスナビゲータは、プロセスエディタで定義したプロセスについてナビゲートするためのツールである。プロセスの各タスクは、ボタンとして表現されていて、ユーザが作業の進行に伴ってボタンを押すことによって、それぞれのタスクについて定義されている支援ツールが活性化される。

プロセスエディタとプロセスナビゲータの関係を図2に示す。

5 適用例

ここで構築した基盤に基づいて、本研究室で提案されたドメイン分析手法であるSSM based

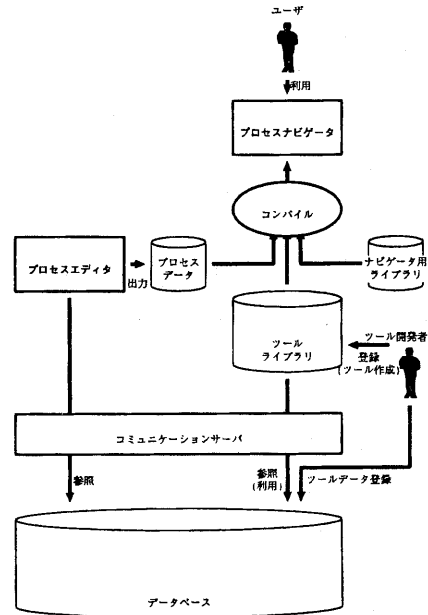


図2: プロセスエディタとプロセスナビゲータ

DAM[15][16]のプロセスについての支援環境を構築した。

5.1 SSM based DAM

SSM based DAM は、システム工学から生まれた方法論であるSSM(Soft Systems Methodology)をドメイン分析に応用した手法である。問題、仕様書/プログラム、およびそれらにあるプロセスから、特定のドメインのモデルを作成するDCM(Domain Component Modeling)と、DCMによって作成された複数のドメインのモデルを集約または抽象化するDA(Domain Aggregation)の大きく分けて2つの部分から成っている。DCM, DAの作業の流れはともに、

1. 問題
2. 問題状況の把握
3. 問題状況の表現
4. 基本定義
5. 概念モデルの作成

6. 比較・検討
7. 改革案の作成
8. 改革案の検討
9. 実行

というプロセスの繰り返しである。

5.2 SSM based DAM 支援用ツール

SSM based DAM のプロセスを支援するために、以下のタスクを支援するためのツールがそれぞれ用意されている。

- 「問題状況の把握」の支援ツール
問題を記述した文章から用語を抽出するツール。
- 「問題状況の表現」の支援ツール
問題領域を表現するための、作画ツール。
- 「基本定義」の支援ツール
顧客・行為者・変換・世界観・所有者・環境制約と基本定義を設定するため、各々をフォーム形式で記述するツール。
- 「概念モデル作成」の支援ツール
開発手順を順番に記述するツール。
- 「比較・検討」の支援ツール
概念モデルであげられた行為の検討を行うための表の作成を支援するツール。
- 「改革案の作成」の支援ツール
「概念モデル作成」の支援ツールと同じものを使用する。

これらのツールによって、SSM based DAM のプロセスを支援する。

5.3 SSM based DAM プロセスの定義

プロセスエディタを用いて、SSM based DAM プロセスの流れを定義する。

まず、SSM based DAM のプロセスの各タスクを定義し、それぞれのタスク間で作業の遷移が可能なるものをフローで定義する。

そして各タスクについて、図 3 に示すように、支援するためのツールが用意されているものは、支援ツールを定義する。支援ツールが用意されていないタスクに関しては、ツールの定義を「None」にしておく。

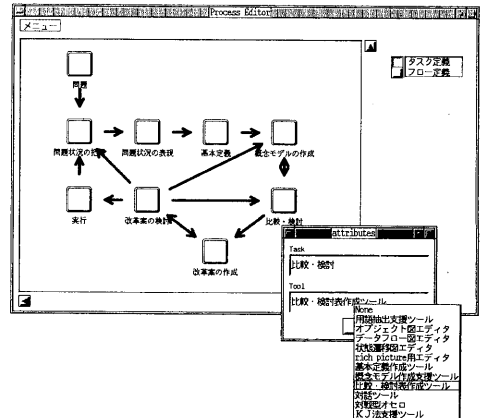


図 3: 支援ツールの定義

プロセスの定義と各タスクの支援ツールの定義が完了したらメニューで「コード生成」を選択し、ナビゲータのためのデータを出力する。

このようにして、プロセスエディタで SSM based DAM のプロセスを定義することにより、ナビゲータを作成できる。

ナビゲータのためのデータの内容は、C++プログラムのソースの一部になっており、これを利用してナビゲータをコンパイルすることになる。このデータには、ナビゲータのためのクラスの「ヘッダ部分」、「クラス宣言部分」、「コンストラクタ部分」、「イベント処理部分」について、プロセスに依存する記述を集めたものが記述されている。

5.4 作業の実行

プロセスエディタによって出力されたデータを用いてナビゲータを作成し、SSM based DAM の作業の流れをナビゲートできる。

ナビゲータでは、図 4 に示すように、タスクに相当するボタンを押すと、そのタスクのための支援ツールが活性化される。開発プロセスの流れに

従って順番にボタンを押してツールを活性化していくことにより、ナビゲーションが得られる。

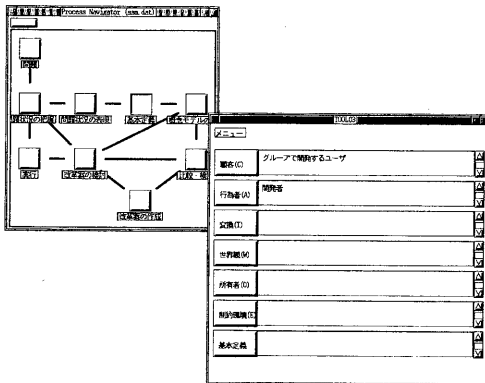


図 4: ナビゲーション

6 評価

本研究で構築した開発基盤によって、以下のようないことが達成できた。

6.1 システム構成

本研究では、データ管理、データベース管理、コミュニケーション管理、ツール管理、プロセス管理の役割を階層化したアーキテクチャによって、機能の分割を明確にし、マルチプラットフォームにも適用しやすい構造を実現できた。

6.2 ツール

開発者間のコミュニケーションというグループ開発に不可欠な機能をツールに持たせ、またツールで用いるデータをすべてデータベースで管理できるようにすることによって、ツールによるグループ開発の支援を実現できた。

6.3 ナビゲーション

ツールを部品として開発プロセスの流れと結びつけておくことによって、開発の流れに従ってツールを起動できるようなプロセスのナビゲーションを実現できた。

7 今後の展開

7.1 グループの管理

現在、コミュニケーション層ではグループの認証が行われていない。このままの機構で複数の開発グループが開発を行うと、同じツールを使った場合にメッセージが混信を起こしてしまう。よって、これを避けるためには、開発グループごとにコミュニケーションサーバを用意しなければならない。これでは無駄が多いし、ネットワークのポート番号の重複を避けるなどの措置も行わなければならない。また、データベースに格納されている情報についても、誰にアクセス権があるのか明確にされていない。

従って、グループの管理機構を組み込むことは、急務といえる。さらに、公開鍵暗号化手法などを用いて、グループ内の情報の機密保護を行うことも必要になるかもしれない。

7.2 柔軟な開発プロセス支援

開発プロセスの支援に関して、本研究では、支援対象のプロセスの規模を規定していない。そのため、開発方法論のような大規模なプロセスの支援をきめ細かく行うためには、プロセスを階層化できるようにすることが望まれる。

また、プロセスの流れを見やすくするためには、プロセスエディタで、分岐や繰り返しなどの制御構造が定義できると便利であろう。

7.3 ツールの管理

今回構築した基盤では、ツールは各プラットフォーム上にライブラリとして実現されているので、データベースで管理されているのは各ツールに関する情報のみで、ツールそのものが管理されている訳ではない。そのため、図 2 で示したように、ツール開発者は、新しいツールを開発した場合やツールに変更を加えた場合に、ライブラリへの登録とデータベースへの登録の両方を行わなければならない。片方でも欠けると、データベースに登録されているのにツールが利用できないということや、新しいツールがあるはずなのに登録されていないなどということ、さらに、プラットフォー

ムによってツールのバージョンが異なるようなことも起こり得る。

また、本来すべての情報がデータベースに保存されるべきなのに、ナビゲータを作成するためにはコンパイルが必要になるので、そのためにローカルにファイルを出力することが必要になってしまう。

今回はツールの実装に C++ を用いているため、各プラットフォーム上でツールをコンパイルして利用する必要があるが、各プラットフォーム上に仮想マシンを用意して、ツールがどのプラットフォーム上でもそのまま動作可能なようにすれば、ツールの集中管理が可能になる。つまり、ツールそのものをデータベースで管理することが可能になり、ツール開発者は、新しいツールを開発した場合に、そのツールをデータベースに登録するだけで、ツールを利用するすべてのプラットフォームに反映されるようになる。また、ナビゲータを作成する場合にも、データベースからツールを読み出してくるだけで、そのまま利用できるようになる。実際、本研究室において、このための研究も進行中である。

8 まとめ

本研究では、開発者間でコミュニケーションをとりながらの協調作業を支援するために、グループ開発のための機能をツールに持たせ、開発に関わるデータをデータベースで集中的に管理するための基盤を構築した。この基盤は、データ管理、データベース管理、コミュニケーション管理、ツール管理、プロセス管理という機能をそれぞれ、物理層、概念層、コミュニケーション層、タスクツール層、プロセス層として階層化し、機能の分担を行った。

また、開発にあたっての意思決定のための支援や、開発プロセスのナビゲートについても考えた。

そして、本開発基盤の適用例として、SSM based DAM の支援環境を構築した。

現在、開発に関する情報の管理はすべてデータベースで行っているが、開発に用いるツールそのものはデータベースで管理されていない。今後、リポジトリの概念を導入して、これらツールに関してもデータベースで管理できるような基盤へと拡

張していくことが課題となる。

謝辞

本研究で用いたオブジェクト指向データベース MATISSE を提供して頂いた SGN Co., Ltd. の方々に感謝します。

参考文献

- [1] 池本 浩幸, 加藤 克己, 楠井 洋一: ビジュアルプロトタイピング, 電子情報通信学会誌, Vol.74, No.7, 1991
- [2] J. ランボー, M. ブラハ, W. プレメラニ, F. エディ, W. ローレンセン (羽生田 栄一 監訳): オブジェクト指向方法論 OMT, トップラン, 1992
- [3] 本位田 真一, 山城 明宏: オブジェクト指向システム開発, 日経 BP 社, 1993
- [4] 落水 浩一郎: ソフトウェア・レポジトリ, 情報処理, Vol.35, No.2, 1994
- [5] Kaj Grønbaek, Jens A. Hem, Ole L. Madsen, Lennert Sloth: Cooperative Hypermedia System: A Dexter-based Architecture, COMMUNICATIONS OF THE ACM, Vol.37, No.2, 1994
- [6] 越塚 登, 坂村 健: 共有対話オブジェクト方式によるマルチユーザインタフェースシステムの設計と実装, 情報処理学会論文誌, Vol.35, No.9, 1994
- [7] 大津 信之, 辻 新太郎, 上田 賀一: ユーザインタフェースモデル記述主導型の実行可能モデル記述支援環境の開発, 情報処理学会第 50 回全国大会 4K-11, 1994
- [8] Rakesh Agarwal: The C++ Interface in Objectivity, ACM SIGPLAN Notices, Vol.29, No.12, 1994
- [9] MATISSE 2.3 TUTORIAL 1st Issue, A.D.B. S.A., 1995
- [10] 田中 俊昭, 山田 満, 羽鳥 好律: 構造化マルチメディア文書を用いた協調編集システム Colleague, 情報処理学会論文誌, Vol.36, No.6, 1995
- [11] Tomás Isakowitz, Edward A. Stohr, P. Balasubramanian: RMM: A Methodology for Structured Hypermedia Design, COMMUNICATIONS OF THE ACM, Vol.38, No.8, 1995
- [12] 岡田 謙一: グループウェアの未来, 情報処理, Vol.36, No.9, 1995
- [13] 新田 恒雄: GUI からマルチモーダル UI (MUI) に向けて, 情報処理, Vol.36, No.11, 1995
- [14] 青山 幹雄: コンポーネントウェア: 部品組立て型ソフトウェア開発技術, 情報処理, Vol.37, No.1, 1996
- [15] 齊藤 恵子, 上田 賀一: ソフトシステム方法論を利用したドメイン分析法の提案, 情報処理学会研究報告 96-IS-58, 1996 年 3 月
- [16] 上田 賀一, 齊藤 恵子: ソフトシステム方法論に基づくドメイン分析法の提案, 情報処理学会ソフトウェア開発のためのドメイン分析・モデリング技術シンポジウム論文集, 1996 年 9 月