

SDA概念モデルに基づいた通信ネットワークシステムのための
プラットフォームの構築

松尾真人 近藤好次
NTT 交換システム研究所

あらまし 我々は、電話網システム、パケット網システム、インテリジェントネットワーク(IN)システム等の通信ネットワークシステムを、統合ネットワーク上へ容易にかつ迅速に構築し、利用状況に応じて動的に再構築可能とする、『Software Defined Architecture (SDA)』の概念を提案している。本稿では、このSDA概念モデルに基づいて構築された通信ネットワークシステムに対して、分散処理のためのメッセージ通信機能と通信ネットワークシステムのインプリメント形態変更機能を提供する、プラットフォームの機能とその実現方式について述べる。また、通信ネットワークシステムへの本プラットフォーム機能の適用法について考察し、例題としてINシステムを取り上げる。

A Platform for Communication Network Systems
based on the SDA Conceptual Model

Masato MATSUO and Yoshitsugu KONDO
NTT Communication Switching Laboratories
3-9-11 Midori-cho Musashino-shi, Tokyo, Japan

Abstract We proposed the "Software Defined Architecture (SDA)" for communication network systems, such as intelligent network (IN) systems and telecom network systems. This SDA concept allows these network systems to be easily and quickly constructed and to be reconstructed according to node load, traffic and signalling delays.

In this paper, we describe about a platform for these network systems based on the SDA conceptual model. This platform provides a message handling function with access, location and duplicate transparencies, a dynamic reallocation function without processing interruption and an equivalent resource control function. And we apply these functions to an IN system as an example of network systems.

1. まえがき

利用者の要求に応じた多種多様なサービスの迅速な開発、ネットワークを構成する部品の統一的管理体系の実現等を目的としたネットワークアーキテクチャの議論が盛んであるとともに、それらの基盤技術として、情報処理技術の通信への適用が試みられている[1]～[4]。将来、ATM(Asynchronous Transfer Mode)網の導入に伴って統合されたネットワーク環境が整備されると、上記技術を利用して、サービスの提供形態や管理形態に応じた様々な通信ネットワークシステムが構築されていくと予想される。

我々はそのような観点から、電話網システム、パケット網システム、インテリジェントネットワーク(IN)システム等の通信ネットワークシステム(以下、単に通信システムと記述する)の、その統合ネットワーク上への、

(1-1) 利用者の要求に応じた迅速かつ容易な構築

(1-2) 通信システムの利用状況に応じた動的な再構築

を目的とするSDA(Software Defined Architecture) [5][6]の概念を提案している。

SDAは通信システムを機能・データ・リソースの種類要素からモデル化し、各要素の関係、使用条件等を個々の要素とは別に定義可能とすることで、設計条件や利用状況に応じた様々な通信システムとそのインプリメント形態を実現する概念である。

SDAを実現する環境は、大きくわけて通信システムの定義部、分散処理実行部、評価部の3つから構成される(図1)。定義部では、通信システムをSDA概念モデルに基づいて、機能・データ・リソースの3種類の要素とそれらの関係で分散システムとして定義する。分散処理実行部ではその定義に基づいてインプリメントされた個々の要素に対して、分散処理のためのメッセージ通信機能とインプリメント形態変更機能をプラットフォーム機能として提供する。評価部では通信システムの利用状況からそのインプリメント形態を評価し、分散処理実行部に対してその変更指示を行なう。

我々はSDA実験システムの構築を進めており、本稿では分散処理実行部が提供するプラットフォーム機能とその実現方式について報告する。またプラットフォーム機能の通信システムの適用について検討を加え、例題としてINシステムを取り上げる。

2. SDA概念モデル

SDAでは通信システムを、E-Rモデルに基づき、機能・データ・リソースの3種類のエンティティとそれらの関係によってモデル化する。このモデル化は以下の3段階の仕様で行なわれる。

(1) コンポーネント仕様

通信システムの構成要素となる機能エンティティ(FE)・データエンティティ(DE)・リソースエンティティ(RE)は各々、ソフトウェアプロセス・データレコー

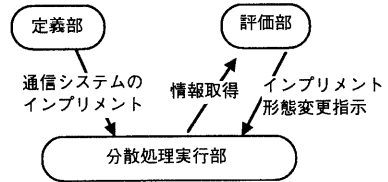


図1 SDAを実現する環境

ド・ハードウェアリソースに相当する。ここでは、各エンティティの特徴をそれが属するエンティティタイプとして規定し、グローバルな論理名を付与する。このとき、機能エンティティタイプ(FET)はそれに属するFEの処理プログラムを規定し、データエンティティタイプ(DET)はデータレコードのカラム構成を規定する。またリソースエンティティタイプ(RET)はハードウェアリソースの種別を表す。

(2) フレーム仕様

通信システムのフレームワークを構成する、エンティティタイプ間の関係(使用する、使用される、管理する等)を規定する。

(3) インプリメンテーション仕様

フレーム仕様の枠組みの中で通信システムのインプリメント形態を規定する。即ち、各エンティティタイプに属するエンティティもしくはその数と、フレーム仕様に基づいた個々のエンティティ間の関係を規定する。ここで特にエンティティ間の関係を規定しなかった場合、該当エンティティは、関係するエンティティタイプに属する全てのエンティティとの関係を持つことになる。この仕様で各FEが管理するDEやRE、およびそのFEのノード配置を規定する。

以上の仕様に従って、DEやREを管理するFEがソフトウェアプロセスとして分散処理実行部にインプリメントされる。

3. プラットフォームとして要求される機能

分散処理実行部に対する要求条件は、まずSDAの目的(1-2)から導出される。即ち、従来の通信システムのノード間機能分担や負荷分散形態が設計段階で固定されるため、利用状況に応じた柔軟な機能分担や負荷分散形態の実現が困難であったのに対して、SDAは通信システムとしてのフレームワークの中で、利用状況に応じた動的なインプリメント形態の変更を可能とする。従って通信システムの動的な再構築のためには、前章より、分散処理実行部において既に運用されている通信システムのインプリメンテーション仕様を柔軟に変更できることが要求される。本仕様においては、FEのノード配置や同じエンティティタイプに属するFEの数がノード間機能分担および負荷分散形態を規定する。また、FEとDEの関係がデータの分散形態を規定する。従ってインプリメント形態の動的な変更機能として以下の機能が要求される。

(3-1)通信システムとしての処理を中断せずに、FEのノード配置を柔軟に変更する機能

(3-2)同じFETに属するFEの数を迅速に変更する機能

(3-3)DEのFEへの割当を柔軟に変更する機能

次にこの前提として、インプリメントされる個々のFE間が疎結合であることが要求される。そこで、FEを分散処理実行部への配置単位とし、それらFE間の関係に基づいたメッセージパッシングによって通信システムとしての処理を遂行するメカニズムを実現することにした。このメッセージ通信には(3-1)(3-2)で可変となるFEのノード配置や数に非依存であることが要求される。

さらに、統合ネットワーク上に複数の通信システムが構築されるため、SDAにおいては統合ネットワークが有するハードウェアリソースを通信システム間でうまく共用することが重要となる。またインプリメント形態に依存せずにどのノードからでも等価にリソース制御ができることが要求される。

従って、(3-1)~(3-3)に以下の2つが加えられる。

(3-4)FEの位置や数に非依存なメッセージ通信機能

(3-5)リソースの通信システムへの割当を変更する機能

および、等価なリソース制御インタフェース

これら(3-1)~(3-5)の機能を分散処理実行部におけるプラットフォームとして構築した。

4. プラットフォーム機能の実現方式

前章で示したプラットフォームが提供する機能の実現方式について、(3-4)FEの位置や数に非依存なメッセージ通信機能、(3-1)~(3-3)インプリメント形態の動的な変更機能、(3-5)リソースの通信システムへの割当を変更する機能、および等価なリソース制御インタフェースの順に述べる。

4. 1 FEの位置や数に非依存なメッセージ通信機能

FE間のメッセージパッシングは、FE間の関係に基づくため、FEの論理名“通信システム名/FET名/FE名” [6]を通信アドレスとした。この論理的な通信アドレス(論理アドレス)により、FEのノード配置に非依存な通信を実現できるだけでなく、FE間の関係でFE間の通信形態を制御できるようになる。なお、この論理アドレスとソフトウェアプロセスに対する実際の通信アドレス(物理アドレス)の対応はネームサーバで管理される。

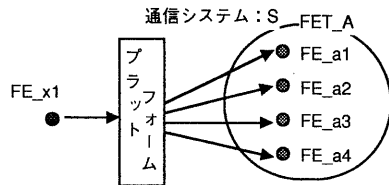
このFE間の通信に対して“通信システム名/FET名”によるマルチキャスト通信を原則とし、以下の2種類のリクエストタイプ設けた。

(1) 本依頼 (standard request)

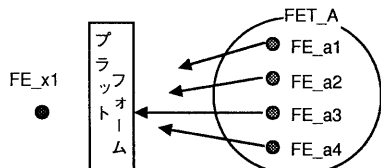
それを受けた全てのFEに対して、スレッドの捕捉とそのスレッドでの処理結果までを期待する。

(2) 仮依頼 (reservation request)

それを受けたFEに対して、当該の処理を行なうスレッドの捕捉だけを期待する。

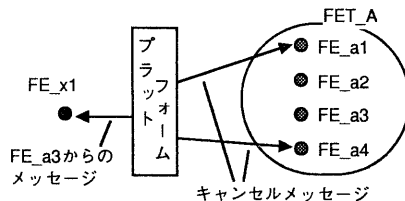


(a) 論理アドレス “S/FET_A” によるマルチキャスト



応答順	FE名	応答結果
1	FE_a3	受付可
2	FE_a4	受付可
3	FE_a2	拒否
4	FE_a1	受付可

(b) 応答待ちテーブル



(c) 応答結果に基づくFEの選択

図2 仮依頼リクエストタイプの処理

仮依頼を受けたFEは、処理できるスレッドが存在していれば受付可の応答メッセージを、そうでなければ拒否の応答メッセージを返す。プラットフォームは仮依頼に対して応答待ちテーブルを作成して、応答の中からある選択ロジックに基づいて特定のFEからの受付可メッセージを選択し、仮依頼元FEに返す。メッセージには発着論理アドレスが設定されているので、仮依頼元FEはこのアドレスを管理することで特定のFEとの論理コネクションを設定することができる。なお、選択されなかったFEには仮依頼キャンセルメッセージがプラットフォームから送信される。(図2)

このFEを選択するロジックには、以下の方法が考えられ、目的に応じて外部から定義かつ指定できることが好ましい。(図2では(4-1)を適用)

(4-1)最初に応答のあったFEを選択する方法

(4-2)処理中のスレッドが少ないFEを選択する方法

(4-3)ある順番で選択する方法

(4-4)ランダムに選択する方法

(4-5)選択履歴に基づいた統計的な評価で選択する方法

このように仮依頼リクエストタイプは、処理を依頼する“通信システム名/FET名”に属する複数のFEから状況に応じて処理相手を選択し、そのFEとの論理コネクションを設定するときに使用される。

1対1の通信を目的とする場合は、仮依頼で適当なFEと論理コネクションを設定した後、本依頼を使用するのが一般的な使用法となる。これによって、通信相手となるFEの数に依存しない、様々な処理依頼形態（例えば(4-1)による負荷の分散）が実現できる。

4. 2インプリメント形態の動的な変更機能

インプリメント形態の変更はすべて、通信システムの利用状況を示すメッセージ通信履歴やエンティティへの負荷等の評価に基づいて、評価部で判断される。

(1) FEの配置変更機能(3-1)

通信システムの処理を中断せずにFEの配置を変更するには、FEが実行時に管理する情報の扱いが重要となる。しかし、FEはスレッドにおける処理の終了とともに、実行時に保持する情報を破棄するので、実現が容易なプログラムマイグレーションによって実現することにした。ただしそこでは、処理の中断を避けるために、移動元FEの処理が全て終了するまで移動元と移動先に同一論理アドレスのFEを並存させ、物理コネクション識別子に基づいてメッセージを配送するメカニズムを構築した。ここでの物理コネクション識別子とは、プラットフォームがFEの配置ノードを識別するもので、ここではノード名を採用した。1回のメッセージのやり取りを通して、プラットフォームによりメッセージのヘッダに付与され、FEが特定のFEと論理コネクションを保持している間は該当FE間でこのヘッダを持ち回る。以下にFEの配置変更手順を示す。

- ①移動元FEのプログラムファイルを指定された移動先ノードにコピーし、プロセスとして起動する。
- ②移動元FEで新規処理依頼を受け付けないようにし、ネームサーバを更新する。
- ③メッセージに物理コネクション識別子が設定されていればそれに基づいて配送し、設定されていない場合はネームサーバに基づいて移動先FEへ配送する。
- ④移動元FEの処理が終り次第、消滅させる。

なお③で、ネームサーバが完全に更新されていないときは移動先FEが新規処理依頼先から外れるが、マルチキャストにより処理が分散されているので問題はない。

(2) FETに属するFE数の変更機能(3-2)

同じFETに属するFEの数の変更には、

- ①FEの絶対数を変更する方法と
- ②絶対数は変更しないが、特定のFEから見たときにアクセス対象となるFE集合を変更する方法がある。

①に関しては、4. 1節に示したようにFE間のコネクションが動的にとられるため、FE単位の追加・削除で容易に実現できる。

②は、マルチキャストによる不要なメッセージの減少に有効となる。通信システムの運用が安定すると、あるFEからの関係するFETに属する個々のFEへの処理

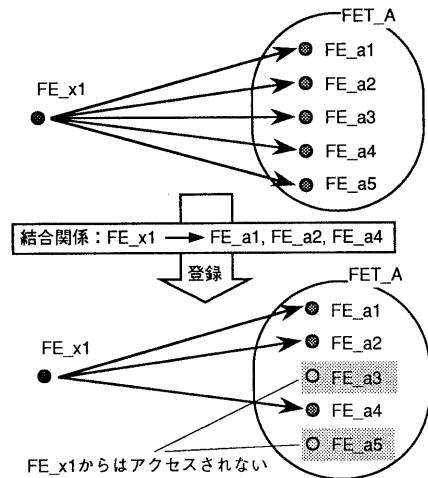


図3 結合関係によるアクセス対象の変更

依存度が明確になってくると考えられる。従って、あるFETに属する全てのFEにマルチキャストを行なうよりも、統計的に処理依存度が高い特定のFEだけにメッセージを送った方が、ノード間トラフィックを減少させることができ、より安定したシステムの運用が可能となる。

このFE毎の対象の変更は、評価部によってFE間のメッセージログから抽出された依存度の高いFE間処理依頼関係（結合関係）をプラットフォームに登録し、その結合関係に従ったメッセージの配送を行なうことにより実現する。（図3）

(3) DEのFEへの割当変更機能(3-3)

割当先FEが変更しうるDEに関して、そのデータ管理を容易にするため、DEのオリジナル情報をプラットフォーム内のデータベースで一元的に管理することにした。FEはその生成時にデータベースからFE-DE関係テーブルを参照して、割当られたDEを読み込む形式をとる。この関係テーブルを変更することで、DEのFEへの割当を変更できる。

4. 3リソースの通信システムへの割当を変更する機能と等価なリソース制御インターフェース

まず、統合ネットワーク上の全てのREに対して、個々のREを管理するFEをリソースマネージャとして独立させ、複数の通信システムから共通にアクセスできるプラットフォームのサブシステムとして位置づけた。この結果FE-FE間のメッセージ通信で任意のリソースにアクセスできるようなる。

また、REの属性としてリソースの使用権/所有権を設けた。これらの権限はREのオリジナルデータとしてDE同様データベースで一元管理しており、FEの生成時にデータベースから読み込まれる。各FEはこの権限に

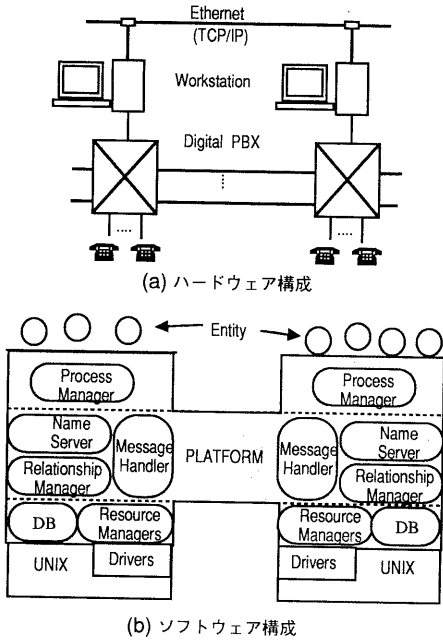


図4 プラットフォームの構成

基づいてリソースに対するアクセス制限を行なう。さらに、これらの権限の変更インタフェースを設け、アクセス権限を動的に変更できるようにした。

5. プラットフォームの構成

以上の機能を分散処理実行部のプラットフォームとしてSDA実験システムに構築した。実験システムの構成を図4に示す。ワークステーションとPBXの組で1ノードを構成した(図4a)。

また、プラットフォームを以下のサブシステムから構成した。

- ネームサーバ
FEの論理アドレス“通信システム名/FE名/FE名”と配置ノードとの対応を管理する。
- プロセスマネージャ
FEをソフトウェアプロセスとして生成/消滅させ、FEの論理アドレスと物理アドレスを管理し、FEとの通信窓口となる。
- リレーションシップマネージャ
評価部の指示に従ってFE間の結合関係を管理する。
- メッセージハンドラ
4. 1節のFE間のメッセージ通信機能を実現する。
- リソースマネージャ群
- データベース

DEとREのオリジナルデータと、FE-DEおよびFE-RE各々の関係テーブルを管理する。またFE間のメッセージ通信履歴を蓄える。

実験システムはプラットフォーム機能の確認が目的

表1 INシステムにおけるFEの例

分散機能プレーン上の機能	SDAにおけるFEの例
サービス制御機能(SCF)	カスタマ網制御FE(CNC_FE) サービス網制御FE(SNC_FE)
サービスデータ機能(SDF)	カスタマ網管理FE(CNM_FE) サービス網管理FE(SNM_FE)
サービススイッチング機能(SSF)	論理端末制御機能FE(LTC_FE) 論理網制御機能FE(LNC_FE) 論理端末管理機能FE(LTM_FE) 論理網制御機能FE(LNM_FE)
特殊リソース機能(SRF)	論理会議ブリッジ制御FE(LBC_FE) 論理音声蓄積装置制御FE(LVC_FE)

であるため、プラットフォームをワークステーションのOS上に位置するカーネルとして構築した。その際、プラットフォーム内における不必要なノード間通信を避けるために、上記のサブシステムを全てのノードに一様に配置した(図4b)。

6. インテリジェントネットワーク(IN)システムへの適用

本章ではINシステムを例題として、プラットフォーム機能の通信システムへの適用について考察する。

6. 1 インテリジェントネットワーク(IN)システム

CCITTのIN概念モデルは4つのプレーンから構成される。[7]

(1) サービスプレーン

サービスユーザにより観測できるプレーン。サービスやサービスフィーチャにより構成される。

(2) グローバル機能プレーン

サービスフィーチャの記述が行なわれるプレーン。SIB(Service Independent Building Block)等で構成される。

(3) 分散機能プレーン

INシステムを構成する機能が規定されるプレーン。サービス制御機能(SCF)、サービスデータ機能(SDF)、サービススイッチング機能(SSF)、制御機能(CCF)、特殊リソース機能(SRF)等から構成される。

(4) 物理プレーン

分散機能プレーン上の機能とノードとの対応が規定されるプレーン。サービス交換ノード(SSP)やサービス制御ノード(SCP)、インテリジェントペリフェラル(IP)等から構成される。

SDAにおけるFEはサービスではなく通信システムの機能アーキテクチャを構成する要素に相当するため、プラットフォーム上に存在するFEは上記の分散機能プレーンのレベルで観測される。ただし、SCF等の機能をさらに複数のFEに分けることも可能である。本稿では上記の分散機能プレーン上の機能に、NTTの提唱するサービス網、カスタマ網からなる階層構造の概念[8]を適用し、FEの例として表1のようにSCF等の機能を構成する機能要素を考える。

6. 2プラットフォーム機能の適用

本稿では、特にメッセージ通信機能とインプリメント形態の変更機能の適用を取り上げる。

(1) メッセージ通信機能の利用

リソースマネージャへのアクセスに仮依頼リクエストタイプを用いることで、ハードウェアリソースの効率的な選択が可能になる。

例えばLVC_FEから仮依頼で必要容量を持つ音声蓄積装置マネージャにアクセスすることで、ネットワークリソースの負荷に応じた選択が可能になる。

(2) インプリメント形態の変更機能の適用

メッセージ通信機能では解決できない特定FEへの過負荷、ノードの過負荷、メッセージ遅延、ノード間信号数の増大がインプリメント形態の変更のトリガとなる。

特定FEへの過負荷が生じた場合は、該当FETに属するFEの絶対数を増やしたり、さらに結合関係を操作することで負荷をうまく分散することができる。逆に、特定FETへの負荷が極端に低いときは、結合関係で特定のFEだけを駆動するようにし、問題がなければその絶対数を減らせばよい。例えば、特定のサービスに対するサービス要求が増大すれば、該当サービスを提供するCNC_FE, SNC_FE, CNM_FE, SNM_FEの絶対数を増やすことで対処できる。これは特に企画型サービス等の輻輳に対する負荷の分散に有効となる。

ノードの過負荷の場合は、その要因となるFEを特定し、そのFEもしくはそれ以外の負荷が小さいFE群を他ノードに配置変更する方法や、要因となるFEへの処理依頼関係を結合関係で規制する方法、そのFEの過負荷を避ける方法等で解決できる。例えば、パーソナル通信サービスにおけるパーソナル番号データベースへのアクセスが増えた場合は、該当データを管理するSNM_FEやSNC_FEを増やし、該当FEへのDEの割当を変更して、特定のSNM_FEとSNC_FEの結合関係を規定する。これによってパーソナル番号移動の地域性に基づく、同サービスの地域的な分化には有効な手段となる。ただし、FEの配置変更の際には、移動先でのノード過負荷を避けなければならない。

メッセージ遅延およびノード間信号数の増大の場合は、その要因となるノードにまたがったFE間の関係を特定して、それらのFEを1ノードに配置するか、結合関係の操作でアクセス先を他ノードのFEへ変更することで対処できる。前者の場合は、要因となるFEとそれに関係する他FEとのアクセス頻度を考慮して、再配置単位および再配置先を決定する必要がある。このようにノード間信号数増大の要因には、処理依存関係の強いFEグループのノード間にまたがった存在が挙げられる。このグループの抽出によって、例えば、サービス要求頻度が大きいサービスに対して、それを提供するSCF, SDF, SSFに属するFEを1ノードに配置することで、従来の機能分担では実現できない利用状況に

じた動的な機能分担の変更が可能となり、SSP~SCP間トラヒックを減少させることができる。

7. むすび

SDA概念モデルに基づいて構築された通信システムに対して、分散処理実行部で必要となるプラットフォーム機能を挙げ、それらの実現方式について述べた。また、通信システムの動的な再構築法について検討を加えた。今後はこの実現方式の評価と、インプリメント形態の評価方法について検討していく必要がある。

前者の検討項目としては、具体的な通信システムとそのサービスに対する、

(7-1)仮依頼・本依頼リクエストタイプの適用法とその有効性、

(7-2)仮依頼における有効な選択ロジックとその適用パターン、

(7-3)結合関係の有効性

の評価が挙げられる。また後者に関しては、

(7-4)結合関係の抽出法

(7-5)FEやノード負荷の評価法とFE移動先ノードの選択方法

(7-6)6章で触れたFEグループの抽出法

について検討する予定である。

【参考文献】

[1] Sudar Akkala, Samir Desai, Ming-Yee Lai, Ken Lehner, Bryab Whittle: "Exploratory work on the Application developer's Perspective of Telecommunications Information Networking Systems", TINA'90.

[2] G.Brruno, L.Cerchio, S.Montesi: "Information Modelling, Software Architectures and Platforms: an example of application of a comprehensive approach in the TMN domain", TINA'91.

[3] 加藤, 藤長, 小花, 鈴木: "INへの分散処理技術の適用に関する考察", 信学会第1回NAシンポジウム 25, 1990.

[4] 中島, 所: "自律分散システムに関する一考察: 大規模システムにおけるネットワークサービス提供のための基盤モデル", 信学技報 COMP 88-67, 1988.

[5] 近藤, 松尾: "ネットワークシステムソフトウェア定義方式", 信学会第1回NAシンポジウム 10, 1990.

[6] Kondo, Matsuo: "Software Defined Architecture Concept: A Network System Model for Information Networking Architecture", 信学論 (掲載予定).

[7] CCITT Com VII, Study Group VII Q19 / VII (DAF) Report, Feb.90

[8] 岡田, 重松, 佐藤: "インテリジェントネットワークにおけるサービス定義・制御の階層的構造", 信学技報 IN90-17, 1990.