

解説

移動体通信システムへの仕様記述技術
の適用上の課題†

荒 野 高 志†† 山 崎 誠 一†† 伊 藤 光 恭††

1. はじめに

通信システムは、より知的に、より個人の生活に密着したものと、多様化、複雑化の一途をたどっている。しかし、それにもかかわらず、システムの信頼性やサービス開発の迅速性の要求は年々高まっており、通信ソフトウェア開発の生産性、信頼性の向上は急務である。この問題を解決するための一つの手段として、近年、形式的仕様記述技術が注目されている。形式的仕様とは、自然言語による仕様と異なり、あらかじめ与えられた利用可能な厳密な規則に従い意味を定めることができる仕様のことを言い¹⁾、形式的仕様記述技術とは形式的仕様を記述するための言語及び仕様の解析を含むその周辺技術のことを言う。

仕様記述技術は通信ソフトウェア開発の生産性／信頼性をドラスティックに変える可能性のある技術である。

(1) 仕様は超高級言語として機能する

要求仕様、設計仕様を直接実行する、あるいは仕様からコードを自動生成することにより、コーディング作業が不要あるいは激減する。生産性、信頼性が飛躍的に向上し、また保守も仕様を直接変更するだけで済むため、保守効率も向上する。

(2) 厳密な仕様により要求者／設計者間のコミュニケーションを促進する

自然言語による仕様は、非常に曖昧であるため、要求者／設計者間での誤解が生じやすく、そのため要求仕様が本来の要求者の要求と異なってしまうことがある。形式的な仕様は、厳密で意味が一意にきまるため、このような誤解の入り込む余地がない。これは、コーディング／テスト段階

以降における仕様変更を未然に防止する。レビュー時間も短縮できる可能性がある。

(3) 実行可能仕様はプロトタイプとして機能する

要求者／設計者間の誤解以外にも、実際の開発では、要求者自身が要求を完全に把握していることは稀である。このことが後の仕様変更による工数増加につながる。プロトタイプは、この問題を解決するための、現状で最も有効な方法である。実行可能仕様はシミュレーションなどにより、プロトタイプとして利用できる。

(4) 通信仕様記述言語は通信モデルを提供する

通信仕様記述言語はプロセス代数、シーケンス図など通信処理に向けた特有のモデルを提示する。汎用プログラミング言語で記述するのに比べ、記述者の負担は格段に少ない。これは単に記述量だけの問題ではない。これにより比較的通信処理の設計に不慣れな人にも適切な記述ができる可能性がある。

(5) 仕様検証は高度なエラーを発見できる

仕様記述言語は上記(4)で述べたように数学的なモデルをその土台としているため、そのモデルに基づいた高度なエラー検出が可能である。デッドロックの検出などがこれにあたる。仕様記述技術を使わない通常の通信システム開発では、このようなエラーは開発早期にはなかなか発見が難しく、一つの大きな問題点となっている。

(6) 仕様記述はデバッグエイドとなる

仕様からテストプログラムを自動生成することができる。これにより、テストプログラム作成工数が削減できる。また、テスト自体もより効率に行える可能性がある。通信処理などの大きなソフトウェアでは、テストプログラム作成工数も非常に大きなものである。

† Specification Techniques for Mobile Telephone Systems by Takashi ARANO, Seiichi YAMAZAKI and Mitsutaka ITO (NTT Software Laboratories).

†† NTT ソフトウェア研究所

(7) 仕様記述の再利用

仕様を再利用することができれば、生産性向上の意味からは、単にコードの再利用よりも遥かに大きい効果が期待できる。

このような可能性を秘めた仕様記述技術であるが、現状の仕様記述技術がそのまま通信ソフトウェア開発の現場に適用しうるかと言うと、なかなか難しい。本解説では、移動体通信システムを例にとり、その記述という側面から、現状の仕様記述技術の適用可能性について解説する。

2. 移動体通信システムの

概要^{1), 25)}

最近街角で携帯電話を使って、電話をかけている人をよく見かけるようになった。移動体システムとは「いつでも、どこでも、どこへでも」という、このようなニーズに応えた通信システムである(図-1)。本章では移動体通信システムが他の通信サービスと比較したときの特徴について述べ、その処理を概観する。

2.1 通常の通信システムとの相違

まず、移動体通信システムを特徴づける点は、2点に集約される。

- I. 無線である
- II. 端末が移動する

以下にソフトウェアに関係する点に絞って、これらを説明する。

I. 無線である

- (1) 資源が有限である

有線のシステムは、銅線や光ファイバ線といった通信媒体を用いている。これは線を増設すれば、それだけ回線容量が増えるという性質を持っている。一方、移動体通信システムはこれとは異なり、おのずと容量が制限されてしまう。

この資源問題を解決するのに、さまざまな技術が開発されている。その代表的なものが、周波数

いつでも、どこでも、どこへでも

図-1 移動体システムとは¹⁾

マルチチャネルアクセス
= 同周波数帯の時分割使用

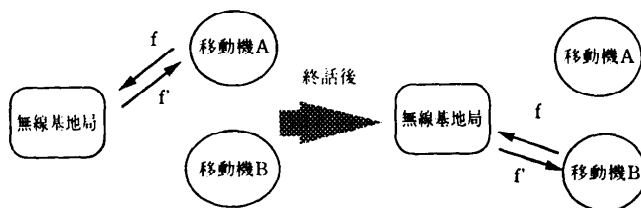
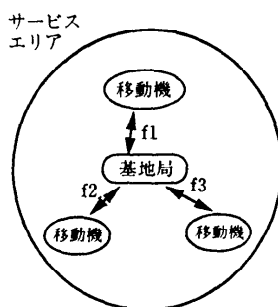


図-2 マルチチャネルアクセス

ゾーンをとらない方式



ゾーン方式
= 同周波数帯の空間分割使用

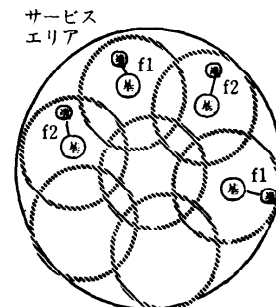


図-3 ゾーン方式

帯の時分割と空間分割である。時分割とは、いわゆるマルチチャネルアクセスと呼ばれる方式であり、同一周波数帯を移動端末固定にせず、その必要になるつど、割当／解放する方法である(図-2)。一方、空間分割は、サービスエリアをいくつかのゾーンに分け、隣接しないエリアでは同一周波数帯を使用するというものである(図-3)。このゾーン方式は後に述べるように、担当無線基地局の動的変更という問題を引き起こす。

(2) 通信前は端末と物理的な接続がない

有線システムは加入者線というハードウェアが存在する。だから、発呼(いわゆる発信)はオフフック(電話機の受話器をあげること)を電氣的に検知できるし、着呼(いわゆる着信)はその加入者線に対し呼出音を鳴らすための信号を送出すればよい。しかし、無線システムでは、通信前は端末と物理的な接続はないため、特別な仕掛が必要となる。

発呼に関しては、マルチチャネルアクセス方式を用いているため、端末はあらかじめ通話チャネルの周波数帯を知ることはできない。そこで、た

例えば特定の周波数帯を発信信号専用チャネルとして割り当てておき、無線基地局が発呼を認識後新しく割り当てた通話チャネルに移行するなどの方式が採用されている。

着呼については、端末が移動可能であるため、より複雑な仕掛けが必要である。これについては次項Ⅱ(3)位置登録で述べる。

(3) 利用者認証が必要である

移動体通信は無線であるから、特定のプロトコルにさえ従っていれば、他人の端末になりすまして、他人の回線を使用することができる。これを防止するために、暗号化技術や利用者認証技術などを用いている。

Ⅱ. 端末が移動する

(1) 移動による障害物などの環境の激しい変化

自動車電話では、障害物による電波の伝播損失の格差、電波の反射や干渉、端末の高速な移動などのため、フェージングと呼ばれる受信波の変動が起こる。これを補償するために、誤り訂正符号などさまざまな技術を用いている。

(2) 担当無線基地局の動的変更

端末ゾーンをまたいで移動することがあるが、このとき、システムは担当無線基地局を通話中に変更しなければならない。この方式をチャンネル切替という(図-4)。

チャンネル切替は、端末のゾーン移行検知、移行先ゾーンの決定、回線切替の手順で行う。ゾーン移行の検知は、受信レベルの低下を検知する方法と、電波到達時間から距離を算出する方法などがある。より詳細な処理については、2.2で述べる。

(3) 位置登録

着呼に関して、着側の端末の存在場所の問題がある。つまり、着側の端末が日本全国のどこかのサービスエリアに存在するとして、それが現在どのゾーンにいるかをどうやって知るか、という問題である。一つの着呼につき全てのゾーンに対し一斉に呼び出すという解は加入者数の増加に対応できないため、現実的ではない。そこで、通常はシステムが常に全端末の位置を知っているような仕掛けになっている。

具体的には、端末ごとにホームとなる移動無線

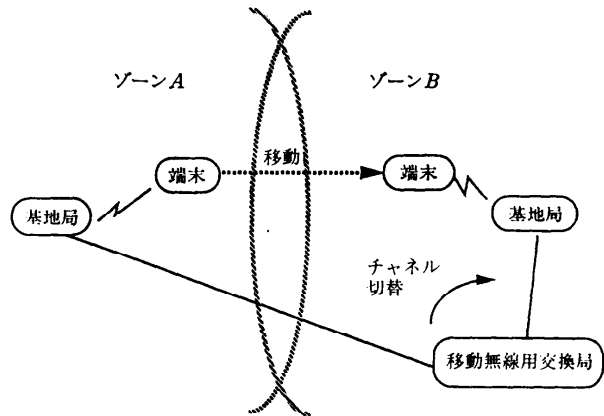


図-4 チャンネル切替

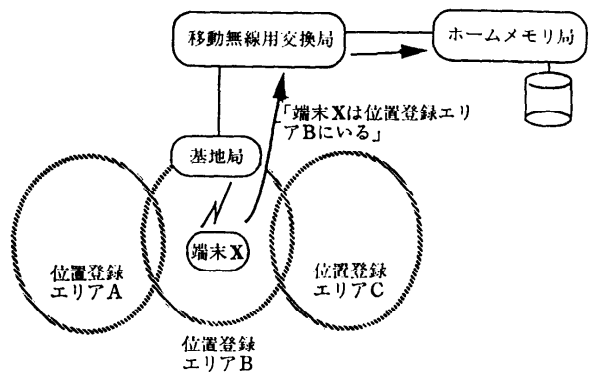


図-5 位置登録

用交換局（ホームメモリ局）をあらかじめ定めておき、端末の位置などの利用者情報の参照、更新はホームメモリ局へ問い合わせるという方式をとる。特に、位置情報の更新処理のことを、位置登録処理と呼ぶ(図-5)。

2.2 処理概要

前節では、移動体システムの特徴を説明した。本節では移動体システムの処理の概要を解説する。

まず、処理の説明の前に移動体通信システムの構成について触れておく(図-6)。自動車電話機や携帯電話機のことを移動端末、端末、あるいは移動局と呼ぶ。移動局と無線で交信する装置が無線基地局である。この無線基地局は前節で述べたゾーンを統括する局である。いくつかの無線基地局を制御するのが無線回線制御局である。また、無線回線制御局には、移動無線用交換局が接続されており、これは公衆交換網とのインタフェース

の役割を果たしている。

なお、蛇足ではあるが、以下で述べる処理は実際の移動体通信システムを理解容易にするために極端に単純化したものであり、場合によっては厳密には実際のシステムとは異なる処理となっている部分もある。ご容赦願いたい。

(1) 位置登録

前節で述べたように、システムは、着端末を常に位置把握するために、位置登録処理を行う必要がある。無線基地局は、電源がオンになっている端末全てに対し、常に報知信号を送出し、端末はその信号を受け、次の契機で自分の位置情報を登録依頼する。

1. 位置登録エリア移動時
2. 端末電源投入時
3. 発信時（必ずしも必要条件ではない）

手順は以下のとおりである（図-7）。端末は、自分の端末番号と自分の現在いるエリアを含む、位置登録信号を送出する。移動無線用交換局では位置登録信号を受け、その発信端末番号からホームメモリ局を割りだし、そこに位置登録信号を転送する。ホームメモリ局から位置登録が完了したことを通知する位置登録確認信号が移動無線用交換局に送られたら、無線基地局を通じて、それを端末に転送する。端末はこれにより、着信可能な状態となる。

(2) 発信

発信処理そのものは、通常の電話と類似している（図-8）。発信者が受話器を上げるなどの操作をトリガに、端末は発信信号を送出する。まず、移動無線用交換局により端末番号からホームメモリ局に利用者情報を問い合わせ、端末を照合する。これが正しく済んだことが確認できると、無線基地局では空いている通話チャンネルを選択する。選択したチャンネルをチャンネル指定信号で端末に通知

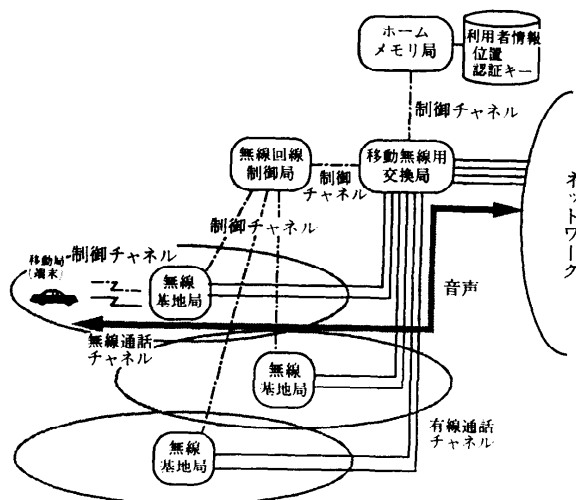


図-6 移動体通信システムの構成

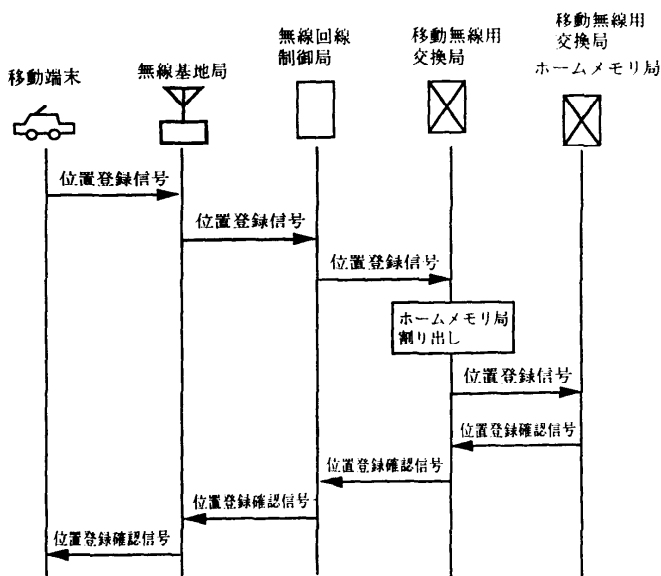


図-7 動作概要 1/5 (位置登録)

し、これにより端末との通話路が設定される。次に端末にダイヤル送出指令信号が送出され、端末は発信者のダイヤルを待つ状態となる。通常の電話では、ツーというダイヤル待機音が鳴っている状態である。発信者からダイヤルされたダイヤル数字は端末から無線基地局に送出され、交換機を経由して、他の端末へ交換接続がなされる。相手端末の状態により、網より呼出音（RBT: Ring Back Tone）または話中音（BT: Busy Tone）が送出され、相手端末の応答により通話が開始される。

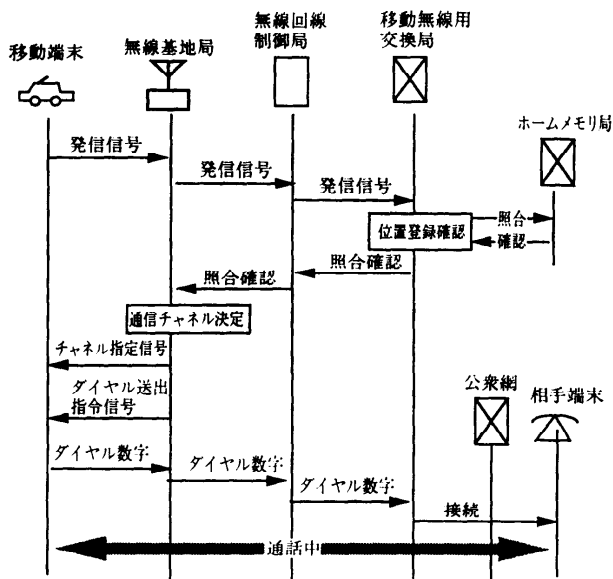


図-8 動作概要 2/5 (発信)

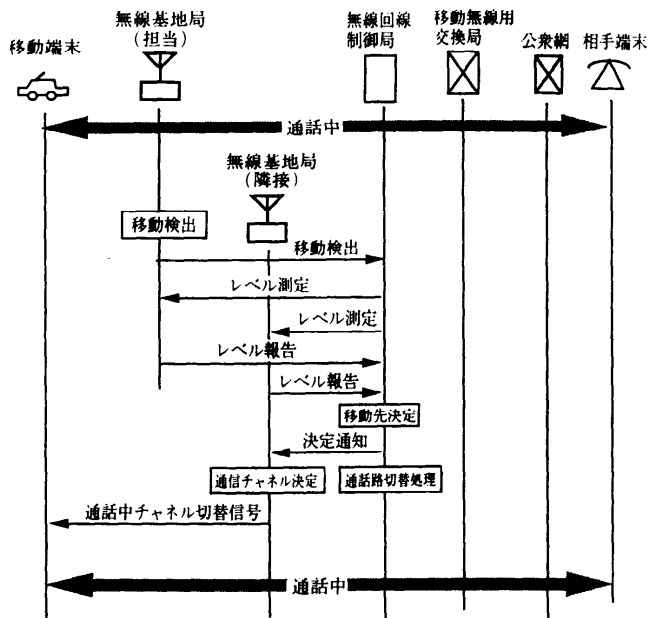


図-9 動作概要 3/5 (チャンネル切替)

(3) チャンネル切替

チャンネル切替の処理概要を図-9に示す。まず、端末があるゾーンから外に出たことを検知する。次に、端末がどのゾーンに移行したかを決定する。それには、移動検知と同様に電波の強さにより判断する。まず通話中ゾーンの周辺ゾーンの無線基地局（隣接局と呼ぶ。なお一般に複数の隣接

局があることに注意）に一斉にレベル測定依頼を行う。それを受けた無線基地局では端末との間で電波の強さを測定し、そのレベルを無線回線制御局に返答する。無線回線制御局では、そのいくつかの返答を受け、最も強い電波を受信したゾーンを移行先ゾーンとする。決定した後は、新しい無線基地局で、空きチャネルを選択し、無線回線制御局と無線基地局との間の回線の切替と、無線基地局と端末の間のチャンネル切替を行う。

(4) 終 話

終話は移動端末側から行われる場合と、網の先の端末から行われる場合があるが、ここでは前者の場合のみ述べる（図-10）。まず、ユーザが受話器を置くなどの操作をトリガに、終了信号が無線基地局に送出される。終了信号を受信す

ると、通話料金の計算のため、通話時間をホームメモリ局へ通知する。ホームメモリ局は通話時間の積算を行い、その情報を記憶する。交換機をとおして、網側の回線を解放し、また同時に端末へ切断信号を送出し、通話チャンネルを解放する。

(5) 着 信

着信は次のような処理となる（図-11）。他の端末から移動端末の局番がダイヤルされたとする。交換機は、最寄りの移動無線用交換局まで接続する。その無線用交換局は移動局番号よりホームメモリ局を割りだし、そこから位置登録情報を参照する。端末の現在位置が自エリアであれば、無線基地局より、端末へ着信信号を一斉呼出する。もし、他エリアであれば、該当エリアの交換局に転送する。移動局番に該当する端末が着信信号を受信すると、それは着信応答信号を返送する。無線基地局では着信応答信号の受信後、空いている通話チャンネルを選択し、端末へチャンネル指定信号を送出することにより、チャンネルが確保される。この時点で、端末に呼出音（RGT: Rin Ging Tone）が送出される。ユーザが移動端末をとり

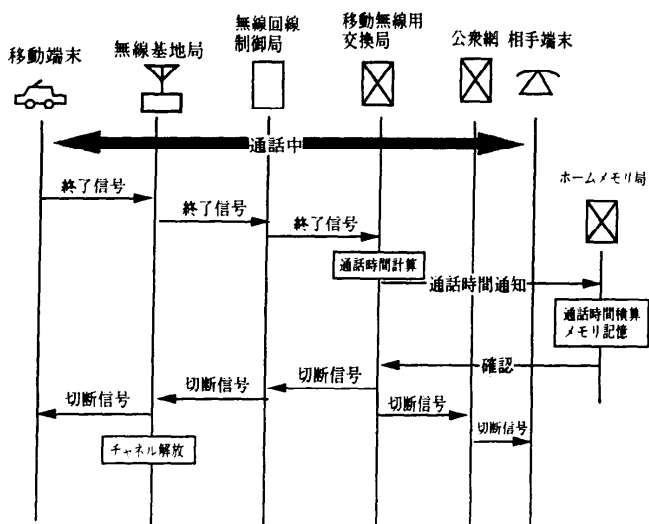


図-10 動作概要 4/5 (終話)

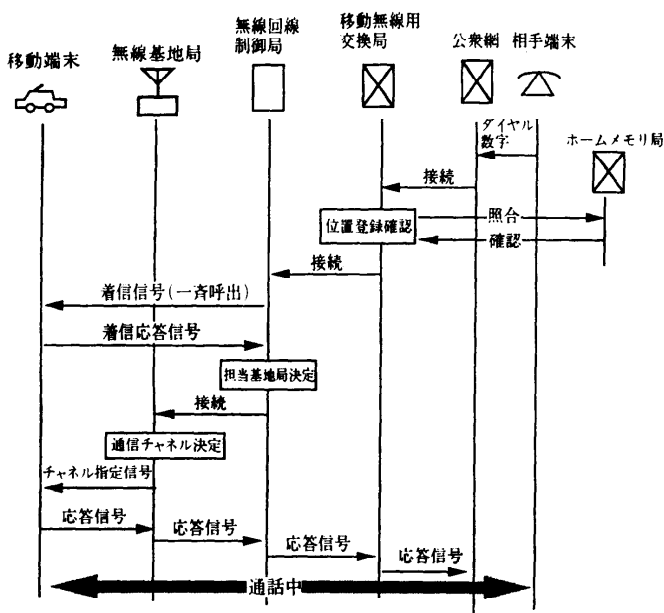


図-11 動作概要 5/5 (着信)

あげると、端末から応答信号が返り、通話中となる。

2.3 準正常系とオペレーション系

以上述べたものは、呼処理の正常系だけであった。実際のシステムには、正常系以外に異常系や準正常系と呼ばれる部分があり、これらの部分はシステムによっては全体の8割近くを占めているとも言われる。また呼処理以外にオペレーション系などの処理もある。オペレーション系は呼処理

よりは規模も大きく、処理も複雑であることが多い。ソフトウェアの生産性や信頼性を左右するのはむしろこれら準正常系やオペレーション系の部分であるといっても過言ではない。本節ではこれらについて簡単に説明する。

準正常という言葉は耳馴染みがない方が多かろうと思う。例をひいて説明する。発信者が発信行為を途中で中止して、受話器を置いた場合を想定する。システム側からみると、端末からオフフック信号の後にダイヤル数字が送られてくるのが、いわゆる正常系である。この場合は、ダイヤル数字の代わりにオンフック信号が送られてきたという状況である。システムは、このような事態を想定して設計されなければならない。これは、正常系ではなく、またシステムが異常や故障状態である「異常系」でもないのだから、準正常系と呼ばれる。また、準正常のトリガとなるイベントを準正常イベント、準正常イベントに続く処理を準正常処理と言う。

別の例をあげよう。無線基地局では、移動端末との接続に際して通話チャンネルを選択しようとするが、そのとき、空きチャンネルがない場合がある。一般に、この例のようなリソース確保の失敗の場合も、システム自体の異常ではないので、準正常である。

以上の例から察せられるように、準正常イベントには2種類ある。前者の例のように、ある時間区間にいつ起こるかが分からないイベント（通常利用者の挙動に由来する）と、後者のようにある処理の結果（通常失敗を意味する）として得られるイベントである。前者はいわゆる割込イベントに相当する。また、後者には、来るべき応答が来ない場合、すなわちタイムアウトも含む。準正常イベントに続く準正常処理としては、確保中のリソースを解放して呼を終了させ

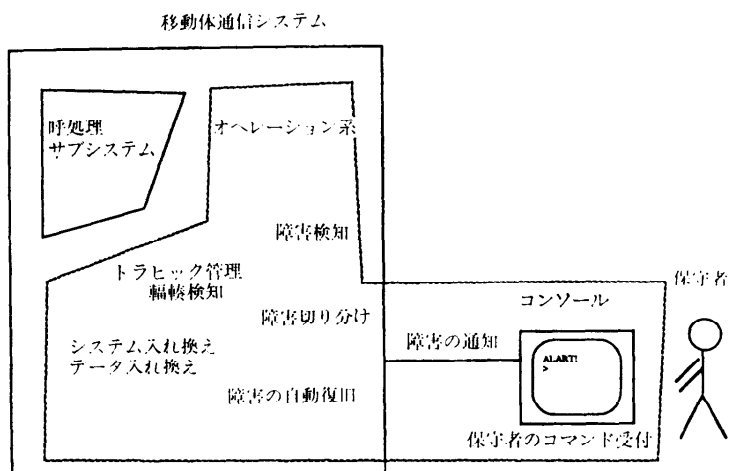


図-12 オペレーション系

るものが一般であるが、呼終了の前に何度かリトライを行うものなどの処理パターンもある。

さて、実際のシステム開発で準正常イベントを決めようとする、なかなか難しい。一般に完全無欠なシステムを設計しようとする、準正常イベントは無限に考えねばならない。まず、システムがオープンであるという意味で、割込の準正常イベントは無限種類考える。また、準正常処理を実行中にも準正常イベントを考慮しなければならないため、準正常処理のネストが無限になる。たとえば、準正常処理で確保リソースを解放中に、解放に失敗し、さらにその準正常処理を実行中にまた別の準正常イベントが起こるというような場合である。どこまで準正常を実現するかは、システムの安全性、準正常イベントの生起頻度、開発工期などを総合的に考慮して、決定する。

サービスの妥当性確認などの目的でプロトタイプに重点がおかれる場合を除いて、一般に仕様には、正常系以外に準正常系も含めて記述するほうがよい。準正常イベントは、正常系部分に対し影響を与えることがあるからである。言い替えると、正常系のみを仕様記述して、仕様を検証し、確定させても、準正常イベントを考慮すると、正常系に手が入る場合がある。現在の仕様記述技術は準正常系に対して、必ずしも書きやすい記述法を提供しているとは言えない。これについては、3.2 で述べる。

オペレーション系について、簡単に一言だけ触れておく。オペレーション系は監視系とも言わ

れ、システム自体の保守・運用・管理を行うためのサブシステムである(図-12)。障害をコンソールに通知し、保守作業者の復旧作業の支援を行ったり、またあるときには障害の自動復旧なども行う。また、システムやシステムデータの入れ換えなどの日常的な保守作業支援も行う。保守作業者のインタフェース以外の部分は、ハードウェア依存の処理も多く、まだまだモデルの整理が必要である。

3. 現状の仕様記述技術の適用上の課題

さて、前章では移動体通信システムの概要について述べた。本章では、仕様記述という立場から移動体通信システムをとらえなおし、それに現状の仕様記述技術がどのように適用できるかについて解説する。なお、ここでは呼処理についてのみを対象とする。

3.1 仕様記述の要件

前章で述べた移動体通信システムの処理を、仕様記述の立場からみると、以下が仕様記述すべき点となる。

I. 移動体通信システムに特有な点

- (1) チャネル切替
- (2) 着信
- (3) 利用者認証
- (4) 無線基地局の役割
- (5) 通信の不確実性

II. 通信システムに共通な点

- (6) タイマ

(7) 準正常割込イベント

(8) 規模の大きさ

これらを順に説明する。

(1) チャネル切替

チャネル切替の契機となるゾーン外移動通知信号はいわゆる割込イベントであり、まずこれを記述できる必要がある。また、レベル測定処理では、無線制御局は複数の無線基地局に対し、レベル測定依頼信号を同報送信（同一内容の信号を同時に送出すること）する。レベル報告の信号の受信に関しては、無線基地局での送信タイミングによって、無線制御局での信号の到着順序がさまざまであるようなプロトコルを記述できる必要がある。次に、担当無線基地局を決定するためには、信号に含まれる受信レベルデータを比較する。数値データの表現とその操作ができることが必要である。

さらに、チャネル切替とは送信先を動的に変更することにほかならない。これについては、(4)で詳述する。

(2) 着 信

着信においては、複数の無線基地局への着信信号の一斉呼出を行う。これはチャネル切替同様、同報通信である。また、移動端末は担当無線基地局を判断後、着信応答信号に無線基地局のIDをのせ、無線制御局はその値によって、どこに通話路を設定するかを決定する。その記述のためには、信号データの定義ができることが必要である。

(3) 利用者認証

利用者認証では、データベース上に登録されているパスワードと入力されたパスワードの一致を調べる。データ構造とその操作を定義できる必要がある。

(4) 無線基地局の役割

無線基地局は一つの呼に対して、二つの役割をもっている。ある場合には移動端末の無線通信を行う担当無線基地局としての役割であり、ある場合には担当無線基地局の隣接局としての役割である。ある特定の無線基地局についてみると、この役割は端末の移動に従って、動的に変更される。このためには、まず役割を区別して記述できることが前提となる。ドキュメントとしてのだけの仕様記述であれば、それで十分であるが、シミュレー

ションやコード生成など実行可能な仕様記述としては、さらに動的に役割を変更できるような記述が望まれる。

一般にこの役割を記述するには、無線基地局を一つのプロセスとして、その中で複数の役割を記述する方法と、一つの役割を一つのプロセスとみなす方法がある。前者では役割を区別して記述することが難しいため、後者の方法で議論する。この場合、動的な役割変更はプロセスの動的な切替、すなわち生成/消滅という問題となる。またプロセスの切替の際には、内部データやチャネルの引継ぎを行う必要もある。逆に、役割を変更したプロセスと通信している他プロセスからみると、通信先を動的に決定するという問題となる。

(5) 通信の不確実性

移動体通信システムでは、信号送受の成功/不成功が環境に左右されやすいため、他の通信システムに比べ、信号送受の失敗の際、リトライを行うことが多い。

II. 通信システムに共通な点

(6) タ イ マ

通信システムでは、入力を無限時間待つということとはほとんどない。たとえば、オンフック後ダイヤル数字が一定時間内に到着しない場合、システムは自動的に接続を切る。つまり、入力待ちの状態では、通常必ずタイマを設定しておく。

(7) 準正常割込イベント

2.3で説明したように、通信システムにおいては、ユーザの切断処理、装置からの異常信号など、準正常割込イベントは数も多い。このような割込イベントを記述できる必要がある。

(8) 規模の大きさ

一般に通信システムのように規模が大きいシステムの記述には、モジュール化機能は不可欠である。さらにモジュール間の階層的な記述も必要である。

以上をまとめて、移動体通信システムの処理を以下のようなプリミティブに分解した²⁾。個々のプリミティブの組合せにより、処理全体が記述可能と考えられる(図-13)。

a) 基本処理 ←(5)

プロセス間の信号の送信/受信、入力信号による分岐は、プロトコルの基本的な処理として、まず最低限必要である。また、リトライを記述す

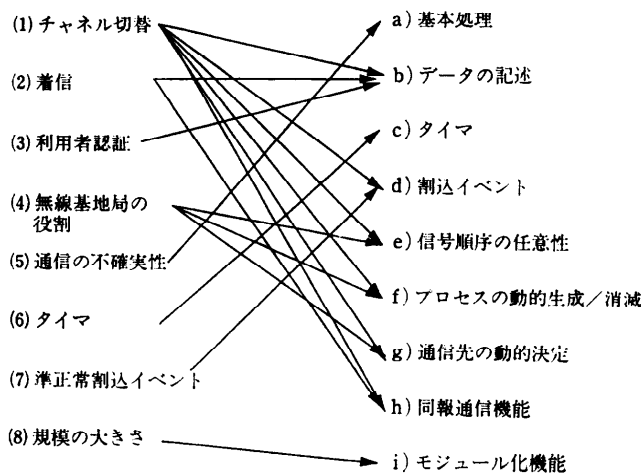


図-13 仕様記述すべき処理とその記述に必要なプリミティブ

るためには、制御のループも記述できる必要がある。図-7～図-11のシーケンス図で記述できるレベルがこの基本処理のレベルである。

b) データの記述 ←(1)(2)(3)

移動体通信システムでは、電話番号などを含む端末情報や、ゾーン情報などのデータ構造を定義し、それを送受する必要がある。また、データの値によって、分岐が行われることもある。

c) タイマ ←(6)

d) 割込イベント ←(1)(7)

単なる大域脱出で十分な場合もあり、また割込処理後、割込地点に戻らなければならない場合もある。割込後リトライを行うような場合は、リトライが成功したら割込地点に復帰するのが通常である。

e) 信号順序の任意性 ←(1)(4)

複数の信号の到着順序が任意であるようなプロトコルを書きたいときがある。つまり、A, B, Cという三つの信号を受けとって処理が進むような場合に、たとえばA, B, Cという受信順序でもC, A, Bという順序でもよいような場合である。

f) プロセスの動的生成/消滅 ←(1)(4)

生成、消滅によって、プロセスの切替を表現したい。内部データやチャネルの引継もできるとよい。

g) 通信先の動的決定 ←(1)(4)

上記f)でプロセスの切替を行うにともない、通信先を動的に決定する必要がある。

h) 同報通信機能 ←(1)(2)

レベル測定依頼信号のように、上記g)も関係

する、つまり、同報通信先のプロセスの集合も動的に決める必要がある場合がある。

i) モジュール化機能 ←(8)

ある程度以上大きなシステムを理解性高く記述するためには、モジュール化機能は不可欠である。

3.2 仕様記述技術の現状

本節では、3.1で述べたプリミティブを現状の仕様記述言語やシステムではどう扱えるかについて解説する。ここでは主に通信プロトコル記述用言語である、SDL^{3),4)}、及び LOTOS^{5),6)}について考える。また、SDL, LOTOS

で記述できない、あるいは記述しにくいものについては、適宜その他の言語などにもふれる。

まず簡単に SDL 及び LOTOS について紹介しておく。SDL は、CCITT において開発、勧告された通信システム向けの形式的仕様記述言語である。拡張状態機械をベースとし、プロトコル記述や通信システムの機能仕様の記述などに適用できる。検証やプログラム生成などの機能を含むツールも数多く開発され、実際の通信システム開発に使用された事例も多い²⁷⁾。一方、LOTOS は ISO により開発、勧告された形式的仕様言語であり、プロセス代数をその数学的モデルとする。プロトコル記述をはじめとして、一般に分散情報処理システムの仕様記述に適用されている。

さて、プリミティブ a)~i) について順に述べる。

a) 基本処理

基本処理は通信用仕様記述言語であれば問題なく記述できる。また、このレベルだけであれば、デッドロック検出などの検証も比較的効率よく行える^{7),8)}。

b) データの記述

SDL や LOTOS では、データは代数的仕様記述、すなわち抽象データ型として記述できる。また、データの送受も問題ない。図-14に SDL のデータ記述部で認証情報を記述した例をあげる。抽象データの操作は、関数と関数の間の関係を等式としてあらわすことにより、規定される。また、Estelle^{9),10)}は、Pascal と同等のデータが記述できる。しかし、データを含む検証となると、現

```

NEWTYPE passwd
LITERALS nil;
OPERATORS
  create: user-name, user-no -> passwd; /* ここでは仮に認証番号をユーザの名前と
                                          ユーザ番号の組で与えられるものとする */
  name: passwd -> user-name;
  num: passwd -> user-no;
  equal: passwd, passwd -> bool;
  .....
AXIOMS
  create(name(p1), num(p1))==p1;
  equal(p1, p2)==and(equal(name(p1), name(p2)), equal(num(p1), num(p2)));
  .....
ENDNEWTYPE passwd;

```

図-14 SDL データ記述部による記述例

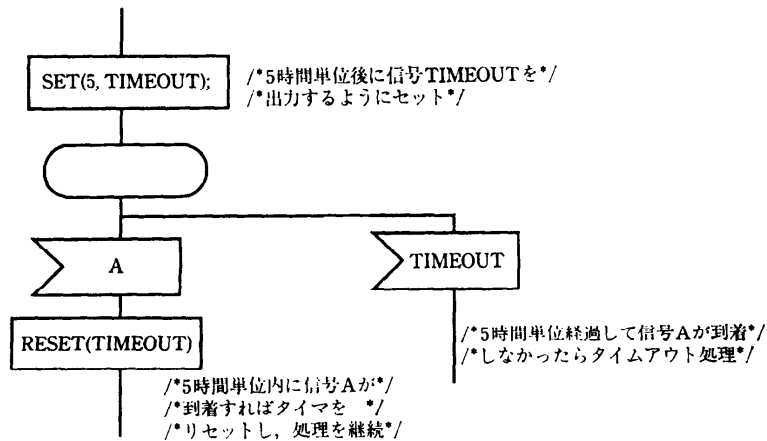


図-15 SDL のタイマ機能

状の技術ではまだまだ難しい。データは往々にして無限の状態をもっており、無限を効率的に扱う方法が今のところないからである。

c) タイマ

SDL はタイマ記述機能をもっている (図-15)。タイマは SET 命令により起動される。SET 命令では、待ち時間とタイムアウトの際に出力される信号名を指定する。タイマは起動後、指定された時間が経過すると指定された信号を発生して、そのプロセスに通知し、停止する。タイムアウトになる前に、RESET 命令が行われれば、タイマはその時点で停止する。

LOTOS など、タイマの概念がない言語でも、図-16 のように時間要素を考慮に入れないプロトコルだけであれば、簡単に記述できるが、時間に依存してシステムの動作が変化するようなものを書こうとした場合にはこれでは不足である。文献 11) では、LOTOS のプロセス代数の枠組の中で、

LOTOS 言語仕様を拡張し、タイマを追加する方法について述べている。

d) 割込イベント

ある時間範囲では常に割込イベントを許すという記述は、現状の仕様記述では結構難しい。まず、SDL で記述した例を図-17 に示す。状態の集合の概念がないため、全ての待ち状態の箇所に割込イベントを記述しなければならない。LOTOS はその点では、簡潔に記述できる (図-18)。しかし、LOTOS では、割込処理後、割込地点に戻るような記述はできない。

また、文献 12) では Z⁽³⁾ (厳密には Object-Z²⁶⁾) による簡単な自動車電話システムの記述例があるが、この例の記述法によれば、割込イベントは非常に簡易に表現できる (図-19)。

e) 信号順序の任意性

任意順序の信号の受信は、SDL, LOTOS で図-20 に示すように記述できる。LOTOS では明快

に記述できる一方、SDL では信号数が増加したときに加速度的に複雑になってしまう。ただし、SDL でも、移動体システムの受信レベル報告の例では、各無線基地局が高々1回ずつしか信号を

```
process timer[set, reset, timeout] :=
  set;
  (reset; stop [ ] timeout; stop)
endproc
```

図-16 LOTOS で記述したタイマ機能

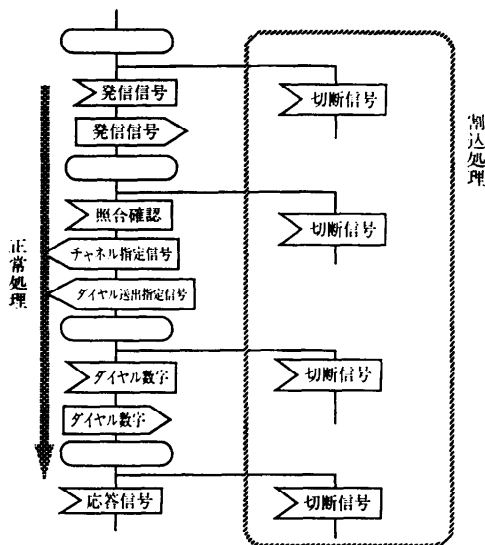


図-17 SDL による割込イベントの記述

送信しないという前提があるので、図-21のように平易に記述することができる。同様のものは LOTOS では、図-22 のようになる。

f) プロセスの動的生成/消滅

図-23 に LOTOS で記述した例を示す。SDL も CREATE 命令を用いて、プロセスを動的に生成することができる。ただし、どちらの言語も内部データの引継や、チャネルやゲート名の引継は、不可能、あるいは可能であっても繁雑である。

g) 通信先の動的決定

基本的に LOTOS は、通信先は静的にしか設定できない。しかし、LOTOS のベース、CCS (Calculus of Communicating Systems)¹⁴⁾の拡張として提案されている、 π カリキュラス¹⁵⁾では、ポートの名前を特にデータ値と区別しないため、ポート名を他のプロセスにパラメタとして渡すこ

```
main[setup, channel, dial, .....]
  >> INT[disconnect] ← 割込処理
where
  process main[setup, channel, dial, ...] :=
    setup; channel; dial; .....
  endproc
  process INT[disconnect] :=
    disconnect; stop;
  endproc
```

図-18 LOTOS による割込イベントの記述

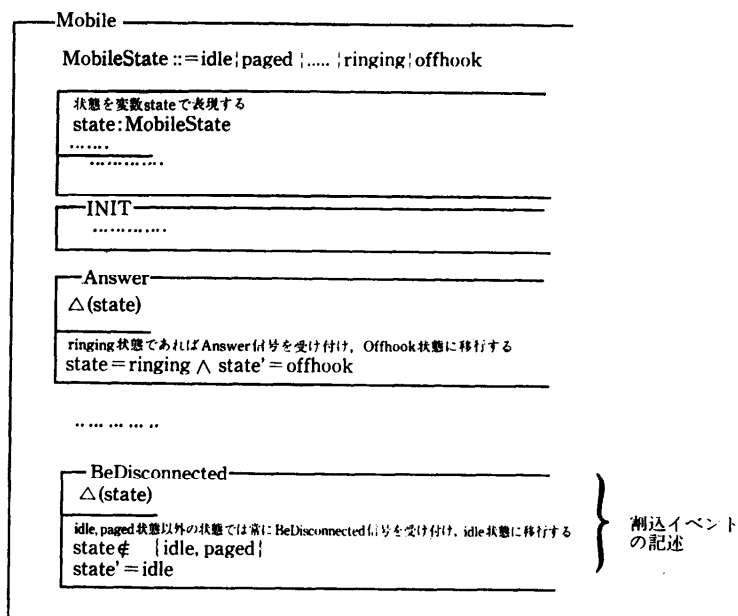


図-19 Z(Object-Z) による割込イベントの記述

とができる。文献16)では、自動車電話システムの通信先動的変更の記述を π カリキュラスを用いて行えることを示した。

また、SDL は通信先を動的に設定することができる(図-24)。通信先はプロセス ID を値とする変数や式を指定できる。しかし、チャンネルなど

の信号路の構成は静的にしかな定義できないため、その範囲でのみ通信先を動的に変えることができるだけである。

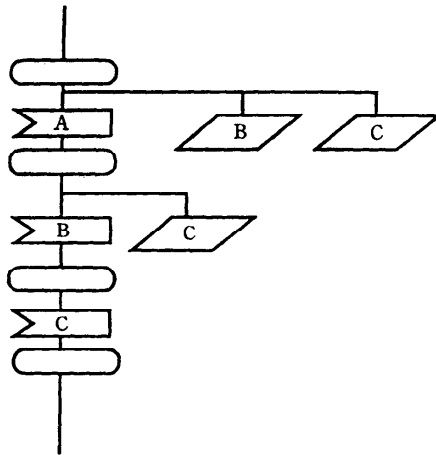
h) 同報通信機能

LOTOS, SDL は同報通信のプリミティブをもたない。

i) モジュール化機能

SDL では、階層的な記述が可能である。システムは一つ以上のブロックからなり、ブロックはいくつかのサブブロック、あるいはいくつかのプロセスからなる。ブロック、プロセス、環境などの間は、チャンネル、シグナルルートなどの信号路を設定できる。図-25 に、それらを表したシステムダイアグラムとブロックダイアグラムの例を示す。ただ、プロセスはそれ自体プロセスを含むことはできず、プロセス自体の階層化機能が欠けているとの意見もある⁴⁾。

一方、LOTOS はプロセスを階層的に記述できるが、モジュールと呼べるものはない。文献17)では、階層的に記述した場合の、動的な振舞い記述部と静的なデータ型記述部とのインタフェースの不都合点を指摘し、LOTOS のモジュール機能追加の提案を行っている。



(a) SDL による記述法

$a \parallel b \parallel c$

(b) LOTOS による記述法

図-20 信号順序任意性の記述

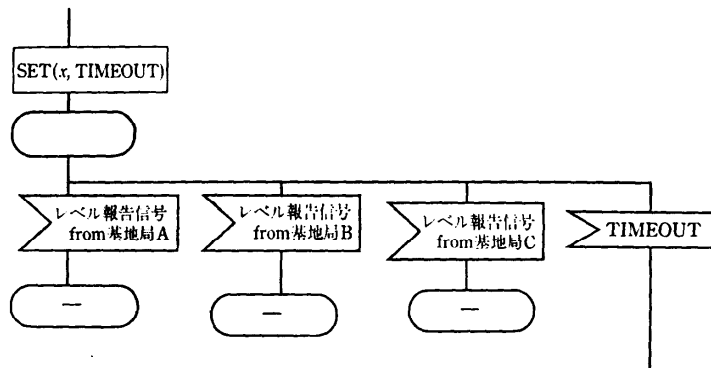


図-21 SDL による信号順序任意性の記述例

```
(level-report[gate-a] ||| level-report[gate-b] ||| level-report[gate-c])
/* 三つのゲートから独立にレベル報告を受信 */
]> (timeout;.....)
...
where
  process level-report[g](x: nat) :=
    g?x; /* レベルxを受け取り */
    ..... /* レベルxに関する処理 */
  endproc
endproc
```

図-22 LOTOS による信号順序任意性の記述例

```

process neighbor-station /*隣接局の定義*/
[examine, measure, report, .....](x: nat):=
.....
examine; /* 制御局からレベル測定依頼 */
measure?x; /* 端末のレベルを測定 */
report!x; /* レベルを制御局へ報告 */
(
  (elected; /* この基地局が担当局として選出された */
   base-station[disc, moved, reqCH]; /* 担当局へ役割切替 */
  )
  [ ]
  (timeout;
   neighbor-station[watch, measure, report, elected, timeout];
  )
)
.....

```

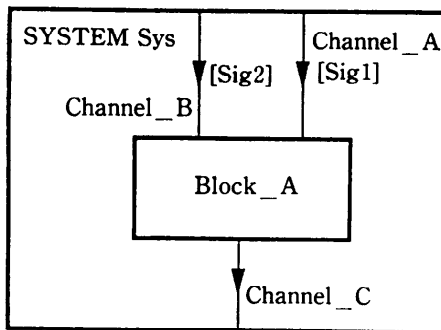
図-23 LOTOS による役割切替の記述

```

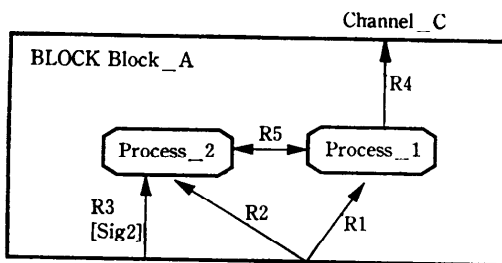
PROCESS Proc(1, 1);
FPAR(to-process PID); /* パラメタとして通信先
                        プロセス ID をもらう */
.....
OUTPUT xxx TO to-process; /* to-process で指定した
                        プロセスに信号 xxx を送信 */
.....

```

図-24 SDL による通信先の動的決定



(a) システムダイアグラム



(b) ブロックダイアグラム

図-25 SDL による階層的記述

4. おわりに

本解説では、形式仕様技術の記述性にポイントをおいて解説したが、このほかにも形式仕様技術

が広く応用されるためには数多くの問題を解決する必要がある。たとえば、方法論、教育、記述支援環境の整備などが問題としてあげられ¹⁸⁾、さまざまな研究がなされている^{19)~21)}。また、文献 23) では、仕様記述研究者コミュニティと開発者コミュニティとの文化のギャップの問題を第一にあげている。

M. Shaw は、文献 22) の中で、産業／開発と科学／理論との融合により進化した段階を「工学」と呼んでおり、現在のソフトウェア工学を真の工学たらしめるためのステップの一つとして、実践と理論の結合の促進をあげている。仕様記述技術もその例外でなく、単に理論のみ偏ることなく、実践的な研究アプローチが今後より必要となると筆者は

考える。特に、通信仕様記述の世界は当面新規で重要なブレイクスルーはない程度の成熟レベルにあり、今後十年は実際の応用に目を向けるべきであるとまで言い切っている人までいる²³⁾。そこでは、文献 24) にも述べられているように、大学をはじめとする研究コミュニティと、企業を中心とする実用化コミュニティとの両社会間での建設的な相互関係がより重要になってくるであろう。

謝辞 情報処理学会「仕様記述の効率的適用と評価」研究グループは、本稿をまとめるきっかけとなりました。また、NTT 移動体通信事業本部 尾上担当課長、安田技師には、移動体通信システムについて、いろいろご教示いただきました。感謝いたします。

参考文献

- 無線技術・音声端末技術，電気通信基礎講座第 3 巻，NTT 中央電気通信学園 (1990)。
- 荒野，堀，伊藤：通信仕様記述の評価，電子情報通信学会 91 年春期全国大会 B-533 (Mar. 1991)。
- CCITT: Functional Specification and Description Language, Recommendation Z. 100 (1989)。
- 若原：SDL 言語の特質と処理系の現状と動向，情報処理，Vol. 31, No. 1, pp. 23-34 (Jan. 1990)。
- ISO: Information Processing Systems—Open Systems Interconnection—LOTOS—A Formal Description Technique Based on the Temporal Ordering of Observational Behavior, ISO 8807 (1989)。

- 6) 高橋, 神長, 白鳥: LOTOS 言語の特質と処理系の現状と動向, 情報処理, Vol. 31, No. 1, pp. 35-46 (Jan. 1990).
- 7) Zafropulo, P. et al.: Towards Analyzing and Synthesizing Protocol, IEEE Trans. of Communications, COM-28, 4, pp. 651-661 (Apr. 1980).
- 8) 伊藤, 市川: 並行プロセスを基本とした交換プログラム仕様の階層的検証法, 電子通信学会論文誌, Vol. J69-B, No. 5, pp. 449-459 (May 1986).
- 9) ISO: Information Processing Systems—Open Systems Interconnection—Estelle—A Formal Description Technique Based on an Extended State Transition Model, ISO/DIS 9074 (1987).
- 10) 岡田: Estelle 言語の特質と処理系の現状と動向, 情報処理, Vol. 31, No. 1, pp. 47-55 (Jan. 1990).
- 11) van Hulzen, W. H. P. et al.: LOTOS Extended with Clocks, Formal Description Techniques II (Vuong, S. T. ed.), pp. 101-115, North-Holland (1990).
- 12) ISO/IEC: Working Document—Architectural Semantics, Specification Techniques and Formalisms, ISO/IEC JTC/SC 21/WG 7 N 314 (Dec. 1990).
- 13) Spivey, J. M.: The Z Notation, Prentice Hall (1989).
- 14) Milner, R.: A Calculus of Communicating Systems, Lecture Notes in Computer Science, Vol. 92, Springer-Verlag (1980).
- 15) Milner, R. et al.: A Calculus of Mobile Process, LFCS Report 89-85, 86, Dept. of Computer Science, Univ. of Edinburgh.
- 16) Orava, F. and Parrow, J.: Algebraic Descriptions of Mobile Networks: An Example, Protocol Specification, Testing and Verification X (Logrippo, L. et al. ed.), pp. 275-291, North-Holland (1990).
- 17) Brinksma, E.: Specification Modules in LOTOS Formal Description Techniques II (Vuong, S. T. ed.), pp. 101-115, North-Holland (1990).
- 18) 二木: ISO における形式記述技法の標準化動向, 情報処理, Vol. 31, No. 1, pp. 3-10 (Jan. 1990).
- 19) Turner, K. J.: A LOTOS-Based Development Strategy, Formal Description Techniques II (Vuong, S. T. ed.), pp. 117-132, North-Holland (1990).
- 20) Martin, M. et al.: The BEST Method for Requirements Capture and Functional Specification, SDL '89: The Language at Work (Faergemand, O. et al. ed.), pp. 23-31, North-Holland (1989).
- 21) Ciccarella, G. and Pignatelli, R.: Experience on Teaching SDL and Using the Language as a Didactic Tool, SDL '89: The Language at Work (Faergemand, O. et al. ed.), pp. 65-73 North-Holland (1989).
- 22) Shaw, M.: Prospects for an Engineering Discipline of Software, IEEE Software, Vol. 8, No. 6, pp. 15-24 (Nov. 1990).
- 23) West, C. H.: The First Ten Years—The Next Ten Years, Protocol Specification, Testing and Verification X (Logrippo, L. et al. ed.), pp. 441-414, North-Holland (1990).
- 24) Chang, C. K. and Trubow, G. B.: Joint Software Research between Industry and Academia, IEEE Software, Vol. 8, No. 6 (Nov. 1990).
- 25) 桑原(監修): 自動車電話, 電子通信学会 (1985).
- 26) R. Duke, P. et al.: The Object-Z Specification Language: Version 1. Technical Report 91-1, Software Verification Research Centre, Dept. of Computer Science, Univ. of Queensland, Australia (May 1991).
- 27) SDL '89 The Language at Work, North Holland, Amsterdam (1989).

(平成4年1月17日受付)



荒野 高志 (正会員)

1962 年生. 1984 年東京大学理学部情報科学科卒業. 1986 年同大学院修士課程修了. 同年日本電信電話(NTT)に入社. 以来, ソフトウェア

研究所において, Common LISP 処理系, 支援系の研究開発, 仕様記述, オブジェクト指向分析/設計法などの研究に従事. 1991 年より, イリノイ大学に客員研究員として渡米中.



山崎 誠一 (正会員)

1958 年生. 1981 年早稲田大学理工学部数学科卒業. 1983 年同大学院博士前期課程修了. 同年日本電信電話公社 (現 NTT) に入社. 以来,

Ada 言語処理系の研究開発, オブジェクト指向分析/設計法などの研究に従事.



伊藤 光恭 (正会員)

1959 年生. 1982 年早稲田大学理工学部数学科卒業. 1984 年同大学院博士前期課程修了. 同年日本電信電話公社 (現 NTT) に入社. 以来,

Ada 言語システムの実用化, 移動通信システムの実用化, オブジェクト指向分析/設計, 分散システム, プロトコル検証などの研究に従事. 電子情報通信学会会員.