

X. 409 支援システムの考察

A Study on X.409 Support System

安部 伸治 中村 能章 中川 透
Shinji ABE Yoshiaki NAKAMURA Toru NAKAGAWA
N T T ヒューマンインタフェース研究所
NTT Human Interface Laboratories

あらまし CCITT-X.409は、人間とのインタフェースを規定する標準記述法、機械間のインタフェースを規定する標準表現法の2つのプロトコルに関する総合的なインタフェースを規定する。X.409は、インタフェースとして、パラメータの組合せおよび判定が容易にできる優れた特徴を有する一方、コーディング規則が複雑なため、インプリメントが難しいという問題がある。そこで、X.409形式のプロトコル開発、システムデバッグなどを統合的に支援するシステムとしてX.409開発支援環境を明かにし、システムを構築する機能の概要を検討する。

Abstract The OSI protocol presentation syntax (CCITT X.409) is applied as the presentation layer protocol in our system. An X.409 support system has been developed for the following functions: ① Supporting design and debugging for the application layer protocols. ② Supporting control data to the online X.409 codec software corresponding to each application. The X.409 support system interprets the edited X.409 notation, checks notation inconsistency, generates the X.409 representation code for debugging, and makes out the data table that controls the processes of the online codec.

1. はじめに

電子メール、データベース・アクセスなどの通信APは、プロトコルに基づいた機能設計が要求される。システムの設計は、プロトタイプリングなどによるフィードバックを考慮して進めるのが一般的である。このため、設計の基本となるプロトコル設計がデバッグも含めて簡易にできる環境が必要となる。

X.409をAPで処理するには、サービスに応じた符号化・復号化を行うコーデックが必須である。しかし、AP対応に設計する方法は、AP内にプロトコル処理を持ち込み、APの機

能の流用性を低下させるとともに、プログラムの設計自動化を妨げる。このため、プロトコル処理をAPと独立に開発できる機能構成が必要である。

本研究では、X.409の文法に基づいたコーデックの機能を検討し、汎用コーデックを中心としたプロトコルの設計、製造、デバッグを統合的に支援できる環境について考察した。次節以降では、まずプロトコルの設計・開発環境を明かにし、設計支援に必要なツールの構成について述べる。さらに、各ツールの機能について概説する。

2. プロトコルの設計支援環境

X. 409は、アプリケーション層におけるプロトコル仕様記述に関する文法を規定する標準記述法 (Standard Representation) と通信データのプレゼンテーション層コーディングの規則を規定する標準表現法 (Standard Notation) からなっている。プロトコル設計は、大別すると仕様設計とAPのプロトコル処理プログラムの設計に分けられる。設計者は、X. 409の標準記述法、標準表現法をそれぞれ仕様設計、プログラム設計で用いる。(図2. 1)

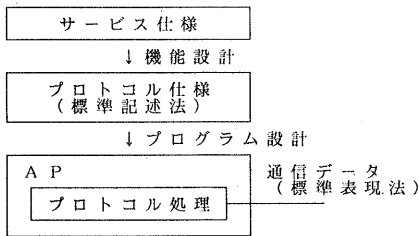


図2. 1 X. 409と設計環境

このように、X. 409は、設計者に2つの文法に沿った設計・開発を要求する。標準記述法は、人間のインタフェースとして重要な意味を持つ。一方、標準表現法は、マシン間のインタフェースであり、一定の規則にしたがった処理が可能である。

標準表現法は、構造化を要求する仕様のため、プログラムの設計・開発の大きな負担となる。このため、標準表現法を設計支援システムで仮想化することにより、設計の簡易化を図ることにする。

プロトコルの設計・開発は、サービス対応のプロトコル設計、インプリメント、デバグの3つの工程からなっている。システム開発の各フェーズにおいて、プロトコルの設計・開発を支援するためのツールとしては、以下のものがある。

- ◎ 仕様エディタ。
- ◎ コーデック。
- ◎ コーデック設計支援ツール。
- ◎ デバグ支援ツール。

プロトコル設計では、X. 409標準記述法

に従った仕様記述を作成する。このため、仕様記述を以後の工程の設計支援ツールに適合した形式で入力する必要がある。

X. 409コーデックに要求される事項として次の2点が挙げられる。

- ① 処理性能が実用上充分であること。
- ② 様々なアプリケーションに対応できること。

①の要求から、コーデックを特定のアプリケーション専用構成することが考えられる。しかしこの場合、サービスプロトコルの仕様変更に対して、コーデックの仕様変更に多大な工数を要することになる。ここでは、②の項目を重視して、コーデックはサービスプロトコル対応に仕様変更が容易な汎用コーデックとすることにした。

一方、プロトコルはAPによって構成が異なるため、汎用コーデックをプロトコル設計対応に専用化する必要がある。このため、コーデックの設計が専用化の作業に置き換えられた。ここでは、プロトコル仕様記述からコーデックの制御テーブルを自動生成するため、コーデックの設計、製造それぞれのフェーズに対応させ、仕様のインタプリタ、コーデックの制御テーブル作成ツールを検討する。

システムの初期デバグは、通信データを用いて単体で実施されるため、多数の通信データが必要となる。一方、通信データを人手でコーデ

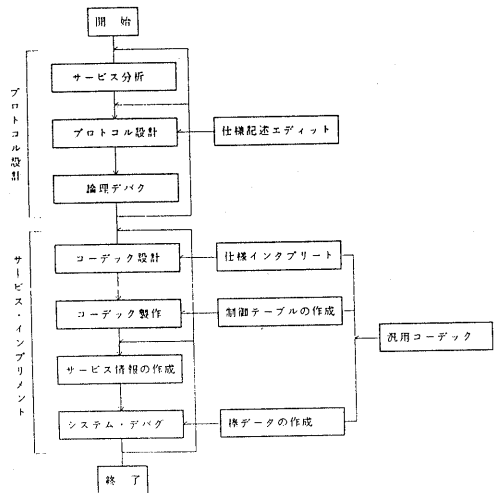


図2. 2 作業フェーズと支援ツールの関係

リングする方法は、多大な工数を要し、システムのデバッグ中に通信データのバグによる2次バグの発生も起きる。したがって、2次バグを回避し、デバッグに要する時間を削減するため、デバッグ用の通信データをインタラクティブに生成するためのツールを検討する。

システムの設計・開発の各工程と各工程の設計支援ツールの関係を図2.2に示す。

3. X.409 支援システムの機能構成

前節で述べた方針に基づき、X.409の設計・開発を支援するシステムとして以下の機能モジュールを考える。X.409支援システムの全体の概略図とデータの流れを図3.1に示す。

- ◎ エディタ。
- ◎ 汎用コーデック。
- ◎ インタプリタ。
- ◎ 制御情報設定ツール。
- ◎ 制御テーブル・ビルダ。
- ◎ 棒データ・ビルダ。

X.409支援ツールは、一連の作業を構成する各々のフェーズを個別に支援する。このため、各ツールは、互いにファイルをインタフェースとして関係付けられている。これにより、各々のツールは個別に機能拡張が行える形式となった。

設計者は、まずサービス要求に従い、プロト

コル仕様をエディタで作成する。作成したプロトコル仕様記述をX.409のインタプリタで構文解析し、エンコードを行ってIDなどの通信に必要な情報を決定する。さらに、仕様で記述できない情報を構文解析の結果を用いてインタラクティブに制御情報設定ツールで決定する。ここまでの作業でサービス仕様に基づいたプロトコル仕様が確定できる。

制御テーブルビルダは、確定したプロトコル仕様に基づいてコーデック制御テーブルを生成する。これにより、サービスプロトコルの変更に対するコーデックの仕様変更は制御テーブルに限定され、コーデック関連のデバッグを解消した。ただし、システム全体のデバッグのため、通信データが必要となる。このため、構文解析の結果より値の入力を行い、システムの単体試験用のテストデータ(棒データ)を棒データ・ビルダで作成する。

X.409開発支援ツールの核に制御テーブルを用いた汎用コーデックを置くことにより、プロトコルの設計・製造を大部分機械化することができた。このため、サービス・プロトコルの開発、プロトコルのインプリメント、システム・デバッグ等に要する時間と工数を大幅に改善することが可能となる。

4. CCITT勧告X.409

CCITT勧告X.409は、メッセージ通信処理システム(MHS)に使用する各種の情

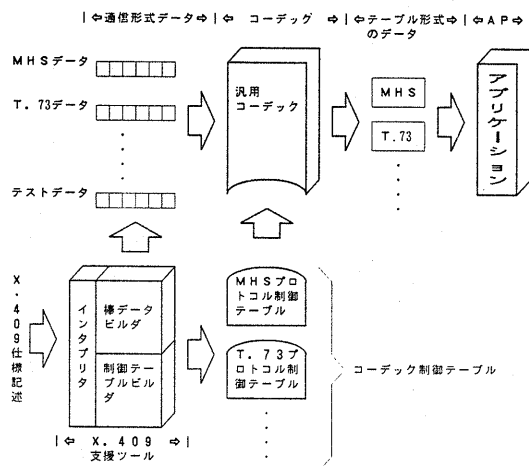


図3.1 X.409 支援システムの構成

報のための転送構文規則を規定している。X. 409は、OSI (OPEN SYSTEM INTERCONNECTION) 参照モデルのアプリケーション層におけるプロトコル仕様記述に関する文法を規定した標準記述法 (Standard Representation) とプレゼンテーション層における通信データのコーディング規則を規定する標準表現法 (Standard Notation) からなっている。X. 409では、各種の情報はその値とともに型 (データタイプ) を持つものとしている。型 (データタイプ) は、"数字", "文字" などのような情報のクラスを記述する。また、値 (データ値) は型 (データタイプ) に従った特定の数値あるいはひとかたまりの文字列である。また、X. 409では、一般的に有用な型として10種類の"組込み型"と7種類の"定義型"を定義している。

4.1 X. 409 標準記述法

標準記述法は、プロトコルの仕様を記述するための言語として、プロトコルのコマンドおよびコマンド・パラメータの仕様の記述に用いられる。標準記述法に沿って記述されたプロトコル仕様は、一般的であり、パラメータの組合せおよび識別に関する情報が網羅されている。標準記述法を用いて記述したプロトコル仕様記述の一例を図4.1に示す。

```

* 個人データ ::= SET {
                                * 氏名,
                                (1) IMPLICIT * 趣味,
                                年齢 (2) IMPLICIT INTEGER }
* 氏名 ::= SEQUENCE {
        姓      PrintableString,
        名      PrintableString }
* 趣味 ::= INTEGER {
        ゴルフ      (0),
        テニス      (1),
        食べること  (2),
        その他      (3) }

```

図4.1 標準記述法によるプロトコル記述例

この例では、"個人データ"が標準記述法に沿って記述されている。ここに現れる情報、"個人

データ", "* 氏名", "趣味", "姓", "名", は、それぞれ"SET", "SEQUENCE", "INTEGER", "PrintableString", "PrintableString"などのデータタイプを持っている。データタイプは、個々の情報間の組合せ (構造) を示すために用いられ、"SET"は、"* 個人データ"が"* 氏名", "* 趣味", "年齢"の集合を代表するものであることを示し、"SEQUENCE"は、"* 氏名"が"姓", "名"の順序付き集合であることを示している。また、"INTEGER"は、それを型として持つ情報の値が整数値であることを示している。

4.2 X. 409 標準表現法

標準表現法は、プロトコルのコーディング規則として、実際の通信路上を伝送するデータの構造を規定している。X. 409標準表現法にしたがったデータは、

```
ID LENGTH DATA ID LENGTH DATA .....
```

の形式で伝送される。ここで"ID"は、データタイプを表すIDであり、"LENGTH"は、データの長さを表すパラメータである。図4.1の例に示したように、ある情報が下位に構造を持つような場合には、"DATA"自身がこのようなデータ構造で表現される場合がある。これらの様子を図4.1の例を用いて標準表現法にしたがって記述した例を図4.2に示す。

```

ID("SET"を表すID)
LENGTH("* 個人データ"全体の長さ)
ID("SEQUENCE"を表すID)
LENGTH("* 氏名"を表す部分の長さ)
ID("PrintableString"を表すID)
LENGTH("姓"を表現する部分の長さ)
DATA("姓"を表現するデータの値)
ID("PrintableString"を表すID)
LENGTH("名"を表す部分の長さ)
DATA("名"を表現するデータの値)

```

図4.2 標準表現法によるデータ構造例

この例では、最初に現れるLENGTHは、"* 個人データ"を表現する全体の長さを意味しており、直後のID以下の全てをDATAとして扱う。同様

に、その次に現れるLENGTHは、“*氏名”を表現する部分(“名”を表現するデータの値まで)をひとつのDATAとして扱うのである。

標準表現法では、ある情報が下位に構造を持つか持たないかは、その情報に対応するIDによって判定が可能な形式となっている。

尚、このような標準表現法にしたがって値が代人されたものを棒データと呼んでいる。

5. 支援ツールの機能概要

ここでは、図2.2の各フェーズにおける人間の作業を自動化するためのツール、X.409支援ツールの機能概要を述べる。

5.1 エディタ

エディタは、プロトコルの仕様記述を編集し、ファイル化する。

5.2 汎用コーデック

X.409形式の通信データは、X.409コーデックによってテーブル形式のデータに変換され、アプリケーションに与えられる。ここで、図中のコーデックでは、T.73, MHSなどのような様々なサービスプロトコルに対して、プロトコルに対応するテーブル(制御テーブル)によって駆動する汎用コーデック形式とした。コーデックをこのように構成することにより、様々なサービスプロトコルに対応してコーデックの仕様変更が容易になる。すなわち、プロトコルの仕様変更に対するコーデックの仕様変更の要求に対して、制御テーブルの差替えのみによって変更を行うことができ、テーブルの作成を自動化することによってコーデックの作成を自動化することが可能となる。

コーデックを汎用化することによって処理性能の低下が予想されるが、汎用コーデックの処理アーキテクチャはサービスプロトコルに依存せず一定であるため、ハードウェア化が可能である。したがって、汎用コーデックをハードウェア化した場合、処理性能の低下は問題にならないと思われる。また、これらの系に対して、サービスプロトコルの開発、システムデバッグなどを系統的に支援するために、X.409開発支援ツールを構成した。

5.3 インタプリタ

図2.2に示した一連の作業フェーズのうち、コーデック設計、コーデック製作、システムデバッグなどの作業フェーズにおいては、これより

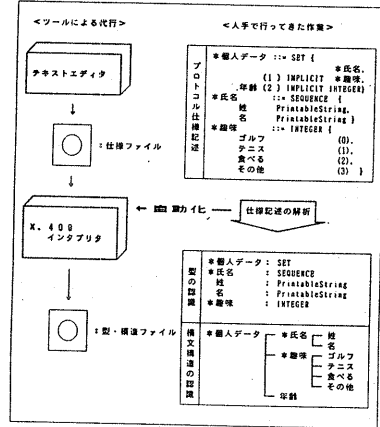


図5.1 構文解析作業とその自動化

先の作業フェーズで設計されたプロトコル仕様記述(図4.1に示したような標準記述法に基づいたもの)を解析する作業を必ず行わなければならない。このような、作業を自動化するためのツールとしてX.409インタプリタを作成した。X.409インタプリタの機能は、X.409標準記述法に沿って記述されたプロトコル仕様記述の構文構造を解析し、構文構造に関する情報を、次節で述べるツールに引渡すためのファイル形式の情報として出力することである。

5.4 制御情報設定ツール

制御情報設定ツールは、X.409標準記述法では、規定できないデータの最大長、要素数などをインタラクティブに設定する。設計者は、プロトコル仕様と制御テーブルの差分を本ツールで設定し、制御テーブルを作成する。

5.5 制御テーブルビルダ

伝送路を経て送られてくるデータが、図4.2で示したような構造を持っているということ、アプリケーションが理解するためには、図4.1に示したプロトコル仕様をアプリケーションに理解させる必要がある。そのために、送

られてきたデータに、プロトコル仕様に関する情報を加えてアプリケーションに引渡す機能としてX.409コーデックを構成した。前述したように、このコーデックは、プロトコル仕

ことである。さらにこのフェーズにおいて重要な作業のひとつとして、プロトコル仕様記述の持つ構文構造を与えられた値とともに表示し、棒データの構造が一目でわかる形式とする必要がある。この作業は、システムデバッグ作業においては重要で、デバッグ作業を行う人間がシステムに対してやり取りを行うデータの構造をすばやく認識できることによって、その負担を軽減することができる。

棒データビルダの機能は、X.409インタプリタによってプロトコル仕様記述を解析した結果を利用して、キーボードから入力された値を代入し棒データを作成することである。さらにプロトコル仕様記述と値を、棒データの構造を認識しやすい形式で表示することである。

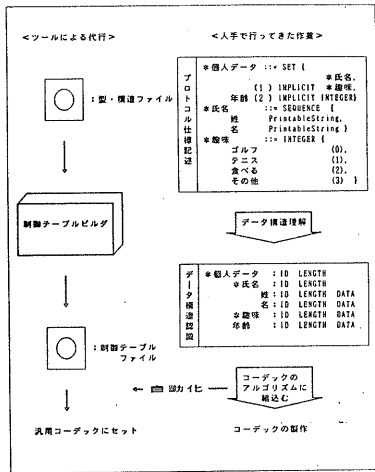


図5.2 コーデック製作作業とその自動化

様に対して汎用性を持った形式として構成した。そのため、コーデックの製作という作業フェーズにおいては、X.409標準記述法に沿って記述されたプロトコル仕様記述を解釈し、その構文構造をアプリケーションが理解できる形式（テーブル形式）に変換するという作業が必要である。この作業を自動化するためのツールとして制御テーブルビルダを作成した。制御テーブルビルダの機能は、設計されたプロトコル仕様に対して、その構文構造を、X.409インタプリタによって解析された結果を用いて制御テーブルに変換することである。

5.6 棒データ・ビルダ

図2.2に示した最終的な作業フェーズとして、システムデバッグという作業がある。ここでは、システムの単体試験を行うため、X.409標準表現法に沿って伝送路上を伝達されるデータ（図4.2の様な形式のデータ）と同じ形式で、値が既知であるテスト用データを作成しなければならない。この作業は、プロトコル仕様記述の構文構造を解析して値を与え、そのプロトコル仕様に適応する形式のデータ構造を持った棒データ（図4.2のような）を作成する

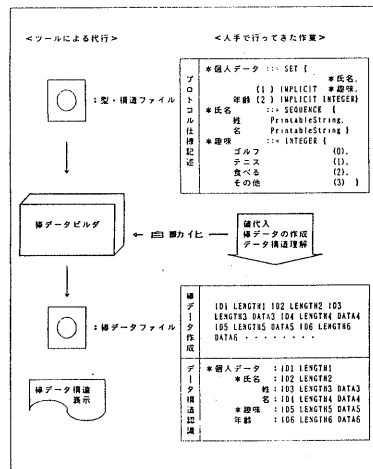


図5.3 棒データ作成作業とその自動化

6. あとがき

本稿では、X.409ベースのプロトコルの設計、開発を支援するシステムの構成法の検討を行った。本稿で明らかにしたX.409開発支援システムは、サービスプロトコルの設計、開発の過程で人間が行ういくつかの作業を、それぞれ個別の機能として自動化したものである。

口頭ご指導いただく安田画像メディア部長、岸本主幹員、笠原主任員に深謝致します。

参考文献

X.409 Message handling Systems : Presentation Transfer Syntax and Notation.