

STEP実用化を容易にするHLDAIの研究と実装

Research and Development of High Level Data Access Interface (HLDAI) for STEP Implementation of Application Object Level

あらまし

本プロジェクトでは、STEP(Standard for the Exchange of Product model data)データをオブジェクトレベルで扱う新しいアプローチを提案している。それは、STEPのAIMデータに直接アクセスするのではなく、HLDAI(High Level Data Access Interface)というアプリケーションのビューを持つユーザに親しみやすいインタフェースを用いるものである。本システムの原理は宣言的なマッピング言語の技術に立脚している。つまりHLDAI生成機能によって、新たに定義されたAPM(Application Program Model)とAIM(Application Interpreted Model)間のマッピング記述がなされる。つぎにこのマッピング記述をコンパイルすることによって、HLDAI部品が自動的に創成され、アプリケーション開発者に提供される。この研究は通産省・情報処理振興事業協会の「企業間高度電子商取引推進事業」プロジェクトの一環として(財)日本情報処理開発協会・STEP推進センターが請け負い、(株)富士通九州システムエンジニアリングが開発したものである。

Abstract

This project proposes a new user-friendly approach with an application view to handling STEP data based on the application object level, instead of STEP AIM using HLDAI. The basic principles of system operation are based on a declarative mapping language technique. The HLDAI generation function creates a mapping description of the schema between a newly defined Application Program Model (APM) and AIM. This description is compiled to automatically generate HLDAI parts for application developers. This paper describes the research conducted by JSTEP for a public project on electronic commerce under the auspices of Japanese Ministry of International Trade and Industry.

井上 和(いのうえ たかし)



1968年九州大学理学部物理学科卒。同年佐世保重工(株)入社。造船設計にて基本計画に従事。1981年(株)富士通九州システムエンジニアリング入社。以来1993年までCAD/CAMシステムの開発に従事。現在STEP研究開発に専任。

岩本卓也(いわたもと たくや)



1988年東京理科大学理学部応用数学科卒。同年(株)富士通九州システムエンジニアリング入社。以来CAD/CAMシステム開発部門にてデータ解析パッケージ、Gui構築支援ツール等の開発に従事。1996年よりSTEP関係の業務に従事。

山本雄樹(やまもと ゆうき)



1989年山口大学工学部建設工学科卒。同年(株)富士通九州システムエンジニアリング入社。以来CAD/CAMシステム開発部門にて主としてCADデータ交換ソフトウェアの開発に従事。1990年よりSTEPに関する業務を開始。

ま え が き

製品のライフサイクル全体の任意の時点で完全なデータ交換がSTEPで実現されるために、世界的に欧米を中心とした各国で実用化に向けた精力的な実証実験が進められてきている^{(1),(2)}。STEP交換システムの実装技術として、これまでSTEPファイルを使ったデータ交換およびSTEPの標準データアクセスインタフェースであるSDAI(Standard Data Access Interface)を介したアプリケーションとのデータ交換の方法がとられている。しかし、これまでの実装技術は、利用者が各アプリケーション独自の複雑なデータ構造を十分に理解したうえで、SDAIとSTEPの製品データモデルであるAIM(Application Interpreted Model)を意識する必要があるため、交換システムの実装を難しくしているのが実状である。

そこで、本研究開発では、各アプリケーション利用者がSDAIを全く意識せずにSTEPデータへの実装をより効率的に行えるための新しいインタフェースHLDAI(High Level Data Access Interface)部品を自動生成することにより、STEPデータへのアクセスを業務レベルで可能にする仕組みを提供することを目的とする。

本システムでは、STEPのAIMとアプリケーションのスキーマであるAPM(Application Program Model)を新たに定義し、両者の関係を記述したマッピング定義を行う。そして、この3者を入力として処理することにより、インタフェースの関数群(HLDAI部品)を生成するものである。実証実験では、CADデータとAP202⁽³⁾間およびPDM(Product Data Management)データとAP203⁽⁴⁾間とのAIMデータ交換の有効性を確認することを目標とする。本稿では、HLDAIを適用した新しいSTEP実装技術に関する報告を行う。

本研究開発は、通産省・情報処理振興事業協会の企業間高度電子商取引推進事業の一環として、財団法人日本情報処理開発協会・STEP推進センターが請け負い、(株)富士通九州システムエンジニアリングが行ったものである。

STEP 実装技術の現状

STEPとは

STEPは“Standard for the Exchange of Product model data”の略である。正式名称は、ISO10303「産業用オートメーションシステムおよびその統合・製品データの表現および交換」といい、製品データのデジタル表現およびその交換に関する国際標準規格である。この規格は、1984年から審議が開始され、1994年12月に第一版が発行された。STEPは生産システム全般において、製品の全ライフ

サイクルを通し、製品に関する技術情報を製品データとして計算機が処理可能な形式で表現し、異なるシステム間で製品データの交換と共有を可能とする。さらに、システムに独立な形式で製品データの蓄積を実現しようとするものである。

STEPと他のデータ交換手段の相違

CADデータ交換の手段として米国標準IGES、デファクト標準DXFなどがある。これらのデータ交換規約とSTEPを比較した場合、つぎのような違いがある。IGESやDXFでは、形状データや注記データ、表示データといったCADシステムが扱うデータしか交換できない。これに対しSTEPは、その正式名称が示すとおり、製品のライフサイクル全体にわたって必要となる情報を対象とする。NCデータ、部品表データ、材料、誤差などが含まれる。専門的⁽⁵⁾には、まずEXPRESS言語⁽⁶⁾と呼ばれる情報モデルの記述言語の存在である。IGES仕様は英文のドキュメントとして、人間が読むために書かれている。一方、STEP仕様は厳密に定義されるEXPRESS言語と英文で記述されている。製品データは、EXPRESS言語により表現定義が行われる。このEXPRESS言語は計算機で解釈可能であり、仕様の冗長性が排除される。また、EXPRESS言語で記述されたスキーマを図式表記する手段として、EXPRESS-Gが存在し、これもSTEP仕様定義に利用される。

第二に、STEPでは特定の応用分野を表すアプリケーションプロトコル⁽⁷⁾(以下、AP)ごとに仕様を定義できる。APIは2次元製図、形状と関係付けを持つ製図、製品構成と製品管理情報、自動車、プラント、建設、造船など種々の分野にわたる。STEPでは、膨大な応用分野の規格を分業で開発できなければならず、これをAPの概念により実現している。EXPRESS言語で記述された各AP仕様をAIMと呼び、STEPでのデータ交換はこのAIM単位に行う。

第三に、STEPはリソースとAPの階層構造を持つ。リソースは基本的な製品データを表現したものであり、各AIMから引用される。このような階層構造とすることによりAP間、すなわち、応用分野間でデータの共有が可能でデータ構造(またはデータの仕組み)を規定する。

STEPの実装方式

【STEPファイルを使ったデータ交換】

STEPでは、データベース仕様をSTEPファイルとして規定し、データ交換を実現できる。これはIGESやDXFの場合と同様である。すなわち、CAD AとCAD B間でデータ交換するとき、STEPファイルを介在させる。この場合、プリプロセッサ、ポストプロセッサをCADシステム側で用意する必要がある。

【SDAIを介したデータ共有と蓄積】

STEPでは、標準のSTEPデータアクセスインタフェースであるSDAI[®]を規定し、STEPデータベースを格納庫として、AIMに沿った製品データをCADやPDMなど各種のシステムで共有することにより、統合環境を実現することが可能である。SDAIは、製品データを保管し共有するためのAPI (Application Programming Interface) として位置づけられる。

HLDAI の提案と開発

マッピングとEXPRESS-X

HLDAIに入る前に、マッピングとEXPRESS-Xについて述べる。システム開発においてはCOBOLやCのような高級手続き言語を用いソースコードを作成して計算機とインタフェースを取る方法が最も一般的である。この場合、あらゆる種類の手続き処理が可能になるという特徴を持っている反面、開発手順が複雑になり開発に要する時間や環境が大がかりになるという欠点を有している。

そこで、出現したのが宣言型言語による処理である。STEPはもともとEXPRESSの強力で厳密なスキーマ定義を根底にしている。このため、宣言型言語で処理するにはきわめて都合の良い環境にある。このような背景から近年、にわかにかこの宣言型言語によるスキーマ変換、つまりデータ交換の優位性が議論され始めてきた。ISO/STEPの場ではこれを通称マッピングと呼び、そのための宣言型言語をEXPRESSを拡張するという意味を含めてEXPRESS-Xと呼んで検討を続けている。⁽⁹⁾

HLDAIの目的と概要

前章で述べたように、SDAIはEXPRESS言語データベースを扱うインタフェース群である。これをSTEP実装者が直接業務レベルで利用するには、生産性の面、使い勝手において効率的ではない。そこで、本研究ではSTEP実装者がSTEPデータへのアクセスを業務レベルで可能にする仕組みを提供することを目的とする。

STEP実装者は、図-1に示すように本研究開発で自動生成されるHLDAI部品を利用することにより、従来のSDAIおよびAIMスキーマの構造を意識することなしにSTEPデータへのアクセスを行えるようになる。STEP実装者は、HLDAI利用の際に、CADやPDMなどのアプリケーションのデータ構造定義を簡単に理解できるレベルに置き換えたAPMスキーマ構造をもとに、STEPデータ交換システムを構築することになる。

開発後の実証実験では、APMの対象をIGESファイルおよびMICRO CADAM^(注1)図面データのテキスト形式であるBMIファイルとするが、その両者のデータ構造を包含し

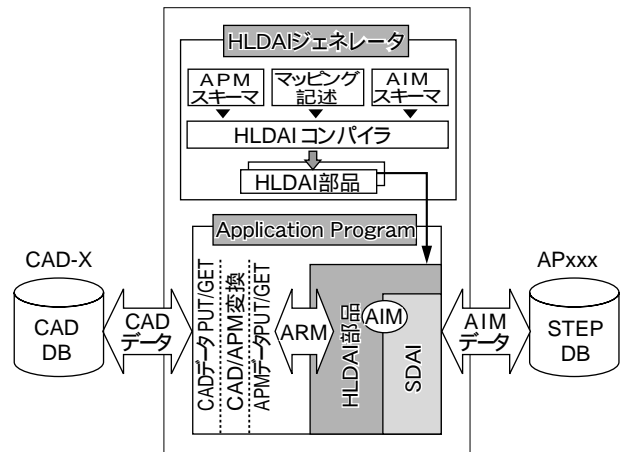


図-1 HLDAIを使ったSTEP DBへのアクセスの仕組み
Fig.1-Mechanism of access to STEP DB using HLDAI.

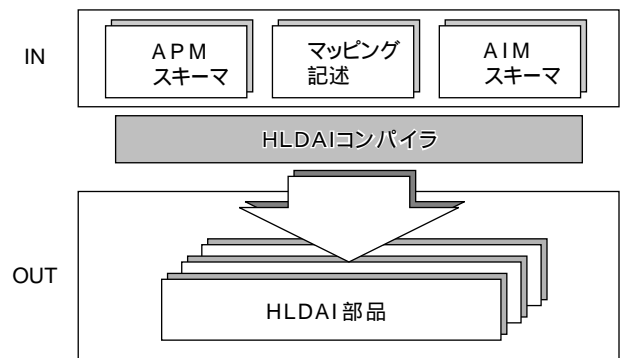


図-2 HLDAIジェネレータ機能
Fig.2-Function of HLDAI generator.

たスキーマとしてAPMを一意に定義する。したがって、IGESファイルおよびBMIファイルの変換とも同じHLDAI部品を使うことになり、IGES用あるいはBMI用と、それぞれを意識した作りを排除できる。

システム動作環境は、つぎのとおりである。

- (1) OS : Windows 95^(注2)またはWindows NT4.0^(注2)
- (2) CPU : Pentium150^(注3) MHz以上
- (3) 主メモリ : 64 Mバイト以上
- (4) ディスク容量 : 2.0 Gバイト以上

HLDAIジェネレータ

HLDAI部品は、HLDAIジェネレータにより自動生成される。HLDAIジェネレータは図-2に示すように、APMスキーマとAIMスキーマおよび両者間のマッピング記述を入力とし、APMエンティティ単位のインタフェースであるHLDAI部品を自動生成するコンパイラ機能である。マッピング言語によりAPMスキーマとAIMスキーマの対

(注1) MICRO CADAMは、キャダムシステム社が専用使用権を所有する登録商標。

(注2) 米国Microsoft, Inc. の米国およびその他の国における登録商標。

(注3) Intel社の登録商標。

応付けを行う。HLDAIジェネレータ機能ではAPMとAIM間のマッピング記述が最も重要な技術となる。

さて、AIMスキーマ、APMスキーマ、マッピング記述を入力とし、HLDAIジェネレータに処理させると、C++で記述されたSDAIへアクセスする(APMスキーマで定義されたプリミティブな、例えば直線や円などの)HLDAI部品が自動出力される。ユーザは、このHLDAI部品を利用することで容易にSTEPデータへアクセスすることができるようになる。

HLDAIでは、従来の実装実験と比較し、以下に挙げる事が大きな特徴と言える。

- (1) STEP実装者は、HLDAI部品を利用することにより、AIMの複雑な構造およびSDAIの利用方法を理解する負荷をなくすることができる。
- (2) 現状のマッピング言語であるEXPRESS-Xでは不可能であった会話型システムに対してもHLDAI部品を使うことで、実装可能となる。
- (3) AIMデータとAPMデータとの双方向のマッピング(オンラインデータ交換)が定義可能である。

APMスキーマ

前章で説明したAIMエンティティに対応するAPMサンプルを以下に示す。

```
SCHEMA APM ;
```

1. ENTITY A ;
2. Name : STRING ;
3. Point : LIST [1:3] OF REAL ;
4. END_ENTITY ;

```
END_SCHEMA ;
```

APMスキーマは、ISOで規定されるARM(Application Reference Model)では十分に定義されていない部分を補充し、よりCAD・PDMシステムに近いモデルとしてEXPRESS言語で定義される。

HLDAIマッピング定義例

マッピング言語としてこれまで主に、EXPRESS-V、EXPRESS-M、BRITYなどが提案され実証実験されてきた。ISOでは、これらの言語形態を“ EXPRESS-X ”として統一し、国際的に標準化しようという動きにある。しかし、この言語はエンティティ単位での一括変換イメージが強く、会話型システムでのマッピング言語には不向きである。また、同時に双方向のマッピング(オンラインデータ交換)が実現できないのが現状である。

HLDAIではPDMシステムの特長上、会話型でまた属性単位での変換が求められる。また、双方向の変換がリアルタイムに実現される必要性があり、世界標準の

EXPRESS-Xより細かな定義が必要となった。それを補うため、EXPRESS-Xをもとに言語拡張し、その結果をISOでのマッピング言語検討会に提案している状況である。

AIMとAPMのマッピングの記述例を図-3に示す。なお、各記述節の概要を以下に示す。

- (1) 2~5行：AIMスキーマとAPMスキーマを宣言。
- (2) 7行目：APMスキーマエンティティに対応するAIMエンティティの宣言。
- (3) 8~12行：APMのエンティティ(構造)をデータ化した際に、対応するAIMデータを作成するための記述。
- (4) 13~15行：APMエンティティデータを消去した際に、対応するAIMデータを削除するための記述。
- (5) 16~20行：AIM既存モデル内に存在するデータをもとに、APMのデータ群を作成(ビューイング)する記述。
- (6) 21~25行：APMエンティティの特定データ属性値が設定された場合に、対応するAIMのデータ反映を記述。
- (7) 26~30行：APMエンティティの特定データ属性値が参照された場合に、対応するAIMのデータ参照を記述。

HLDAI部品の自動生成

HLDAIジェネレータ機能で自動生成されたHLDAI部品は、C++のクラスライブラリとして利用することができる。HLDAI部品は、APMエンティティ単位に以下のアク

```
1.  SCHEMA_MAP sample01 ;
2.  GLOBAL
3.  DECLARE aim INSTANCE OF associative_draughting ;
4.  DECLARE apm INSTANCE OF APM ;
5.  END_GLOBAL ;
6.  VIEW map_A : apm::A ;
7.  DECLARE $point INSTANCE OF aim::cartesian_point ;
8.  A() ; BEGIN
9.      NEW A.$point ; $point.name := ' ';
10.     $point.coordinates += 1.0 ;
11.     $point.coordinates += 1.0 ;
12. END ;
13. ^A() ; BEGIN
14.     DELETE $point ;
15. END ;
16. FROM( itr : aim::cartesian_point )
17. WHEN TRUE ; BEGIN
18.     $point := itr ;
19.     Name := itr.name ; Point := itr.cartesian_point ;
20. END ;
21. SET CASE $$ OF
22.     Name : $point.name := Name ;
23.     Point : $point.cartesian_point := Point ;
24. END_CASE ;
25. END_SET ;
26. GET CASE $$ OF
27.     Name : Name := $point.name ;
28.     Point : Point := $point.cartesian_point ;
29. END_CASE ;
30. END_GET ;
31. END_VIEW ;
32. END_SCHEMA_MAP ;
```

図-3 マッピング記述の例
Fig.3-Example of mapping description.

```

2次元座標(1.0,1.0)を持つ点のインスタンス生成をサンプルとして以下に記載する。
1. #include "hldai.h"
2. #include "hldaiDataC.h"
3. #include "hldai_A_C.h"
4. int main( int argc, char *argv[] ) {
5.     hldaiSessionC session;
6.     HldaiErrorCode error;
7.     hldai_A_C* aEntity;
8.     HldaiRepositoryC repo( session,"step" );
9.     // データベースのオープン
10.    if( ( error=repo.OpenRepository( ) ) ) {
11.        printf( "error Open Repository\n" );
12.        return ( 1 );
13.    }
14.    // モデルを作成
15.    hldaiModelC model( repo, "Model01","sample01" );
16.    if( ( error=model.CreateModel( ) ) ) {
17.        printf( "error Open Model\n" );
18.        return ( 1 );
19.    }
20.    // インスタンスを生成
21.    aEntity = new hldai_A_C( &model );
22.    // インスタンスに属性を設定
23.    aEntity->Point.Add( 1.0 );
24.    aEntity->Point.Add( 1.0 );
25.    aEntity->Name = "aaa";
26.    // モデルの保存およびデータベースのクローズ
27.    model.Commit( );
28.    return ( 0 );
29. }
    
```

図-4 HLDAI部品の利用例
Fig.4-Example of HLDAI parts usage.

セスメソッドを提供する。

- (1) インスタンス生成
- (2) インスタンス削除
- (3) インスタンス群取得
- (4) 属性値設定
- (5) 属性値取得

「点」のAIMスキーマ, APMスキーマおよびマッピング記述(図-3)を入力とし, HLDAIジェネレータ部品を利用した点エンティティのインスタンス生成例として2次元座標(1.0, 1.0)を持つ点のインスタンス生成をサンプルとして図-4に示す。

HLDAIによるSTEP実装の優位性

本研究開発の目的は, HLDAIの提供によりSTEPデータへのアクセスを業務レベルで可能にすることである。具体的には, 以下のとおり, HLDAIによる実装の優位性あるいは効果を期待する。

アクセス方式の面から

アプリケーション開発者は, STEP DBをアクセスするためのアプリケーションプログラムの開発において, HLDAI部品を呼ぶことによりAIMの構造ならびにSDAIをまったく意識せずにSTEP処理システムを構築することができる。

STEPデータ操作の面から

STEP実装へのHLDAIの適用により, STEPデータの操

マニュアル	SDAIによる実装	(数字の単位は頁) HLDAIによる実装
・AIM(AP202, AP203)	1,227	-
・SDAI	344	-
・EXPRESS	208	62データ定義のみ
・APM(ユーザズガイド)	-	} 100
・HLDAI部品(#)	-	
total	1,779	162
参考 Application Program	1	0.25

図-5 アプリケーション開発者の負担面の有効性
Fig.5-HLDAI validity in load of application developer.

作が容易になる。

アプリケーション開発者は, アプリケーションプログラミングにおいて, HLDAI部品のインタフェースのみを考えればよい。また, HLDAIジェネレータによりHLDAI部品が自動生成されるため, STEP技術者はSDAIインタフェースを意識する必要がない。

アプリケーション開発者の負担の面から

図-5は, STEP実装に当たり, 参照するマニュアルのボリュームについて, 従来のSDAIによる実装の場合とHLDAIを適用した実装の場合とを比較したものである。

従来のSDAIをベースとした実装の場合, AP, SDAI, EXPRESSの約1800頁もの規格書を参照する必要がある。一方, HLDAIによる実装の場合, HLDAIのユーザズガイドに加えEXPRESSの一部の参照でよいことになる。明らかに, HLDAIを適用した実装の場合では, アプリケーション開発者はマニュアル参照の負担が軽くなるのが分かる。

また, 1例として点インスタンスの実装に当たり, SDAIインタフェースを使った場合では71ステップを要する。一方, HLDAI部品を適用した場合では, 29ステップである。この場合, 生成のみの比較であるが, データの参照・更新などのロジックを考慮すると, アプリケーションプログラムの総ステップ数は, SDAIを使った場合の四分の一以下になるものと予測される。

開発体制の面から

HLDAIの適用による作業運営において, アプリケーション開発者とSTEP技術者を明確に分離した開発体制を組むことが可能になる。

アプリケーション開発者によって決定されたAPIに基づき, STEP技術者は, アプリケーション開発者のサポートを受けてAPMスキーマ定義を行う。そして, APMとAIMのマッピング記述を行い, HLDAIジェネレータを使ってHLDAI部品を自動生成させる。アプリケーション開発者は, それらのHLDAI部品を利用してアプリケーションプログラムの開発を行う。

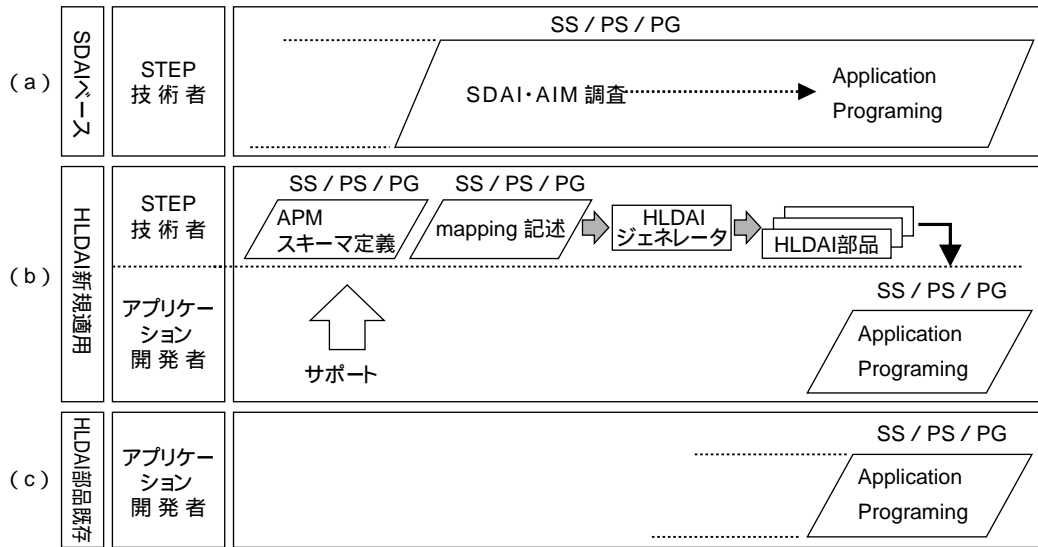


図-6 開発工程からみた有効性
Fig.6-HLDAI validity by process of development.

開発工程の面から

図-6は、(a)従来のSDAIベースによる開発工程、(b)HLDAI新規適用による開発工程、(c)HLDAI部品流用による開発工程について比較したものである。

まず、(a)の場合では、STEP実装者（STEP技術者またはアプリケーション開発者）が、2章の考察で述べたように多大な労力を払いSDAIとAIMを調査する。そして、やっとアプリケーションプログラムの構築に至る。(b)の場合、前章の「開発体制の面」で述べたようにSTEP技術者とアプリケーション開発者が各々の作業分担をもって、HLDAI部品生成の手順にのっとる。この場合、APMスキーマ定義とAPM/AIM間のマッピング記述が主な作業タスクとなる。アプリケーションプログラム作成までのトータル工数の評価については、(a)との比較で実証実験において検証されるべき課題のひとつである。(c)の場合、特定APIに対するHLDAI部品がいったんできてしまえば、アプリケーションプログラムの開発期間は、大幅に短縮されるものと考えられる。

む す び

本研究開発は、より効率的にSTEPデータの実装が行えるための環境整備の第一歩である。今回の開発では実装の対象をIGES、BMIおよびPDMシステムとしたが、今後、すべてのCADシステム・PDMシステムはもちろん、その他のデータベース系への拡張が十分に期待される。今回開発のHLDAIの双方向のマッピング技法は世界でもまだ例がない。今後、この研究が国際的に十分貢献できる題材になることは言うまでもない。

世界的にSTEP規格の標準化が急速に進んでいることは明らかである。このHLDAIを足掛かりに、日本国が技術的にSTEPの先進国としての仲間入りを果たすと同時に、国際標準をもとにした製品化を実現する第一歩としての期待が持てるものと考えられる。

最後に、このHLDAIの開発に当たり、貴重な助言を頂いた中央大学の平岡先生、北海道大学の岸浪先生、東京大学の安藤氏、弊社と技術協定を結びSTEP教育に関して多大な協力を頂いたノルウエーのEPM社に対し、ここに感謝の意を表す。

参考文献

- (1) - : STEPセミナーテキスト . (財)日本情報処理開発協会・STEP推進センター, 1997 .
- (2) 八木, 安藤, 森 : STEPの最新動向 . 日経CG, p.14(1995 . 6)
- (3) ISO 10303-202 Part202 : Application protocol : Associative draughting .
- (4) ISO 10303-203 Part203 : Application protocol : Configuration controlled design .
- (5) 木村ほか : 製品モデル表現とその利用技術-STEP . 日本規格協会, 1995 .
- (6) ISO 10303-11 Part11 : Description methods : The EXPRESS language reference manual, 1994 . 12 . 15 .
- (7) 特集 : プロダクトモデルとCADデータ交換国際標準STEP . 精密工学会誌, 1993 . 3 .
- (8) ISO/DIS 10303-22 Part22 : Implementation methods : Standard Data Access Interface specification .
- (9) ISO TC184/SC4/WG5/N259 EXPRESS-X Reference Manual .