STEP を用いた境界表現ソリッドモデルデータ 交換システムの試作と実験*

小島俊雄** 久貝 稔*** 中村伊知郎† 木村文彦††

An Implementation of B-rep Solid Model Data Exchange System Based on STEP

Toshio KOJIMA, Yutaka KUGAI, Ichiro NAKAMURA and Fumihiko KIMURA

This paper describes a STEP data exchange system developed by a program generation method. In the method, data exchange specifications are defined using EXPRESS data description language as a set of STEP data model definition of an Application Protocol, CAD data model definitions of manifold solid B-rep model and the conversion algorithms between the data models. Principal parts of the data exchange system, using CAD data file as input and output, is generated from the EXPRESS definitions with a common data repository. The data exchange system is proved to be applicable easily to each application protocol in a general manner and effective in its exact implementation of STEP, through the data exchange experiments between two commercial CAD systems.

Key words: STEP, CAD/CAM data exchange, IGES, ISO, data archival, manufacturing software

1. 緒 言

CAD データの交換は、交換対象の製品モデルデータの構造とそのファイル形式の基準を定めて行う方式が実用化されている。しかしながら、図面データを基準としており製品モデルデータの表現として不十分であることや解釈に曖昧さがあることなどに起因する問題点が指摘されている¹⁾。CAD データ交換の国際規格案 STEP²⁾ は、製品モデルデータの完全な交換を目指して審議が進められてきた。そして、STEP を利用した CAD データ交換システム（以下では STEP 交換システムと呼ぶ）もいくつかの方式が研究されている^{3)~5)}。本研究は、STEP を正確に実現している STEP 交換システムの実装方式について試作と実験を通して評価する。試作した STEP 交換システムはデータ構造が類似の 2 種の CAD システムにおける境界表現ソリッドモデルデータを対象としており、一方のデータ交換実験の結果をまとめる。

2. STEP 交換システムの実装方針

STEP はデータ要素の集合からなる製品モデルを 2 次元図面データ、境界表現ソリッドモデルなど交換対象の単位ごとにアプリケーションプロトコル（以下では AP と呼ぶ）として定め、その AP をデータ記述言語 EXPRESS⁶⁾ で形式定義している。STEP 交換システムに関するこれまでの研究では、AP をシステムのデータ構造定義に翻訳する方式が多い³⁾⁴⁾。EXPRESS は STEP 交換システムへの利用を直接意図したものではないが、EXPRESS が形式言語で、その言語トランスレータ（以下では EXPRESS トランスレータと呼ぶ）を実現できるからである。菊地ら⁵⁾ は、CAD データモデルを EXPRESS で記述する CAD データの交換システムについて発表している。これらの研究では、STEP 交換システムにおけるデータ構造の記述のみを取り扱っており、データの検査やデータ変換アルゴリズムなどはプログラム言語で直接記述する。この方式では、実装対象の CAD システムごとに変換プログラムを効率のよいアルゴリズムで記述できるなどの利点がある。しかしながら、個々のプログラミング作業で規格を解釈することから曖昧さが生じる可能性が高いなどの問題点があり、STEP の正確な実現に関する検証が困難である。

筆者ら⁷⁾ は、STEP を用いたデータ交換仕様全体について考察を加え、それを AP、CAD データモデル

* 原稿受付 平成 4 年 7 月 22 日

** 正 会 員 機械技術研究所（つくば市並木 1-2）

*** 正 会 員 シャープ(株)総合デザイン本部（大阪市阿倍野区長池町 22-22）

† 正 会 員 三菱電機(株)情報通信システムエンジニアリングセンター（鎌倉市上町屋 325）

†† 正 会 員 東京大学工学部（東京都文京区本郷 7-3-1）

```

SCHEMA <schema_name>; (* schema definition *)
<external_reference>; ...
ENTITY <entity_name>; (* entity definition *)
  <attr_name: attr_value_class>; ...
  (* attribute_name & data_type, ... *)
WHERE (* constraint on an entity *)
  <constraint_predicate>; ...
END_ENTITY; (* end of entity definition *)
... (* another entity definition *)
FUNCTION ... END_FUNCTION;
... (* function definition *)
RULE ... END_RULE;
... (* constraint on data model *)
END_SCHEMA; (* end of schema definition *)

```

Fig. 1 Product model definition in EXPRESS

およびそれらの間の変換アルゴリズムの組からなるデータモデル集合として定式化した。そして、データ交換仕様全体が EXPRESS で定義できることを示した。本文は、EXPRESS トランスレータを用いて EXPRESS 定義されたデータ交換仕様から STEP 交換システムをプログラム生成方式で試作した結果を示す。この方式においては、仕様の追加や変更に対しては EXPRESS トランスレータを用いて、STEP 交換システム全体を再構成することになるが、STEP を正確に実現するという観点からはデータモデルの意味に関する検査が EXPRESS の利用により直接的に行える特徴を有している。

3. EXPRESS 記述を利用した STEP 交換システム

3.1 構成原理⁷⁾

STEP で AP を記述するために開発された EXPRESS はデータモデルの論理構造を図 1 で例示する形式で記述する言語である。同図において、大文字とセミコロンは終端文字で、それ以外は非終端言語要素を示している。データ要素は ENTITY 単位で定義され、属性は属性名とそのデータ型の対からなる集合で定義される。なお、WHERE 節で属性値に関する拘束条件を記述する。データ要素間の関係などデータモデル全体の性質に関する拘束条件は RULE ブロックで記述する。CAD システムのデータモデルも同様に記述でき⁸⁾、まとめると式 (1) および式 (2) が得られる。一般に、AP や CAD データモデルはデータ要素の階層構造になっており、これらの式ではスキーマ全体の親データ要素の存在を仮定している。

```

SCHEMA step;
(* STEP Application Protocol *)
  <root-step-entity-definition>;
  <entity-definition>; ...
END_SCHEMA

```

(1)

```

SCHEMA cad;
(* CAD Data Definitions *)
  <root-cad-entity-definition>;
  <entity-definition>; ...
END_SCHEMA;

```

(2)

さて、両者のモデルのデータ要素間には、意味的な 1 対 1 の対応が明示的に存在しない場合もある。しかしながら、データモデル全体としての 1 対 1 の対応関係の存在をデータ交換の前提として仮定でき、それを式 (3) および式 (4) で示すように親データ要素間の変換アルゴリズムとして FUNCTION ブロックの関数定義を用いて形式的に表現することが可能である。

```

SCHEMA cs-conv;
  REFERENCE FROM cad (...);
  REFERENCE FROM step (...);
  (* external reference *)
  FUNCTION cs-conv(<root-cad-entity>):
    <root-step-entity>;
  END_FUNCTION;
  ...
END_SCHEMA;

```

(3)

```

SCHEMA sc-conv;
  REFERENCE FROM step (...);
  REFERENCE FROM cad (...);
  (* external reference *)
  FUNCTION sc-conv(<root-step-entity>):
    <root-cad-entity>;
  END_FUNCTION;
  ...
END_SCHEMA;

```

(4)

以上の式 (1) ないし (4) の EXPRESS 記述をデータ交換仕様と考えることができる。この記述から、EXPRESS トランスレータで STEP 交換システムのプログラムを生成することが可能である。EXPRESS トランスレータが正しく作成されているとき、STEP 交換システムが STEP を正確に実装しているか否かは式 (1) ないし (4) の EXPRESS 記述に依存する。すなわち、CAD データから STEP データへの変換、そして、STEP データから CAD データへの変換というループ状のデータ変換による交換データの比較、およびデータが CAD データモデルと AP、それぞれの拘束条件の記述に関して正しいデータであるか否かの交換前後における検査によって STEP 交換システムの検証が間接的に可能である。

3.2 システム構成の具体化

式 (1) ないし (4) の EXPRESS 定義のデータ交換

Fig. 2 Implementation of the data exchange system (program generation and system operation)

仕様からは様々な構成のSTEP交換システムを導出することが可能である。本文ではSTEPを正確に実現するSTEP交換システムを具体化するため、以下のような構成とした。

1) ファイルを介したデータ交換

CADシステムのデータベースアクセスインタフェースは様々である。本文では、STEP交換システムの可搬性を高めるため、CADデータの入出力をCADシステムのデータベースを直接利用するのではなく、テキスト形式で用意するCADデータファイルを利用する方式とした。このとき、STEP交換システムの中でCADシステム個々に直接プログラミングする部分はCADデータ入出力プログラムのみとなる。なお、STEPデータ入出力プログラムはすべてのSTEP交換システムに共通である。

2) データ貯蔵庫の採用

本文では、データ入出力、データ検査、データ変換で構成されるSTEP交換システムに必要な各処理を逐次的かつ段階的に実施するためにデータ貯蔵庫を用いるシステム構成を採用した。すなわち、データ貯蔵庫はEXPRESS記述のデータ交換仕様が動作する環境で、システム構成が簡潔であることやデータ変換前後の総合的なデータ検査が可能になるなどの点を考慮したものである。データ貯蔵庫ではSTEPデータとCADデータがそれぞれのEXPRESS記述に対応する構造の表現形式で共存している。したがって、データの検査やデータ変換の処理はデータ貯蔵庫を共通のデータ領域とする関数処理として逐次的に実行することができる。また、変換精度の検査や変換速度などの

性能評価に利用できる。

4. システム構築の概要

4.1 データ交換仕様のプログラムへの変換

3章で考察したように、データ記述言語EXPRESSによるデータ交換仕様の定義式(1)ないし(4)をシステム仕様記述と見なし、図2に示すEXPRESSトランスレータを利用してSTEP交換システムの主要部をプログラム生成する。EXPRESSトランスレータは、EXPRESS文の字句/構文解析と意味解析を行い、シンボル表、データ要素の定義に従うデータ貯蔵庫の構造定義、データの拘束条件などの整合性の検査プログラムおよびデータ変換プログラムを生成する。シンボル表は、交換ファイルの解釈におけるデータ要素名の文字列の照合などに便利な形式でEXPRESS記述のデータ構造を表現したものである。EXPRESSトランスレータの生成プログラムはC++プログラムである。EXPRESSトランスレータ自身は24277行のCプログラムである。

STEPではEXPRESSによって定義するデータモデルとデータの実現値を表現するSTEP交換ファイル書式とのマッピングを定義しており、EXPRESSではSTEP交換ファイルを直接記述することができない²⁾。このため、以下に示すプログラムを別途作成する。

STEP交換ファイル入出力プログラムは、CADシステム個々の仕様に依存しない共通プログラムである。CADファイル入出力プログラムは個々のCADシステムに依存するために個別に作成する。

これらのファイル入出力プログラムにおいて、ファイルデータの解析や合成で利用するデータ要素名や属性値のデータ型の検査には EXPRESS 記述から得られるシンボル表を利用する。ファイル入出力プログラムとデータ変換で用いられる幾何データ変換プログラムは C++ プログラムライブラリとして準備され、他のシステム構成要素とリンクされて、次節で示す STEP 交換システムのプリプロセッサおよびポストプロセッサとなる。

4.2 データ交換システムの動作

STEP 交換システムの運用時の動作は図 2 に示すように 2 段階で実行される。

1) プリプロセッサ

CAD ファイル入力プログラムは、CAD データファイルを読み込み、データの型検査を行いながら、データ貯蔵庫に CAD データを格納する。そして、すべてのデータを格納した後、データの前参照の問題を処理する。

CAD データ検査プログラムは、データ貯蔵庫内にある CAD データを読み込み、図 1 や式 (1) ないし (4) で示した EXPRESS 記述における WHERE 節や RULE ブロックに記述された拘束条件に関してデータの正しさを検査する。

CAD/STEP データ変換プログラムは、データ貯蔵庫内にある CAD データを読み込み、EXPRESS の関数記述として定義された CAD から STEP データへの変換アルゴリズムに従って、変換を実行し、データ貯蔵庫内に STEP データ構造として格納する。

STEP データ検査プログラムは、データ貯蔵庫内にある STEP データを読み込み、CAD データ検査プログラムと同様の検査を行う。

STEP 交換ファイル出力プログラムは、データ貯蔵庫の STEP データを STEP 交換ファイル仕様で出力する。

2) ポストプロセッサ

ポストプロセッサはプリプロセッサの処理における対象データに関して、CAD データと STEP データを入れ換えた形になる。すなわち、STEP 交換ファイル入力プログラム、STEP データ検査プログラム、STEP/CAD データ変換プログラム、CAD データ検査プログラム、そして CAD ファイル出力プログラムの順に実行される。

5. 境界表現ソリッドモデルの交換実験

5.1 アプリケーションプロトコル仕様

データ交換実験は、manifold-solid-brep ソリッド

```
SCHEMA step;
...
ENTITY edge
SUBTYPE OF ( topology );
  edge_start, edge_end : vertex;
  edge_curve : curve; (* 204 *)
WHERE
  WR1 : dimensionality( edge ) = 1;
  WR2 : qualitative_extent( edge ) = 3;
  WR3 : NOT self_intersect( edge );
  WR4 : arcwise_connected( edge );
  WR5 : empty_set(geometric_intersection
    (edge_start, step_domain(edge)));
  WR6 : empty_set( geometric_intersection
    (edge_end, step_domain(edge)));
  WR7 : (0<=genus(edge)) AND (genus(edge)<=1);
  WR8 : constraints_geometry_edge( edge );
END_ENTITY;
...
END_SCHEMA;

(a) STEP Data Schema (a subset of AP 204 (part))
```

```
SCHEMA cad ;
...
ENTITY winged_edge ; (* winged edge structure *)
  left_loop, right_loop : cad_loop; (* loop *)
  left_edge, right_edge : winged_edge; (* edge *)
  edge_s, edge_e : cad_vertex; (* vertex *)
  c_points : OPTIONAL ARRAY [1:2] OF cad_point ;
WHERE
  WR1 : ((left_edge.edge_e :=: edge_e) AND
    (left_edge.right_loop :=: left_loop)) XOR
    ((left_edge.edge_s :=: edge_e) AND
    (left_edge.left_loop :=: left_loop));
  WR2 : ((right_edge.edge_s :=: edge_s) AND
    (right_edge.left_loop :=: right_loop)) XOR
    ((right_edge.edge_e :=: edge_s) AND
    (right_edge.right_loop :=: right_loop));
  WR3 : edge_s <:> edge_e ;
END_ENTITY ;
...
END_SCHEMA ;

(b) CAD Data Schema (part)
```

Fig. 3 Data schema definition in EXPRESS

モデルに関する AP (以下では AP 204 と呼ぶ)⁹⁾ の原案文書のうち、幾何データとして平面のみを含む部分仕様を対象とした。AP 204 から除外した仕様は曲面・曲線の幾何データ部分で、本文の目的に十分な仕様である。EXPRESS 記述は全体で 671 行である。例として、edge データ要素定義を図 3 (a) に示す。

5.2 CAD システムのデータ構造記述

実験対象とした CAD システムはソリッドモデルを組み込んだ 2 機種の CAD システム (以下では、A および B システムと呼ぶ) である⁸⁾。両システムのデータモデルはウィングドエッジデータ構造を基礎とする境界表現ソリッドモデルで基本構造は同一であり、A システムでは 298 行、B システムでは 164 行の EXPRESS 記述としてそれぞれ定義される。winged-edge データ要素の定義例を図 3 (b) に示す。

Fig. 4 Correspondence between STEP and CAD data

図3(b)に示す winged-edge データ間の相互参照に関する WHERE 節である WR1 ないし WR3 の拘束条件はデータ検査プログラムとして実行される。

5.3 データの対応関係と変換アルゴリズム

図4に示す位相データを中心に、式(4)に対応する STEP データから B システムの winged-edge データ構造への変換を例に説明する。

1) AP 204 の親データ要素 manifold-solid-brep (以下 msb) と CAD システム B の親データ cad-object が 1 個の立体に対応する。ただし、msb は複数の shell 要素から構成される集合になっており、最初の shell が立体の外殻を、それ以外の shell が空洞を示す殻に対応する。実験においては空洞を含むデータを想定していないため、B システム側では shell 要素に対応するデータを省略することができる。cad-object を msb と shell データの 1 個ずつの組に対応させる。また、この条件に相当する拘束条件は変換後のデータ貯蔵庫内のデータに対するデータ検査プログラムとして実現される。

2) AP 204 の closed-shell データはそれを構成する複数の face を属性データとする。AP 204 の face は複数の loop よりなり、外形状を示す loop と内側(穴)を示す 0 個以上の loop を属性データとして定義する。一方、winged-edge データ構造では face をデータ構造として明示的に表現せず、object データが cad-loop, winged-edge, cad-vertex それぞれの集合を属性データとして保有する。すなわち、cad-loop データの属性によって内外周ループを識別し、外周ループを AP 204 の face に対応する位相データとしている。また、このループに内周ループが存在する場合にはその loop データを属性として保有する。さらに、内周ループデータはそれが属する外周データ

を属性として参照することが可能である。そこで closed-shell を構成する face 集合要素を list of cad-loop に変換する。

3) AP 204 の loop はそれを構成する edge をリストとしてもつ。cad-loop データは構成要素のうちの任意の 1 つの winged-edge を属性にもつ。したがって、loop に対応するデータは list of winged-edge となる。そこで、1 つの loop を list of winged-edge データに変換する。また、loop を構成する edge に対して、それに対応する winged-edge データが生成済か否か (edge は 2 つの loop に共有される) を検査する。このような手順で winged-edge データが作成されて左右 cad-loop データがセットされる。

4) AP 204 の edge は 2 つの vertex を属性としてもつ。一方、winged-edge データは 2 つの cad-vertex と左右両側の cad-loop, そして両ループにおける次の winged-edge データを属性としてもつ。AP 204 の vertex と cad-vertex はほぼ対応する概念である。未生成の場合、各 edge データに対して、winged-edge データをデータ貯蔵庫に作成する。winged-edge データの始点・終点属性値が未確定の場合、そのデータをセットする。対応する cad-vertex が未生成の場合には、cad-vertex データを作成して座標値をセットする。以上の winged-edge データ生成プログラムの一部を図5に示す。A システムから AP 204 への変換関数の EXPRESS 記述は 319 行、AP 204 から B システムへの変換関数の EXPRESS 記述は 383 行であった。

6. STEP 交換システムの生成と交換実験

5章で示した、境界表現ソリッドモデリングシステムに対してシステム構築とデータ交換実験を行った。使用した環境は、SUN Sparc Station 2 (28.5 MIPS) である。データ貯蔵庫にはオブジェクト指向データベースシステム¹¹⁾を利用した。表1に交換実験システムの規模を示す。EXPRESS トランスレータによって、ほぼ7倍の行数のC++およびCプログラムが得られており、EXPRESS 記述が一般性をもっていること、また、CAD データ入出力プログラムのみを個別にプログラム化すればよいことなど、システム構築が容易であることを確認できた。データ交換実験に用いた例題データは電話器のハンドセットで、A システムの CAD データファイルは STEP 交換ファイルと類似で 1112 行である。プリプロセサを用いて、図6に一部を示す STEP 交換ファイルに変換した。さらにポストプロセサにより 321 行からなる B

```

SCHEMA sc_conv;
...
FUNCTION edge_to_winged_edge
(e:edge; VAR s:LIST OF vertex;
VAR c:LIST OF cad_vertex) :winged_edge;
LOCAL
  out : winged_edge;
  v_s, v_e : cad_vertex;
  i : INTEGER := 1;
END_LOCAL;
(* check if cad_vertex already exists *)
REPEAT UNTIL i > SIZEOF(s);
  IF (e.edge_start := s[i]) THEN v_s := c[i];
  END_IF;
  IF (e.edge_end := s[i]) THEN v_e := c[i];
  END_IF;
  i = i + 1;
END_REPEAT;
(* create cad_vertex if needed *)
IF EXISTS(v_s) THEN
  v_s := vertex_to_cad_vertex(e.edge_start);
  s := s + e.edge_start;
  c := c + v_s;
END_IF;
IF EXISTS(v_e) THEN
  v_e := vertex_to_cad_vertex(e.edge_end);
  s := s + e.edge_end;
  c := c + v_e;
END_IF;
(* constructor *)
out := winged_edge{?, ?, ?, v_s, v_e, ?};
RETURN(out);
END_FUNCTION;
END_SCHEMA;

```

Fig. 5 STEP to CAD conversion functions (part)

Table 1 Program size of data exchange system

System component	EXPRESS lines	C++/C lines
Preprocessor A		7 614
Data schema	298	
CAD/STEP conv.	319	4 118
System dependent	—	3 496
Postprocessor B		5 889
Data schema	164	
STEP/CAD conv.	383	2 567
System dependent	—	3 322
STEP schema	671	6 100
Generic program	—	9 313

システムのCADファイルに変換した。そして、Bシステム上のデータとAシステムで作成したデータの内容を比較して、データが損失なく交換されたことを確認した。表2に処理時間を、図7にSTEP交換ファイルの表示例をそれぞれ示す。

7. ま と め

EXPRESS言語で定義した仕様記述からプログラム生成によってSTEP交換システムを実現する手法に関して、

```

STEP;
HEADER; FILE_NAME('tel2', ... ); ...; ENDSEC;
DATA; ...
#845=LINE(#299, #592); #846=LINE(#301, #595); ...
#1006=VERTEX(#275); ...
#1162=EDGE(#1003, #984, #848);
#1163=EDGE_LOGICAL_STRUCTURE(#1007, .F.); ...
#1545=EDGE_LOOP((#1459, #1460, #1461, #1462)); ...
#1696=SURFACE_LOGICAL_STRUCTURE(#922, .T.); ...
#1697=CLOSED_SHELL((#1549, #1550, #1551, ...
#1615, #1617, #1618, #1619, #1620, #1621, #1622));
#1698=MANIFOLD_SOLID_BREP(#1697);
ENDSEC;
ENDSTEP;

```

Fig. 6 STEP exchange file (total 1 708 lines)

Table 2 Processing time* of the experiment

Preprocessor	s	Postprocessor	s
CAD in	124	STEP in	150
CAD check	10	STEP check	15
CAD/STEP conv.	56	STEP/CAD conv.	81
STEP check	11	CAD check	6
STEP out	22	CAD out	6

* Elapse time (s)

Fig. 7 Telephone Handset (STEP data) on the CRT

(1) CADデータファイルの利用,
 (2) データ貯蔵庫を利用した、データ入力、データ検査、データ変換、そしてデータ出力の逐次的な処理順序の動作、
 を用いて具体化する方式を提案した。手法はEXPRESS表現に基づいており、システム構築手法として一般性を持ち、STEPを正確に実現していることの確認が容易であることが特徴である。

さらに、境界表現ソリッドモデルのSTEP交換システムを試作し、データ交換実験によってSTEP交換システムの基本性能を評価した結果、

(1) CADシステム依存部の局所化により可搬性が高い、

(2) 構成が簡潔でシステム構築が比較的容易である，
の特徴を有することを確認した。

謝 辞

本研究は，工業技術院標準部機械規格課の標準化調査研究に関連する。関係各位に厚くお礼申し上げます。

参 考 文 献

- 1) 小島俊雄：CADの標準の動向，精密工学会誌，52，5（1986）801。
- 2) 日本設計工学会ほか共催：「CAD標準化とSTEP」セミナーテキスト，（1991），その他ISO/TC 184/SC 4審議資料。
- 3) PDES Inc.：A High-level Architecture for Implementing a PDES/STEP Data Sharing Environment，PTI 017. 03. 00，（1991）。
- 4) CADEX：Ways to STEP Data Exchange Processors，Esprit Project 2195 CADEX，（1991）。
- 5) 菊地慶仁ほか：3階層スキーマ構造に基づくCADデータ交換，精密工学会誌，58，1（1992）145。
- 6) ISO/TC 184/SC 4 N 64：The EXPRESS Language，（1990）。
- 7) 小島俊雄ほか：STEPデータ交換システムの基本設計，1992年度精密工学会春季大会学術講演会講演論文集，（1992）71。
- 8) NICOGRAPH：プラントデータ交換システムの標準化に関する調査研究成果報告書，（1991）。
- 9) Mechanical Design Using Boundary Representation（AP 204），ISO/TC 184/SC 4/WG 3 N 59，（1991）。
- 10) 島谷浩志，千代倉弘明編著：3次元CADの基礎と応用，共立出版，（1991）31。
- 11) VERSANT：OODBMS Reference Manual，（1991）。

変換効率 12% 世界最高アモルファス太陽電池達成

新エネルギー産業技術総合開発機構（NEDO）はこのほど 10 cm 角のアモルファス（非晶質）シリコン太陽電池で 12.0% の実効変換効率を達成したと発表した。

従来の 11.1% を上回り，世界最高の効率といわれている。今後は低コストで生産する技術やアモルファス太陽電池に特有の劣化を抑える技術に開発の主眼を移し実用化を目指す。

通産省工業技術院によるサンシャイン計画の一環として NEDO が三洋電機に委託して開発を進めてきた。

基板の上にアモルファスシリコンを透明電極で挟んだ単層のセルを並べ，各セルを 14 個直列につないで約 13 V の電圧を取り出した。

変換効率を改善できたのは，① 高真空中で原料ガスの精密な制御ができる装置を用い，光を吸収する i 層の品質を高めたこと，② i 層と p 層の間に設けた界面層を水素プラズマにさらして欠陥を低減，両層を電氣的にうまく接続した——などが理由である。

すでに開発していた p 層の特性改善技術，加工性に

優れたレーザによるパターン形成技術などを組み合わせたものである。

アモルファス太陽電池は結晶シリコンを用いるタイプに比べ面積の大きなものを比較的安く作れる半面，変換効率が低く，使用中にアモルファスシリコンが結晶化して特性が劣化するのが難点であった。

今回のデータは変換効率では理論値に近い水準を達成したが，屋外で 1 年間使用した場合，効率が 10～15% 程度劣化するなど，まだ改良の余地はある。

また，コスト面では年間 100 MW の生産を想定した場合，1 W 当たりの発電コストは 630 円程度になるという。

このため，今後は変換効率だけでなく，劣化性能の向上やコスト低減に向けた技術開発に力を入れる方針である。

通産省は 2000 年に太陽電池の発電コストを 1 W 当たり 100～200 円に下げることが目標にしている。

（服部 敏夫）

編集後記

新しい年度になり，会社にも学校にも新人の姿が目立つころである。この号が出る時期にはかなり慣れてはきているだろうが，それでもまだ不安と期待が交錯した状態がいくばくかは残っているだろう。十何年か前は自分もそうだった，夢もいっぱいあった，と彼らを見るたび，ちょっぴりうらやましくなる。研究に年齢は関係ない，そこは経験の差で勝負とは思ってはみても，若々しい発想を見せられたりすると，うーんとうなってしまう。何だか愚痴わいてしまったが，これは近ごろの若者どもに負けてたまるかという自分を鼓舞する裏返しの表現である。彼らと協調し，互いに切磋琢磨し，新しいことを学んで研究開発に

良い結果を残したい，とつくづく考える今日このごろである。

本誌の編集委員も，5 月から任期を終えた人の代わりに新しい人が入ってくる。新鮮な発想を出していただいて会誌を今以上に魅力あるものにしていきたいものである。現在の編集委員の構成をみると，企業の方，および若年層がやや少ないように見受けられる。会員の比率で企業の方がどの程度かはよく知らないが，もう少し増やして先端の実用化研究をアピールし，会誌に反映できるようになったらいいのではないかと。我と思う方はぜひ来年度からでも立候補していただきたい。「特集」や「グラビアとインタビュー」記事のテーマ選択で，世の中を見渡そうという意欲が湧いてくることうけあいである。

（西本克史）