ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 10303-512— 2008

Системы автоматизации производства и их интеграция

ПРЕДСТАВЛЕНИЕ ДАННЫХ ОБ ИЗДЕЛИИ И ОБМЕН ЭТИМИ ДАННЫМИ

Часть 512

Прикладные интерпретированные конструкции. Многогранное граничное представление

ISO 10303-512:1999
Industrial automation systems and integration — Product data representation and exchange — Part 512: Application interpreted construct: Faceted boundary representation
(IDT)

Издание официальное

БЗ 2—2008/13

Москва
Стандартинформ
2008

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 ПОДГОТОВЛЕН Государственным научным учреждением «Центральный научно-исследовательский и опытно-конструкторский институт робототехники и технической кибернетики» на основе собственного аутентичного перевода стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 459 «Информационная поддержка жизненного цикла изделий»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 11 марта 2008 г. № 39-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 10303-512:1999 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 512. Прикладные интерпретированные конструкции. Многогранное граничное представление» (ISO 10303-512:1999 «Industrial automation systems and integration — Product data representation and exchange — Part 512: Application interpreted construct: Faceted boundary representation»).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении F

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2008

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины и определения . . . 2
3.1 Термины, определенные в ИСО 10303-1 . . . 2
3.2 Термины, определенные в ИСО 10303-42 . . . 2
3.3 Термин, определенный в ИСО 10303-202 . . . 2
3.4 Другие определения . . . 2
4 Сокращенный листинг на языке EXPRESS . . . 3
4.1 Основные понятия и допущения . . . 4
4.2 Определение объекта faceted_brep_shape_representation схемы aic_faceted_brep . . . 4
Приложение A (обязательное) Сокращенное наименование объекта . . . 7
Приложение B (обязательное) Регистрация информационного объекта . . . 8
Приложение C (справочное) Машинно-интерпретируемые листинги . . . 9
Приложение D (справочное) EXPRESS-G диаграммы . . . 10
Приложение E (справочное) Требования соответствия ПИК и цели испытаний . . . 13
Приложение F (справочное) Сведения о соответствии национальных стандартов Российской Федерации ссылочным международным стандартам . . . 26

Введение

Стандарты комплекса ИСО 10303 распространяются на компьютерное представление информации об изделиях и обмен данными об изделиях. Их целью является обеспечение нейтрального механизма, способного описывать изделия на всем протяжении их жизненного цикла. Этот механизм применим не только для нейтрального обмена файлами, но является также основой для реализации и совместного доступа к базам данных об изделиях и организации архивирования.

Стандарты комплекса ИСО 10303 представляют собой набор отдельно издаваемых стандартов (частей). Стандарты данного комплекса относятся к одной из следующих тематических групп: методы описания, методы реализации, методология и основы аттестационного тестирования, интегрированные обобщенные ресурсы, интегрированные прикладные ресурсы, прикладные протоколы, комплекты абстрактных тестов, прикладные интерпретированные конструкции и прикладные модули. Настоящий стандарт входит в группу прикладных интерпретированных конструкций.

Прикладная интерпретированная конструкция (ПИК) обеспечивает логическую группировку интерпретированных конструкций, поддерживающих конкретную функциональность для использования данных об изделии в разнообразных прикладных контекстах. Интерпретированная конструкция представляет собой обычную интерпретацию интегрированных ресурсов, поддерживающую требования совместного использования информации прикладными протоколами.

Настоящий стандарт определяет прикладную интерпретированную конструкцию для описания граничного представления твердого тела с плоскими гранями и неявной топологией.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы автоматизации производства и их интеграция

ПРЕДСТАВЛЕНИЕ ДАННЫХ ОБ ИЗДЕЛИИ И ОБМЕН ЭТИМИ ДАННЫМИ

Часть 512

Прикладные интерпретированные конструкции. Многогранное граничное представление

Industrial automation systems and integration. Product data representation and exchange.
Part 512. Application interpreted constructions. Faceted boundary representation

Дата введения — 2008—09—01

1 Область применения

Настоящий стандарт определяет интерпретацию обобщенных ресурсов, обеспечивающую соответствие требованиям к определению трехмерной формы посредством модели граничного представления с плоскими гранями и неявно заданными прямолинейными ребрами и представлению одной или нескольких таких форм объектом **faceted_brep_shape_representation**.

Требования настоящего стандарта распространяются на:

- трехмерную геометрию;
- граничные представления;
- B-rep модели;
- многогранные B-rep модели;
- поликонтуры;
- неограниченную геометрию;
- использование топологии для ограничения геометрических объектов;
- геометрические преобразования.

Требования настоящего стандарта не распространяются на:

- двумерную геометрию;
- кривые;
- явные определения ребер;
- поверхности, не являющиеся плоскостями;
- вынесенные кривые и поверхности.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие международные стандарты:

ИСО/МЭК 8824-1:1995 Информационные технологии. Взаимосвязь открытых систем. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации

ИСО 10303-1:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 1. Общие представления и основополагающие принципы

ИСО 10303-11:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS

ИСО 10303-12:1997 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 12. Методы описания. Справочное руководство по языку EXPRESS-I

Издание официальное

ИСО 10303-41:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 41. Интегрированные обобщенные ресурсы. Основы описания и поддержки изделий

ИСО 10303-42:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 42. Интегрированные обобщенные ресурсы. Геометрическое и топологическое представление

ИСО 10303-43:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 43. Интегрированные обобщенные ресурсы. Структуры представлений

ИСО 10303-202:1996 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 202. Прикладные протоколы. Ассоциативные чертежи

3 Термины и определения

3.1 Термины, определенные в ИСО 10303-1

В настоящем стандарте применены следующие термины:

- **приложение** (application);
- **прикладной контекст** (application context);
- **прикладной протокол**, ПП (application protocol; AP);
- **метод реализации** (implementation method);
- **интегрированный ресурс** (integrated resource);
- **интерпретация** (interpretation);
- **данные об изделии** (product data).

3.2 Термины, определенные в ИСО 10303-42

В настоящем стандарте применены следующие термины:

- **линейно связанный** (arcwise connected);
- **граница** (boundary);
- **пространственная модель граничного представления**, B-rep модель (boundary representation solid model, B-rep);
- **ограничения** (bounds);
- **координатное пространство** (coordinate space);
- **кривая** (curve);
- **ориентируемый** (orientable);
- **поверхность** (surface);
- **топологическое значение** (topological sense).

3.3 Термин, определенный в ИСО 10303-202

В настоящем стандарте применен следующий термин:

прикладная интерпретированная конструкция, ПИК (application interpreted construct, AIC): Логическая группировка интерпретируемых конструкций, которая поддерживает определенную функцию для использования данных об изделии в контекстах различных приложений.

3.4 Другие определения

В настоящем стандарте также применены следующие термины с соответствующими определениями:

представление формы многогранной B-rep моделью (faceted B-rep shape representation): Представление формы, состоящее из одной или более B-rep моделей многообразия многогранных тел. Каждая входящая в него B-rep модель должна иметь плоские грани и ребра, определенные в неявной форме.

многогранная B-rep модель многообразия (manifold faceted B-rep): Линейно связанное многогранное тело, характеризующееся тем, что при размещении центра очень маленькой сферы в любой точке на границе этого тела ее внутренний объем делится точно на две области. При этом одна из этих областей находится внутри тела, а другая — снаружи.

поликонтур (polyloop): Контур, расположенный на плоской грани и состоящий из линейных сегментов. Геометрия ребер и топология определены в неявном виде списком точек в декартовом пространстве.

4 Сокращенный листинг на языке EXPRESS

В настоящем разделе определена EXPRESS-схема, в которой используются элементы интегрированных ресурсов и содержатся типы, конкретизации объектов и функции, относящиеся к настоящему стандарту.

Примечание — В интегрированных ресурсах допускается существование подтипов и элементов списков выбора, не импортированных в данную ПИК. Такие конструкции исключают из дерева подтипов или из списка выбора посредством правил неявного интерфейса, определенных в ИСО 10303-11. Ссылки на исключенные конструкции находятся вне области применения данной ПИК. В некоторых случаях исключаются все элементы списка выбора. Поскольку ПИК предназначены для реализации в контексте прикладного протокола, элементы списка выбора будут определяться областью применения прикладного протокола.

Данная прикладная интерпретированная конструкция предоставляет непротиворечивое множество геометрических и топологических объектов для определения моделей многообразия тел с плоскими гранями, ребрами, определенными в неявной форме, и вершинами. Границы граней в B-rep моделях задаются объектами **poly_loop**, и каждая грань должна иметь внешнюю границу, заданную в явной форме.

Объектом самого верхнего уровня в данной ПИК является **faceted_brep_shape_representation**. Он является конкретизацией объекта **shape_representation** (см. ИСО 10303-41), состоящего из объектов **faceted_brep** и **mapped_item**, определенных как преобразованные копии объектов **faceted_brep**.

EXPRESS-спецификация

```
* )
SCHEMA aic_faceted_brep;

  USE FROM geometry_schema
        (axis2_placement_3d,
         cartesian_point,
         cartesian_transformation_operator_3d,
         elementary_surface,
         plane);

  USE FROM geometric_model_schema
        (brep_with_voids,
         faceted_brep,
         manifold_solid_brep);

  REFERENCE FROM geometric_model_schema(msb_shells);

  USE FROM topology_schema
        (closed_shell,
         connected_face_set,
         face_bound,
         face_outer_bound,
         face_surface,
         oriented_closed_shell,
         poly_loop);

  USE FROM representation_schema(mapped_item);

  USE FROM product_property_representation_schema(shape_representation);
( *
```

Примечания

1 Для объекта **connected_face_set** установлен явный интерфейс, т.е. он включен в списки операторов USE FROM для того, чтобы правила, определенные для объекта **faceted_brep_shape_representation**, имели доступ к

атрибутам этого объекта. При использовании данной ПИК этот объект должен реализовываться только в виде одного из своих подтипов.

2 Для объекта **manifold_solid_brep** установлен явный интерфейс, чтобы обеспечить компиляцию функции **msb_shells**. Для объекта **elementary_surface** установлен явный интерфейс, чтобы позволить объекту **faceted_brep_shape_representation** иметь доступ к атрибутам этого объекта. При использовании данной ПИК эти объекты должны реализовываться только в виде одного из своих подтипов.

3 Схемы, ссылки на которые приведены выше, можно найти в следующих стандартах комплекса ИСО 10303:

geometric_model_schema — ИСО 10303-42;
geometry_schema — ИСО 10303-42,
topology_schema — ИСО 10303-42;
representation_schema — ИСО 10303-43,
product_property_representation_schema — ИСО 10303-41.

4.1 Основные понятия и допущения

Для независимой реализации в схемах прикладных протоколов, в которых используется данная ПИК, предназначены следующие объекты:

- axis2_placement_3d;
- brep_with_voids;
- cartesian_point;
- cartesian_transformation_operator_3d;
- closed_shell;
- direction;
- face_bound;
- face_outer_bound;
- face_surface;
- faceted_brep;
- mapped_item;
- oriented_closed_shell;
- plane;
- poly_loop;
- representation_map.

Прикладной протокол, использующий данную ПИК, должен обеспечивать, чтобы объект **shape_representation** реализовывался как объект **faceted_brep_shape_representation.**

4.2 Определение объекта faceted_brep_shape_representation схемы aic_faceted_brep

Объект **faceted_brep_shape_representation** является подтипом объекта **shape_representation**, в котором элементы представления являются конкретизациями объектов **faceted_brep**. Они отличаются от более общей B-rep модели тем, что у них имеются только плоские грани и неявная геометрия ребер.

EXPRESS-спецификация

```
*)
ENTITY faceted_brep_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1 : SIZEOF (QUERY  (it  <*  items  |
        NOT  (SIZEOF ( [ ' AIC_FACETED_BREP.FACETED_BREP',
            ' AIC_FACETED_BREP.MAPPED_ITEM',
            ' AIC_FACETED_BREP.AXIS2_PLACEMENT_3D' ]  *
             TYPEOF (it) )  =  1 ) ) ) = 0;
  WR2 : SIZEOF (QUERY  (it  <*  items  |
        SIZEOF( [ ' AIC_FACETED_BREP.FACETED_BREP',
        ' AIC_FACETED_BREP.MAPPED_ITEM' ]  *  TYPEOF (it) )  = 1) ) > 0;
  WR3 : SIZEOF (QUERY  (fbrep  <*  QUERY ( it  <*  items  |
        ' AIC_FACETED_BREP.FACETED_BREP' IN  TYPEOF (it) )  |
        NOT (SIZEOF  (QUERY (csh  <* msb_shells (fbrep)  |
        NOT (SIZEOF  (QUERY (fcs  <* csh\connected_face_set.cfs_faces |
```

```
        NOT ( ( 'AIC_FACETED_BREP.FACE_SURFACE ' IN TYPEOF (fcs) ) AND
        ( ( 'AIC_FACETED_BREP.PLANE' IN TYPEOF
              (fcs\ face_surface.face_geometry ) ) AND
        ( 'AIC_FACETED_BREP.CARTESIAN_POINT' IN TYPEOF (
              fcs\face_surface.face_geometry\
                 elementary_surface.position.location))))))
                   = 0) ) ) = 0) ) ) = 0;
  WR4 : SIZEOF (QUERY (fbrep <* QUERY ( it <* items |
        'AIC_FACETED_BREP.FACETED_BREP' IN TYPEOF(it) ) |
        NOT (SIZEOF (QUERY (csh <* msb_shells (fbrep) |
        NOT (SIZEOF (QUERY (fcs <* csh\connected_face_set.cfs_faces |
        NOT (SIZEOF (QUERY (bnds <* fcs.bounds |
          'AIC_FACETED_BREP.FACE_OUTER_BOUND' IN TYPEOF(bnds) ) )
            = 1 ) ) ) = 0 ) ) ) = 0 ) ) ) = 0;
  WR5 : SIZEOF (QUERY (msb <* QUERY (it <* items |
        'AIC_FACETED_BREP.MANIFOLD_SOLID_BREP' IN TYPEOF (it) ) |
        'AIC_FACETED_BREP.ORIENTED_CLOSED_SHELL' IN
           TYPEOF (msb\manifold_solid_brep.outer) ) ) = 0;
  WR6 : SIZEOF (QUERY (brv <* QUERY (it <* items |
        'AIC_FACETED_BREP.BREP_WITH_VOIDS' IN TYPEOF(it) ) |
        NOT (SIZEOF (QUERY (csh <* brv\brep_with_voids.voids |
          csh\oriented_closed_shell.orientation) ) = 0 ) ) ) = 0;
  WR7 : SIZEOF (QUERY (mi <* QUERY (it <* items |
                'AIC_FACETED_BREP.MAPPED_ITEM' IN TYPEOF (it) ) |
        NOT ( 'AIC_FACETED_BREP.FACETED_BREP_SHAPE_REPRESENTATION' IN
          TYPEOF(mi\mapped_item.mapping_source.mapped_representation))))
                = 0;
END_ENTITY;
( *
```

Формальные утверждения

WR1 — атрибут **items** супертипа **representation** должен содержать только объекты **faceted_brep, mapped_item** и **axis2_placement_3d.**

WR2 — по крайней мере один из элементов атрибута **items** должен быть объектом **faceted_brep** или **mapped_item** (см. также WR7).

WR3 — у каждого объекта **faceted_brep** из **items** грани должны быть объектами **face_surface**, ассоциированная поверхность для каждой грани должна быть объектом **plane** и каждый объект **plane** должен использовать объект **cartesian_point** для определения своего местоположения.

Примечание — Вызов функции **msb_shells** в WR3 и в WR4 корректен, так как, несмотря на то, что обобщенным типом аргумента 'fbrep' является **representation_item**, оператором QUERY для 'fbrep' был определен подтип **manifold_solid_brep**.

WR4 — явно заданная внешняя граница должна быть определена для каждого объекта **face** из **faceted_brep**.

WR5 — для каждого объекта **manifold_solid_brep** из **items** атрибут **outer** не должен принадлежать к ориентированному подтипу.

WR6 — если объект **manifold_solid_brep** является также объектом **brep_with_void**, то каждая оболочка из множества **voids** должна быть объектом **oriented_closed_shell** со значением ориентации FALSE.

WR7 — если в **items** присутствует объект **mapped_item**, то объект **mapped_representation** атрибута **mapping_source** должен быть объектом **faceted_brep_shape_representation.**

Примечание — Если объект **cartesian_transformation_operator_3d** включен как объект **mapped_item.mapping_target** с атрибутом **axis2_placement_3d**, соответствующим начальной системе координат как объект **mapped_representation.mapping_origin**, то результирующий объект **mapped_item** является преобразованной копией объекта **faceted_brep_shape_representation**. Точное определение преобразования, включая пе-

ремещение, вращение, масштабирование и, если необходимо, зеркальное отображение, задается оператором преобразования.

EXPRESS-спецификация

```
*)
END_SCHEMA; -- конец схемы AIC_FACETED_BREP
(*
```

Приложение А
(обязательное)

Сокращенное наименование объекта

Сокращенное наименование объекта, установленного в настоящем стандарте, приведено в таблице А.1. Требования к использованию сокращенных наименований содержатся в методах реализации, описанных в соответствующих стандартах комплекса ИСО 10303.

Таблица А.1 — Сокращенное наименование объекта

Полное наименование	Сокращенное наименование
FACETED_BREP_SHAPE_REPRESENTATION	FBSR

Приложение В
(обязательное)

Регистрация информационного объекта

В.1 Обозначение документа

Для обеспечения однозначного обозначения информационного объекта в открытой системе настоящему стандарту присвоен следующий идентификатор объекта:

{ iso standard 10303 part(512) version(1) }

Смысл данного обозначения установлен в ИСО/МЭК 8824-1 и описан в ИСО 10303-1.

В.2 Обозначение схемы

Для обеспечения однозначного обозначения в открытой информационной системе схеме aic_faceted_brep (см. раздел 4) присвоен следующий идентификатор объекта:

{ iso standard 10303 part(512) version(1) object(1) aic-faceted-brep(1) }

Смысл данного обозначения установлен в ИСО/МЭК 8824-1 и описан в ИСО 10303-1.

Приложение С
(справочное)

Машинно-интерпретируемые листинги

В данном приложении приведены ссылки на сайты, на которых находятся листинги наименований объектов на языке EXPRESS и соответствующих сокращенных наименований, установленных в настоящем стандарте. На этих же сайтах находятся листинги всех EXPRESS-схем, установленных в настоящем стандарте, без комментариев и другого поясняющего текста. Эти листинги доступны в машинно-интерпретируемой форме и могут быть получены по следующим адресам URL:

Сокращенные наименования: <http://www.tc184-sc4.org/Short_Names/>

EXPRESS. <http://www.tc184-sc4.org/EXPRESS/>

При невозможности доступа к этим сайтам необходимо обратиться в центральный секретариат ИСО или непосредственно в секретариат ИСО ТК184/ПК4 по адресу электронной почты: sc4sec@ tc184-sc4.org.

Примечание — Информация, представленная в машинно-интерпретируемой форме на указанных выше URL, является справочной. Обязательным является текст настоящего стандарта.

Приложение D
(справочное)

EXPRESS-G диаграммы

EXPRESS-G диаграммы, представленные на рисунках D.1 и D.2, получены из сокращенного листинга, приведенного в разделе 4, с использованием спецификаций интерфейса стандарта ИСО 10303-11. В диаграммах использована графическая нотация EXPRESS-G языка EXPRESS. Описание EXPRESS-G установлено в ИСО 10303-11, приложение D.

Примечания

1 Выбранные типы geometric_set_select, pcurve_or_surface, shell, trimming_select, vector_or_direction импортируются в расширенный листинг ПИК в соответствии с правилами неявных интерфейсов по ИСО 10303-11. В настоящем стандарте эти выбранные типы не используются в других объектах.

2 В правилах неявных интерфейсов ИСО 10303-11 присутствуют некоторые объекты, реализация которых запрещена правилами, относящимися к **faceted_brep_shape_representation**. На EXPRESS-G диаграммах эти объекты отмечены символом «*».

3 С тем, чтобы избежать загромождения рисунков, некоторые производные атрибуты на рисунках D.1 и D.2 опущены.

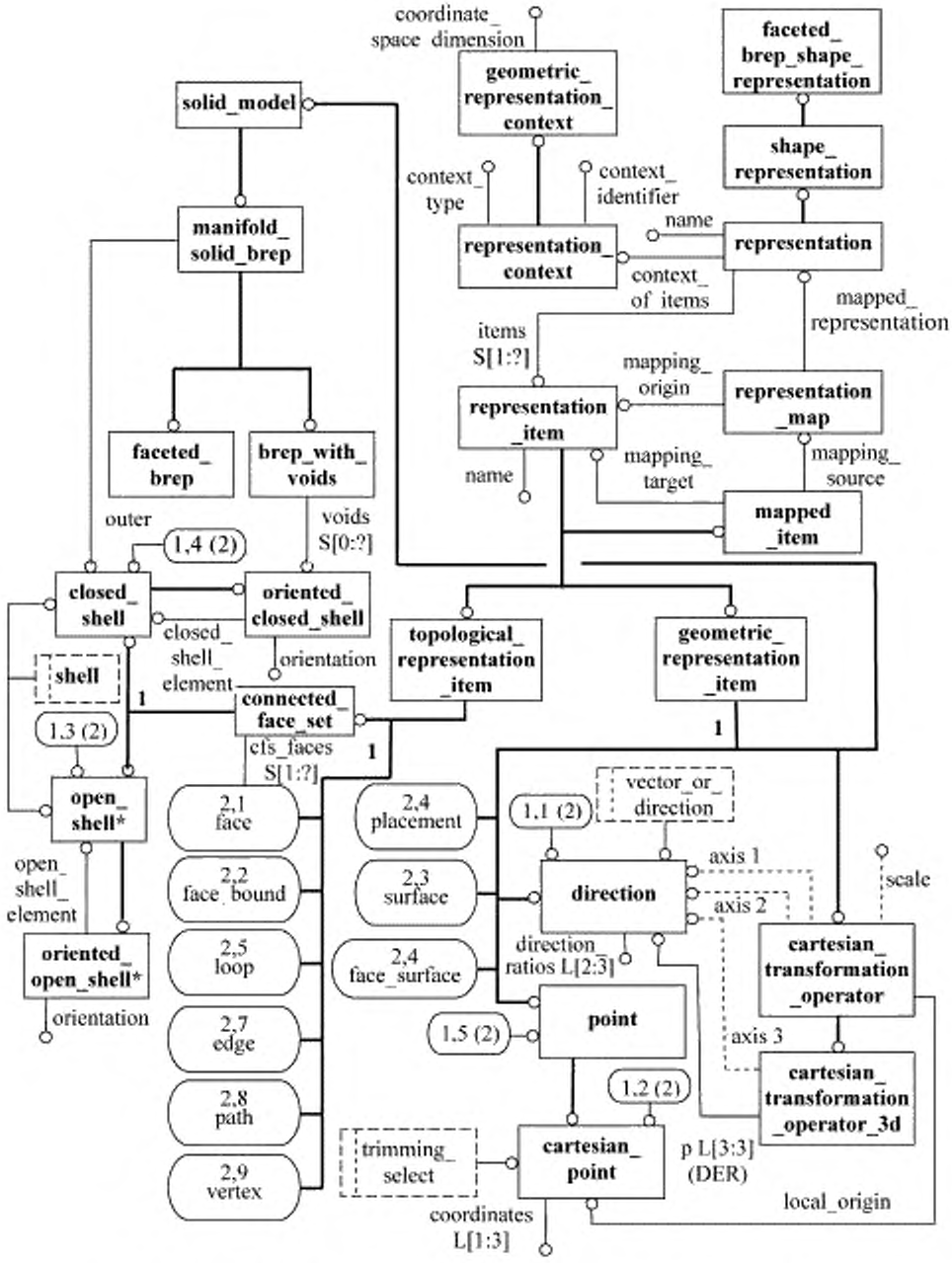

Рисунок D.1 — ПИК **aic_faceted_brep** в формате EXPRESS-G (диаграмма 1 из 2)

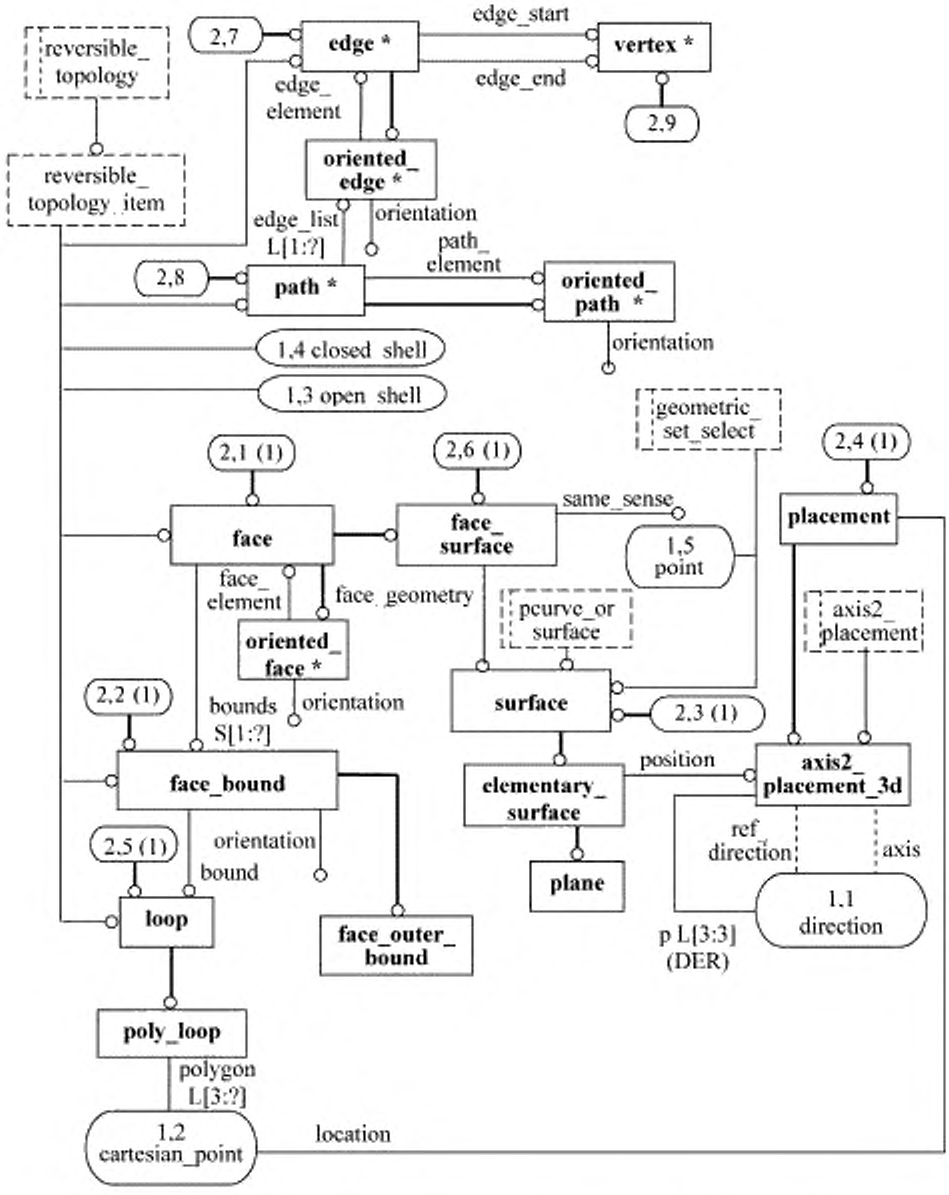

Рисунок D.2 — ПИК **aic_faceted_brep** в формате EXPRESS-G (диаграмма 2 из 2)

Приложение Е
(справочное)

Требования соответствия ПИК и цели испытаний

Е.1 Требования соответствия ПИК: многогранные B-rep модели

Любой прикладной протокол, в котором используется данная ПИК, может потребовать соответствия приведенным ниже требованиям соответствия ПИК при реализации объекта **faceted_brep_shape_representation**.

Соответствие данной ПИК означает, что должны поддерживаться все определяемые типы и типы объектов, присутствующие в EXPRESS-листинге ПИК. Единственным допустимым в рамках контекста данной ПИК использованием экземпляра геометрического или топологического объекта является его использование для определения объекта **faceted_brep_shape_representation**.

Следующие объекты реализуются как часть определения объекта **faceted_brep_shape_representation:**

- axis2_placement_3d;
- brep_with_voids;
- cartesian_point;
- cartesian_transformation_operator_3d;
- closed_shell;
- direction;
- face_bound;
- face_outer_bound;
- face_surface;
- faceted_brep;
- mapped_item,
- oriented_closed_shell,
- plane;
- poly_loop;
- representation_map.

Е.2 Цели тестирования для ПИК многогранной B-rep модели

В настоящем разделе определены цели тестирования для ПИК многогранной B-rep модели. Критерии основаны на конструкциях, определенных в разделе 4 настоящего стандарта.

Примечание — Для любого прикладного протокола, использующего данную ПИК, могут потребоваться дополнительные цели испытаний для объекта 'самого верхнего уровня', который относится к типу **shape_representation**. Кроме того, прикладной протокол потребует определения контекста изделия для этих целей тестирования.

Е.2.1 Объект faceted_brep_shape_representation

Следующие цели тестирования сформулированы, исходя из определения этого объекта:

FB1: Объект **representation** представлен как **shape_representation**, который в свою очередь представлен как **faceted_brep_shape_representation** (см. Е.3.1).

FB2: Объект **faceted_brep_shape_representation** с контекстом представлен как **geometric_context** с элементами **item**, представленными как **faceted_brep** (см. Е.3.1).

FB3: Объект **faceted_brep_shape_representation** с контекстом представлен как **geometric_context** с элементами **item**, представленными как **mapped_item** (см. Е.3.4).

FB4: Объект **faceted_brep_shape_representation** с контекстом представлен как **geometric_context** с двумя или более элементами **item**, представленными как **faceted_brep**, **mapped_item** или **axis2_placement_3d**. При этом по крайней мере одним из них должен быть объект **axis2_placement_3d** (см. Е.3.4).

Е.2.2 Объект faceted_brep

Следующие цели тестирования сформулированы, исходя из определения этого объекта:

FB5: Объект **faceted_brep** с атрибутом **outer** (атрибут **voids** отсутствует) представлен как **closed_shell** (но не как подтип **oriented_closed_shell**) (см. Е.3.1).

FB6: Объект **faceted_brep** и подтип **brep_with_voids** с атрибутом **outer** представлены как **closed_shell**, а множество элементов **voids** представлено множеством, состоящим из одного объекта **oriented_closed_shell** (атрибут **voids** присутствует) (см. Е.3.2).

FB7: Объект **faceted_brep** и подтип **brep_with_voids** с атрибутом **outer** представлены как **closed_shell**, а множество элементов **voids** представлено множеством, состоящим более чем из одного объекта **oriented_closed_shell** (атрибут **voids** присутствует) (см. Е.3.2).

E.2.3 Объект oriented_closed_shell

Следующая цель тестирования сформулирована, исходя из определения этого объекта и ограничений, наложенных на объект **faceted_brep_shape_representation**:

FB8: Значением объекта **oriented_closed_shell** с атрибутом **orientation** является FALSE (см. E.3.2).

E.2.4 Объект closed_shell

Следующая цель тестирования сформулирована, исходя из определения этого объекта и ограничений, наложенных на объект **faceted_brep_shape_representation**:

FB9: Объект **closed_shell** с атрибутом **cfs_faces** представлен как множество, состоящее более чем из одного объекта **face_surface** (см. E. 3.1).

E.2.5 Объект face_surface

Следующие цели тестирования сформулированы, исходя из определения этого объекта и ограничений, наложенных на объект **faceted_brep_shape_representation**:

FB10: Объект **face_surface** с атрибутом **face_geometry** представлен как объект **plane** (см. E.3.1).

FB11: Значением объекта **face_surface** с атрибутом **same_sense** является TRUE (см. E.3.1).

FB12: Значением объекта **face_surface** с **same_sense** является FALSE (см. E.3.3).

E.2.6 Объект face

Следующие цели тестирования сформулированы, исходя из определения этого объекта и ограничений, наложенных на объект **faceted_brep_shape_representation**:

FB13: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds**, представленным множеством, состоящим из одного объекта **face_bound**, представленного как **face_outer_bound** с атрибутом **orientation**, является TRUE (см. E.3.1).

FB14: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds**, представленным множеством, состоящим из одного объекта **face_bound**, представленного как **face_outer_bound** с атрибутом **bound**, представленным как **poly_loop** и **orientation**, является FALSE (см. E.3.3).

FB15: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds**, представленным множеством, состоящим по крайней мере из двух объектов **face_bound**, один из которых является объектом **face_outer_bound**, с атрибутом **bound**, представленным как **poly_loop** и **orientation**, является TRUE (см. E.3.3).

FB16: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds,** представленным множеством, состоящим по крайней мере из двух объектов **face_bound**, один из которых является объектом **face_outer_bound**, с атрибутом **bound**, представленным как **poly_loop** и **orientation**, является FALSE (см. E.3.3).

E.2.7 Объект surface

Следующие критерии сформулированы, исходя из определения этого объекта и ограничений, наложенных на объект **faceted_brep_shape_representation**:

FB17: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** с атрибутом **axis** (см. E.3.1).

FB18: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** без атрибута **axis** (см. E.3.3).

FB19: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** с атрибутом **ref_direction** (см. E.3.1).

FB20: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** без атрибута **ref_direction** (см. E.3.3).

E.2.8 Объект cartesian_transformation_operator_3d

Следующие критерии сформулированы, исходя из определения этого объекта, объекта **mapped_item** и ограничений, наложенных на объект **faceted_brep_shape_representation**:

FB21: Объект **mapped_item** с атрибутом **mapping_target** представлен как **cartesian_transformation_operator_3d** (см. E.3.5).

FB22: Объект **cartesian_transformation_operator** представлен как **cartesian_transformation_operator_3d** с масштабом, представленным значением типа REAL, не равным 1.0 (см. E.3.5).

E.3 Контрольные примеры для ПИК многогранной B-rep модели

E.3.1 Контрольный пример fb1

Контрольный пример fb1 является основным. Объектом тестирования является одиночный тетраэдр, одна из вершин которого расположена в начале системы координат. Вся геометрия задана в явной форме без значений, принимаемых по умолчанию, и без необходимости изменения направлений на обратные. Задание внешней оболочки осуществляется посредством контекста **tetrashell_instance** с использованием исходных параметров.

E.3.1.1 Реализуемые цели тестирования ПИК

FB1: Объект **representation** представлен как **shape_representation**, который в свою очередь представлен как **faceted_brep_shape_representation**.

FB2: Объект **faceted_brep_shape_representation** с контекстом представлен как **geometric_context** с элементами **item**, представленными как **faceted_brep**.

FB5: Объект **faceted_brep** с атрибутом **outer** (атрибут **voids** отсутствует) представлен как **closed_shell** (но не как подтип **oriented_closed_shell**).

FB9: Объект **closed_shell** с атрибутом **cfs_faces** представлен как множество, состоящее более чем из одного объекта **face_surface**.

FB10: Объект **face_surface** с атрибутом **face_geometry** представлен как объект **plane**.

FB11: Значением объекта **face_surface** с атрибутом **same_sense** является TRUE.

FB13: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds**, представленным множеством, состоящим из одного объекта **face_bound**, представленного как **face_outer_bound** с атрибутом **orientation**, является TRUE.

FB17: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** с атрибутом **axis**.

FB19: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** с атрибутом **ref_direction**.

E.3.1.2 Спецификация ввода в постпроцессор

Предметом тестирования является объект **faceted_brep_shape_representation**, состоящий из единственного объекта **faceted_brep**. Объект **faceted_brep** представляет собой тетраэдр, одна из вершин которого расположена в начале системы координат, а смежные ребра направлены вдоль координатных осей. Все грани заданы объектами **poly_loop** и имеют тип **face_surface** с геометрией поверхности, определенной как плоскость. Соответствующее множество размеров определено в приведенной ниже спецификации на языке EXPRESS-I, которая представляет полную спецификацию объекта.

E.3.1.2.1 EXPRESS-I-спецификация fbrep_1

```
*)
TEST_CASE example_fbrep_1; WITH aic_faceted_brep;

  REALIZATION

  LOCAL
    shell_object : closed_shell;
    tetrahedron : faceted_brep;
    fbsr1 : faceted_brep_shape_representation;
    its_units : named_unit;
    prod1_context : representation_context;
  END_LOCAL;

    CALL tetrashell_instance ; -- используются значения по умолчанию,
                               -- поэтому нет WITH
      IMPORT (shell_object := @tetrashell; );
    END_CALL;

      its_units := length_unit ( ) || si_unit ( 'milli', 'metre' );

      prod1_context := geometric_representation_context
                       ( 'context_1', 'context_for_tetrahedron', 3) ||
                       global_unit_assigned_context ( [its_units] );

      tetrahedron := faceted_brep ( 'tetrahedron', shell_object);

      fbsr1 := faceted_brep_shape_representation
                 ( 'fbsr1', [tetrahedron], its_context );

  END_REALIZATION;
END_TEST_CASE;
(*
```

Примечания

1 В этом контрольном примере для определения формы используется самая простая форма контекста **tetrashell_instance** со значениями по умолчанию.

2 Глобальные единицы длины определены в **global_units_assigned_context** как миллиметры.

E.3.1.3 Критерии решения постпроцессора

FB1: Все формальные утверждения (WR) для объекта **faceted_brep_shape_representation** должны быть проверены.

FB2: Созданная модель не должна содержать вершин, граней, кривых, а единицы длины и объект **global_unit_assigned_context** должны быть правильно интерпретированы.

FB5: Объект **faceted_brep** должен быть правильно интерпретирован.

FB9: Атрибут **cfs_faces** должен быть множеством, состоящим из более чем одного объекта **face_surface**; грани должны пересекаться по линиям, соединяющим точки.

FB10: Объект **plane**, определяющий **face_geometry** для каждого объекта **face_surface**, должен проходить через все точки, ограничивающие объект **poly_loop**.

FB11: Перпендикуляр к каждому объекту **plane** должен быть направлен от тела.

FB19: Множество **P**, состоящее из трех осей, для каждого объекта **axis2_placement_3d** должно быть корректно выведено.

E.3.2 Контрольный пример fb2

Контрольный пример fb2 разработан для проверки определения многогранной B-rep модели тела, внутри которого имеется одна или несколько пустот. Контекст **tetrashell_instance** используется с различными параметрами для определения внешней оболочки и оболочек пустот. Результатом является полое четырехгранное тело с расположенной внутри пустотой (или пустотами) той же формы.

Примечание — При необходимости этот тест может быть легко модифицирован для проверки геометрической точности путем изменения параметров с целью определения пустот, расположенных очень близко друг к другу или к внешней оболочке. Однако в данной версии этого контрольного примера возможности для такого вмешательства не предусмотрено.

E.3.2.1 Реализуемые цели тестирования ПИК

FB6: Объект **faceted_brep** и подтип **brep_with_voids** с атрибутом **outer** представлены как **closed_shell**, а множество элементов **voids** представлено множеством, состоящим из одного объекта **oriented_closed_shell** (атрибут **voids** присутствует).

FB7: Объект **faceted_brep** и подтип **brep_with_voids** с атрибутом **outer** представлены как **closed_shell**, а множество элементов **voids** представлено множеством, состоящим более чем из одного объекта **oriented_closed_shell** (атрибут **voids** присутствует).

FB8: Значением объекта **oriented_closed_shell** с атрибутом **orientation** является FALSE.

E.3.2.2 Спецификация ввода в постпроцессор

Предметом тестирования является объект **faceted_brep_shape_representation**, состоящий из единственного объекта **faceted_brep**. Объект **faceted_brep** представляет собой полый тетраэдр, одна из вершин которого расположена в начале системы координат, а смежные ребра направлены вдоль координатных осей. Должны быть созданы два отдельных экземпляра, один из которых имеет одну внутреннюю пустоту, а другой — две. Оболочка каждой пустоты по форме аналогична внешней оболочке и расположена внутри тела. Пустоты не должны пересекаться друг с другом или внешней оболочкой. Соответствующее множество размеров определено в приведенной ниже спецификации на языке EXPRESS-I, которая представляет полную спецификацию объекта.

E.3.2.2.1 EXPRESS-I-спецификация fbrep_2

```
*)
TEST_CASE example_fbrep_2; WITH faceted_brep_arc;

  REALIZATION

  LOCAL
    shell_object, hollow1, hollow2 : closed_shell;
    void1, void2 : oriented_closed_shell;
    tetra_with_void : manifold_solid_brep;
    tetra_with_voids : manifold_solid_brep;
    fbsr1, fbsr2 : faceted_brep_shape_representation;
    its_context : representation_context;
    its_units : named_unit;
  END_LOCAL;

    CALL tetrashell_instance;   -- используются значения по умолчанию,
                                -- поэтому нет WITH
      IMPORT (shell_object := @tetrashell; );
    END_CALL;

    CALL tetrashell_instance;   -- устанавливаются значения
                                -- размеров (большая пустота)
      IMPORT (hollow1 := @tetrashell; );
      WITH (orc := 20; lx := 50; ly := 50; lz := 50; );
    END_CALL;

    CALL tetrashell_instance;   -- устанавливаются значения
                                -- размеров (малая пустота)
```

```
        IMPORT (hollow2 := @tetrashell; );
        WITH (orc := 5; lx := 20; ly := 20; lz := 20; );
    END_CALL;

    CALL basic_product_structure ;   -- параметры для второго изделия
        IMPORT (shape_2_def := @prod_def_shape; );
        WITH (prod_name  :=  @prod2_name;
              pdef_desc  :=  ' test product definition 2 ' ,
              propd_desc :=  ' shape of test product 2 ' ;
              prod_name  :=  ' second test product' ;
              prod_id    :=  ' P02 ' ;
              pdf_id     :=  ' PDF02 ' ; );
    END_CALL;

    void1 := oriented_closed_shell ( ' void1', hollow1, FALSE);
    void2 := oriented_closed_shell ( ' void2', hollow2, FALSE);

    tetra_with_void  :=  faceted_brep ( ' tetra_with_void', shell_object)  | |
                         brep_with_voids ( [void1] );

    tetra_with_voids := faceted_brep ( ' tetra_with_voids', shell_object) | |
                         brep_with_voids ( [void1, void2] );

    its_units  :=  length_unit ( ) | |  si_unit ( 'milli', 'metre' );

    its_context := geometric_representation_context
                       ( 'context_1', 'context_for_tetrahedron',  3)  | |
                       global_unit_assigned_context ( [its_units] );

    fbsr1  :=  faceted_brep_shape_representation
               ( ' fbsr1',  [tetra_with_void], its_context);

    fbsr2  :=  faceted_brep_shape_representation
               ( 'fbsr2 ',  [tetra_with_voids], its_context);

  END_REALIZATION;
END_TEST_CASE;
( *
```

Примечания

1 Применяются все критерии для fb1, кроме тех, которые касаются цели FB5.

2 В контрольном примере для определения пустот используется объект **oriented_closed_shell**, значением атрибута **orientation** которого должно быть FALSE.

3 Контекст повторно используются с различными параметрами для определения оболочек пустот.

E.3.2.3 Критерии решения постпроцессора

FB6: Комплексные подтипы **faceted_brep** и **brep_with_voids** должны быть правильно интерпретированы.

FB7: в объекте **faceted_brep** с более чем одной правильно интерпретированной пустотой оболочки пустот не должны пересекаться друг с другом или с внешней оболочкой.

FB8: Перпендикуляр к оболочке каждой пустоты должен быть направлен внутрь пустоты.

E.3.3 Контрольный пример fb3

Контрольный пример fb3 разработан для проверки возможности задания граней с внутренними контурами, а также для проверки использования значений по умолчанию и изменения направлений на обратные. Тестируемый объект представлен в форме прямоугольного блока со сквозным треугольным отверстием и треугольным углублением на верхней грани. В контрольном примере определены геометрия и топология объекта.

E.3.3.1 Реализуемые цели тестирования ПИК

FB12: Значением объекта **face_surface** с **same_sense** является FALSE.

FB14: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds,** представленным множеством, состоящим из одного объекта **face_bound**, представленного как **face_outer_bound** с атрибутом **bound**, представленным как **poly_loop** и **orientation**, является FALSE.

FB15: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds**, представленным множеством, состоящим по крайней мере из двух объектов **face_bound**, один из которых является объектом **face_outer_bound**, с атрибутом **bound**, представленным как **poly_loop** и **orientation**, является TRUE.

FB16: Значением объекта **face**, представленного как **face_surface** с атрибутом **bounds**, представленным множеством, состоящим по крайней мере из двух объектов **face_bound**, один из которых является объектом **face_outer_bound**, с атрибутом **bound**, представленным как **poly_loop** и **orientation**, является FALSE.

FB18: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** без атрибута **axis**.

FB20: Объект **surface** (как объект **elementary_surface**) представлен как объект **plane** с атрибутом **position**, представленным как **axis2_placement_3d** без атрибута **ref_direction**.

E.3.3.2 Спецификация ввода в постпроцессор

Тестируемой моделью является объект **faceted_brep_shape_representation**, состоящий из единственного объекта **faceted_brep**. Объект **faceted_brep** представлен в форме прямоугольного блока с ребрами, параллельными осям координат. Блок имеет треугольное сквозное отверстие и треугольное углубление на одной грани. Для некоторых контуров значением атрибута **orientation** является FALSE. Для некоторых граней значением атрибута **face_geometry** является FALSE. При создании модели обмена для некоторых атрибутов объекта **axis2_placement_3d** используются значения по умолчанию. Полная спецификация приведена на языке EXPRESS-I.

E.3.3.2.1 EXPRESS-I-спецификация fbrep_3

```
*)
TEST_CASE example_fbrep_3; WITH faceted_brep_aic;

  REALIZATION

  LOCAL
    origin, px, py, pxy, pz, pxz, pyz, pzyz        : cartesian_point;
    q, qx, qy, qz, qxz, qyz                        : cartesian_point;
    r, rx, ry, rz, rxz, ryz                        : cartesian_point;
    neg_x, neg_y, neg_z, slope, pos_x, pos_y : direction;
    loopb, loopt, loopf, loopbk, loopl, loopr      : poly_loop;
    loopbi, loopmid, loopt1, loopt2, loopqf        : poly_loop;
    loopql, loopqs, looprb, looprr, looprs         : poly_loop;
    a1, a2, a3, ap3, ar3, ap1, ap2, ar1, ar2       : axis2_placement_3d;
    ar4, aq1, aq2, aq4                             : axis2_placement_3d;
    pa1, pa2, pa3, pap1, pap2, pap3, paq1          : plane;
    paq2, paq4, par1, par2, par3, par4             : plane;
    bottom, top, front, back, left, right          : face_outer_bound;
    qfbd, qlbd, qsbd, rbbd, rrbd, rsbd, midbd      : face_outer_bound;
    bibd, t1bd, t2bd                               : face_bound;
    fs1, fs2, fs3, fsp1, fsp2, fsp3, fsq1          : face_surface;
    fsq2, fsq4, fsr1, fsr2, fsr3, fsr4             : face_surface;
    blockshell                                     : closed_shell;
    block                                          : faceted_brep;
    fbsr                    : faceted_brep_shape_representation;
    its_context : representation_context;
    its_units : named_unit;
  END_LOCAL;

  ( * Декартовы точки на границе: * )
  origin := cartesian_point ( ' origin ', [0, 0, 0]);
  px := cartesian_point ( ' px ', [50, 0, 0]);
  py := cartesian_point ( ' py ', [0, 50, 0]);
  pxy := cartesian_point ( ' pxy ', [50, 50, 0]);
  pz := cartesian_point ( ' pz ', [0, 0, 100]);
  pxz := cartesian_point ( ' pxz ', [50, 0, 100]);
  pyz := cartesian_point ( ' pyz ', [0, 50, 100]);
  pxyz := cartesian_point ( ' pxyz ', [0, 0, 100]);

  q   := cartesian_point ( ' q ', [10,10, 0]);
  qx := cartesian_point ( 'qx ', [25,10, 0]);
  qy := cartesian_point ( ' qy ', [10, 25, 0]);
  qz := cartesian_point ( ' qz ', [10, 10, 100]);
  qxz := cartesian_point ( ' qxz ', [25, 10, 100]);
  qyz := cartesian_point ( ' qyz ', [10, 25, 100]);

  r   := cartesian_point ( ' r ', [45, 45, 50]);
  rx := cartesian_point ( ' rx ', [35, 45, 50]);
  ry := cartesian_point ( ' ry ', [45, 35, 50]);
```

```
    rz := cartesian_point ( ' rz ', [45, 45, 100]);
    rxz := cartesian_point ( ' rxz ', [35, 45, 100]);
    ryz := cartesian_point ( ' ryz ', [45, 35, 100]);

    neg_x := direction ( ' neg_x ', [–1, 0, 0]);
    neg_y := direction ( ' neg_y ', [0, –1, 0]);
    neg_z := direction ( ' neg_z ', [0, 0, –1]);
    slope  := direction ( ' slope ', [1, 1, 0 ]);
    pos_x := direction ( ' pos_x ', [1, 0, 0]);
    pos_y := direction ( ' pos_y ', [0, 1, 0 ]);

(* внешние контуры: *)
    loopb := poly_loop ( ' loopb ', [origin, px, pxy, py]);
    loopt := poly_loop ( ' loopt ', [origin, pxz, pxyz, pyz]);
    loopf := poly_loop ( ' loopf ', [origin, px, pxz, pz]);
    loopbk := poly_loop ( ' loopbk ', [py, pxy, pxyz, pyz]);
    loopl := poly_loop ( ' loopl ', [origin, pz, pyz, py]);
    loopr := poly_loop ( ' loopr ', [px, pxz, pxyz, pxy]);

(* внутренние контуры (треугольные): *)
    loopbi := poly_loop ( ' loopbi ', [q, qx, qy]);
    loopmid := poly_loop ( ' loopmid ', [r, rx, ry]);
    loopt1 := poly_loop ( ' loopt1', [qz, qxz, qyz]);
    loopt2 := poly_loop ( ' loopt2 ', [rz, rxz, ryz]),

(*внутренние контуры (прямоугольные): *)
    loopqf := poly_loop ( ' loopqf ', [q, qx, qxz, qz]);
    loopql := poly_loop ( ' loopql ', [q, qz, qyz, qy]);
    loopqs := poly_loop ( ' loopqs ' , [qx, qxz, qyz, qy]),
    looprb := poly_loop ( ' looprb ', [r, rx, rxz, rz]);
    looprr  := poly_loop ( ' looprr ', [r, rz, ryz, ry]);
    looprs  := poly_loop ( ' looprs ', [rx, rxz, ryz, ry]);

(* axis_placements (примечание — число определяет параллельна ли ось осям x, y или z) *)
    a1 := axis2_placement_3d ( ' a1', origin, neg_x, neg_y);
    a2 := axis2_placement_3d ( ' a2 ', origin, neg_y, neg_x);
    a3 := axis2_placement_3d ( ' a3 ', origin, neg_z, ?);
    ap3 := axis2_placement_3d ( ' ap3 ', pz, ?, ?);
    ar3 := axis2_placement_3d ( ' ar3 ', r, ?, ?);
    ap1 := axis2_placement_3d ( ' ap1 ', px, pos_x, ?);
    ap2 := axis2_placement_3d ( ' ap2 ', py, pos_y, ?);
    ar1 := axis2_placement_3d ( ' ar1 ', r, pos_x, ?);
    ar2 := axis2_placement_3d ( ' ar2 ', r, pos_y, ?);
    ar4 := axis2_placement_3d ( ' ar4 ', r, slope, ?);
    aq1 := axis2_placement_3d ( ' aq1 ', q, pos_x, ?);
    aq2 := axis2_placement_3d ( ' aq2 ', q, pos_y, ?);
    aq4 := axis2_placement_3d ( ' aq4 ', r, slope, ?);

(* плоскость, определенная для расположения каждой оси: *)
    pa1 := plane ( ' pa1 ', a1);
    pa2 := plane ( ' pa2 ', a2);
    pa3 := plane ( ' pa3 ', a3);
    pap1 := plane ( ' pap1 ', ap1);
    pap2 := plane ( ' pap2 ', ap2);
    pap3 := plane ( ' pap3 ', ap3);
    paq1 := plane ( ' paq1 ', aq1);
    paq2 := plane ( ' paq2 ', aq2);
    paq4 := plane ( ' paq4 ', aq4);
    par1 := plane ( ' par1 ', ar1);
    par2 := plane ( ' par2 ', ar2);
    par3 := plane ( ' par3 ', ar3);
    par4 := plane ( ' par4 ', ar4);
```

```
(* внешние границы блока *)
   bottom := face_outer_bound ( ' bottom ', loopb, FALSE);
   top    := face_outer_bound ( ' top ', loopt, TRUE);
   front  := face_outer_bound ( ' front ', loopf, TRUE);
   back   := face_outer_bound ( ' back ', loopbk, FALSE);
   left   := face_outer_bound ( ' left ', loopl, FALSE),
   right  := face_outer_bound ( ' right ', loopr, TRUE);

(* внешние границы внутренней грани *)
   qfbd := face_outer_bound ( ' qfbd ', loopqf, FALSE);
   qlbd  := face_outer_bound ( ' qlbd ', loopql, TRUE);
   qsbd := face_outer_bound ( ' qsbd ', loopqs, FALSE);
   rbbd  := face_outer_bound ( ' rbbd ', looprb, FALSE),
   rrbd := face_outer_bound ( ' rrbd ', looprr, TRUE);
   rsbd  := face_outer_bound ( ' rsbd ', looprs, TRUE),
   midbd := face_outer_bound ( ' midbd ', loopmid, TRUE);

(* внутренние границы *)
   bibd  := face_bound ( ' bibd ', loopbd, TRUE);
   t1bd  := face_bound ( ' t1bd ', loopt1, FALSE);
   t2bd  := face_bound ( ' t2bd ', loopt2, FALSE);

(* внешние грани блока *)
   fs1 := face_surface ( ' fs1 ', [left], pa1, TRUE);
   fs2 := face_surface ( ' fs2 ', [front], pa2, TRUE);
   fs3 := face_surface ( ' fs3 ', [bottom, bibd], pa3, TRUE);
   fsp1 := face_surface ( ' fsp1 ', [right], pap1, TRUE);
   fsp2 := face_surface ( ' fsp2 ', [back], pap2, TRUE);
   fsp3 := face_surface ( ' fsp3 ', [top, t1bd, t2bd], pap3, TRUE),

(* внутренние грани
   отверстие: *)
   fsq1 := face_surface ( ' fsq1 ', [qlbd], paq1, TRUE);
   fsq2 := face_surface ( ' fsq2 ', [qfbd], paq2, TRUE);
   fsq4 := face_surface ( ' fsq4 ', [qsbd], paq4, FALSE);
(* углубление: *)
   fsr1 := face_surface ( ' fsr1 ', [rrbd], par1, FALSE);
   fsr2 := face_surface ( ' fsr2 ', [rbbd], par2, FALSE);
   fsr3 := face_surface ( ' fsr3 ', [midbd], par3, TRUE);
   fsr4 := face_surface ( ' fsr4 ', [rsbd], par4, TRUE);

blockshell := closed_shell ( ' blockshell ', [fs1, fs2, fs3, fsp1, fsp2, fsp3, fsq1, fsq2, fsq4, fsr1, fsr2, fsr3, fsr4]),

block := faceted_brep ( ' block ', blockshell);

    its_units := length_unit() || si_unit ( ' milli ', ' metre ');
    its_context := geometric_representation_context
                    ( ' its_context ', ' context_for_block ', 3)    | |
                    global_unit_assigned_context ( [its_units] ),

      fbsr := faceted_brep_shape_representation
       ( ' fbsr ', [block], its_context );

  END_REALIZATION;
END_TEST_CASE;
( *
```

Примечания

1 Внутренние контуры используются для определения отверстий и углублений.

2 Тестируются изменение направления контуров на обратное и геометрия поверхности.

3 При определении **axis2_lacement_3d** используются значения по умолчанию.

E.3.3.3 Критерии решения постпроцессора

FB12: объект **face_surface** со значением атрибута **orientation** FALSE должен быть правильно обработан, т.е. у определяемого объекта **closed_shell** перпендикуляры ко всем граням должны быть направлены от тела.

FB14: объект **face_surface** со значением атрибута **orientation** FALSE должен быть правильно обработан, т.е. конечная ориентация должна всегда быть правильной с гранью слева.

FB15: объекты **face** с внутренней границей должны быть правильно обработаны, т.е. границы должны быть проверены на пересечения.

FB16: объект **loop** со значением атрибута **orientation** FALSE должен быть правильно интерпретирован.

FB18: при обработке **axis2_placement_3d** должны быть правильно установлены значения по умолчанию для объекта **axis**.

FB19: при обработке **axis2_placement_3d** должны быть правильно установлены значения по умолчанию для объекта **ref_direction**.

E.3.4 Контрольный пример fb4

Контрольный пример fb4 разработан для того, чтобы протестировать использование объектов **mapped_item** при создании простой сборки многогранных B-rep моделей. Он также обеспечивает проверку непротиворечивого поведения объектов **geometric_representation_context** при различении координатных плоскостей. В этом тесте для определения геометрии и топологии используется контекст **tetrashell_instance**.

E.3.4.1 Реализуемые цели тестирования ПИК

FB3: Объект **faceted_brep_shape_representation** с контекстом представлен как **geometric_context** с элементами **item**, представленными как **mapped_item**.

FB4: Объект **faceted_brep_shape_representation** с контекстом представлен как **geometric_context** с двумя или более элементами **item**, представленными как **faceted_brep**, **mapped_item** или **axis2_placement_3d**. При этом по крайней мере одним из них должен быть объект **axis2_placement_3d**.

E.3.4.2 Спецификация ввода в постпроцессор

Тестируемой моделью является объект **faceted_brep_shape_representation**, состоящий из единственного объекта **faceted_brep**. Объект **faceted_brep** представляет собой сплошной тетраэдр, одна из вершин которого расположена в начале системы координат, а смежные ребра направлены вдоль координатных осей. Это представление затем используется в связке с объектом **mapped_item** для создания в том же самом объекте **representation_context** представления, являющегося повернутой копией исходного представления. В отдельном объекте **representation_context** создается представление, состоящее из исходного объекта **faceted_brep** и отображенной копии исходного представления. Полная спецификация приведена на языке EXPRESS-I.

E.3.4.2.1 EXPRESS-I-спецификация fbrep_4

```
* )
TEST_CASE example_fbrep_4; WITH faceted_brep_aic;

  REALIZATION

  LOCAL
    origin : cartesian_point;
    pos_z, neg_y : direction;
    refaxes, oldaxes, newaxes : axis_placement_3d;
    shell_object : closed_shell;
    tetrahedron : faceted_brep;
    fbsr, fbsr1, fbsrass : faceted_brep_shape_representation;
    grc1, grc2 . representation_context;
    its_units : named_unit;
    tetrarot1, tetrarot2 : mapped_item;
    mapping1, mapping2 : representation_map;
  END_LOCAL;

    CALL tetrashell_instance ; -- используются значения по умолчанию,
                               -- поэтому нет WITH
      IMPORT (shell_object := @tetrashell;
              origin := @origin;
              neg_y .= @ neg_y; refaxes := @a1;);
    END_CALL;

tetrahedron  .=  faceted_brep ( ' tetrahedron ', shell_object);
its_units  :=  length_unit ( )  || si_unit  ( ' milli ',  ' metre ');

grc1  :=  geometric_representation_context ( ' grc1 ',
      ' context for tetrahedron ', 3)  ||
  global_unit_assigned_context ( [its_units] );

grc2  :=  geometric_representation_context ( ' grc2 ',
      ' context for rotated tetrahedron ',  3)  ||
  global_unit_assigned_context ( [its_units] );

fbsr := faceted_brep_shape_representation ( ' fbsr ', [tetrahedron], grc1 );
```

```
(* Определяются смещения осей для использования при отображении *)
oldaxes := axis2_placement_3d ( ' oldaxes ', origin, ?, ?);

pos_z := direction ( ' pos_z ', [0, 0, 1]);

newaxes := axis2_placement_3d ( ' newaxes ', origin, pos_z, neg_y),

mapping1 := representation_map (refaxes, fbsr );
tetrarot1 := mapped_item ( ' tetrarot1 ', mapping1, newaxes );

(* Определяется представление с использованием только tetrarot1 *)
fbsr1 := faceted_brep_shape_representation ( ' fbsr1 ', [tetrarot1], grc1 );

(* Определяется представление, являющееся сборкой тетраэдра и
 отображенной копии *)

  mapping2 := representation_map (oldaxes, fbsr);
  tetrarot2 := mapped_item ( ' tetrarot2 ', mapping2, newaxes);

  fbsrass := faceted_brep_shape_representation
          ( ' fbsrass ', [tetrahedron, tetrarot2, oldaxes], grc2);

  END_REALIZATION;
END_TEST_CASE;
( *
```

Примечания

1 Контрольный пример для **mapped_item** и «сборки» с использованием простого сплошного тетраэдра.

2 Оболочка тетраэдра создана с использованием контекста с параметрами по умолчанию.

3 Внешняя оболочка объекта **faceted_brep** является объектом **closed_shell**, но не является объектом **oriented_closed_shell**.

4 **fbsr1** должен быть повернутой копией **fbsr**.

5 **fbsrass** должен быть эквивалентен двум копиям **fbsr**, «склеенным» вместе.

E.3.4.3 Критерии решения постпроцессора

FB3: После обработки сплошной тетраэдр, определенный объектом **mapped_item**, должен быть правильно определен и позиционирован.

FB4: Созданы контексты двух отдельных геометрических представлений. Модели **fbsr** и **fbsr1** пространственно связаны и должны соприкасаться только в одной точке. Модели **fbsr1** и **fbsrass** пространственно не связаны. Оболочки в **fbsrass** не должны пересекаться, но должны совмещаться по общей грани, которая неявно является совместно используемой.

E.3.5 Контрольный пример fb5

Контрольный пример fb5 разработан для тестирования использования объектов **mapped_item** совместно с **cartesian_transformation_operator** при создании простой сборки многогранных B-rep моделей. Тестируется использование коэффициента масштабирования. В этом тесте для определения геометрии и топологии используется контекст **tetrashell_instance**.

E.3.5.1 Реализуемые цели тестирования ПИК

FB21: Объект **mapped_item** с атрибутом **mapping_target** представлен как **cartesian_transformation_operator_3d**.

FB22: Объект **cartesian_transformation_operator** представлен как **cartesian_transformation_operator_3d** с масштабом, представленным значением типа REAL, не равным 1.0.

E.3.5.2 Спецификация ввода в постпроцессор

Тестируемой моделью является объект **faceted_brep_shape_representation**, состоящий из единственного объекта **faceted_brep**. Объект **faceted_brep** представляет собой сплошной тетраэдр, одна из вершин которого расположена в начале системы координат, а смежные ребра направлены вдоль координатных осей. Это представление затем используется в связке с объектами **mapped_item** и **cartesian_transformation_operator** для создания в том же самом объекте **representation_context** представления, состоящего из повернутой и масштабированной (но не с коэффициентом 1.0) копией исходного представления и исходного объекта **faceted_brep**. Полная спецификация приведена на языке EXPRESS-I.

E.3.5.2.1 EXPRESS-I-спецификация fbrep_5

```
*)
TEST_CASE example_fbrep_5; WITH faceted_brep_aic;

  REALIZATION
```

```
    LOCAL
        origin : cartesian_point ;
        pos_z, neg_y, pos_x : direction;
        oldaxes : axis_placement_3d;
        transform : cartesian_transformation_operator_3d;
        shell_object : closed_shell;
        tetrahedron . faceted_brep;
        fbsr, fbsrass : faceted_brep_shape_representation;
        its_units : named_unit;
        grc1, grc2 : representation_context;
        tetratrans : mapped_item;
        mapping1 : representation_map;
    END_LOCAL;

        CALL tetrashell_instance ; -- используются значения по умолчанию,
                                   -- поэтому нет WITH
            IMPORT (shell_object := @tetrashell;
                    origin := @origin;
                    neg_y .= @ neg_y; );
        END_CALL ;

tetrahedron := faceted_brep ( ' tetrahedron ', shell_object);

its_units := length_unit() || si_unit ( ' milli ', ' metre ');

grc1 := geometric_representation_context ( ' grc1 ',
      ' context for tetrahedron ',  3)   | |
   global_unit_assigned_context ( [its_units] );

grc2 := geometric_representation_context ( ' grc2 ',
       ' context for assembly ',   3)   | |
global_unit_assigned_context ( [its_units] );

(* Определяются axis_placement и cartesian_transformation_operator для использования при отображении *)

pos_x := direction ( ' pos_x ', [1, 0, 0]),

pos_z := direction ( ' pos_z ', [0, 0, 1]),

oldaxes := axis2_placement_3d ( ' oldaxes ', origin, pos_z, pos_x);

transform := cartesian_transformation_operator_3d ( ' transform ',
                                    pos_x, neg_y, origin, 0.75, pos_z );

fbsr := faceted_brep_shape_representation ( ' fbsr ',
                                    [tetrahedron, oldaxes], grc1 );

mapping1 := representation_map (oldaxes, fbsr);
(*  tetratrans является 75% масштабированной копией оригинала,
   отраженного на плоскость ZX *)
tetratrans := mapped_item  ( ' tetrarot1 ', mapping1, transform);

(* Определяется представление, являющееся сборкой тетраэдра и преобразованной (масштабированной и отра-
женной) копии *)

fbsrass := faceted_brep_shape_representation
          ( ' fbsrass ', [tetrahedron, tetratrans], grc2);

    END_REALIZATION;
END_TEST_CASE;
( *
```

Примечания

1 Контрольный пример для объекта **mapped_item** и «сборки» с использованием простого сплошного тетраэдра.

2 Оболочка тетраэдра создана с использованием контекста с параметрами по умолчанию.

3 Объект **tetratrans** должен быть масштабированной копией **fbsr** после отражения на плоскость OZX.

E.3.5.3 Критерии решения постпроцессора

FB21: После обработки сплошной тетраэдр, определенный объектом **mapped_item**, должен быть правильно определен и позиционирован.

FB22: Объект **fbsrass** должен состоять из двух сплошных тетраэдров, которые соприкасаются в плоскости ZX. Один из них является копией размером 3/4 другого после отражения на плоскость.

Е.4 Контекст tetrashell_instance, используемый в тестах многогранной B-rep модели

Приведенный ниже контекст на языке EXPRESS-I используется в контрольных примерах, описанных в разделе Е.3. Этот контекст дает возможность определить простой объект **closed_shell** четырехгранной формы с вершинами, расположенными в точках (orc, orc, orc), (lx, orc, orc), (orc, ly, orc) и (orc, orc, lz). Все границы определяются объектами **poly_loop**.

```
*)
CONTEXT tetrashell_instance ;

PARAMETER
   orc        : length_measure := 0;
   lx         : length_measure := 100;
   ly         : length_measure := 100;
   lz         : length_measure := 100;

   origin     : cartesian_point := cartesian_point ( ' origin ', [orc, orc, orc]);
   p_x        : cartesian_point := cartesian_point ( ' p_x ', [lx, orc, orc]);
   p_y        : cartesian_point := cartesian_point ( ' p_y ', [orc, ly, orc]);
   p_z        : cartesian_point := cartesian_point ( ' p_z ', [orc, orc, lz]);

   neg_x      : direction := direction ( ' neg_x ', [–1, 0, 0]),
   neg_y      : direction := direction ( ' neg_y ', [0, –1, 0]),
   neg_z      : direction := direction ( ' neg_z ', [0, 0, –1]);
   dslope     . direction := direction ( ' dslope ', [1, 1, 1]),
   dperp      : direction := direction ( ' dperp ', [1, –1, 0]);

   loop_x     : poly_loop := poly_loop ( ' loop_x ', [origin, p_z, p_y]);
   loop_y     : poly_loop := poly_loop ( ' loop_y ', [origin, p_x, p_z]);
   loop_z     : poly_loop := poly_loop ( ' loop_z ', [origin, p_y, p_x]);
  loop_slope : poly_loop := poly_loop ( ' loop_slope ', [p_z, p_x, p_y]);

   a1         : axis2_placement_3d := axis2_placement_3d ( ' a1 ', origin,
                                                           neg_x, neg_y);
   a2         : axis2_placement_3d := axis2_placement_3d ( ' a2 ', origin,
                                                           neg_y, neg_x);
   a3         : axis2_placement_3d := axis2_placement_3d ( ' a3 ', origin,
                                                           neg_z, neg_y);
   a4         : axis2_placement_3d := axis2_placement_3d ( ' a4 ', p_x, dslope,
                                                                dperp);

   p1         : plane := plane ( ' p1 ', a1);
   p2         : plane := plane ( ' p2 ', a2);
   p3         : plane := plane ( ' p3 ', a3);
   p4         : plane := plane ( ' p4 ', a4);

   b1         : face_outer_bound := face_outer_bound ( ' b1 ', loop_x, TRUE);
   b2         . face_outer_bound := face_outer_bound ( ' b2 ', loop_y, TRUE);
   b3         : face_outer_bound := face_outer_bound ( ' b3 ', loop_z, TRUE);
   b4         . face_outer_bound := face_outer_bound ( ' b4 ', loop_slope, TRUE);

   fs1 : face_surface := face_surface ( ' fs1 ', [b1], p1, TRUE);
   fs2 : face_surface := face_surface ( ' fs2 ', [b2], p2, TRUE);
   fs3 : face_surface := face_surface ( ' fs3 ', [b3], p3, TRUE);
   fs4 : face_surface := face_surface ( ' fs4 ', [b4], p4, TRUE);
END_PARAMETER;

SCHEMA_DATA tetra_ctxt;

cfs = connected_face_set {SUBOF(@tri); cfs_faces ->
                                  ( [@fs1, @fs2, @fs3, @fs4] );
                          SUPOF(@tetrashell);};

tri = topological_representation_item {SUBOF(@ri); SUPOF(@cfs);},

ri = representation_item {name -> ' tetrashell ';
```

```
                                        SUPOF(@tri);};
tetrashell = closed_shell {SUBOF(@cfs);};
END_SCHEMA_DATA;
END_CONTEXT;
(*
```

Приложение F
(справочное)

Сведения о соответствии национальных стандартов Российской Федерации ссылочным международным стандартам

Таблица F.1

Обозначение ссылочного международного стандарта	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 8824-1:1995	ГОСТ Р ИСО/МЭК 8824-1—2001 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации
ИСО 10303-1:1994	ГОСТ Р ИСО 10303-1—99 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 1. Общие представления и основополагающие принципы
ИСО 10303-11:1994	ГОСТ Р ИСО 10303-11—2000 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS
ИСО/ТО 10303-12:1997	ГОСТ Р ИСО/ТО 10303-12—2000 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 12. Методы описания. Справочное руководство по языку EXPRESS-I
ИСО 10303-41:1994	ГОСТ Р ИСО 10303-41—99 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 41. Интегрированные обобщенные ресурсы. Основы описания и поддержки изделий
ИСО 10303-42:1994	*
ИСО 10303-43:1994	ГОСТ Р ИСО 10303-43—2002 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 43. Интегрированные обобщенные ресурсы. Структуры представлений
ИСО 10303-202:1996	*

* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.

УДК 656.072:681.3:006.354 ОКС 25.040.40 ОКСТУ 4002 П87

Ключевые слова: автоматизация производства, средства автоматизации, интеграция систем автоматизации, промышленные изделия, представление данных, обмен данными, прикладные интерпретированные конструкции, описание формы тела, многогранное граничное представление

Редактор *В.Н. Копысов*
Технический редактор *Н.С. Гришанова*
Корректор *М.В. Бучная*
Компьютерная верстка *В.И. Грищенко*

Сдано в набор 11.08.2008. Подписано в печать 18.09.2008. Формат 60x84 1/8. Бумага офсетная. Гарнитура Ариал.
Печать офсетная. Усл. печ. л. 3,72. Уч.-изд. л. 2,70. Тираж 145 экз. Зак. 1141.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru gost.e-ecolog.ru
Набрано во ФГУП «СТАНДАРТИНФОРМ» на ПЭВМ
Отпечатано в филиале ФГУП «СТАНДАРТИНФОРМ» — тип. «Московский печатник», 105062 Москва, Лялин пер., 6

gost.e-ecolog.ru