

1.

1.1

(distributed processing environment)

가

가

가

, 가 가 .  
(reliability)

가 (availability)

. 가 가

가 [1, 2, 31].

. , 가

, ISO ODP (Open Distributed Processing)

[4, 5, 6]. ODP

,  
(distribution

transparency)

(broker)

ODP

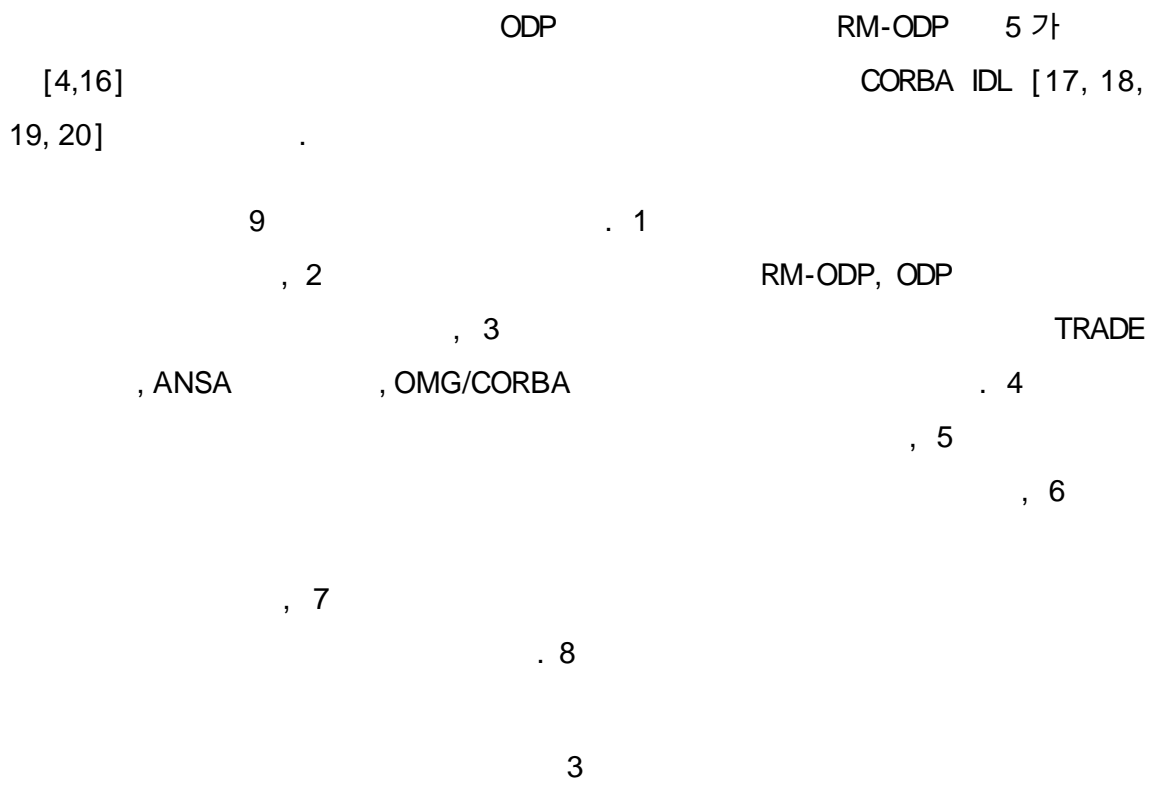
(trading function)

[3].

(trader) . ODP  
. ,  
.  
가 [8]. 가 ODP  
가 .  
(type management) [9]  
(type safety) 가 . (data  
typing) . , ,  
(behavior) .  
(description)  
(relationship) (management)  
(type manager)  
[10, 11].  
가 ODP  
. [12, 13, 14] 가  
ODP  
[15]. (Quality)

(compatibility)

## 1.2





RM-ODP

ODP

. ISO ITU-T  
[4, 5, 6].

### 2.1.1 RM-ODP

(portability) 가

(transparency)

### 2.1.2 RM-ODP

(specification)

. RM-ODP 5 가 (viewpoint)  
[16].

□ (Enterprise viewpoint)

(policy)

□ (Information viewpoint)

가 , 가  
ERD (Entity -Relationship Diagram), (conceptual schema)  
가

□ (Computational viewpoint)

. RM-ODP

(object based)

□ (Engineering viewpoint)

(semantics) 2

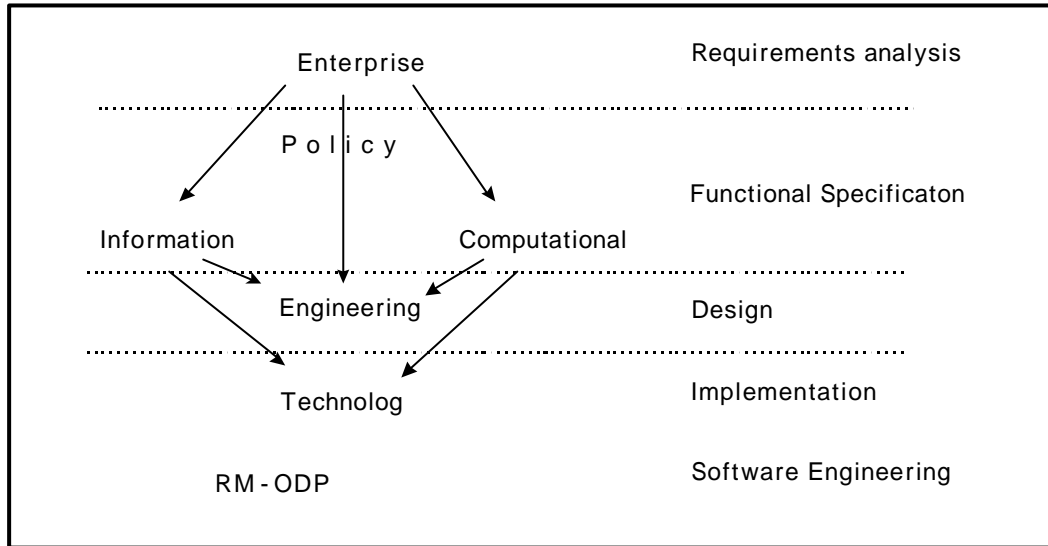
가

□ (Technology viewpoint)

가

1 RM-ODP

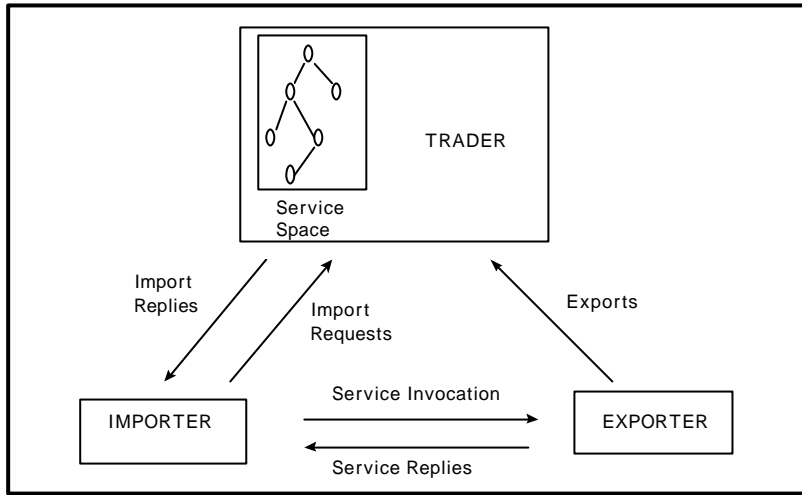
(specification)



### 1. RM-ODP

## 2.2 ODP

(importer), (exporter),  
 (service offer)  
 (service type)  
 (constraints) ,  
 가  
 [7, 21].



2.

2  
 ( )가 ( ) 3  
 (third party object) . 가  
 (advertise) .  
 (provider) 가 (property)  
 .  
 가  
 . (Request) 가  
 (match) .  
 가



(binding)

가

ODP

(autonomy)

가

(directed trading graph)

3 5

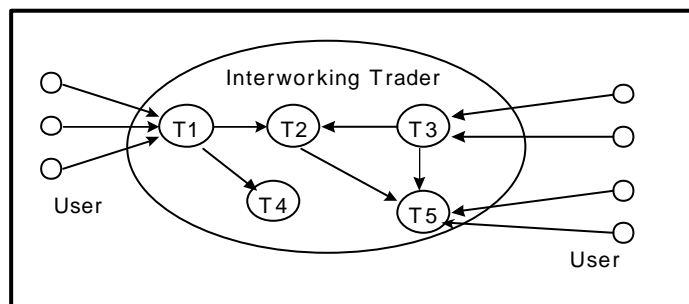
가

ISO ITU

Final Draft

ODP Trading Function Standard [7]

DCE [12], CORBA [13,22]



3



### 3.

TRADE

ANSA

CORBA

RFP (Request For Proposal) [29]

#### 3.1 TRADE

TRADE (Trading and Coordination

Environment)

DCE

[23].

TRADEr (TRADE trader)

ODP

OSF DCE [24]

DCE

가 DCE

TRADER

(service offer manager), (service selection manager), (trader interworking manager), (type manager), (access control manager)가

(comparison) (common understanding) (basis)

(explicit) (subtype polymorphism) 가

가

A가 B

, B가 A

DCE RPC

DCE Cell Directory Service Global Directory

Service

### 3.2 ANSA

ANSA (Advanced Network System Architecture) ODP

ODP

가 ANSAware [25].

ANSAware  
 [26]. RM-ODP  
 (description)  
 (flat name space) . 가 . ANSAware  
 (inclusion polymorphism)  
 “is compatible with” , ,  
 A is\_compatible\_with B , B 가  
 A .  
 (compatibility graph) .

가

ANSAware

ODP

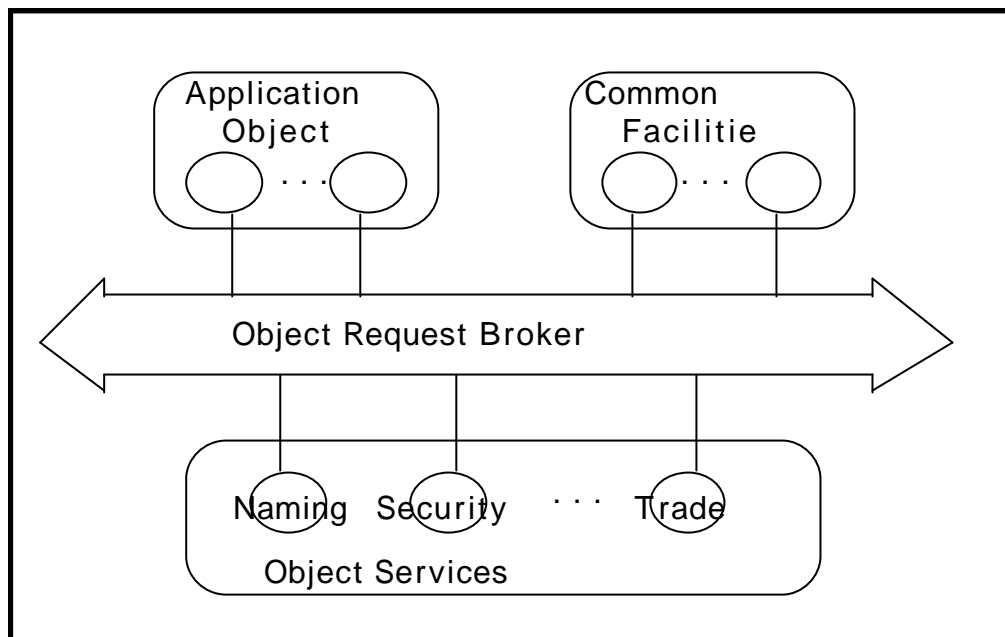
### 3.3 OMA CORBA

OMG (Object Management Group) OMA (Object Management Architecture)  
 (interoperable) 가

CORBA [17, 18]. OMA (Distributed  
 Object-Oriented Computing) (facility)

4

4



#### 4. CORBA

- ORB (Object Request Broker)

(object bus) . ORB (static method invocation) (dynamic method invocation) .

- OS (Object Services)

. OMG COSS  
 (Common Object Service Specification) . RFP1 (Request for Proposal)  
 RFP4  
 RFP5 .

- CF (Common Facilities)

Service) 가 (vendor) CF . ORB OS (Object

□ AO (Application Object)

OMG

OMA RM-ODP 가 (predicate) (module) , . CORBA

ORB 가 가

가 DII (Dynamic Interface Invocation)

가

OMA CORBA OMG ODP (infrastructure) (behavior description)

4.

(policy)

가 가

#### 4.1

가

(property)

(classification)

가

(search)

(select)

□

(type safety)

□

(interworking)

가

□

(resource)

“ 가 가” , “ 가 가” , 가 가



□ (evolution)  
 가 가 가 , 가  
 .  
 가 가 .

## 4.2

, , [28]  
 .

□ (run time type checking)  
 . 가 가  
 가 .  
 (interaction) .  
 (compatibility) 가

□ (matching)  
 (relationship) 가

□ (dynamic selection)  
가 가  
가 가

### 4.3

(policy) (behavior) 가

□ , , 가

□ 가

□ (SDL, Service type Description Language) 가

□ ,

□ 가

가 .

□

가

#### 4.4

가 .

(cooperation)

가

가

( )

가

□

□

□ 가

## 5. (Information Viewpoint)

가

가

### 5.1

(boolean) (predicate) [6].

□ (inheritance) 가 .

(subtype) .

가

(supertype)

□

(polymorphism)

가

가

□

(encapsulation)

### 5.1.1

(service type)

가

가

(service property type)

(Quality of Service, e.g.,

response time, throughput)

가

(mode) . , (value),  
가 가  
(compound) 가 [27].

### 5.1.2

(interface type) (abstraction)  
가 가  
( ) . 2 가  
가 가  
, 가  
(stream interface) (information flow)  
[6].

open() “ read() write() ” .

### 5.1.3

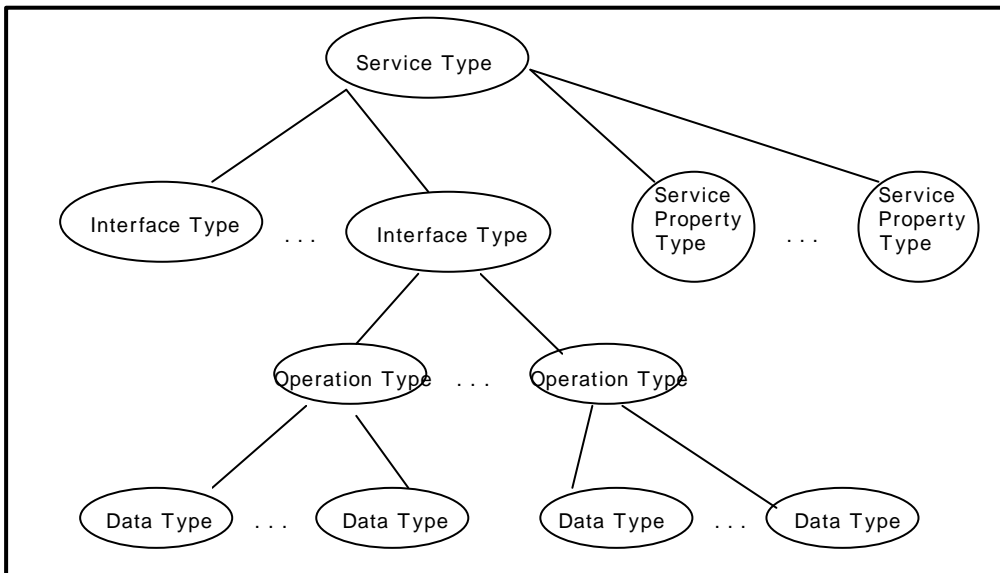
(operation type) ,

(argument) , (termination)  
 2 가 transactional non-transactional  
 RM-ODP transactional properties ACID (Atomicity, Consistency, Isolation, Durability) [6]

### 5.1.4

(data type) ( , Boolean type, integer type ) ( , enumerated type, structure type )

5



5.

## 5.2

(relationship)

가 가 가

가 가

(meta level)

가 (predicate)

RM-

ODP

[6].

### 5.2.1

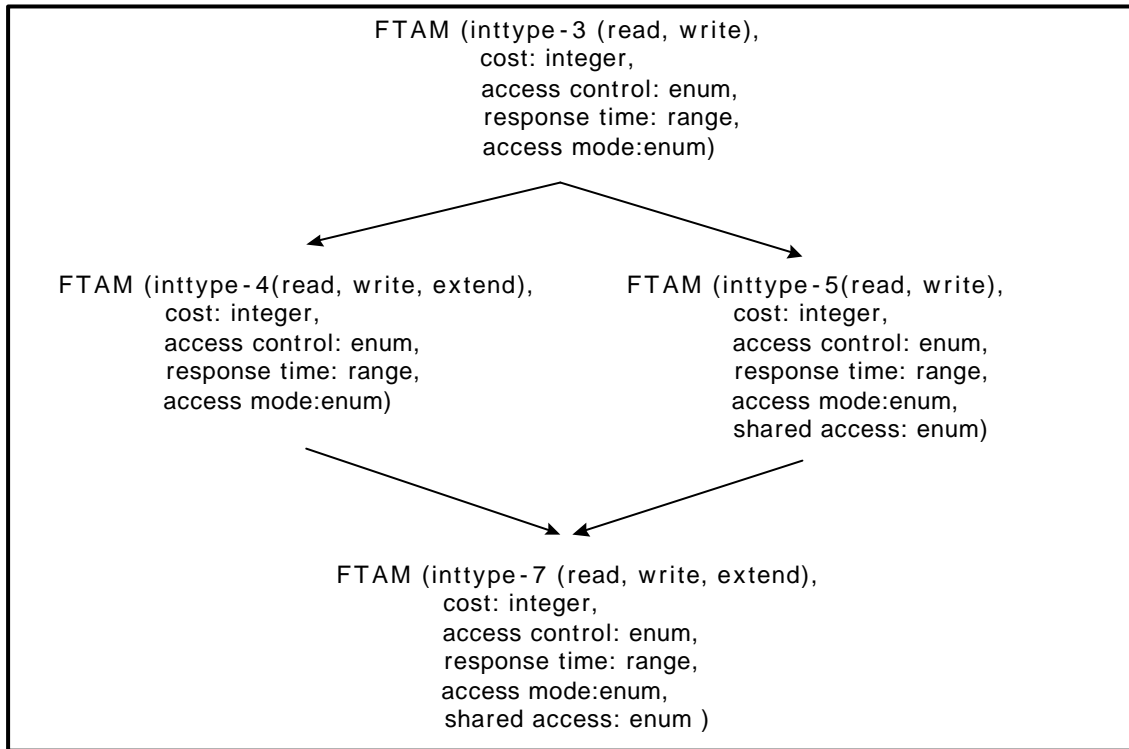
B B가 A 가 A 가

B B가 A

6 FTAM (File Transfer, Access and Management)



[10].



6.

### 5.2.2

(conversion)

가 .  
 .  
 (mapping) , . 가 가 ‘  
 ‘ (mapping) ,  
 (symmetric) .  
 가 (asymmetric) .  
 (stable) (unstable)

### 5.3

(repository)  
 . SDL (Service type  
 Definition Language) .  
 .  
 가  
 (reusability) .

#### 5.3.1

□

(header) (body) .  
 (keyword) "servicetype"  
 ( )가 . 7  
 PrintService .  
 □ (property) , , 가  
 . 1  
 (static) .  
 (load)

Property Mode	Meaning.
Normal	Property can be updated or deleted.
Read_only	Property cannot be updated, but can be deleted.
Fixed_normal	Property can be updated, but not deleted.
Fixed_readonly	Property cannot be updated, but can not be deleted.
Undefined	Not relevant; only used in exceptions.

1. (Property mode)

□  
 "interfacetype"  
 (signature) 가 . 8 PrintService  
 Printing .

□

8 Printing  
open(), write(), close()

```
servicetype PrintService {  
    normal Cost integer;  
    fixed_normal Response integer;  
    read_only Kind string;  
    interfacetype Printing{  
        open();  
        write();  
        close();  
    }  
}
```

7.

### 5.3.2

“relationship”  
가 가 가  
8  
DotPrint 가 PrintService

```

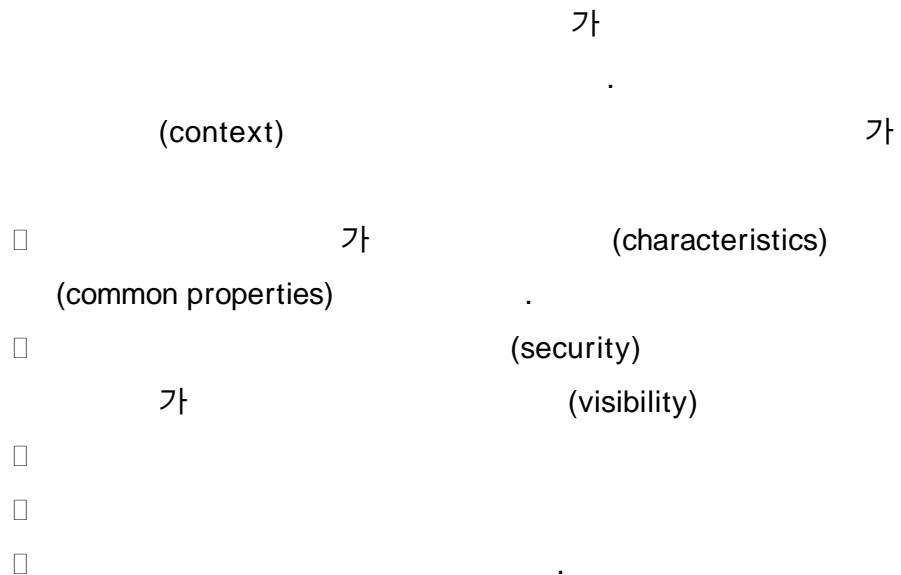
relationship RelationshipName{
    relationkind subtype;
    sourcetype PrintService;
    targettype DotPrint;
}

```

8.

## 5.4

### 5.4.1



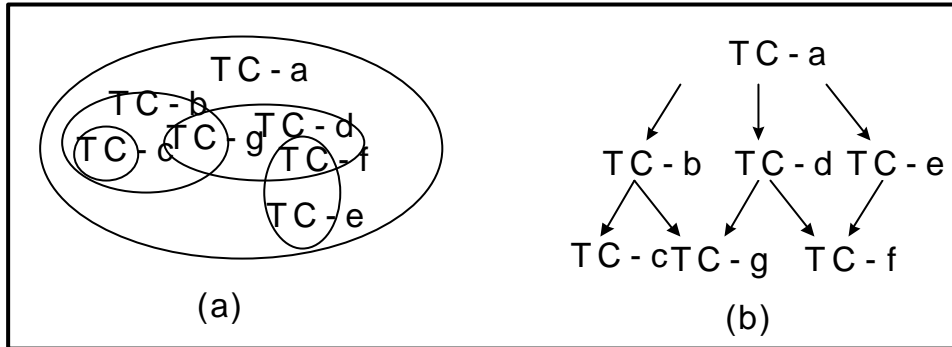
(subset)

9 (a)

TC-a

가

9 (b)



9.

(Representation)

## 6. (Computational Viewpoint)

가 가

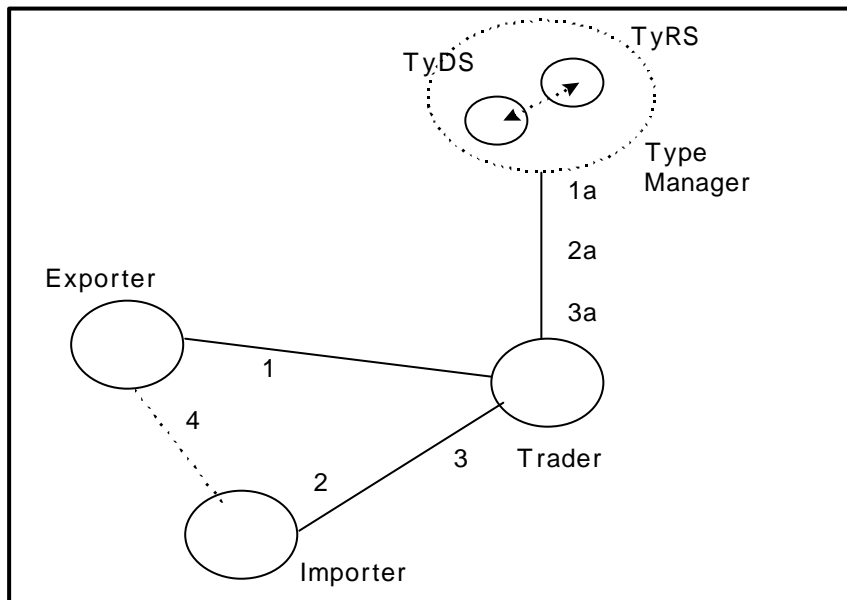
### 6.1

(interaction)

□ 1. (export) - 가  
( 10 1a).

□ 2. (import) - 가,  
가 ( 10 2a)

□ 3. (offer match) -  
 (identifier, ) ( 10  
 3a). 가  
 2  
 TyDS (Type Description Service) , TyRS (Type  
 Relationship Service)



10.

## 6.2

가



가 (query) 가 ,

### 6.2.1 (Type Description Operation)

□ 가 (add type)  
가

(type identifier)가

□ (delete type)

□ (modify type)

### 6.2.2

( ) ( , symmetry)

(integrity)



[30].

가

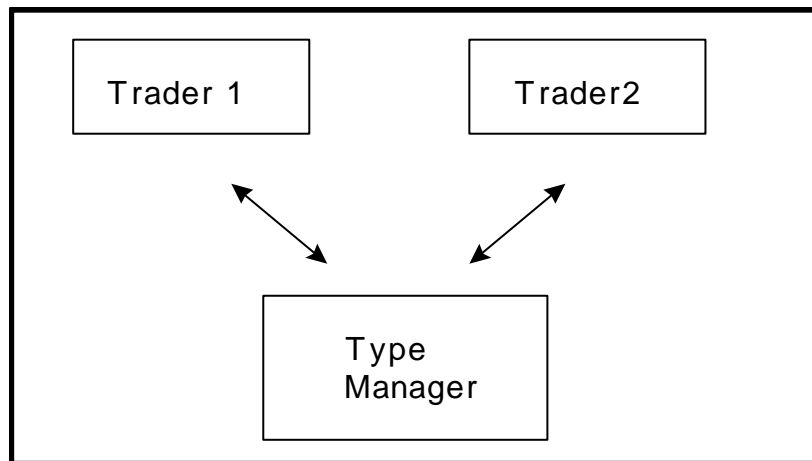
6.3.1

가

11

(boundary)가

(homogeneous)



11.

6.3.2

가

가

□

가

□

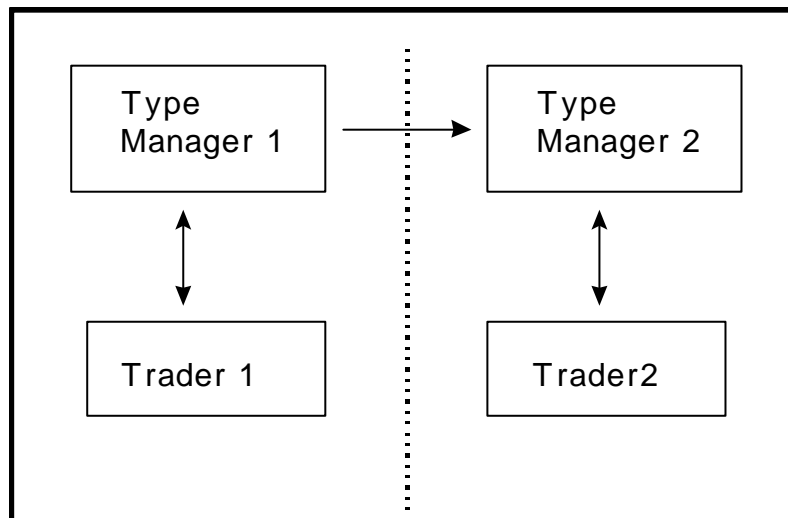
가

□

□

(security)

가



12.

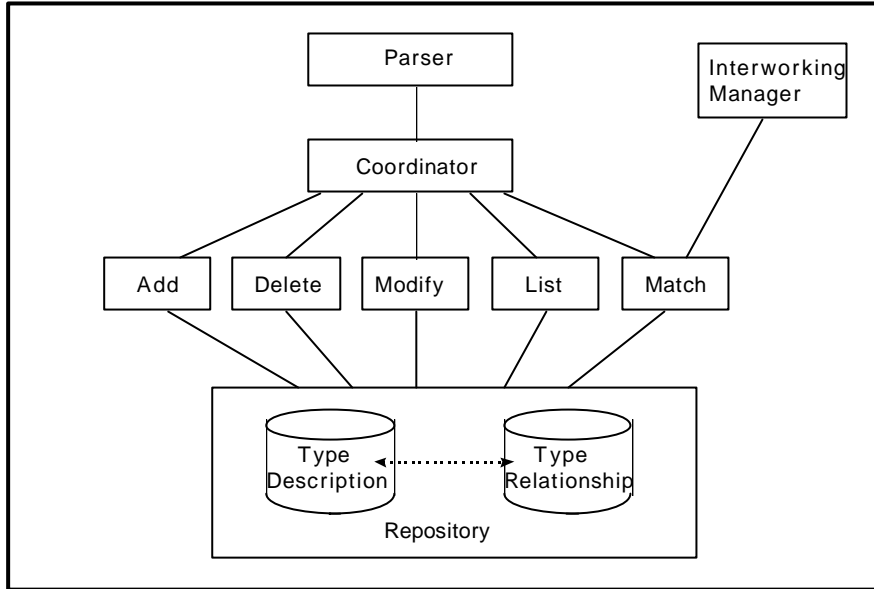
가

graph) . 가 12 (link) (directed acyclic  
 . 1 2  
 1 2  
 .

## 7. (Engineering)

### 7.1

가, , 13 (parser), (coordinator), , ,



13.

□ (parser)

(invoke)

□ (coordinator)

가

□ 가 (add)

가 .

가 .

□ (delete)

(integrity)

□ (modify)

□ (list)

□ (match)

□

□

(integrity)

□





14

( )가

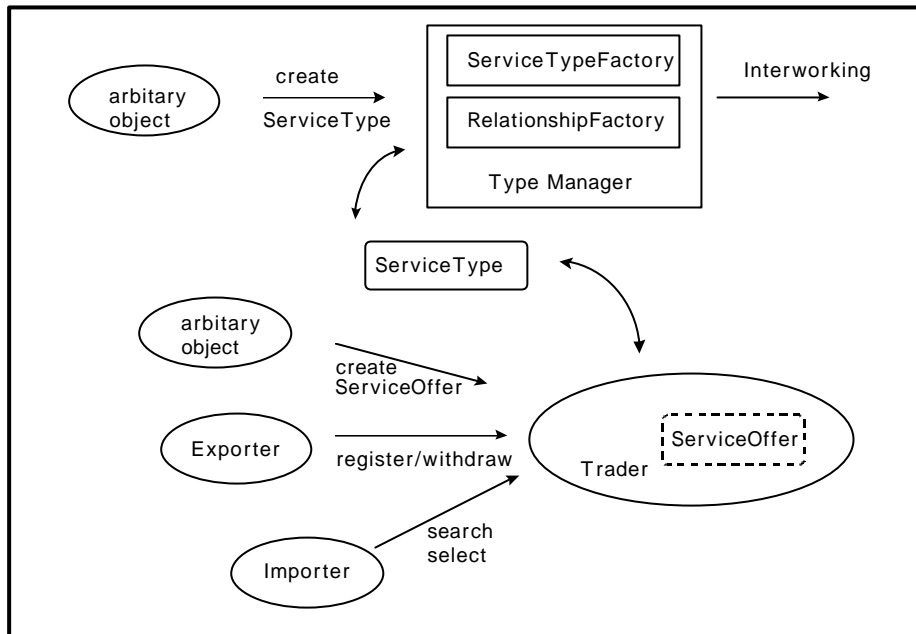
가

(community)

가

가

(constraint)



14.

## 8.2

IDL

IDL

### 8.2.1 Service Type Factory

(modify), (list, find, findall) 가 (add), (delete), .

## □ Add Service Type

```
IncarnationNumber add_servicetype (  
    in ServiceTypeName name,  
    in ContextName context_name,  
    in InterfaceStructSeq inters,  
    in PropStructSeq props  
) raises (  
    InvalidServiceType,  
    InvalidContextName,  
    InvalidInterfaceName,  
    InvalidProperty  
);
```

```
가 . , ,  
 , 가 .  
 가 .  
 .  
 "InvalidServiceType"  
 , "InvalidContextName",  
 "InvalidInterfaceName",  
 "InvalidProperty" .
```

## □ Delete Service Type

“UnKnownServiceType”

가 “HasRelationship”

```
void delete_servicetype (  
    in ServiceTypeName name,  
    in ContextName context_name  
  
) raises (  
    InvalidServiceType,  
    UnKnownServiceType,  
    InvalidContextName,  
    HasRelationship  
);
```

#### □ Modify Service Type

```
void modify_servicetype (  
    in ServiceTypeName name,  
    in ContextName context_name,  
    in InterfaceStructSeq inters,  
    in PropStructSeq props  
  
) raises (  
    InvalidServiceType,  
    UnKnownServiceType,  
    InvalidContextName,  
    InvalidProperty  
);
```

#### □ List Service Type

```
ServiceTypeNameSeq list_servicetype (  
    in SpeciedServiceTypes which_types  
);
```

. which\_types

#### □ Find Service Type

```
TypeStruct find_servicetype (  
    in ServiceTypeName name,  
    in ContextName context_name  
) raises (  
    UnknownServiceType,  
    InvalidServiceType,  
    InvalidContextName  
);
```

가

#### □ Findall Service Type

```
TypeStructSeq findall_servicetype (  
    in ServiceTypeName name,  
    in ContextName context_name  
) raises (  
    InvalidServiceType,  
    InvalidContextName  
);
```

## 8.2.2 Relationship Factory

가 (add), (delete), (modify)  
가 .

### □ Add Service Type Relationship

```
void add_relationship (
    in RelationshipName r_name,
    in ContextName context_name,
    in SourceType f_name,
    in TargetType t_name,
    in ServiceRelKind rel_type
) raises (
    InvalidRelationship,
    InvalidRelationshipKind,
    InvalidContextName,
    UnknownServiceType
);
```

가 . (source) ,  
(target) ( , )

“UnknownServiceType”

“InvalidRelationship”

“InvalidRelKind”

#### □ Delete Service Type Relationship

```
void delete_relationship(  
    in RelationshipName r_name,  
    in ContextName context_name  
) raises (  
    UnknownRelationship,  
    InvalidContextName  
);
```

“UnKnownRelationship”

#### □ Find Relationship

```
RelStruct find_Relationship (  
    in RelationshipName rel_name,  
    in ContestName context_name  
) raises (  
    InvalidRelationship,  
    UnknownRelationship,  
    InvalidContextName  
);
```

가

□ Type Matches

가 . (match)

```
TypeStruct matches (  
    in ServiceTypeName name,  
    in ContextName context_name  
) raises (  
    InvalidServiceType,  
    InvalidContextName  
);
```





가

(SDL, Service

type Declaration Language)

□

가

□

(specification)

□

가

□

(query)

가

□

가

[1] A. Beitz, M. Bearman and A. Vogel, "Service Location in an Open Distributed Environment", Proceedings of Second International Workshop on Services in Distributed and Networked Environments. ed. pp. 28-34 IEEE Computer Society Press, Los Alamitos, CA, (1995).

[2] A. Warren Pratten, James W. Hong, J. Michael Bennett, Michael A. Bauer and Hanan Lutfiyya, "Design and Implementation of a Trader-Based Resource Management System", Proceedings of CASCON'94, Oct 31 - Nov 3, 1994, Toronto, Ontario, Canada.

[3] Mirion Bearman and Kerry Raymond, "Federating Traders: An ODP Adventure", Proceedings of the IFIP TC6/WG6.4 International Workshop on Open Distributed Processing, Berlin, Germany, 8-11 October, 1991, publ. North-Holland, pp.125-141.

- [4] ISO/IEC 10746-1 ODP Reference Model-Part1: Overview, DIS, 1995.
- [5] ISO/IEC 10746-2 ODP Reference Model-Part2: Foundation, 1995.
- [6] ISO/IEC 10746-3 ODP Reference Model-Part3. Architecture, 1995.
- [7] ISO/IEC "ODP Trading Function", Final Draft 13235 June 1995.
- [8] M. Merz, K. Mueller and W. Lamersdorf, "Service Trading and Mediation in Distributed Computing Systems", Proceedings of the 14th International Conference on Distributed Computing Systems, Poznan, Poland, June 21-24, 1994, publ. IEEE Computer Society Press, p. 450-457.
- [9] Wayne Brookes and Jadwiga Indulska, "ODP Types and Their Management: an Object-Z Specification", Proceedings of the Second International Conference on Open Distributed Processing (ICODP'95), Brisbane, Australia, February 1995.
- [10] Wayne Brookes and Jadwiga Indulska, "A Type Management System for Open Distributed Processing", Technical Report 285, Department of Computer Science, The University of Queensland, February 1994.
- [11] Wayne Brookes and Jadwiga Indulska, "A Formal Specification of a Basic ODP-based Type Manager", Technical Report 286, Department of Computer Science, The University of Queensland, February 1994.
- [12] Ashley Beitz and Mirion Bearman, "An ODP Trading Service for DCE", Proceedings of the First International Workshop on Services in Distributed and Networked Environments (SDNE), June 1994, Prague, Czech Republic, publ. IEEE Computer Society Press, pp 42-49.
- [13] K. Duddy, K. Raymond and A. Vogel, "Trader Down Under: Upside Down and Inside Out", TAPOS, Special issue on CORBA, to appear. Technical Report #25, CRC for

Distributed Systems Technology, Brisbane, 1995.

[14] Lea Kutvonen, "Overview of the DRAYD Trading System Implementation". The IFIP/IEEE International Conference on Distributed Platforms, February 1996, Chapman and Hall, pp314-326.

[15] Andrew Waugh and Mirion Bearman, "Designing an ODP Trade Implementation using X.500", Proceedings of International Conference on Open Distributed Processing, February 1995, Brisbane, Australia.

[16] Kerry Raymond, "RM-ODP Tutorial",  
<http://www.dstc.edu.au/AU/research.news/odp/ref-model/papers.html>

[17] Thomas J.Mowbray and Ron Zahavi, "The Essential CORBA", pp35-54, OMG, 1995.

[18] Robert Orfali, Dan Harkey, Jeri Edwards, "The Essential Distributed Objects Survival Guide", pp91-122, Wiley, 1996

[19] IONA Technologies, Orbix Reference, IONA Technologies Ltd.,1995.

[20] IONA Technologies, Orbix Programming Guide, IONA Technologies Ltd.,1995.

[21] Mirion Bearman, "ODP-Trader", Proceedings of the IFIP TC6/WG6.1 International Conference on Open Distributed Processing, Berlin, Germany, 13-16 September, 1993, North-Holland, pp. 37-51.

[22] Luiz Augusto de Paula Lima Jr. and Edmundo Roberto Mauro Madeira, "A Model for a Federated Trader", Proceedings of the International Conference on Open Distributed Processing, February 1995, Brisbane, Australia.

[23] K. Mueller-Jones, M. Merz and W. Lamersdorf, "The TRADER: Integrating Trading into

DCE”, Proceedings of the International Conference on Open Distributed Processing, February 1995, Brisbane, Australia.

[24] W. Rosenbery, D. Kenny, G. Fisher, Understanding DCE, O'Reilly & Assoc. Inc, 1992.

[25] Rob van der Linden, An Overview of ANSA, APM.1000.01, July, 1993  
<http://www.ansa.co.uk/phase3-doc.root/sponsors/index.html>

[26] Jean\_Pierre Deschrevel, The ANSA Model for Trading and Federation, APM.1005.01, July 1993.

[27] Claudia Popien and Miguel Heineken, “Trading Enhancement by Service Combination in ODP”, Proceedings of the IFIP TC6/WG6.1 International Conference on Open Distributed Processing, Berlin, Germany, 13-16 September, 1993, pp384-386

[28] Jaga Indulska, Mirion Bearman and Kerry Raymond, “A Type Management System for an ODP Trader”, Proceedings of the IFIP TC6/WG6.1 International Conference on Open Distributed Processing, Berlin, Germany, 13-16 September, 1993, . North-Holland, pp. 169-180.

[29] OMG, “Object Service RFP5”, OMG TC Document 95-6-18, June 1995.

[30] Andreas Vogel, Mirion Bearman and Ashley Beitz, “Enabling Interworking of Traders”, Proceedings of the International Conference on Open Distributed Processing, February 1995, Brisbane, Australia.

[31] , , “CORBA ODP ”,  
96 가 , 23 2 , pp.1073-1084, 1996.

```

/* -----*/
/*  Type Manager CORBA IDL Specification          */
/* -----*/

#include <orb.idl>

module TypeManager
{
    /*-----*/
    /*  DATA Types          */
    /*-----*/

    const short max_size = 30; /* operation name max size */
    const short no_of_oper = 100; /* number of max operations */
    typedef string ServiceTypeName;

```

```

typedef string ContextName;
typedef string PropertyName;
typedef string InterfaceName;
typedef string OperationArray[max_size][no_of_oper];
typedef string RelationshipName;
typedef string SourceType;
typedef string TargetType;

enum PropertyMode {
    normal,          /* can be update or delated */
    read_only,      /* cannot be updated, can be deleted */
    fixed_normal,   /* can be updated, but not deleted */
    fixed_readonly, /* cannot be updated, cannot be deleted */
    undefined       /* used in exception */
};

struct PropStruct { /* property structure */
    PropertyName property_name;
    any Property_value;
    PropertyMode property_mode;
};
typedef sequence<PropStruct> PropStructSeq;

struct InterfaceStruct { /* interface structure */
    InterfaceName inter_name;
    OperationArrayopers;
};
typedef sequence<InterfaceStruct> InterfaceStructSeq;

struct IncarnationNumber {

```



```

        unsigned long high;
        unsigned long low;
};

struct TypeStruct { /* service type structure */
    ServiceTypeName service_name;
    PropStructSeq props;
    InterfaceStructSeq inters;
    IncarnationNumber incarnation;
};

typedef sequence<TypeStruct> TypeStructSeq;

enum ServiceRelKind {subtype, conversion};
struct RelStruct { /* service type relationship structure */
    RelationshipName r_name;
    SourceType f_type;
    TargetType t_type;
    ServiceRelKind rel_kind;
};

typedef sequence<RelStruct> RelStructSeq;

typedef sequence<ServiceTypeName> ServiceTypeNameSeq;

/*-----*/
/* exceptions for type manager */
/*-----*/

exception InvalidServiceType {};
exception UnKnownServiceType {};
exception InvalidInterfaceType {};
exception InvalidProperty {};

```

```

exception HasRelationship {};
exception InvalidRelationship {};
exception UnknownRelationship {};
exception InvalidRelationshipKind {};
exception InvalidContextName {};

interface ServiceTypeFactory;
interface RelationshipFactory;
interface LinkFactory;

/*-----*/
/* interface definition      */
/*-----*/

interface ServiceTypeFactory
{

    /* Add Service Type */
    enum ListOption {all,since};
    union SpeciedServiceTypes switch (ListOption) {
        case since: IncarnationNumber incarnation;
    };

    IncarnationNumber add_servicetype (
        in ServiceTypeName name,
        in ContextName context_name,
        in InterfaceStructSeq inters,
        in PropStructSeq props
    ) raises (
        InvalidServiceType,

```

```

        InvalidContextName,
        InvalidInterfaceType,
        InvalidProperty
    );

/* Delete Service Type */
void delete_servicetype (
    in ServiceTypeName name,
    in ContextName context_name
) raises (
    InvalidServiceType,
    UnKnownServiceType,
    InvalidContextName,
    HasRelationship
);

/* Modify Service Type */
void modify_servicetype (
    in ServiceTypeName name,
    in ContextName context_name,
    in InterfaceStructSeq inters,
    in PropStructSeq props
) raises (
    InvalidServiceType,
    InvalidContextName,
    UnKnownServiceType,
    InvalidProperty
);

```

```

/* List Service Type */
ServiceTypeNameSeq list_servicetype (
    in SpeciedServiceTypes which_types
);

/* Find Service Type */
TypeStruct find_servicetype (
    in ServiceTypeName name,
    in ContextName context_name
) raises (
    UnKnownServiceType,
    InvalidServiceType,
    InvalidContextName
);

/* Findall Service Type */
TypeStructSeq findall_servicetype (
    in ServiceTypeName name,
    in ContextName context_name
) raises (
    InvalidServiceType,
    InvalidContextName
);

}; /* End of ServiceTypeFactory */

```

```

interface RelationshipFactory

```

```
{
```

```
/* Add Service Type Relationship */
```

```
void add_relationship (
```

```
    in RelationshipName r_name,
```

```
    in SourceType f_name,
```

```
    in TargetType t_name,
```

```
    in ServiceRelKind rel_type,
```

```
    in ContextName context_name
```

```
) raises (
```

```
    InvalidRelationship,
```

```
    InvalidRelationshipKind,
```

```
    InvalidContextName,
```

```
    UnknownServiceType
```

```
);
```

```
/* Delete Service Type Relationship */
```

```
void delete_relationship(
```

```
    in RelationshipName r_name,
```

```
    in ContextName context_name
```

```
) raises (
```

```
    UnknownRelationship,
```

```
    InvalidContextName
```

```
);
```

```
/* Find Relationship */
```

```
RelStruct find_Relationship (
```

```
    in RelationshipName rel_name,
```

```
    in ContextName context_name
```

```
) raises (
```

```
        InvalidRelationship,  
        UnknownRelationship,  
        InvalidContextName  
    );  
  
    /* Type Matches */  
    TypeStruct matches (  
        in ServiceTypeName name,  
        in ContextName context_name  
    ) raises (  
        InvalidServiceType,  
        InvalidContextName  
    );  
}; /* End of Relationship Factory Interface */  
}; /* End of Type Manager Module */
```