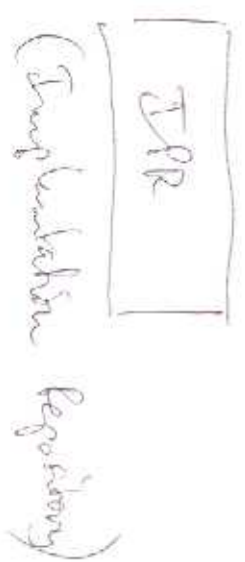
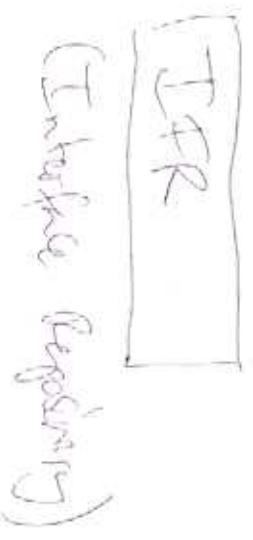
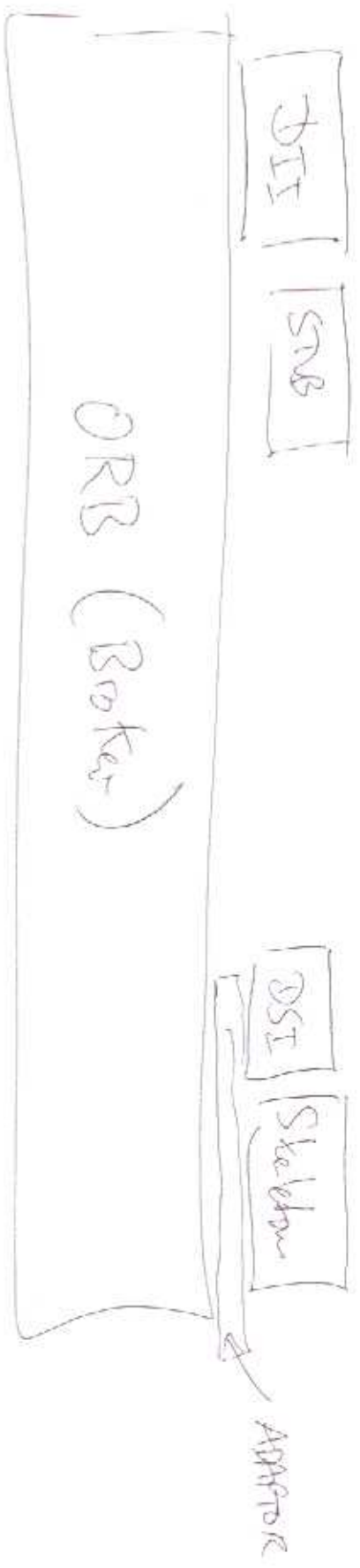
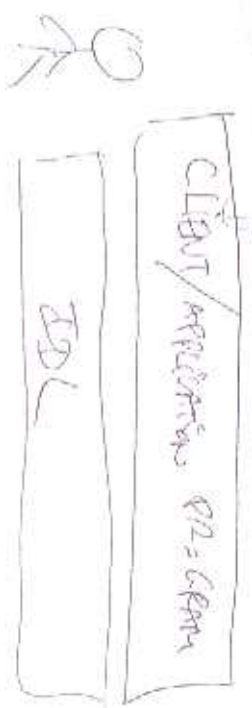


refl. get Name ()



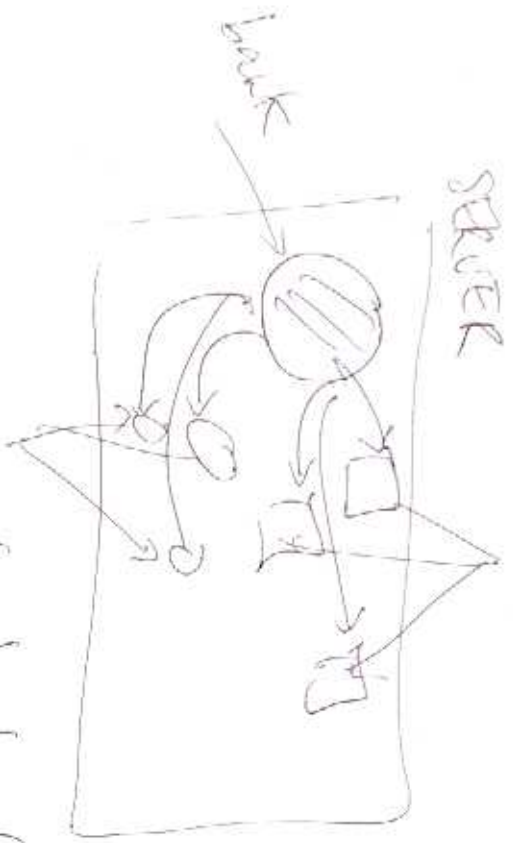
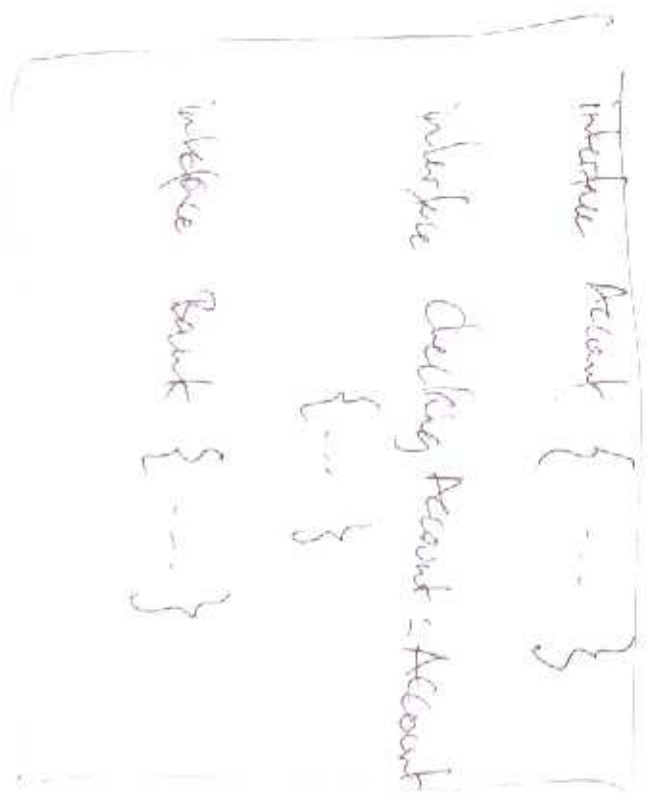
Compile Time IDL => Populate/Instantiate

- all information about server

The make-types of IFR (which describe

- interfaces
- attributes
- operations

SQL



create objects of type
CheckingAccount

13-2

objects of type
Account

Server (file)

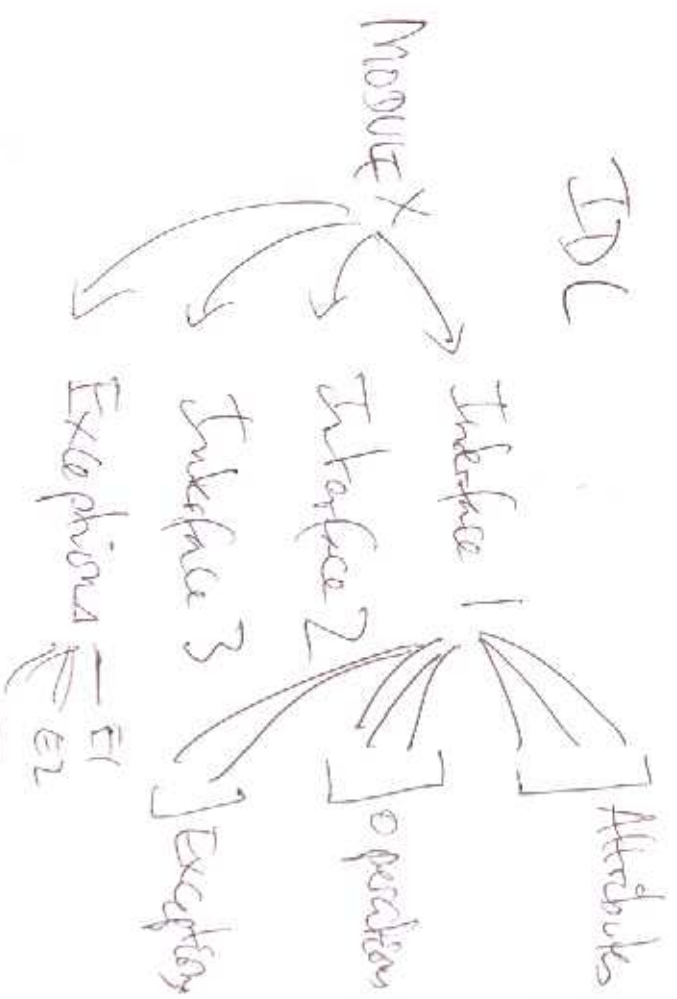
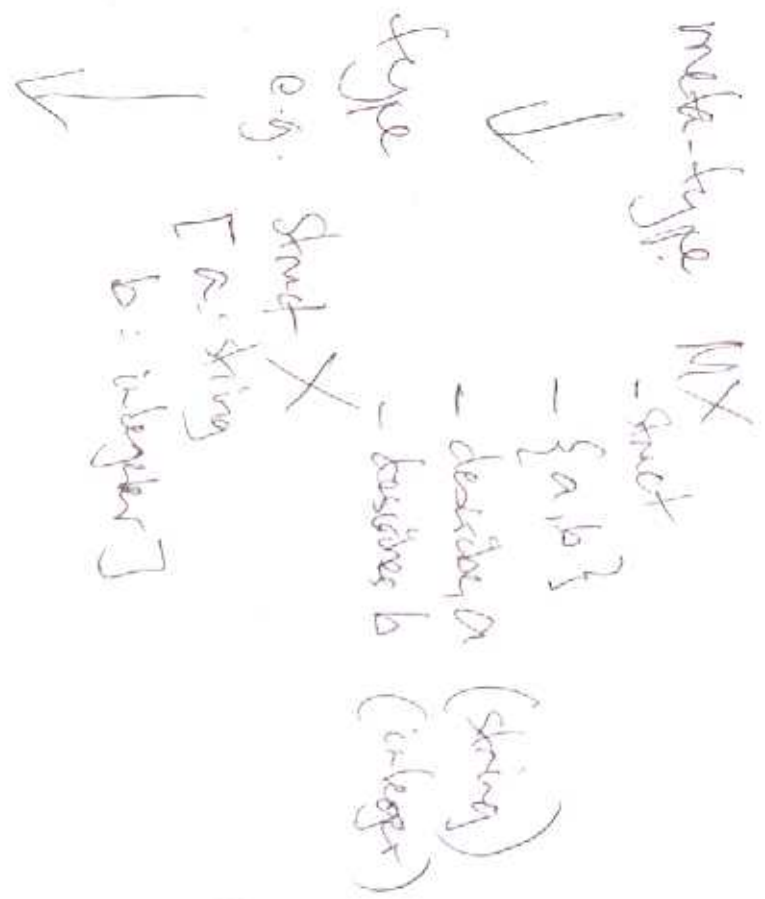
- initialise the ORB
- create ONE object of type bank

Server

- [Server implements Account
- [bank i - from CheckingAccount
- [Java [] of Bank

Idea of JFR is to fully describe all the information

within IDL files



Meta-Module X = INSTANCE OF META-TYPE

- list of interfaces = { Interface 1, Interface 2, ... }
- list of exceptions = { E1, E2, E3 }

IFR

Meta-type

CONTAINER

(describe all containers, like interfaces, modules, ...)

CONTAINED

(describe all elements that are members of a given container, e.g. interfaces, attributes, operations, exceptions)

Interface Def

(describe the information in a given interface)

Operation Def

Attribute Def

operations that browse the content of a container, content() look-up()...

operations that ~~also~~ describe the content of each element

List of Instances for operations

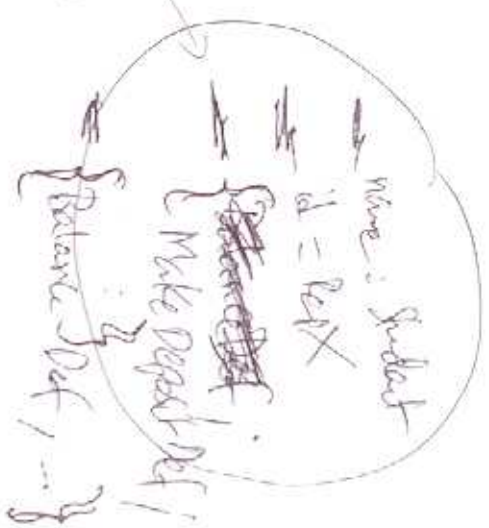
StudentDef = Instance of

the meta-type of

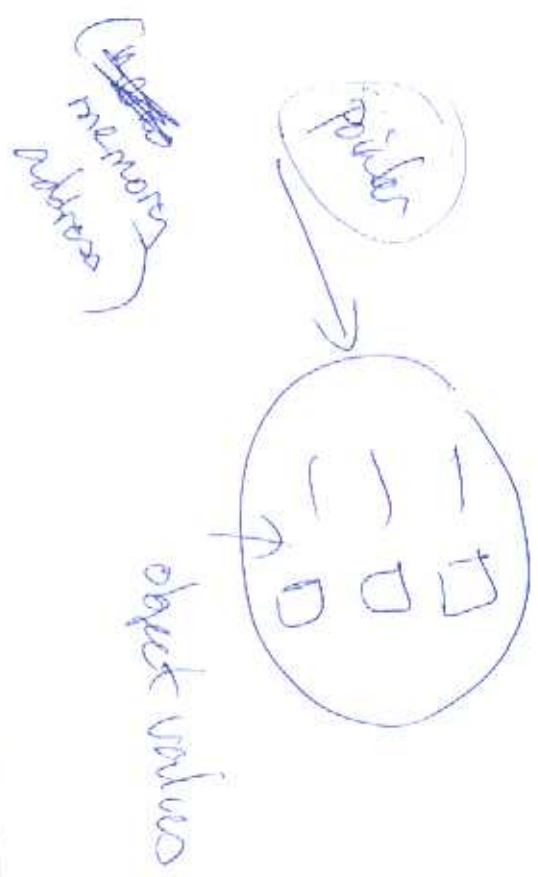
InterfaceDef for the interface Student

BalanceDef

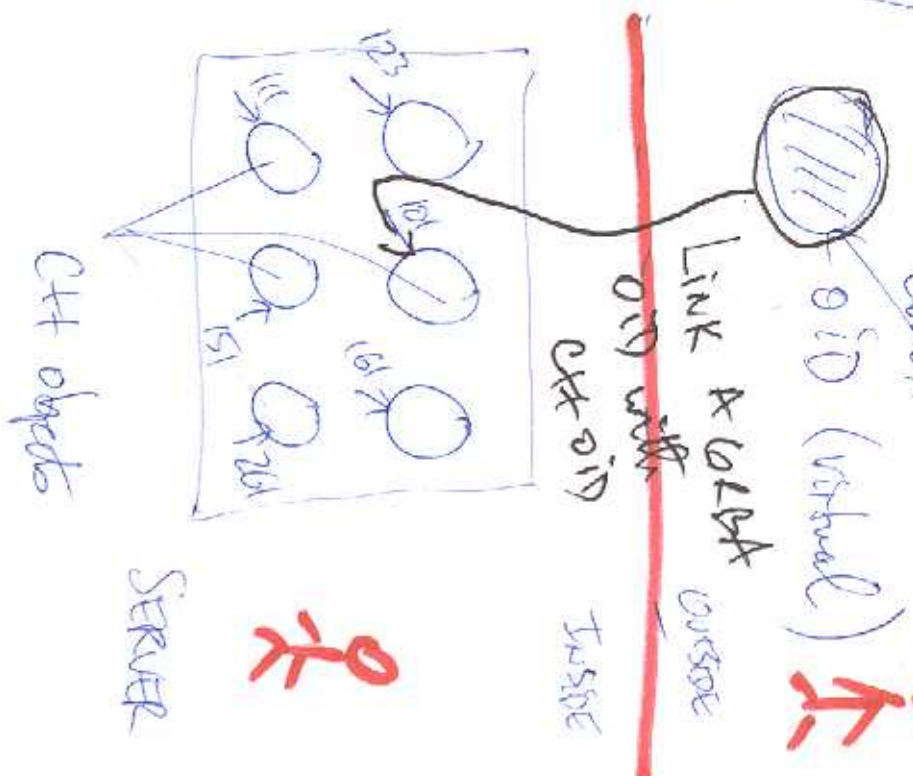
StudentDef



OBJECTS IDENTIFY (oid) → (points)



LINKS
ENVIRONMENT



DISTRIBUTED ENVIRONMENT

SERVER

9-7

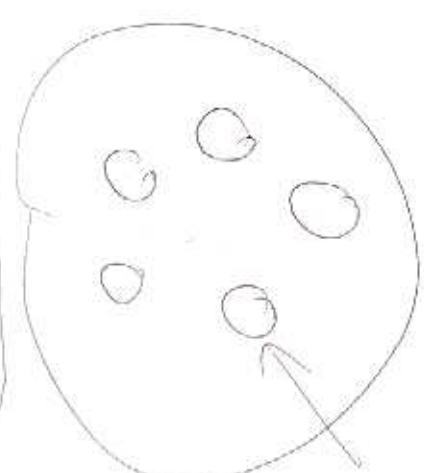
BINDING process

(Translating)

NO IMPLEMENTATION

1st step of demultiplexing

REPOSITORY

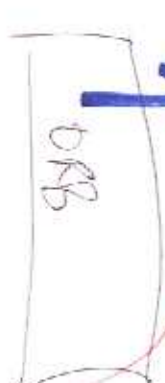


implementable objects

REF → OP()



IOP
(translate into CDR)



ADAPTER



ORB
Translate FWD(): translate into CDR
Translate BWD(): translate from CDR into Native representation

13-8

2nd Step



~~ADAPTER~~
~~ADAPTER~~
X

when created, you need specify the policy for processing requests



~~ADAPTER~~



1st WAY: I don't have a lot of objects

⇒ Share the data ^{implementation} within the objects

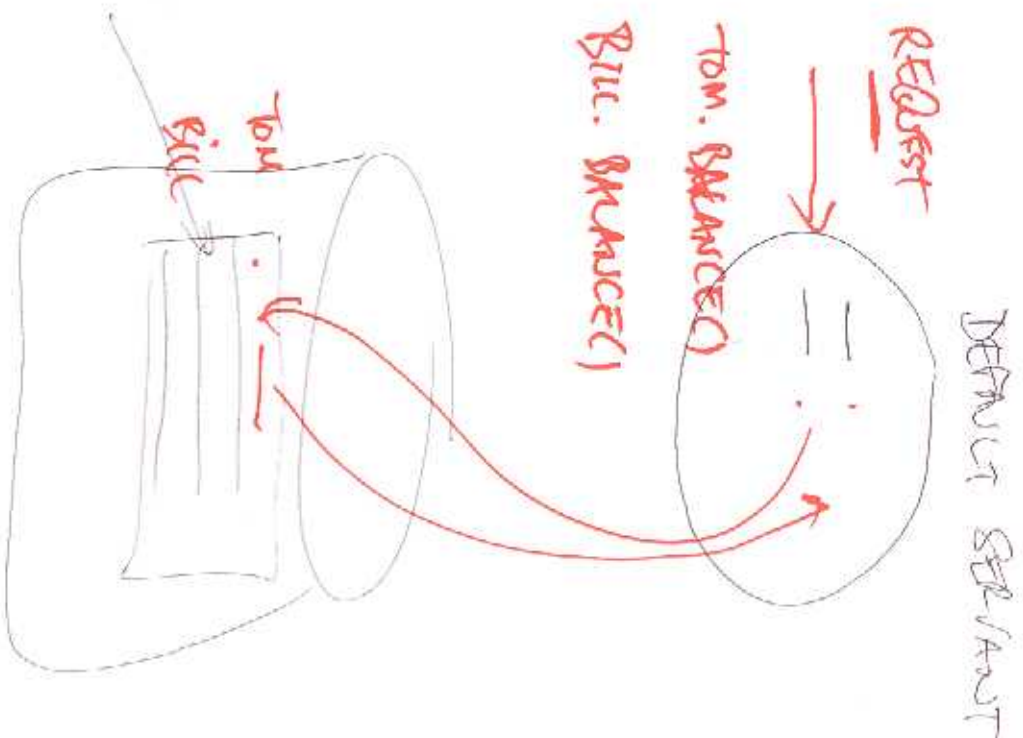
+ MAP TABLE

2nd WAY: Cannot use

MAP-TABLE WHEN LARGE/HEAVY OF SMALL OBJECTS ARE USED (e.g. bank application)

we do NOT store data
 within compilation objects.
 Instead, ~~open~~ will have just
 ONE DEFAULT OBJECTS
 to create/insert the
 in manipulation objects

REQUEST = REFERENCE + OPERATIONS
 Account table



TOM = reference ⇒ DEFAULT SERVANT

It needs to know
 the key (e.g name + surname)

HAS A-WAY WHICH
DATA to fetch from
 Database

13-10

A policy to how to create

CO RSA ~~OTD~~ OTD

~~OTD~~ OTD

TRANS PARENT
TO THE USER

(MAY TRIPLE)

NON-TRANSPARENT

= USER passes some information
to the adaptor

(e.g. Key information like
name and surname)

13-11