CORBA & Interpreted Languages

Duncan Temple Lang

July 18, 1998

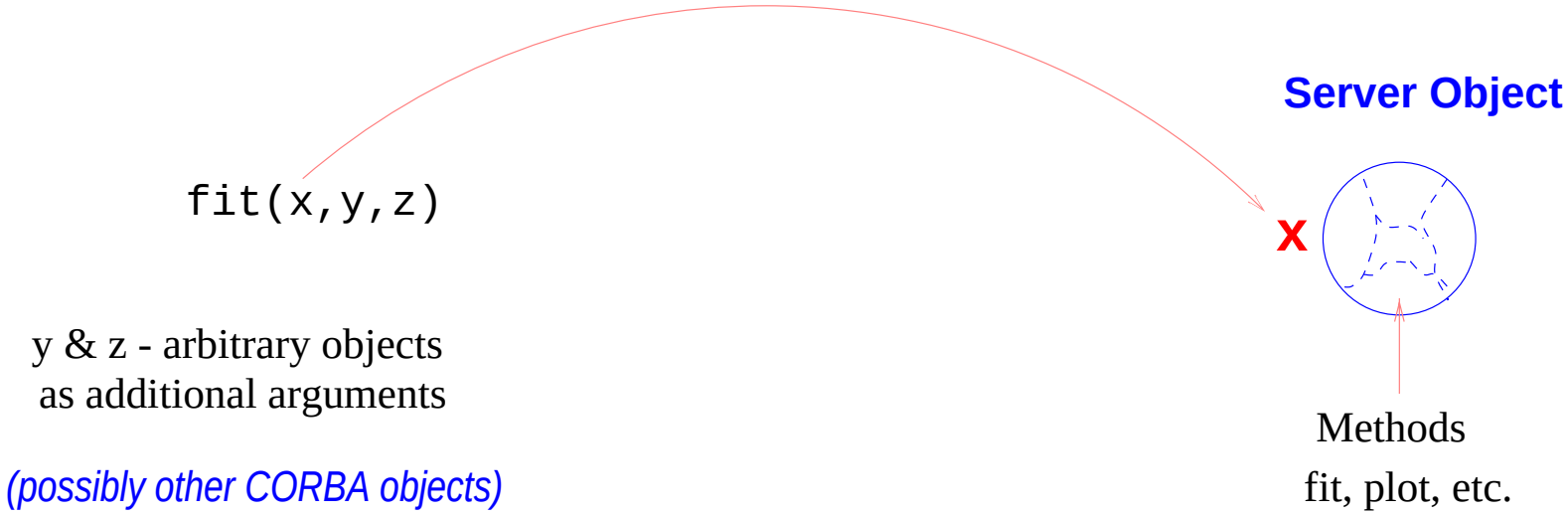
`cm.bell-labs.com/stat/duncan/DistComp/DynamicCORBA.html`

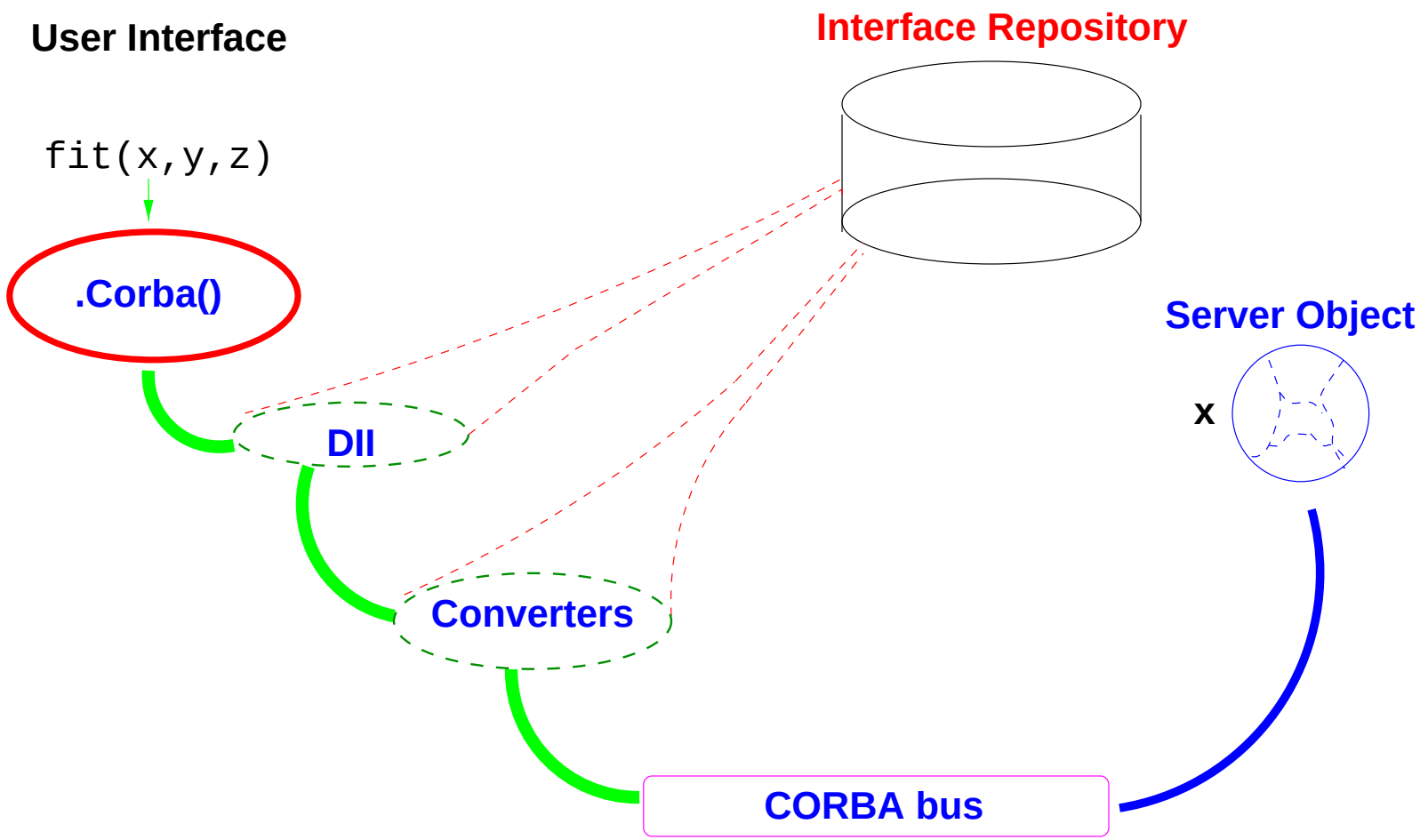
Lucent Technologies
Bell Labs Innovations



Goals

- Provide transparent access to and from interpreted languages in CORBA framework.
- S, Xlisp-Stat, Matlab, Java, etc.
- Make legacy code available
  with minimum amount of work.
- New packages be CORBA compliant without enforcing an unnatural programming style.
- Ideally, hide IDL specification from user.
- 2 scenarios - as Server and Client


Server Object
fit(x,y,z)
x
y & z - arbitrary objects
as additional arguments
(possibly other CORBA objects)
Methods
fit, plot, etc.


User Interface
Interface Repository
fit(x,y,z)
.Corba()
Server Object
x
DII
Converters
CORBA bus

As Client

S

- `CORBAObject` class - contains IOR.
- `plot(x,a,b,c)` interpreted as call to CORBA method `plot()` on object `x` with arguments **a,b,c**.
- Interface function `.Corba()`
  uses CORBA C++ bindings.
- Call `.Corba(x,"plot", y,z)`
- Use Dynamic Invocation Interface to create request at run time.
- Convert S arguments to CORBA objects - types specified by method's interface.
- Convert return value to S object.

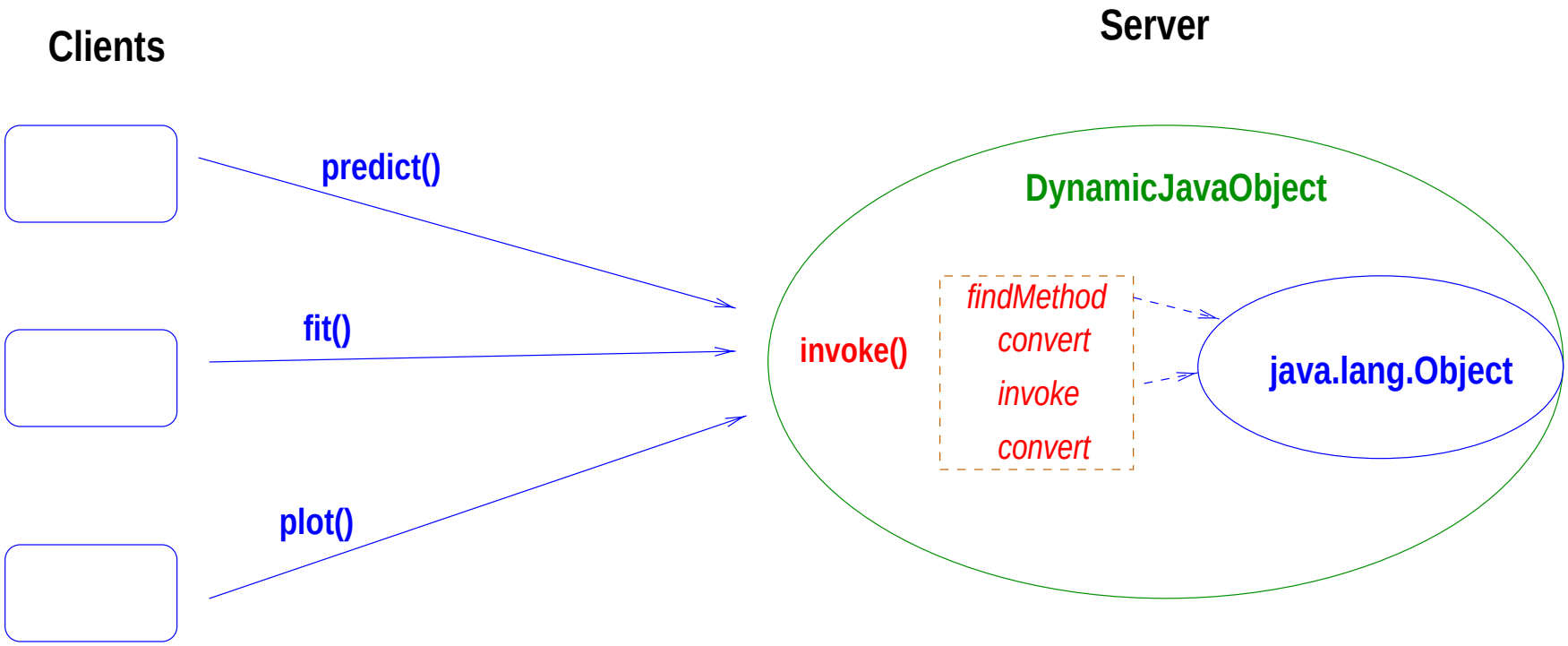
Conversion of CORBA and S Objects

To S

- Read description of target class from *IR*.
- Rely on attributes.
- Construct named list from object's attributes - recursive, terminating at primitive types, or
- create class corresponding to IDL and instantiate, or
- keep as CORBA reference (`CORBAObject`), or
- user-level converter functions or routines for different IDL interfaces.

Java Server

- Create Interface Repository (*IR*) entries. **`java2idl`** or **`java2iiop`**/**`Caffeine`**
- Static implementation of server for each IDL interface - manual or auto-generated code, or
- use *DynamicJavaObject* class - contains object in a generic CORBA class.
- Employs Dynamic Skeleton Interface (DSI) - `invoke()` method.
- Argument of `invoke()` identifies
  - name of method
  - named argument list
- Use `findMethod()` and `Method.invoke()`.
- Convert arguments from and results to CORBA objects.


Clients
Server
predict()
fit()
plot()
DynamicJavaObject
invoke()
findMethod
convert
invoke
convert
java.lang.Object

S Server

- Publish class attributes and selected methods into *IR*.
- Code instance of server object (C++), or
- single S-level function for arbitrary objects.
- Employs DSI.
- Pass object, name and interface type (from *IR*).
- Event loop and `invoke()` calls corresponding server object's S method.
- Convert arguments to S objects

Complications

- No polymorphism in IDL to match source languages.
- Different semantics of languages.
- Argument names not available in Java.
- Instantiaion of objects not known at compile time.

DynamicJavaObject

- Augment `invoke()` to implement specific invocations.
- Conversion of objects
  non-direct mapping of methods
- Override method selection.