

Understanding Applications Built on Middleware Using CORBA Packet Traces

September 18, 1998

Dr. John Zinky

jzinky@bbn.com

Richard Shapiro

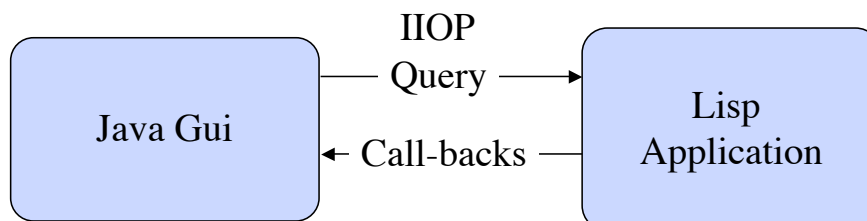
rshapiro@bbn.com

Dr. Ken Anderson

kanderson@bbn.com

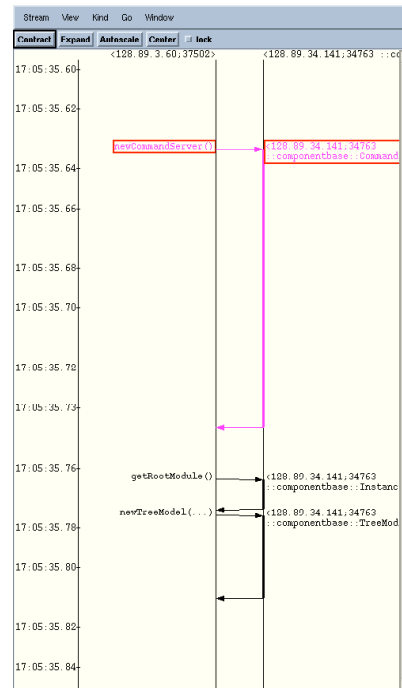
Application Interactions are to Many Objects Going Over two IIOP Connections

The screenshot shows a SpyGlass CORBA packet trace window. The window title is "File View Go Window" and it has buttons for "Load", "More", "Hidden", "Refresh", and "Lock". The main area displays a list of network packets. A specific packet is highlighted with a red box, showing details: "ROOT -> 192.168.1.104 -> ETHERNET -> IP -> ATOM APT. BBN.COM 32828.32830 -> TCP -> 32828-32830 -> IIOP -> CORBA". The packet list contains many entries, each starting with a timestamp and showing the flow of data between the Java GUI and the Lisp Application.



The First Call Is to the Factory Object Which Creates Objects

- **The Factory interface** must be specified because the Java Client got the IOR out of band
- **The IOR for all other objects** are in the parameter list of other method calls. Hence the interfaces for these objects will be determined automatically.



3 02/98 Demo

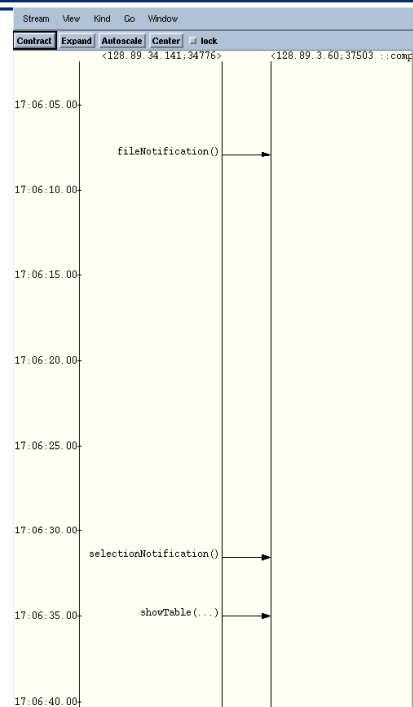
SpyGlass CORBA Example

John Zinky

BBN Technologies

When the Results of Operations are Completed the Java GUI is Called-back

- These Calls are “one-way” Calls and have no return message.



4 02/98 Demo

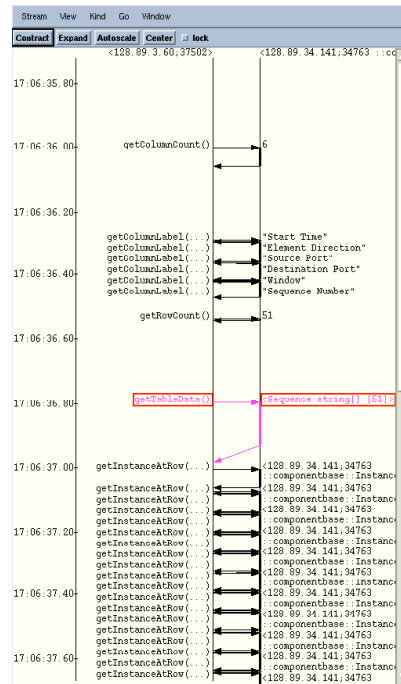
SpyGlass CORBA Example

John Zinky

BBN Technologies

A Table Model Manages a Sequence of Objects, such as a Stream of PDUs or a List of Fields for a PDU

- getColumnLabel gets a string for each Column Label
- GetTableData gets a 2D list of strings for table data
- Each row is an Object



5 02/98 Demo

SpyGlass CORBA Example

John Zinky

BBN Technologies

Magnifying a CORBA Method Call Displays the Parameter List and Basic Call Timing.

Method Call Has:

- In Arguments
- In Context
- Out Arguments
- Return Value

| Element | Kind | Window |
|---------------------|------|---------------------|
| Start Time | | 10:51:47.06 |
| Element Direction | | NIL |
| Element Class | | CORBA-CALL |
| Object | | RemoteObj |
| Requestor | | <128.11.0.15:44268> |
| Request | | someMethod 11 |
| Arguments | | () |
| Reply | | NO_EXCEPTION 11 |
| Return Value | | () |
| Return Args | | NIL |
| Request Duration | | 0.0001 |
| Reply Duration | | 0.0001 |
| Processing Duration | | 0.0117 |

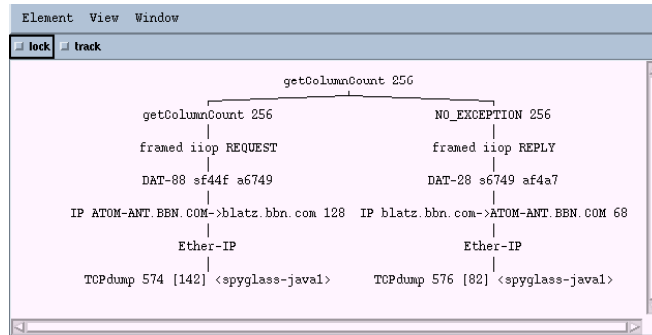
6 02/98 Demo

SpyGlass CORBA Example

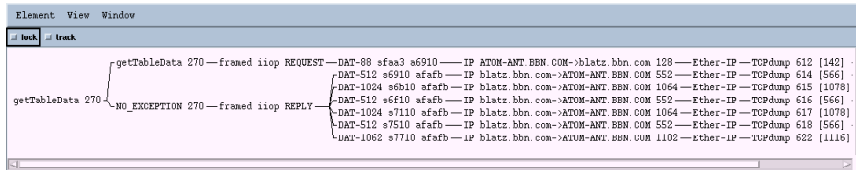
John Zinky

BBN Technologies

A CORBA Method Call Combines an IIOB Request with an IIOB Reply



- Method calls with large parameter list can combine 100's of TCP packets.



7 02/98 Demo

SpyGlass CORBA Example

John Zinky

BBN Technologies

The Application's IDL is Loaded into the Interface Repository

- The Factory interface
- Command Server
- Tree Model
- Table Model
- Instance
- Callback

```

Kind Window
lock track
Repository
Module componentbase
<typedef table>
Interface Factory
<operation newTreeModel>
<operation newTableModel>
<operation newCommandServer>
<operation getModule>
Interface CommandServer
<operation loadTraceFile>
Interface Callback
<operation showTree>
<operation showTable>
<operation selectionNotification>
<operation fileNotification>
Interface TreeModel
<operation setShowHidden>
<operation isLeaf>
<operation getIndexofChild>
<operation getChildCount>
<operation getChild>
<operation getRoot>
Interface TableModel
<operation removeColumn>
<operation addColumn>
<operation getColumnAttributeIndex>
<operation getAttributeAt>
<operation getInstanceAtRow>
<operation getTableData>
<operation getColumnLabel>
<operation getValueAt>
<operation getRowCount>
<operation getColumnCount>
Interface Instance
<operation doOperation>
<operation select>
<operation getOperations>
<operation isSelected>
<operation getDisplayName>
<typedef strings>
  
```

8 02/98 Demo

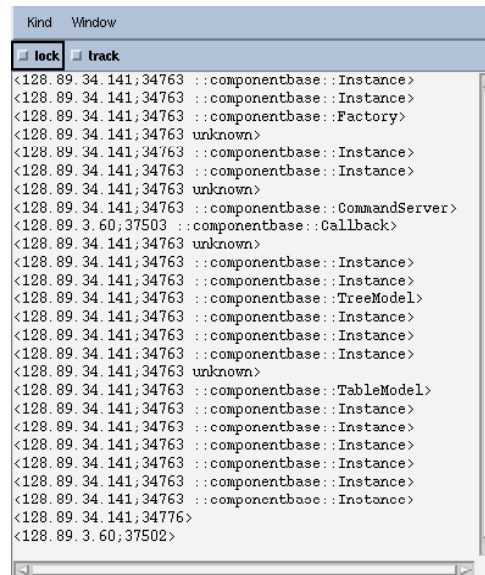
SpyGlass CORBA Example

John Zinky

BBN Technologies

The Object Repository Keeps Track of All the Object That have been Identified

- **Objects** can be defined by:
 - IIOP connections
 - IORs in Parameter list
 - Explicitly Reading an IOR
 - User Specification
- **Client processes** are also identified.
 - Note that these processes can contain CORBA Objects.
- Parameter lists must be **displayed** in order to parse IORs.

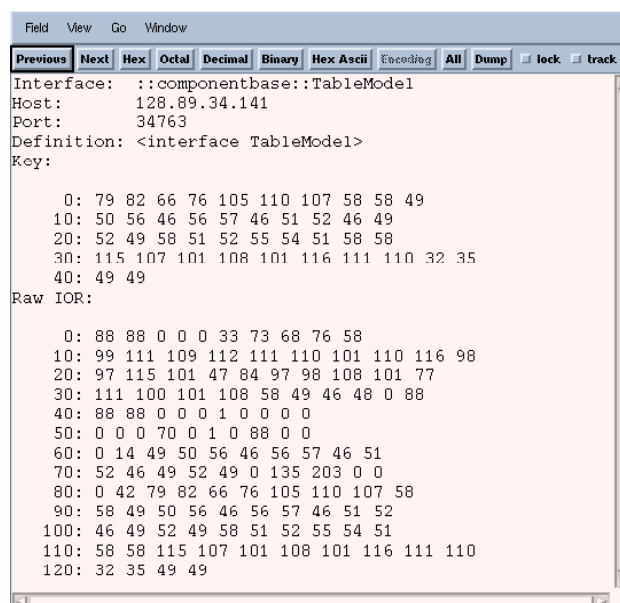


```
Kind Window
lock track
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Factory>
<128.89.34.141;34763 unknown>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 unknown>
<128.89.34.141;34763 ::componentbase::CommandServer>
<128.89.3.60;37503 ::componentbase::Callback>
<128.89.34.141;34763 unknown>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::TreeModel>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 unknown>
<128.89.34.141;34763 ::componentbase::TableModel>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.34.141;34763 ::componentbase::Instance>
<128.89.3.60;37502>
```

Objects Definitions can be Edited Some of the Attributes can be Unknown.

Objects Contain:

- Host
- Port
- Object Key
- Interface Type
- Display Name



```
Field View Go Window
Previous Next Hex Octal Decimal Binary Hex Ascii Encoding All Dump lock track
Interface: ::componentbase::TableModel
Host: 128.89.34.141
Port: 34763
Definition: <interface TableModel>
Key:
  0: 79 82 66 76 105 110 107 58 58 49
  10: 50 56 46 56 57 46 51 52 46 49
  20: 52 49 58 51 52 55 54 51 58 58
  30: 115 107 101 108 101 116 111 110 32 35
  40: 49 49
Raw IOR:
  0: 88 88 0 0 0 33 73 68 76 58
  10: 99 111 109 112 111 110 101 110 116 98
  20: 97 115 101 47 84 97 98 108 101 77
  30: 111 100 101 108 58 49 46 48 0 88
  40: 88 88 0 0 0 1 0 0 0 0
  50: 0 0 0 70 0 1 0 88 0 0
  60: 0 14 49 50 56 46 56 57 46 51
  70: 52 46 49 52 49 0 135 203 0 0
  80: 0 42 79 82 66 76 105 110 107 58
  90: 58 49 50 56 46 56 57 46 51 52
  100: 46 49 52 49 58 51 52 55 54 51
  110: 58 58 115 107 101 108 101 116 111 110
  120: 32 35 49 49
```