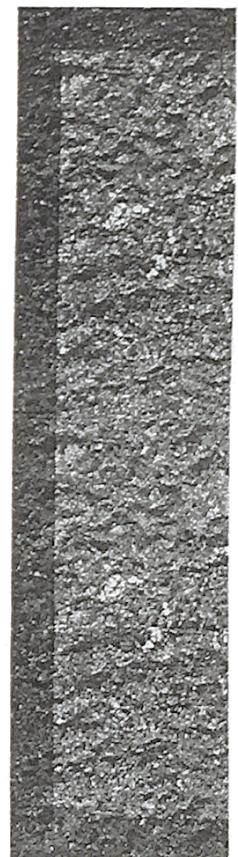
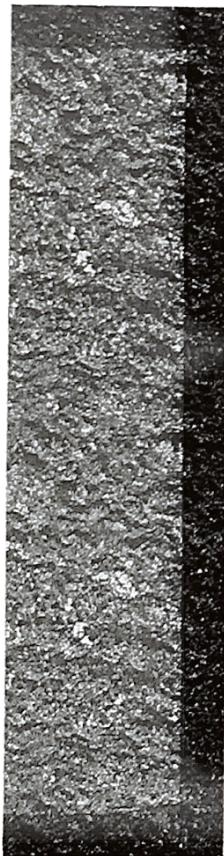


## *Part II*

### *The Solution*

## *Conclusion: Avoid explicit Bridging !*

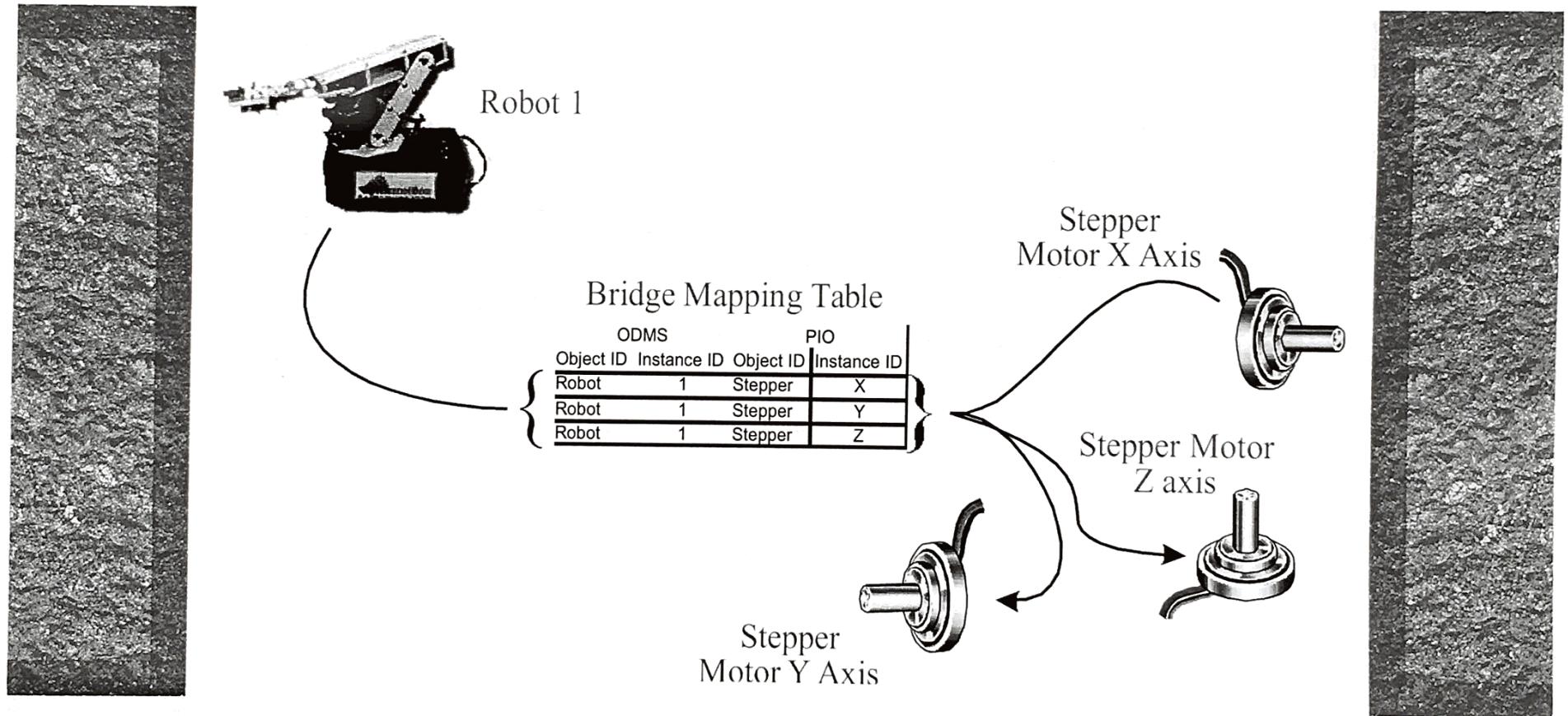
- *BUT, Q: What do we do instead ?*
- *A: The Architecture/Translator automatically inserts the necessary Bridging.*
- *Q: Excuse me? How ?*
- *A: By following either a mapping rule or an enumerated mapping table called a Half Table.*

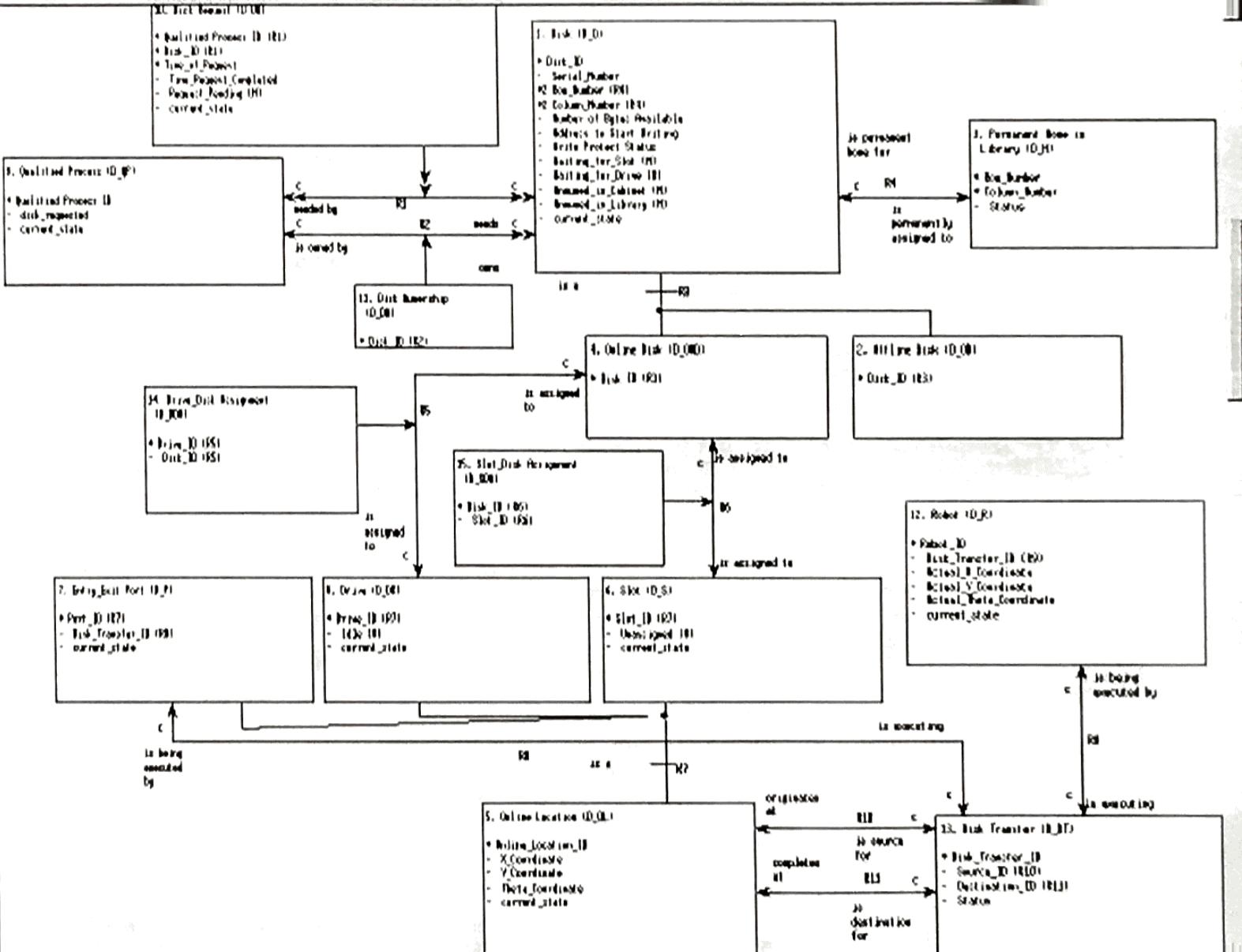


## *Example*

- *The code automation process generates `get{attribute.Name}` and `set{attribute.Name}` accessor functions.*
- *These normally just set or return the appropriate value in the memory allocated to the instance.*
- *But we have control over the translator, it could read a table and sometimes generate something different if we wished.*

## *Example - The ODMS*





14. Locking Door on Eject

D\_P2:EE Port  
Locked

D\_P11:Robot Removal/  
Set S  
Assigned

Select a  
Generate

D\_P1:Operator  
Closes Door

D\_P6:Operator  
Opens Door

13. Waiting for Operator to Close  
Door on Eject

D\_P5:Operator  
Removes Disk

D\_P4:Operator  
Inserts Disk

12. Waiting for Operator to  
Remove Disk

D\_P6:Operator  
Opens Door

D\_P1:Operator  
Closes Door

11. Waiting for Operator to Open  
Door on Eject

D\_P3:Port Unlocked

10. Unlocking the Door on Eject

Select one disk\_tfr related by  
Self->D\_DT[R8];

1. Idle

Select many disk\_tfr\_set from instances of D\_DT;  
For each tmp\_disk\_tfr in disk\_tfr\_set

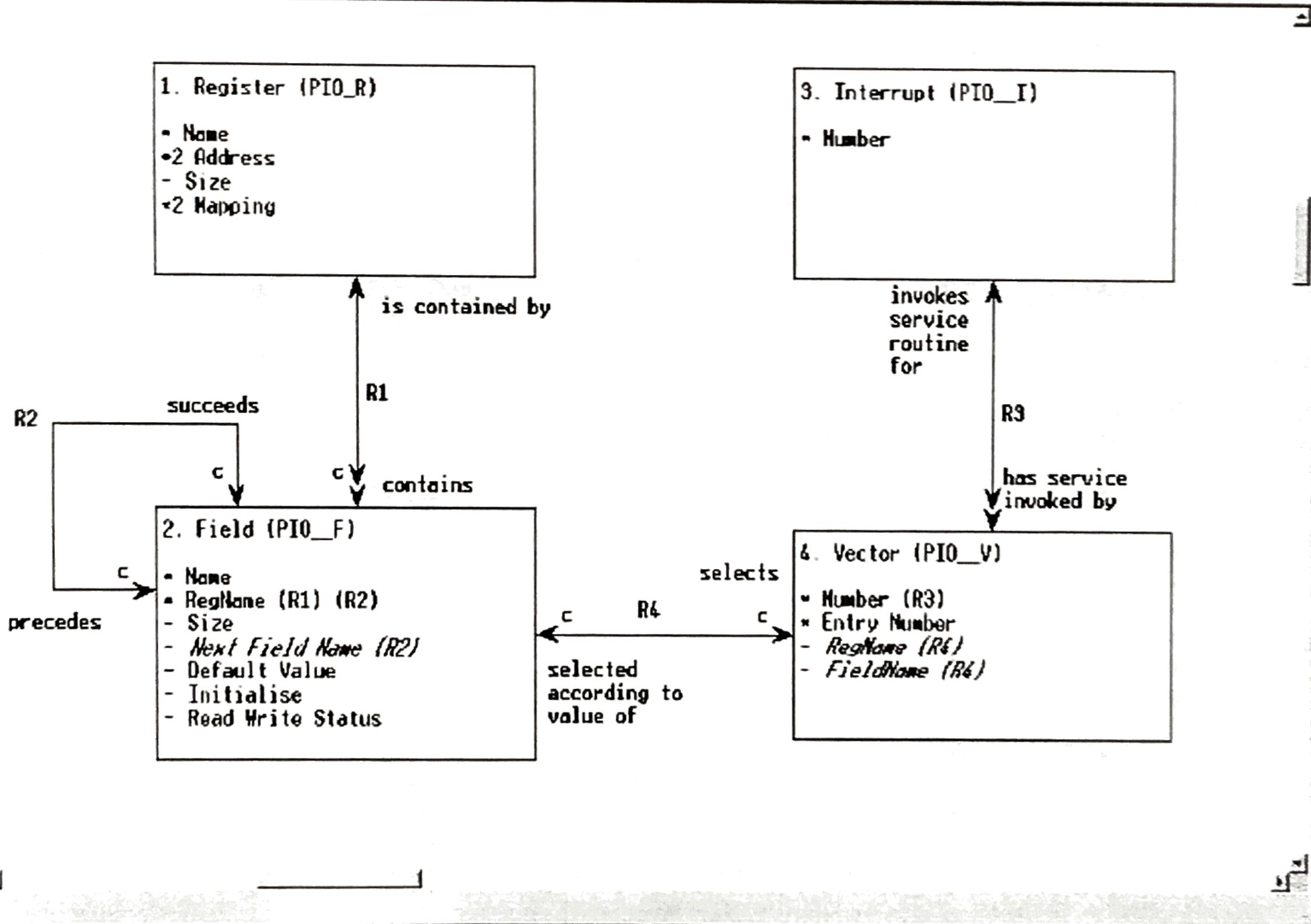
```
// Status is Ready For Port  
If (tmp_disk_tfr.Status == 0)  
    Generate D_P10:'EE Port Request Pending' () To Self:  
    Select one disk_tfr related by Self->D_DT[R8];  
    Unrelate Self from disk_tfr across R8;  
End If;  
End for;
```

D\_P10:EE Port  
Request Pending

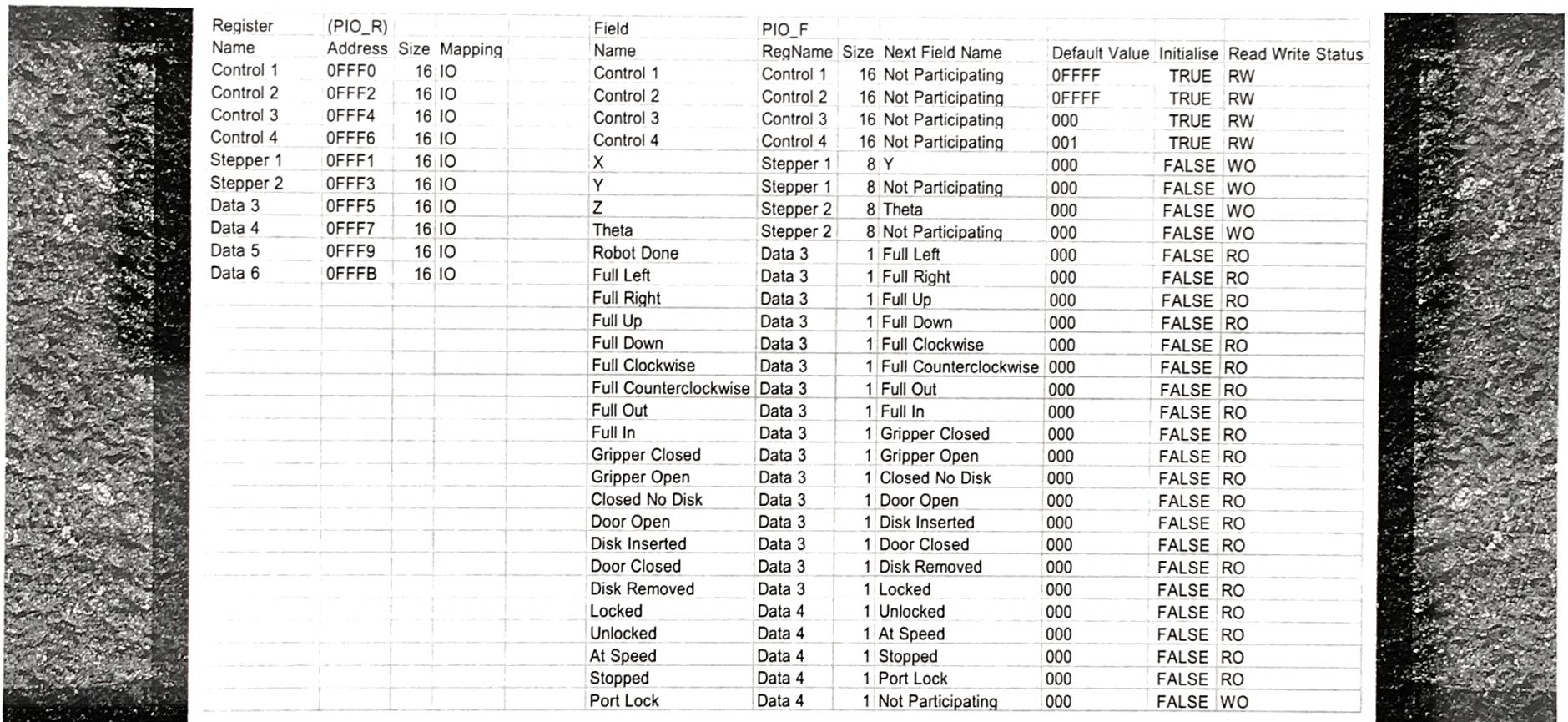
2. Selecting a Disk Transfer

Select many disk\_tfr\_set from instances of D\_DT;  
For each disk\_tfr in disk\_tfr\_set

```
// Select Status = Ready For Port  
If (disk_tfr.Status == 0)  
    If (disk_tfr.Source_ID == Self.Port_ID)  
        Generate D_P7:'Insert Request Selected' () to Self:  
    Else  
        Generate D_P8:'Eject Request Selected' () to Self:  
    End if;  
    Relate Self to disk_tfr across R8;
```

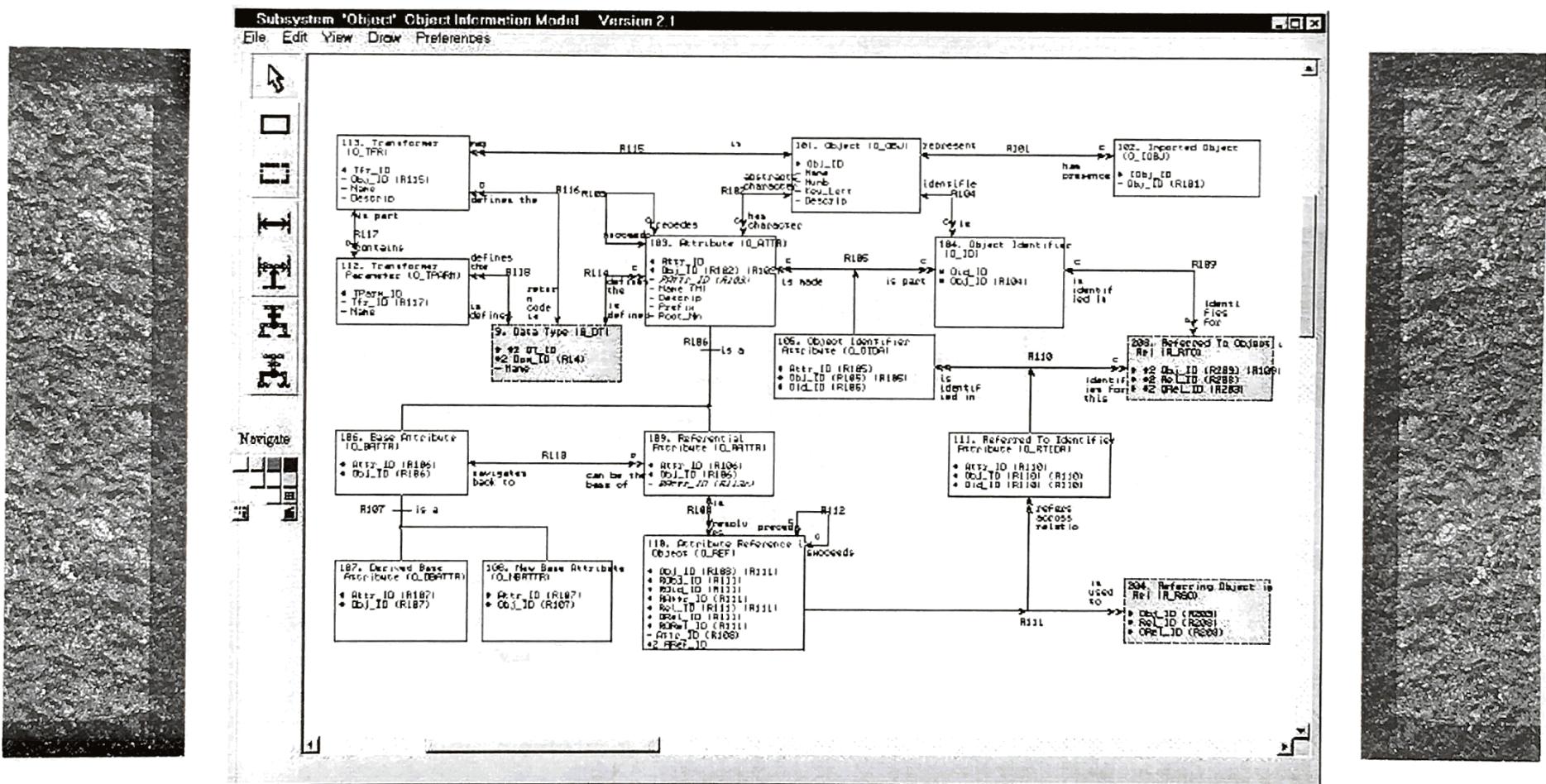


## *PIO Domain Instances*

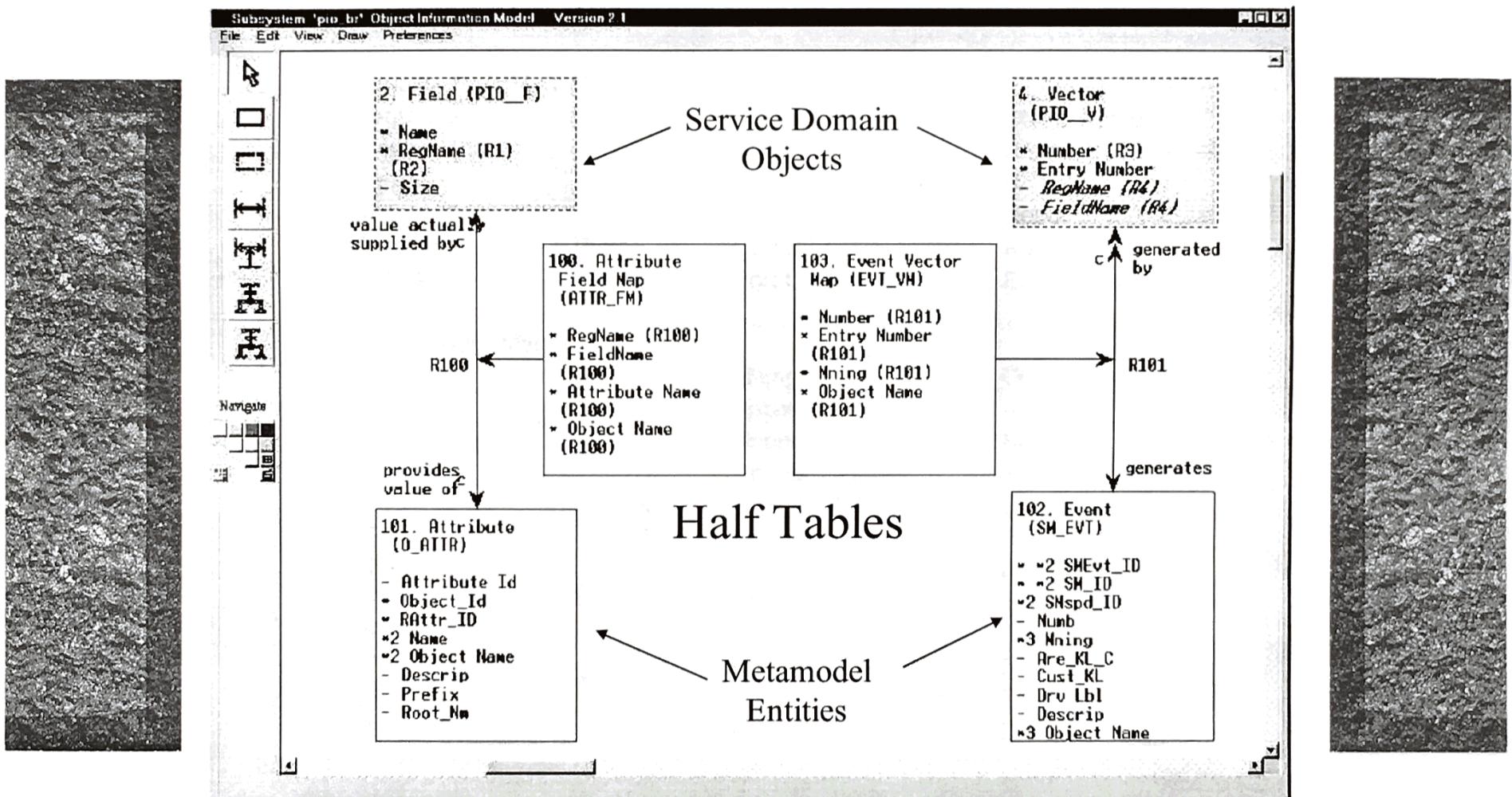


Register	(PIO_R)	Field	PIO_F	Default Value	Initialise	Read	Write	Status
Name	Address	Size	Mapping	Name	RegName	Size	Next Field Name	
Control 1	0FFF0	16	IO	Control 1	Control 1	16	Not Participating	0FFFF TRUE RW
Control 2	0FFF2	16	IO	Control 2	Control 2	16	Not Participating	0FFFF TRUE RW
Control 3	0FFF4	16	IO	Control 3	Control 3	16	Not Participating	000 TRUE RW
Control 4	0FFF6	16	IO	Control 4	Control 4	16	Not Participating	001 TRUE RW
Stepper 1	0FFF1	16	IO	X	Stepper 1	8	Y	000 FALSE WO
Stepper 2	0FFF3	16	IO	Y	Stepper 1	8	Not Participating	000 FALSE WO
Data 3	0FFF5	16	IO	Z	Stepper 2	8	Theta	000 FALSE WO
Data 4	0FFF7	16	IO	Theta	Stepper 2	8	Not Participating	000 FALSE WO
Data 5	0FFF9	16	IO	Robot Done	Data 3	1	Full Left	000 FALSE RO
Data 6	0FFFB	16	IO	Full Left	Data 3	1	Full Right	000 FALSE RO
				Full Right	Data 3	1	Full Up	000 FALSE RO
				Full Up	Data 3	1	Full Down	000 FALSE RO
				Full Down	Data 3	1	Full Clockwise	000 FALSE RO
				Full Clockwise	Data 3	1	Full Counterclockwise	000 FALSE RO
				Full Counterclockwise	Data 3	1	Full Out	000 FALSE RO
				Full Out	Data 3	1	Full In	000 FALSE RO
				Full In	Data 3	1	Gripper Closed	000 FALSE RO
				Gripper Closed	Data 3	1	Gripper Open	000 FALSE RO
				Gripper Open	Data 3	1	Closed No Disk	000 FALSE RO
				Closed No Disk	Data 3	1	Door Open	000 FALSE RO
				Door Open	Data 3	1	Disk Inserted	000 FALSE RO
				Disk Inserted	Data 3	1	Door Closed	000 FALSE RO
				Door Closed	Data 3	1	Disk Removed	000 FALSE RO
				Disk Removed	Data 3	1	Locked	000 FALSE RO
				Locked	Data 4	1	Unlocked	000 FALSE RO
				Unlocked	Data 4	1	At Speed	000 FALSE RO
				At Speed	Data 4	1	Stopped	000 FALSE RO
				Stopped	Data 4	1	Port Lock	000 FALSE RO
				Port Lock	Data 4	1	Not Participating	000 FALSE WO

# *The Object Metamodel Subsystem*



# *Introducing . . . Half Tables*



## *The Half Table Instances*

Attribute Field Map (ATTR_FM)			
RegName	FieldName	Attribute Name	Object Name
Stepper 1	X	Current X Position	Robot
Stepper 1	Y	Current Y Position	Robot
Stepper 2	Z	Current Z Position	Robot
Stepper 2	Theta	Current Theta Position	Robot
Data 4	Port Lock	Port Locked	Entry Exit Port

Event Vector Map			
Number	Entry Number	Mning	Object Name
0F7	1	Robot Move Complete	Robot
0F7	2	Gripper Closed On Disk	Robot
0F7	3	Gripper Opened	Robot
0F7	4	Operator Opens Door	Entry Exit Port
0F7	5	Operator Inserts Disk	Entry Exit Port
0F7	6	Operator Closes Door	Entry Exit Port
0F7	7	Operator Removes Disk	Entry Exit Port
0F7	8	EE Port Locked	Entry Exit Port
0F7	9	Port Unlocked	Entry Exit Port
0F7	10	Drive At Spin Speed	Drive
0F7	11	Drive Stopped	Drive

## *Code without the Bridge:*

```
*****
Function name : D_R_set_Actual_x_coordinate
Author       :
Date        :
Return Desc. :
Description  :
*****
void D_R_set_Actual_x_coordinate(unsigned int Robot_id, unsigned int
value)
{
    Robots [Robot_id] .Actual_x_coordinate = value ;
}
```

## *Code with the Bridge :*

```
*****
Function name : D_R_set_Actual_x_coordinate
Author       :
Date        :
Return Desc.  :
Description   :
*****
void D_R_set_Actual_x_coordinate(unsigned int Robot_id, unsigned int
value)
{
    set_pio_data_1_x(value) ;
}
```

## *Code Generated from PIO Instances:*

```
typedef struct {
    int pio_x : 8 ;
    int pio_y : 8 ;
} pio_data_1_type ;

void set_pio_data_1_x(int new_value)
{
    pio_data_1_type register_value ;
    register_value = (pio_data_1_type)input(0xFFFF1) ;
    register_value.pio_x = new_value ;
    output(register_value, 0xFFFF1) ;
}

int get_pio_data_1_x(void)
{
    return ((pio_data_1_type)input(0xFFFF).pio_x) ;
}
```

# *The Archetype :*

```
*****
Function name : $U{object.Key_Lett}_set_$cr{attribute.Name}
Author       :
Date        :
Return Desc. :
Description  :
*****
.select one data_type related by attribute->S_DT[R114]
.assign my_type = "unsigned int"
.if (data_type.Name == "integer")
void $U{object.Key_Lett}_set_$cr{attribute.Name}(unsigned int $cr{object.Name}_id, unsigned int value)
{
.//
// Attribute Process Input/Output Singleton Field Map
// ATTR_PIO_SFM Bridge defined in PIO_singleton_schema.sql
// Mappings in PIO_singleton_instances.sql
.//
.select any pio_sfm_bridge_mapping from instances of ATTR_FM where ((selected.ObjectName == object.Name)
                                         AND (selected.AttributeName == attribute.Name))
.if ((not_empty pio_sfm_bridge_mapping)
    AND ((Field.Read_Write_Status == "RW") OR (Field.Read_Write_Status == "WO")))
    set_pio$_1_{pio_sfm_bridge_mapping.RegName}$_1_{pio_sfm_bridge_mapping.FieldName}(value) ;
.else
    $cr{object.Name}s[$cr{object.Name}_id].$cr{attribute.Name} = value ;
.end if
}
```

## *More Advanced Stuff*

- *The preceding example is in use today in real products*
- *However, it is a simple case called a ‘Singleton Mapping’*
- *No instance resolution is needed.*
- *There are a number of ways to implement instance resolution.*

