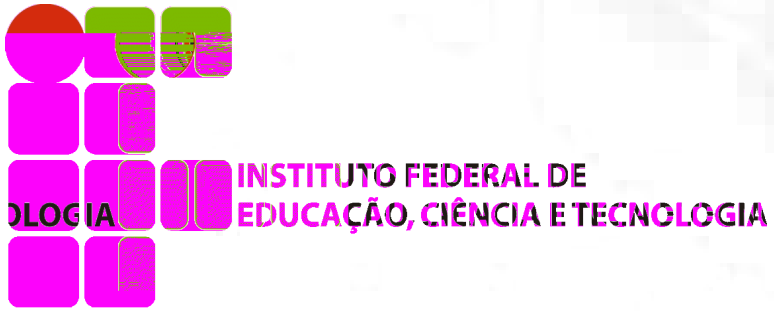
Bridge

Sandro Santos Andrade

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas



Bridge

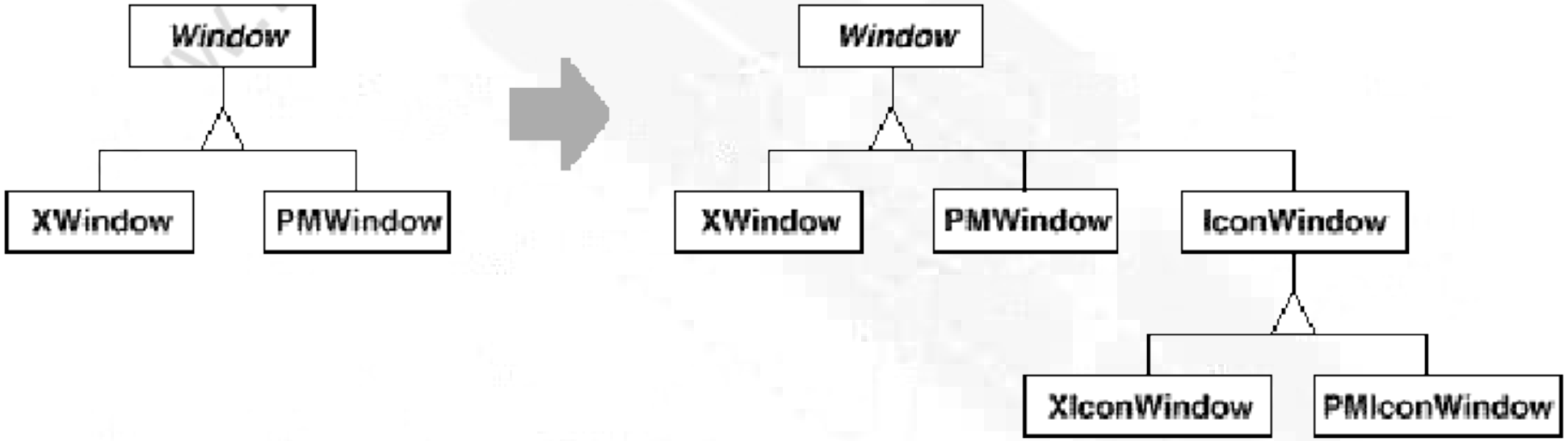
-
  -
-
-
  -
  -

Handle/Body

Bridge

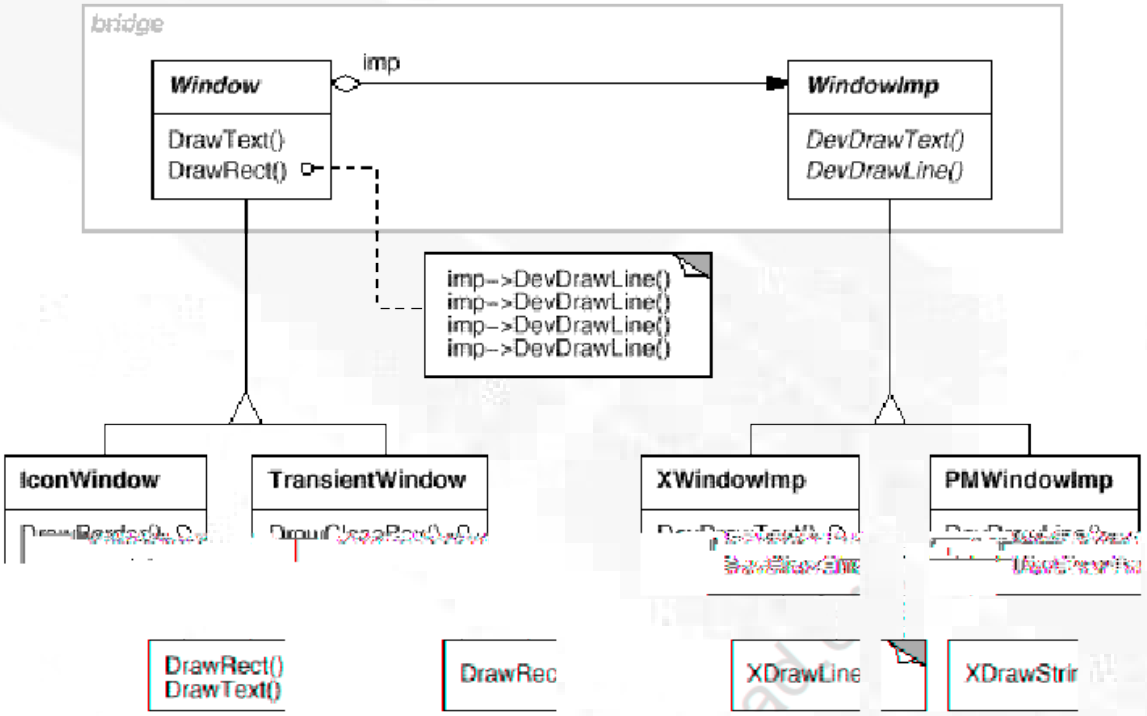
-
  - *Window* *toolkit*
  - *X Window* *Presentation Manager*
  - *Window* *XWindow* *PMWindow*

Bridge


Window
XWindow
PMWindow
Window
XWindow
PMWindow
IconWindow
XIconWindow
PMIconWindow

Bridge

Bridge

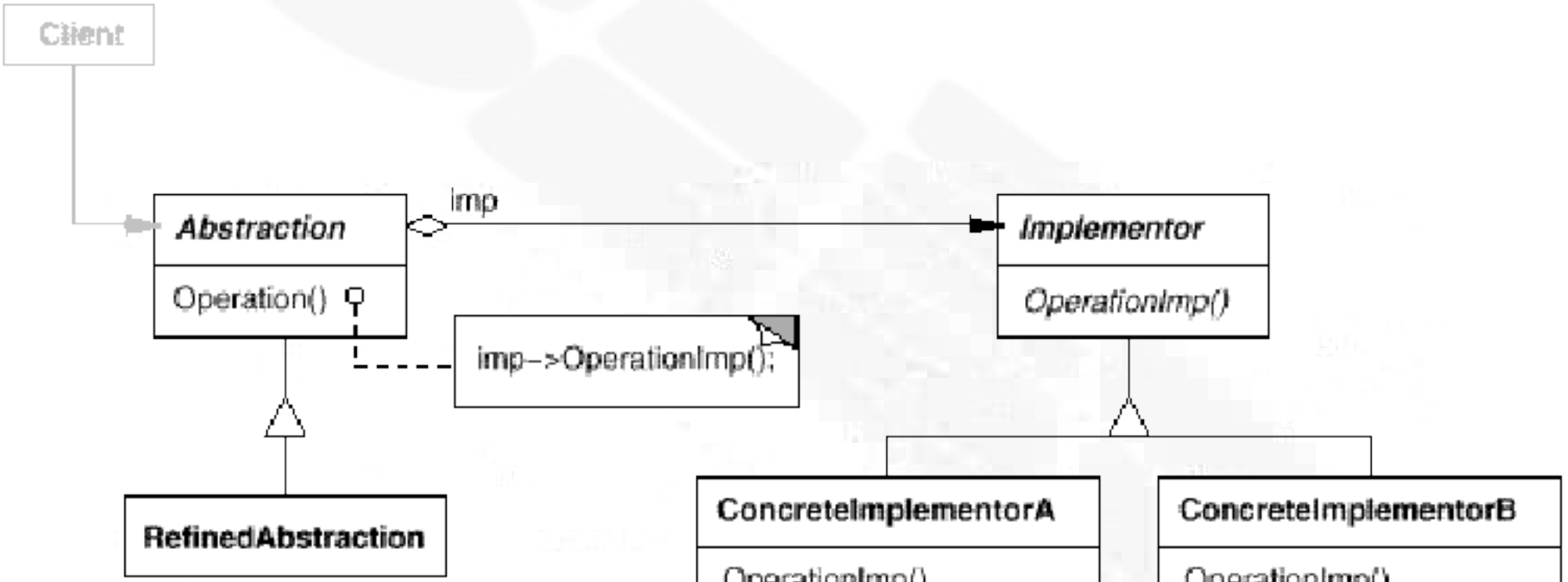


Window

WindowsImpl

Bridge

Bridge


Client
Abstraction
Operation()
imp
Implementor
OperationImp()
imp->OperationImp();
RefinedAbstraction
ConcreteImplementorA
OperationImp()
ConcreteImplementorB
OperationImp()

Bridge

-
  - *Abstraction*
  - *RefinedAbstraction*
    Abstraction
  - *Implementor*
  - *ConcreteImplementor*
    Implementor

Bridge

-
  - *Abstraction*
    Implementor

Bridge

-
  -
    -
    -
  -
    -
  -

run-time

Abstraction

Implementor

Bridge

-
  - *Implementor*
    Implementor *Bridge*

    link
  - *Implementor*
    - *Abstraction*
    - *default*
      run-time
    -

Bridge

```
Handle& Handle::operator= (const Handle& other)  {
    other._body->Ref();
    _body->Unref();

    if (_body->RefCount() == 0) {
        delete _body;
    }
    _body = other._body;

    return *this;
}
```

Bridge

```
class Window {
public:
    Window(View* contents);

    // requests handled by window
    virtual void DrawContents();

    virtual void Open();
    virtual void Close();
    virtual void Iconify();
    virtual void Deiconify();

    // requests forwarded to implementation
    virtual void SetOrigin(const Point& at);
    virtual void SetExtent(const Point& extent);
    virtual void Raise();
    virtual void Lower();

virtual void DrawLine(const Point&, const Point&);
virtual void DrawRect(const Point&, const Point&);
virtual void DrawPolygon(const Point[], int n);
virtual void DrawText(const char*, const Point&);

tected:                                                    pro
WindowImp* GetWindowImp();
View* GetView();

vate:                                                      pri
WindowImp* _imp;
View* _contents; // the window's contents
                                                           };
```

Bridge

```
class WindowImp {
public:
    virtual void ImpTop() = 0;
    virtual void ImpBottom() = 0;
    virtual void ImpSetExtent(const Point&) = 0;
    virtual void ImpSetOrigin(const Point&) = 0;

    virtual void DeviceRect(Coord, Coord, Coord, Coord) = 0;
    virtual void DeviceText(const char*, Coord, Coord) = 0;
    virtual void DeviceBitmap(const char*, Coord, Coord) = 0;
    // lots more functions for drawing on windows...
protected:
    WindowImp();
};
```

Bridge

```
class ApplicationWindow : public Window {
public:
    // ...
    virtual void DrawContents();
};

void ApplicationWindow::DrawContents () {
    GetView()->DrawOn(this);
}
```

```
class IconWindow : public Window {
public:
    // ...
    virtual void DrawContents();
private:
    const char* _bitmapName;
};
```

Bridge

```
void IconWindow::DrawContents() {
    WindowImp* imp = GetWindowImp();
    if (imp != 0) {
        imp->DeviceBitmap(_bitmapName, 0.0, 0.0);
    }
}
```

```
void Window::DrawRect (const Point& p1, const Point& p2) {
    WindowImp* imp = GetWindowImp();
    imp->DeviceRect(p1.X(), p1.Y(), p2.X(), p2.Y());
}
```

Bridge

```
class XWindowImp : public WindowImp {
public:
    XWindowImp();

    virtual void DeviceRect(Coord, Coord, Coord, Coord);
    // remainder of public interface...
private:
    // lots of X-window-system-specific state, including
    Display* _dpy;
    Drawable _winid; // window id
    GC _gc;          // window graphic context
};
```

Bridge

Bridge

Bridge

-
  - *Abstract Factory* *Bridge*
  - *Adapter*
  - *Bridge*

Bridge

Sandro Santos Andrade

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas