

CAP

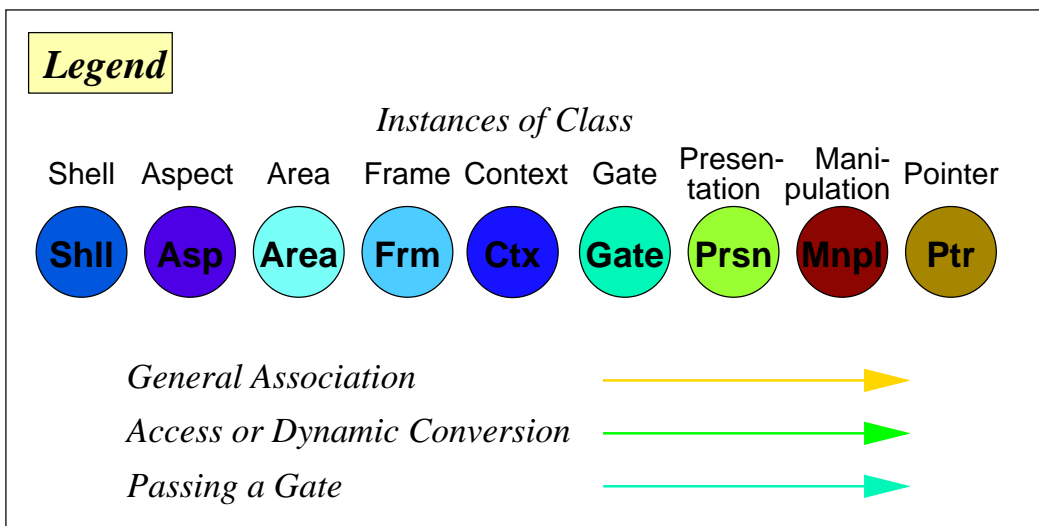
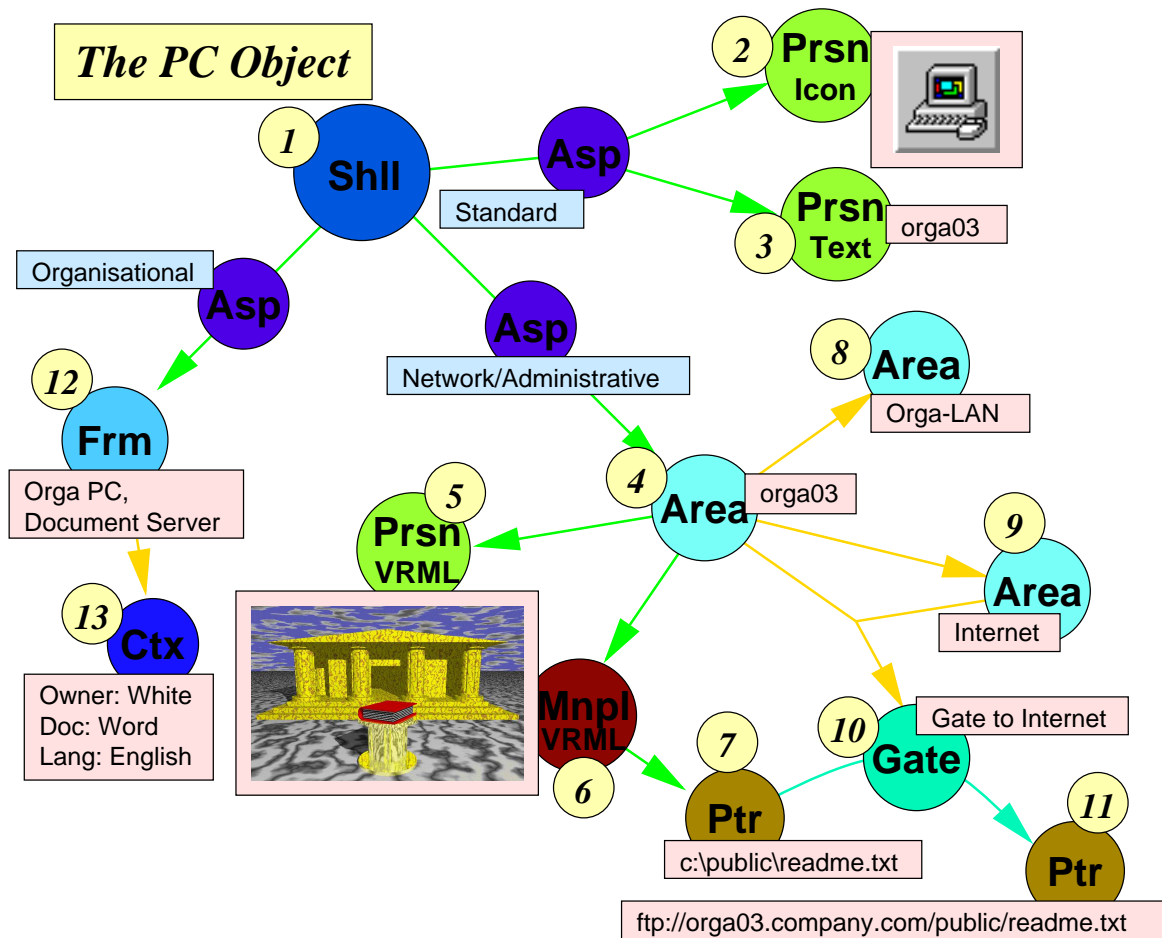
The CyberSpace Architecture Project

©1994,95 by Andreas Leue

EXAMPLE

A SIMPLE CASE STUDY

Example: Working with a COA Object



Commentary and C++ Code Examples

1

At the beginning, you get an object representing a personal computer. Since you work with CAP, your first access to that object is through the **Shell** interface.

```
Shell * PC = GetPC(); // get PC Shell from somewhere
```

2

Your favourite CyberSpace Visualizer requests a representation from the **Shell**. To do that, it uses *Dynamic Conversion*. As the target the class **Presentation/Icon** was specified. You haven't stated any special **Aspect** yet, so the standard one is used. As a result, an icon is returned.

```
Prsn * Icon = DC(                // dynamically convert (DC)
    PC,                          // from PC
    PrsnIcon,                    // to class "PrsnIcon"
    DEFAULT_ASPECT              // with default aspect
);                               // (DC is a macro)
```

3

Additionally, the visualizer requests an object of class **Presentation/Text** to put a text under the icon.

```
Prsn * Text = DC(                // dynamically convert (DC)
    PC,                          // from PC
    PrsnIcon,                    // to class "PrsnText"
    DEFAULT_ASPECT              // with default aspect
);
```

4

Now you want to have a look into the PC. You wonder, whether there are any network services available. Hence you request an **Area** corresponding to the PC by using again *Dynamic Conversion*. As an **Aspect** you specify "Network/Administrative". You get an **Area** object.

```
Area * PCArea = DC(PC, Area, Aspect("Network", "Administrative"));
// again, DC is used, the target class is "Area"
// the aspect is created as a temporary object
```

5

Again, the CyberSpace Visualizer requests a presentation from the Area. There is some VRML (Virtual Reality Markup Language) information available. The visualizer shows a 3D scene.

```
Prsn * Graphic = DC(PC, PrsnGraphic, DEFAULT_ASPECT);
                // a graphical representation is requested
```

6

The scene is interactive, so a Manipulator object is also created. In the graphic you see a nice greek building (surely there are documents stored inside: a file server). In front of it, there is a small pillar with a book on it's top. It looks like that you should read this book. So you select it by clicking on it with your mouse.

```
Mnpl * Manipulator = Graphic.GetMnpl(); // request associated Manipulator from
                                         // graphic object
```

7

The Manipulator object translates the mouse selection to a Pointer object representing that selection. It points to a file on the PC, named "readme.txt".

```
Ptr * Selection = Manipulator.GetSelection(); // get the currently selected object
                                              // from the scene
```

8

Since you do not want to read the file now, you consider sending the Pointer to your PC at home. To perform this, you examine the environment of the PC whether there is an exit. First, you find that the PC's Area is contained in another Area called "Orga-LAN".

```
Area * SuperArea = PCArea.GetFirstSuperArea(); // get the first superarea from
                                                // the PC area (if any)
Prsn * AreaName = DC(SuperArea, PrsnText, DEFAULT_ASPECT); // request a simple text
// AreaName now contains "Orga-LAN"
```

9

Then you discover that there is another super Area, the Internet. It seems more appropriate for your purpose.

```
Area * SuperArea = PCArea.GetNextSuperArea(); // get the next superarea from
                                                // the PC area (if any)
Prsn * AreaName = DC(SuperArea, PrsnText, DEFAULT_ASPECT); // request a simple text
// AreaName now contains "Internet"
```

10

To pass the **Pointer** to the Internet, you request a **Gate** between the PC's Area and the Internet Area. You succeed and get a **Gate** object.

```
Gate * ExitToInternet = PCArea.GetGateTo(SuperArea); // get gate from PCArea
                                                       // to SuperArea
```

11

Passing the **Pointer** through the **Gate** automatically converts the **Pointer** to a format usable outside the PC in the Internet. Now you can send it to your PC at home.

```
Ptr * InternetPtr = ExitToInternet.Pass(Ptr); // pass Ptr through the gate to
                                              // get a new pointer
```

12

Before you release the object, you want to know for what purposes the PC is used in the company. Since you are interested in contextual informations, you request a **Frame** from the **Shell** and specify an "organisational" **Aspect**. You get a **Frame** called "Orga PC, Document Server".

```
Frame * PCFrame = DC(PC, Frame, Aspect("Organisational"));
// again, get a frame using DC
```

13

Associated with each **Frame** is a **Context** (That's the reason why you requested a **Frame**). The context states that there's a person named "White" who is responsible for the PC. You also get informed that documents stored on the PC are in "Word" format and that the language to be used is "English".

```
Ctx * Context = Frame.GetContext(); // get context associated with frame
cout << Context["Owner"] << Context["Doc"] << Context["Lang"]; // put some
    // contextual values to cout; you get "WhiteWordEnglish"
```