

Cake: a tool for adaptation of object code

Stephen Kell

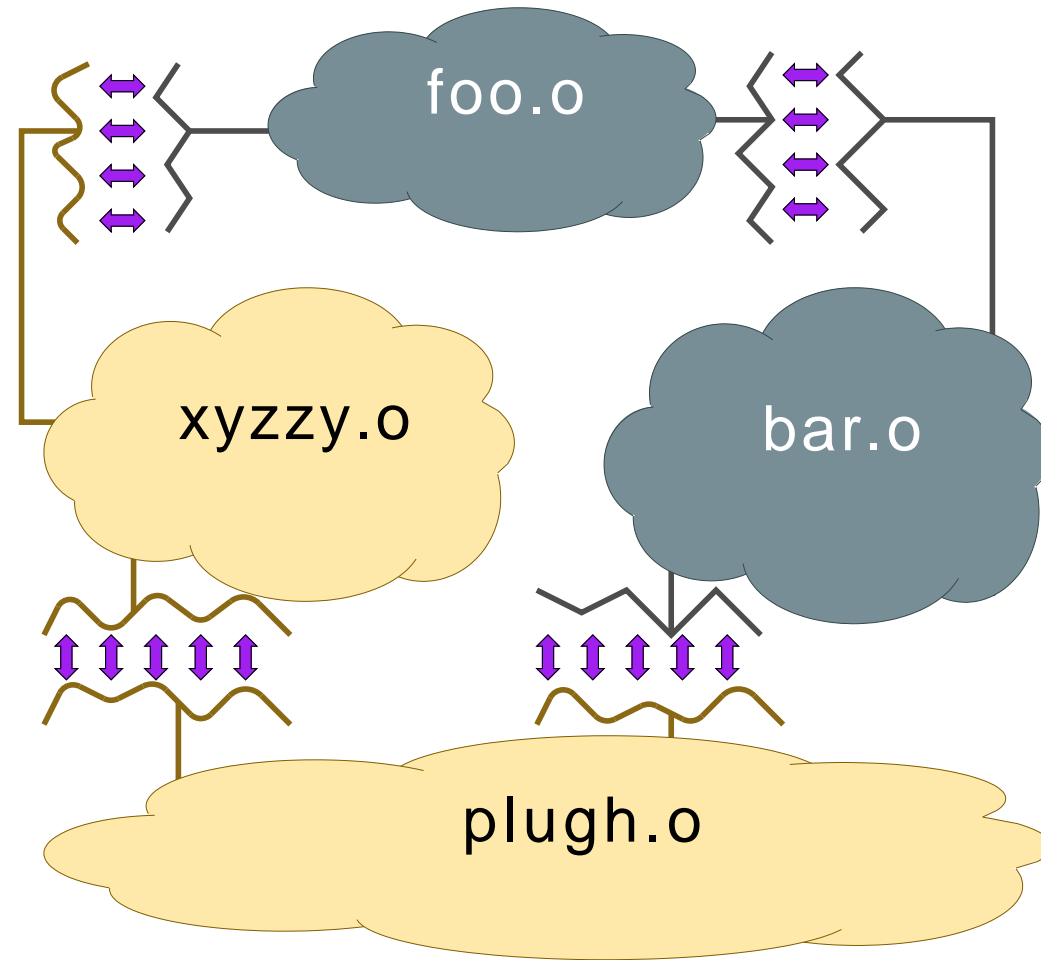
`Stephen.Kell@cl.cam.ac.uk`

Computer Laboratory



UNIVERSITY OF
CAMBRIDGE

Starting point: Cake's big picture



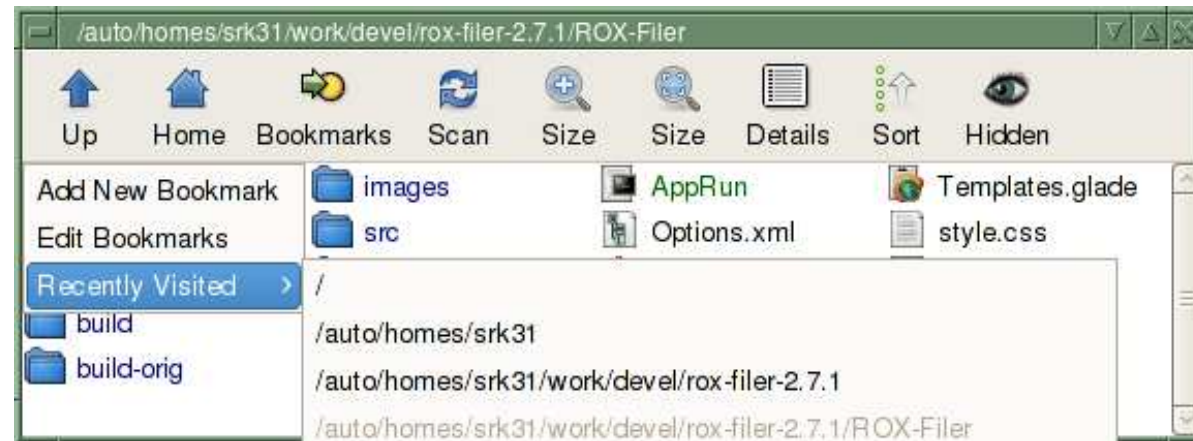
- new programming languages → **mismatch**
- decentralised development → **mismatch**
- unanticipated composition → **mismatch**

Cake is

- a language expressing *compositions* of software
- a productive tool for overcoming *mismatch*
- operating on *binaries*
- a *black-box* approach
- designed around practical experience
- ongoing work

Wanted: a tool for helping with tasks like...

Unanticipated composition: port feature P from app X to Y



Case study: Konqueror + ROX-Filer

Evolution: link client version 1 against library version 2



Case study: gtk-theme-switch

A taste of the Cake language

Two mismatched structures...

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    GtkWidget type;  
    guint window_has_focus:1;  
};
```

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    gchar *wm_role;  
    guint type:4; /* GtkWidget */  
    guint has_focus :1;  
};
```

A taste of the Cake language

Two mismatched structures...

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    GtkWidget type;  
    guint window_has_focus:1;  
};
```

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    gchar *wm_role;  
    guint type :4; /* GtkWidget */  
    guint has_focus :1;  
};
```

```
switch12 ↔ libgtk20 {  
    values GtkWidget ↔ GtkWidget {
```

A taste of the Cake language

Two mismatched structures...

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    GtkWidgetType type;  
    guint window_has_focus:1;  
};
```

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    gchar *wm_role;  
    guint type :4; /* GtkWidgetType */  
    guint has_focus :1;  
};
```

```
switch12 ↔ libgtk20 {  
    values GtkWidget ↔ GtkWidget {  
        void → .wm_role;
```

A taste of the Cake language

Two mismatched structures...

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    GtkWidget type;  
    guint window_has_focus:1;  
};
```

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    gchar *wm_role;  
    guint type:4; /* GtkWidget */  
    guint has_focus :1;  
};
```

```
switch12 ↔ libgtk20 {  
    values GtkWidget ↔ GtkWidget {  
        void → .wm_role;  
        .type as .GtkWidget <--> .type as .GtkWidget;
```


A taste of the Cake language

Two mismatched structures...

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    GtkWidgetType type;  
    guint window_has_focus:1;  
};
```

```
struct _GtkWindow {  
    GtkBin bin; gchar * title ; // ...  
    gchar *wm_role;  
    guint type:4; /* GtkWidgetType */  
    guint has_focus :1;  
};
```

```
switch12 ↔ libgtk20 {  
    values GtkWidget ↔ GtkWidget {  
        void → .wm_role;  
        .type as .GtkWidgetType <--> .type as .GtkWidgetType;  
        .window_has_focus ↔ .has_focus ;  
    }  
}
```

Status:

- case-studies: done, hand-written glue code
 - ◆ Gtk+ glue generation is partly automated
- compiler: current work
 - ◆ ...
- runtime library: most developed piece
 - ◆ used in the Gtk+ case study

Thanks for your attention. Any questions?