



TEND TO YOUR KNITTING

EVOLUTION OF A DINOSAUR

THE KNITTING:

- Selectra Industries makers of 'Active Wear'
- Knitted on high tech machines
- Management of production a problem
 - yarn to nearly complete garment different from cut 'n' sew
 - tried available software – hosiery was close
 - called in dinosaurs

A KNITTING MACHINE

Knitting is done inside the box.

The metal hoops support some of the spools of microfiber yarn



SOME OF THE KNITTING MACHINES AT SELECTRA

Each machine produces a single size garment. Each garment can be knitted with varying degrees of compression. Selectra refers to this as 'body engineering'.



THE DINOSAURS

- systems programmers IBM mainframe
- Rexx a known quantity
- firm's platform was Windows
- so... oorexx
- also oodialog
rexSQL

THE NEW SYSTEM

- dialogs for each major step in the production path
 - sales order entry
 - knitting
 - sewing
 - dyeing
 - &c

THE NEW SYSTEM

- each dialog tended to be large – 2-4K statements
- business, presentation and data logic in every program
 - data retrieved from database
 - presented to user
 - updated in database for each function
- it worked, but...

OBJECT ORIENTING A DINOSAUR

- pace of change picks up
 - user experience with early system points up need for changes
 - users stories changed significantly – the 90% problem
 - fixed capacities overrun
 - miscommunication between users and developers

A DINOSAUR EVOLVES

- factoring business logic out of dialog code
- creating objects to model processes
- sharing objects between dialogs
- class methods to create instances

typical UI function

```
::class dyeOrder subclass rcDialog
```

```
::initDialog
```

```
expose or.
```

```
ord.=self~getOrder(okey)
```

```
self~displayOrder(ord.)
```

```
::method getOrder
```

```
dx='select ... .. from dye_order_table'
```

```
sql_rc=SQLCommand('or',do)
```

```
... .. error checking, handling
```

```
return dx.
```

```
::method displayOrder
```

```
use arg dor.
```

```
... .. set fields, display to user
```

```
::method saveOrder
```

AFTER SOME EVOLUTION...

```
::class dyeOrder
::method orderfromSQL class
  use arg key
  doq='select ... .. from table ... .. where index=key'
  inst=.dyeOrder~new(... ..)
  return inst
::attribute ...
::attribute ...
::method init
  use arg ..., ..., ...=default
```

OBJECTS

- eliminate duplicate code
- enable and simplify communication between dialog elements
- enable (somewhat) simpler changes for new user ideas
- help eliminate hard constraints on numbers

QUESTIONS?

- Thanks