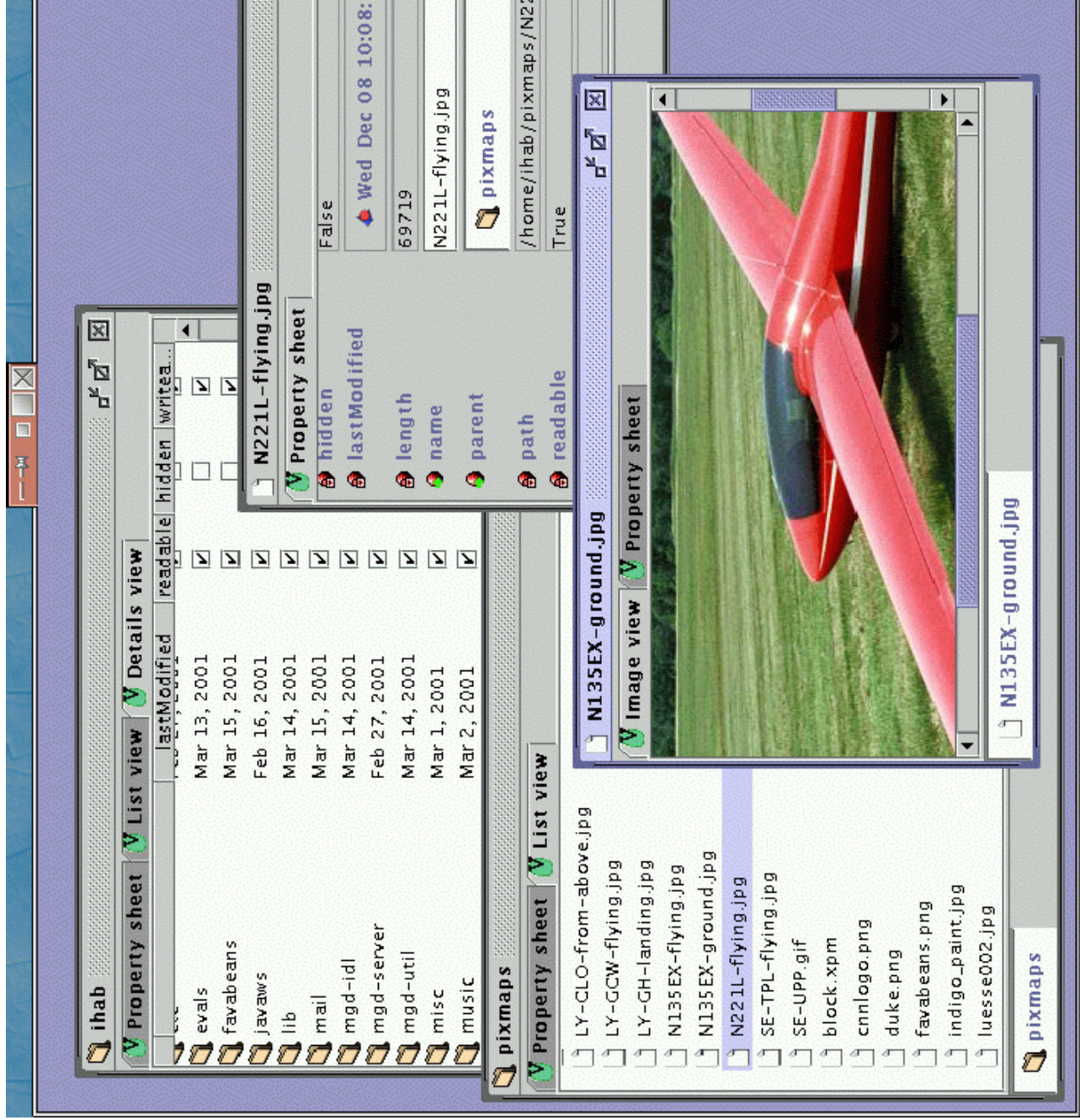
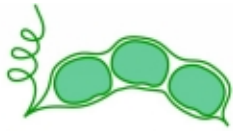


FavaBeans



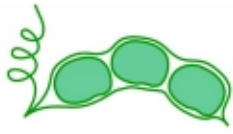


End-Users Can Be Object Geeks Too

The Story of FavaBeans, an Open Source Object
Oriented User Interface Framework

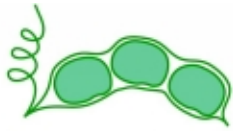
Ihab A.B. Awad <ihab@umn.edu>

Center for Computational Genomics and Bioinformatics
University of Minnesota



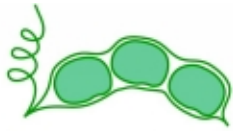
Messages

- There's no such thing as language independence
 - (Broadly speaking)
- Object Oriented User Interfaces (OOUIs) are
 - *Not* [quite] Windows/MacOS/KDE/Gnome/...
 - *Not* a UI written in Java/C++/Python/Smalltalk/...
 - Convenient for end-users
 - Profitable for you
 - An all-round Good Thing™
- You can use FavaBeans to build OOUIs in Java
 - Conversely, FB needs your help

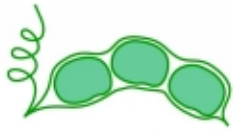


Outline

- OOUIs (and some terminology) defined
- History of OOUIs; related work
- How FavaBeans came about
- Quick summary of FavaBeans architecture
- Where I'd like FavaBeans to go

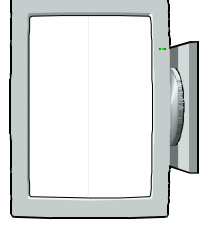
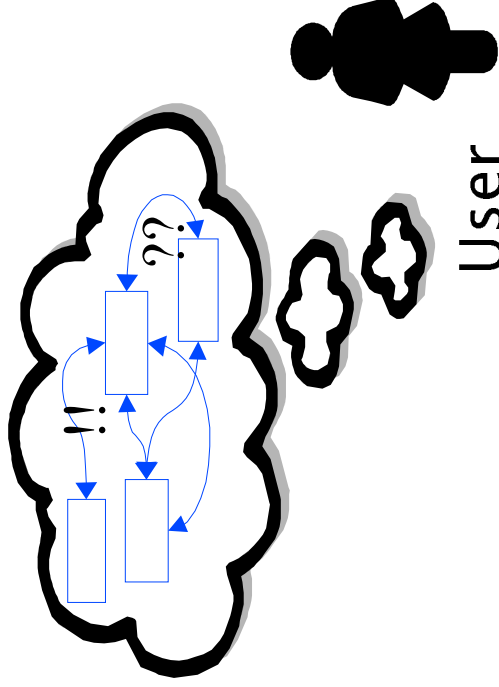


1. What's an OOUI?

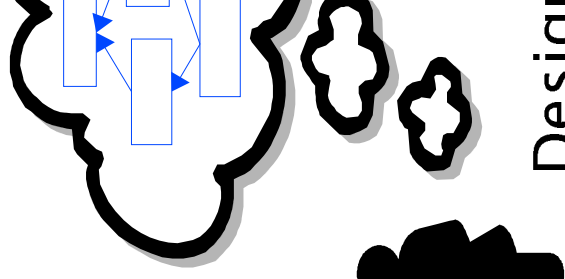


Models

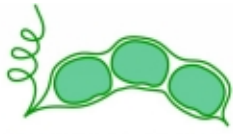
- Every user has a mental model
 - How they *thinks* things work
- Contrast this with the designer's intention
 - How they think and/or *want* the user to think
- Frequently not at all similar!!
 - The system seems mysterious and inconsistent
 - Users may entertain *mythologies*



User

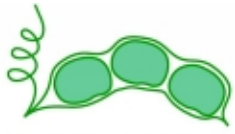


Designer



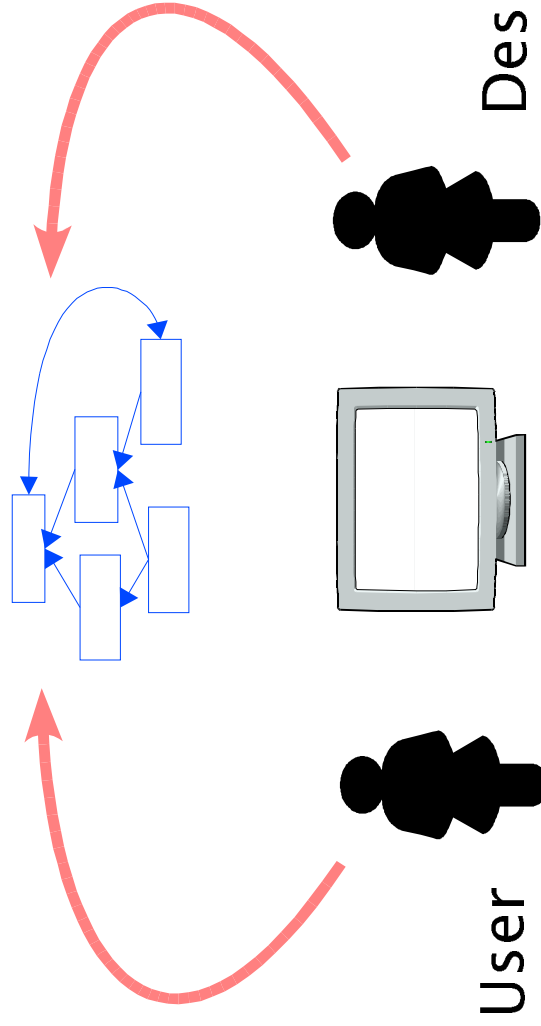
Reasons for Inconsistency

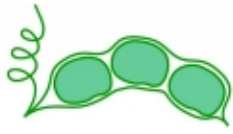
- Developers and users often work in different ways
 - "Let's design the screens right now; all the rest is implementation"
 - "This implementation is really cool; oh, about that"
- If we don't have a *language* to communicate
 - We can't/won't!
- But UI and implementation are based on a *language*
 - Can the spec be "neutral"; "language-independent"?



Sharing the Models

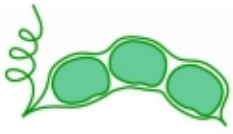
- OOUI -> designer's & user's models both
 - Can be shared!
- But, more precisely:
 - The models are instances of [the/an] OO meta-





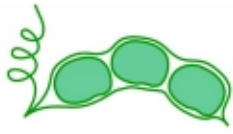
Cool Book & Definition

- Collins, Dave. *Designing Object-Oriented User Interfaces*. Benjamin/Cummings, 1995.
- A Definition of OOUI (pg. 89)
 1. Users perceive and act on [visual] objects.
 2. Users can classify objects based on how they behave.
 3. In the context of what the users are trying to accomplish, the interface objects fit together into a coherent representation (user's conceptual model). ... No island.
- None of this is absolute
 - Maybe an "OOUIIQ" metric?



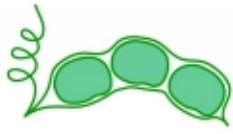
What does this all mean?

- What am I claiming?
 - Users get a class browser? Just give 'em a debugger
 - No!
- However...
 - The results of analysis drive the implementation
 - They should also drive the UI
- So...
 - An OOA maps well to an OOPL
 - So should an OOA to an OOUI
- What's it like for the user without an OOUI?
 - Just like a programmer without an OOPL
 - Possible, but messy



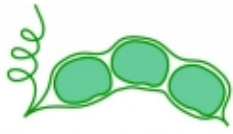
Visual Representations

- Given a conceptual metamodel...
 - Can we build a visual representation of instances
- Answer: Yes
 - UML, OMT, Booch, ...
 - But surely not for end-users!
- Therefore an OOUI:
 - Friendly visualization based on OO metamodel
- Many ways to do this...
 - The most obvious being the all-too-familiar WIN
 - Others include zooming, 3D, ...



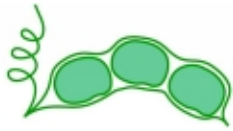
Things to Represent

- Elements of OO metamodel to represent:
 1. Fundamentals
 - Objects, identity
 2. Features of objects
 - Operations, messages
 - Properties
 3. Connections between objects
 - Relationships (~ properties)
 - Inheritance, polymorphism (Aristotelean classification)
 4. Common patterns
 - Collections, composition
- Notably absent:
 - Applications, programs, data [files], ...

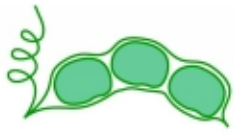


What's an OOU? -- summary

- No such thing as language "neutrality"
 - In this context, OO is a "language"
- OOU -> user's conceptual model is OO
 - Users perceive, act on and classify objects which behavior and form a coherent model

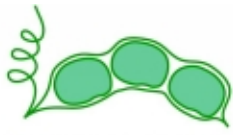


2. OOUI History & Related Work



OS/2 Warp Screenshot





OS/2 Warp Screenshot

Desktop "tree view"

Location: <http://euxxx.uland.com/html32/index.htm>

visitors from 13/06/1996

0 0 0 0 0 0 0 3

Person1
My name is Eugen Kuleshov. I'm 23 years old.

Interst

From 1995 year I use IBM OS/2. I create program - hardware solution and WAN too.

Hobby

VC

win FC

[Messages] [M:0/P:0] [irc2.magic.ca]

BinkD

IBM Internet Connection Server 4.1.3

Netscape - [Kris Kwilas' Unfinished Home Page]

Netscape News - [Re: Netscape plugin for Wav ?]

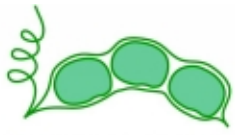
T-Mail

irc/2. script part v8.81 [%4.3#1]

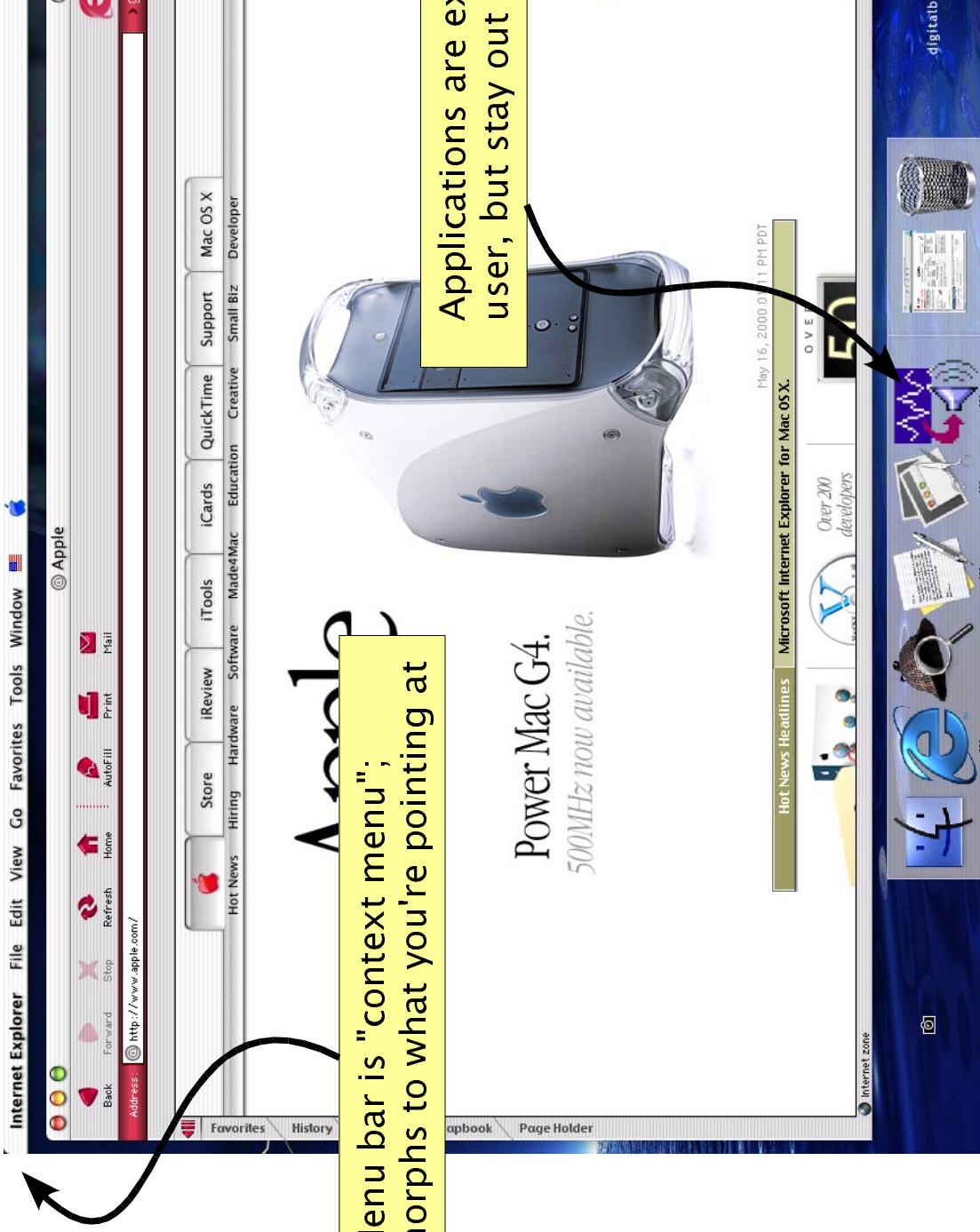
[18:02] Euxx (+1) on #os2fun (+nt) [UP: 8d0h8h]

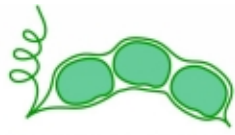
<#os2fun>

18:02 10/4/97 Shredder



Mac OS X (NeXTStep) Screenshot



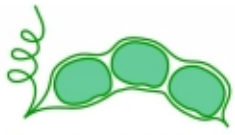


Together (Java IDE with UML)

The screenshot shows the Together IDE interface with several annotations:

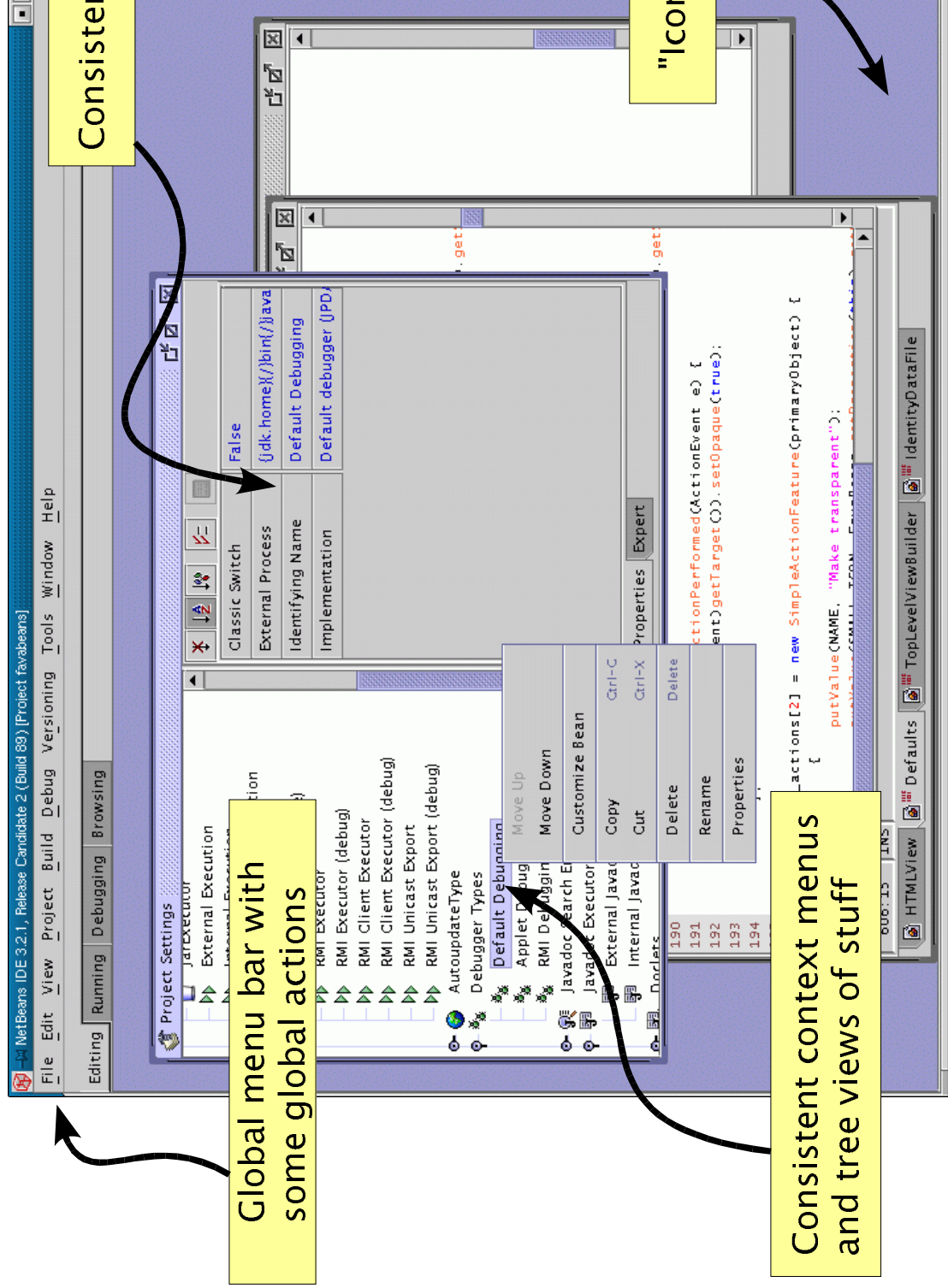
- Task-centered "folders of structure":** A yellow box with an arrow pointing to the "Facet" interface in the "interfaces-FacetsViewPrototypes" view.
- UML:** A yellow box with an arrow pointing to the UML diagram in the "UML" view.
- Consistent:** A yellow box with an arrow pointing to the "Layout" menu option in the "View" menu.
- Still have a menu bar, with some "global" actions:** A yellow box with an arrow pointing to the "File" menu bar.

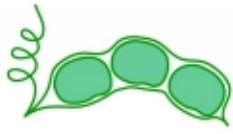
The IDE interface includes a menu bar (File, Edit, Object, Search, View, Select, Options, Tools, Help), a toolbar, and several views: "Project Explorer" (showing a tree structure), "UML" (showing a class diagram), "FacetsViewPrototypes" (showing an interface "Facet"), "Code Editor" (showing Java code), "Messages" (showing license information), and "Progress" (showing progress bars).



NetBeans (Java IDE)

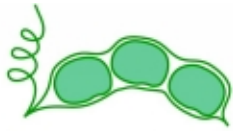
- Was *the* most influential on FavaBeans



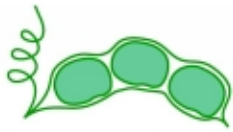


OOUI History & Related Work -- *st*

- Lots of existing implementations
 - The fundamental idea is not new
- We can do *much* better!
 - Drag and drop, reuse of views & settings
 - Across traditional "application" boundaries
 - Do away with the file/application split
 - Will require some OO persistence underneath
 - Global undo/redo, global "save", ...
 - Drag-and-drop "programming"

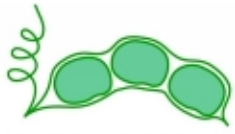


3. The Origins of FavaBeans



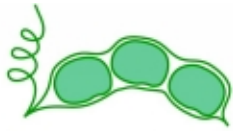
Complex Information Model

- Biologists need to understand genomes
 - Complex interactions among proteins
- We can visualize some concrete models
 - But not always; sometimes changes too fast!
- We need a visualization of the *metamodel*
 - Hence OOU
 - Biologists are scientists; exploratory
 - Linnean classification (Order, Genus, ...) ~ class

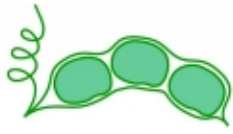


The Origins of FavaBeans -- *summary*

- Scientific data
 - Sophisticated end-users
 - Complex information model
- Not the *only* place where useful...
 - But a good place to start

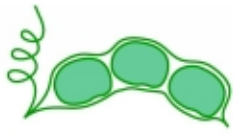


4. FavaBeans Architecture



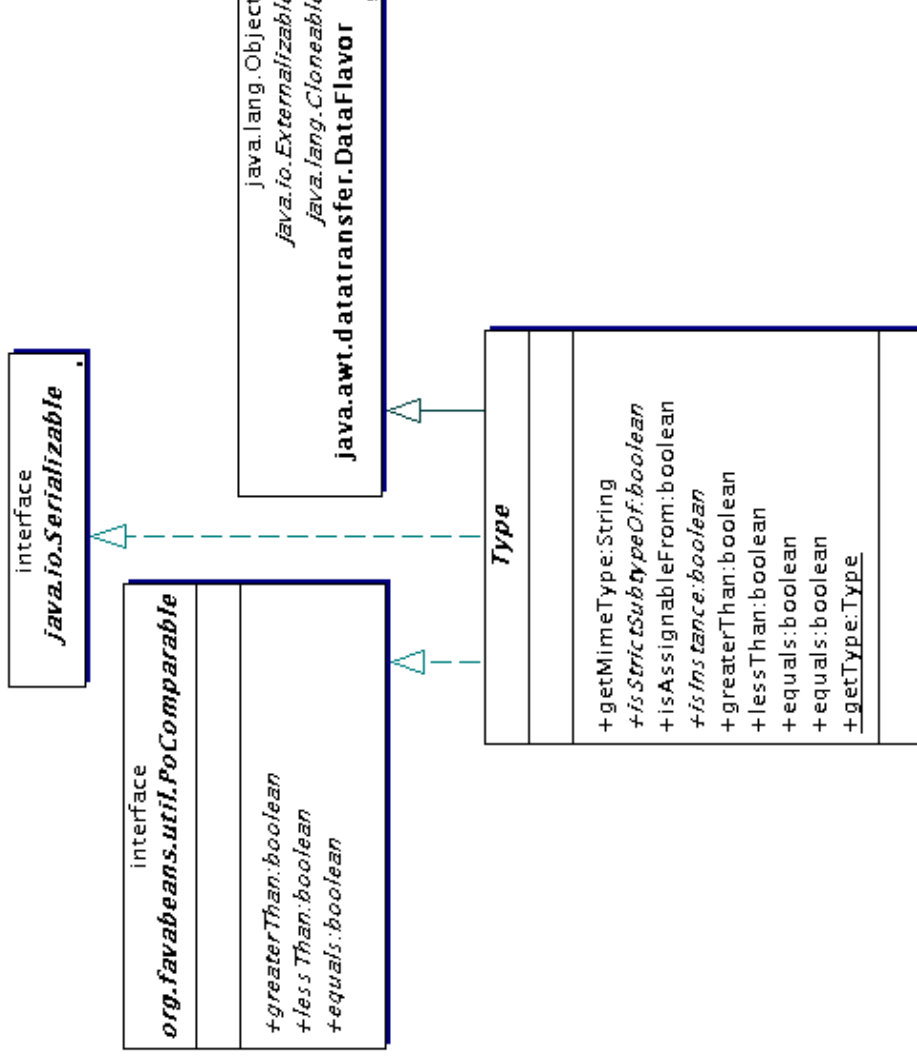
GoF Adapters

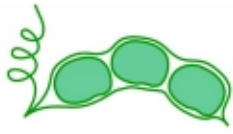
- FB = adapters, adapters, and more adapters
 - As far as the eye can see
- We call them "facets"
 - Name directly stolen from ObjectSpace Voyage
- Facets adapt your domain objects
 - Present standard interfaces to general UI widgets
 - Properties, Actions, Folder, ...
 - Plug in your custom UI widgets where you must
 - ... and *only* where you must
- Objects need not be facet-aware
 - But they *can* be -> tightly integrated into FB



Types

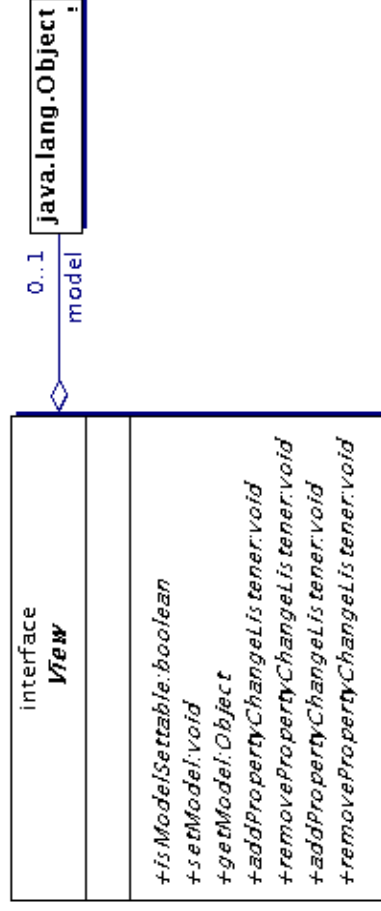
- Recognize an object: `Type.isInstance()`
 - Can use any criteria you want

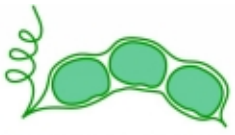




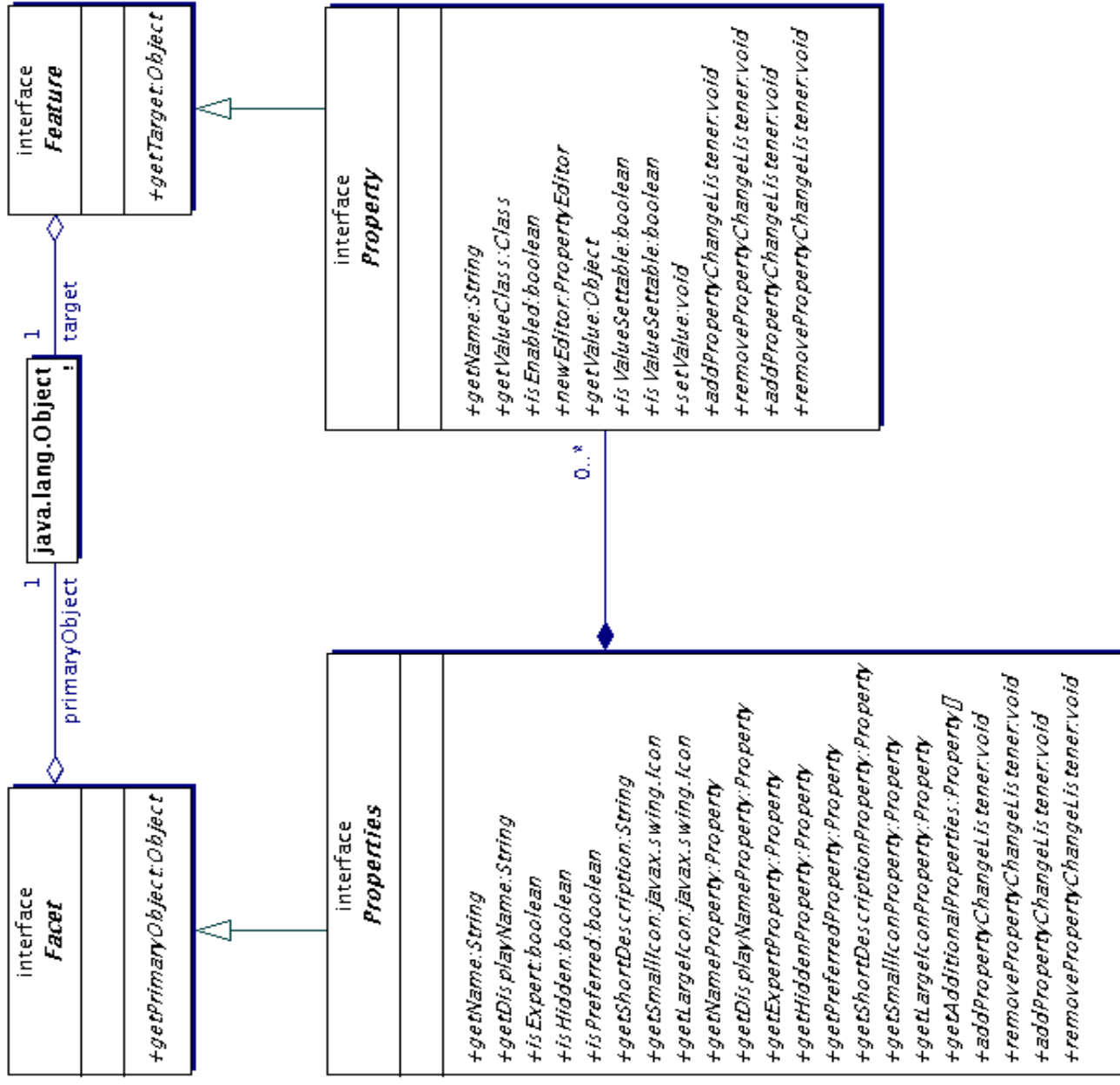
Views

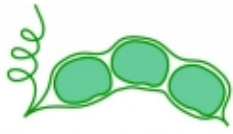
- Fundamental GUI widget interface
- *Can* extend `java.awt.Component`





Facets & Features

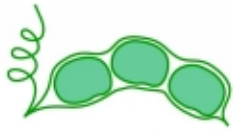




Interface Faceted

- Allows you to create Facets of yourself
- Tight integration into the framework

```
interface
  Faceted
+getFace(type: Type): Facet
```

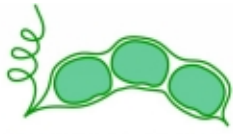


The FacetRegistry

- Create Facets of non-Facet-aware objects
 - By adding FacetFactory objects
 - Implements "inheritance" of Types
 - `fr.addFactory(srcType, destType, aFacetFactory)`
 - `Facet f = fr.getFacet(anObject, destType);`
- Loose integration (or coupling) with FB

interface FacetRegistry
<code>+clear():void</code> <code>+addFactory(primaryObjectType:Type,facetType:Type,factory:FacetFactory):boolean</code> <code>+removeFactory(primaryObjectType:Type,facetType:Type):boolean</code> <code>+getFacetTypes(primaryObject:Object):Set</code> <code>+getFacet(primaryObject:Object,facetType:Type):Facet</code> <code>+clearFacets(primaryObject:Object):boolean</code> <code>+addTypeInfo(isTener(:TypeInfoIsTener):void</code> <code>+removeTypeInfo(isTener(:TypeInfoIsTener):void</code>

interface FacetFactory
<code>+newFacet(primaryObject:Object,</code>

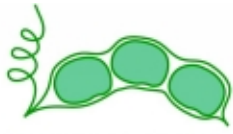


The TypeMetadataRegistry

- Add arbitrary information about a Type

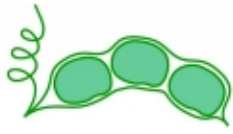
```
- tmr.put(aType,  
    "org.favabeans.property.name",  
    "A name for this type of thing");  
  
- tmr.put(aType,  
    "org.favabeans.view.propertySheet",  
    aPropertySheetImplementation);  
  
- View aPropertySheet = (View)  
  tmr.getForObject(anObject,  
    "org.favabeans.views.propere
```

- Information retrieved later
- By the default Facet implementations (can bypass)
- Can be dynamic
- put() a TypeAttribute; calls evaluate() on it
- Information is inherited along Type hierarchy



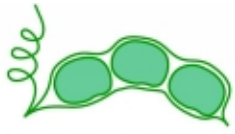
Other Facets

- Change
 - Publishes coarse-grained change notifications
- Prototype
 - Provides method to clone its primary object
- DnDTargetContainer
 - What happens when drag-drop onto self when in
- DataTransfer
 - Data flavors/Types available for cut/paste & Dr



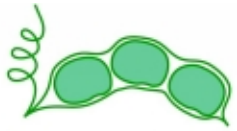
More Facets

- Actions
 - User-invokeable actions for toolbar or context menu
- ViewPrototypes
 - Views of the object that can be cloned
- Folder
 - Represents a collection as an ordered list
 - Contains a set of Columns
 - Can use these to sort and [soon] search the collection

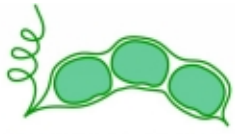


FavaBeans Architecture -- *summary*

- Facets
 - Adapters for generic UI widgets
- FacetRegistry
 - Facets for non-Facet-aware objects
- Metadata registry
 - Provides backing information with "inheritance"
- Types
 - Categorize objects for Facets & metadata

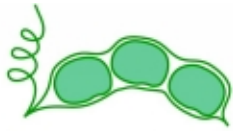


5. Future Directions



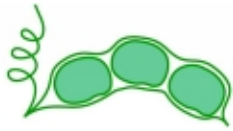
Improve Base

- Better direct manipulation
 - In-place editing of object names
- Fully robust views for collections
 - List, Details, Icon
- A "property editor" mini-framework
 - Based on Type of property
 - To be used for tables (Details View) & property



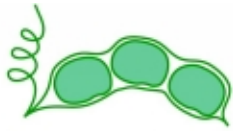
Add Nontrivial Content

- Email, Web, Newsgroups
- Gnutella client
- Aimed at evangelizing the framework



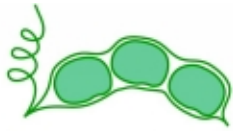
Push for Commercial Adoption

- Initial market: tech-savvy users, complex applications
 - Server management console
 - Network monitoring & management workstation
 - Scientific data analysis and visualization



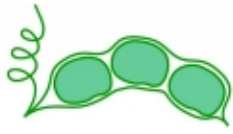
Future Improvements

- Support for "modules"
 - Loadable via Java Web Start
- Scripting support
 - Jython, Rhino [JavaScript], Jacl [Tcl], ...
- Persistence
 - Framework using an open source Java OODBMS
- Support for handhelds
 - J2SE 1.3 on 206MHz StrongArm, 32MB RAM/ROM
 - Compaq iPAQ, Samsung YOPY, ...
 - <http://www.savaje.com/>



Future Directions -- summary

- Deepen, but mostly broaden
 - Push for adoption
 - Seek funding
- Questions
 - How would *you* use FavaBeans in your work?
 - What problems would it solve?
 - Who would be the target developers? Users?
 - What business models are appropriate?



Acknowledgements

- Center for Computational Genomics & Bioinformatics
 - Ernest F. Retzel, PhD, Director
 - Jim E. Johnson
 - <http://www.cbc.umn.edu/>
- University of Minnesota Academic Health Center
 - <http://www.ahc.umn.edu/>
- United States Department of Agriculture, Agricultural Research Service
 - <http://www.usda.gov/>
 - <http://www.ars.usda.gov/>