Architecting for User Extensibility

Russell Miller
CTO, SunView Software, Inc
@RussMiller123
Russ-Miller.com

About Me

- MS in Computer Science: Northern Illinois University
- Worked for many large software companies
  (e.g. IBM, Intuit, Quark, Compuware)
- 20+ years architecting software and leading teams
- Co-founded SunView Software 10 years ago
- IT Service Management / Business Process Automation
- Geeking out on Meta for about 15 years
  (following: Johnson, Yodel, Wirfs-Brock, an others)

What sort of extensibility?

Core extensibility—not just surface level

What's new?

- Customer expectations!!!
- Extensibility itself is not new, what is new is the breath and depth of the expected adaptability.
- Virtualization—cheaper to deal with overhead of meta
- Cloud—scale of economy, large non-captive user base
- Dynamic languages and NoSQL
- Acceptability of separating "what" from "how"

What's not new…

- Too many apps being crafted from hand with hard coded logic and model
- Why? Adaptive Object Models noted over a decade ago?
- It is much easier with current technology and scale of economy is there.
- We need industry wide, open source solution on the level of Open Stack.

Architecture

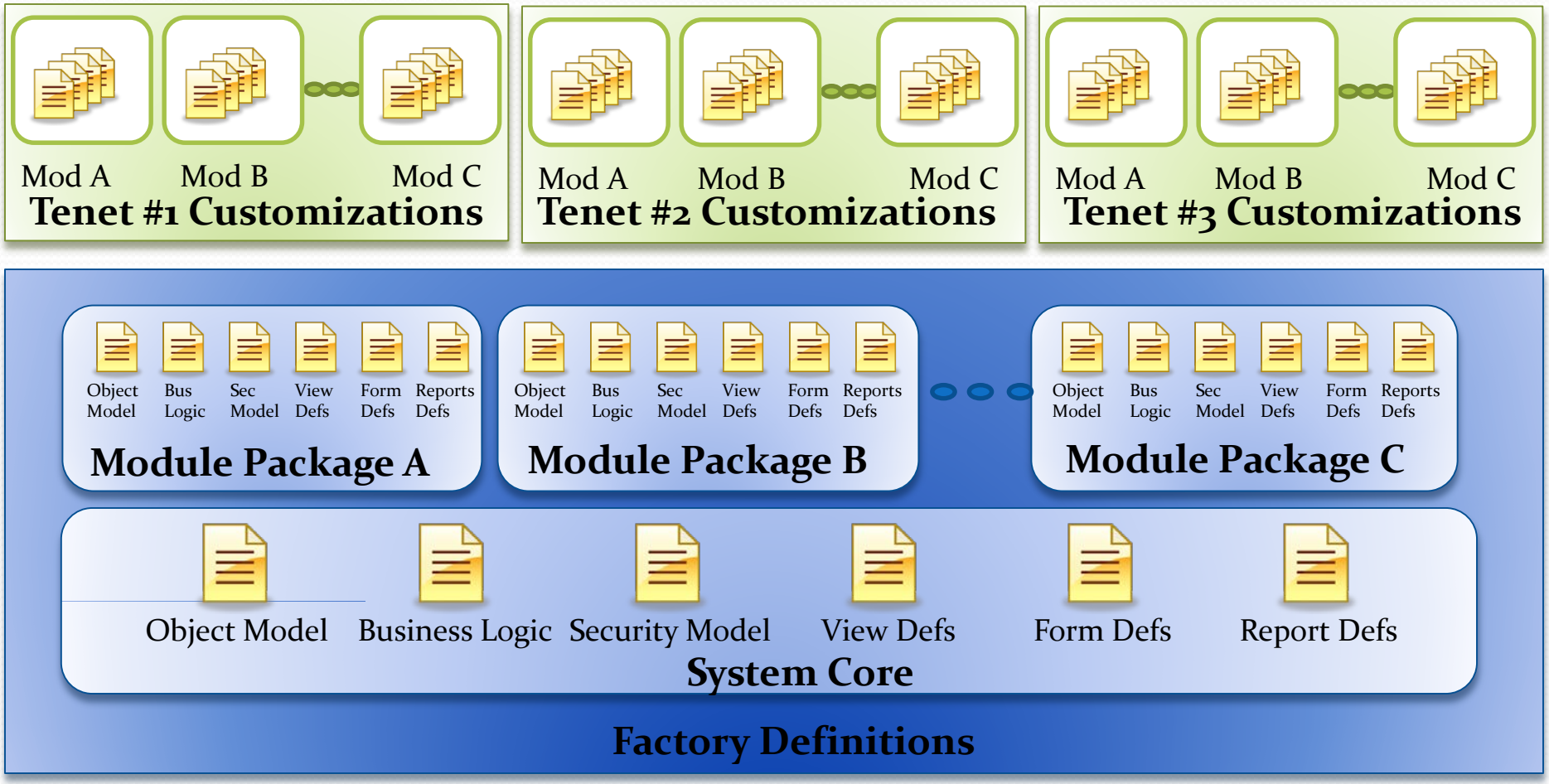
- Don't bake in the "what" and the "how".
- Separate the "what" from the "how".
- Make it easy for the customer to evolve both.
- Leverage dynamic nature of latest runtimes

Areas of Extensibility

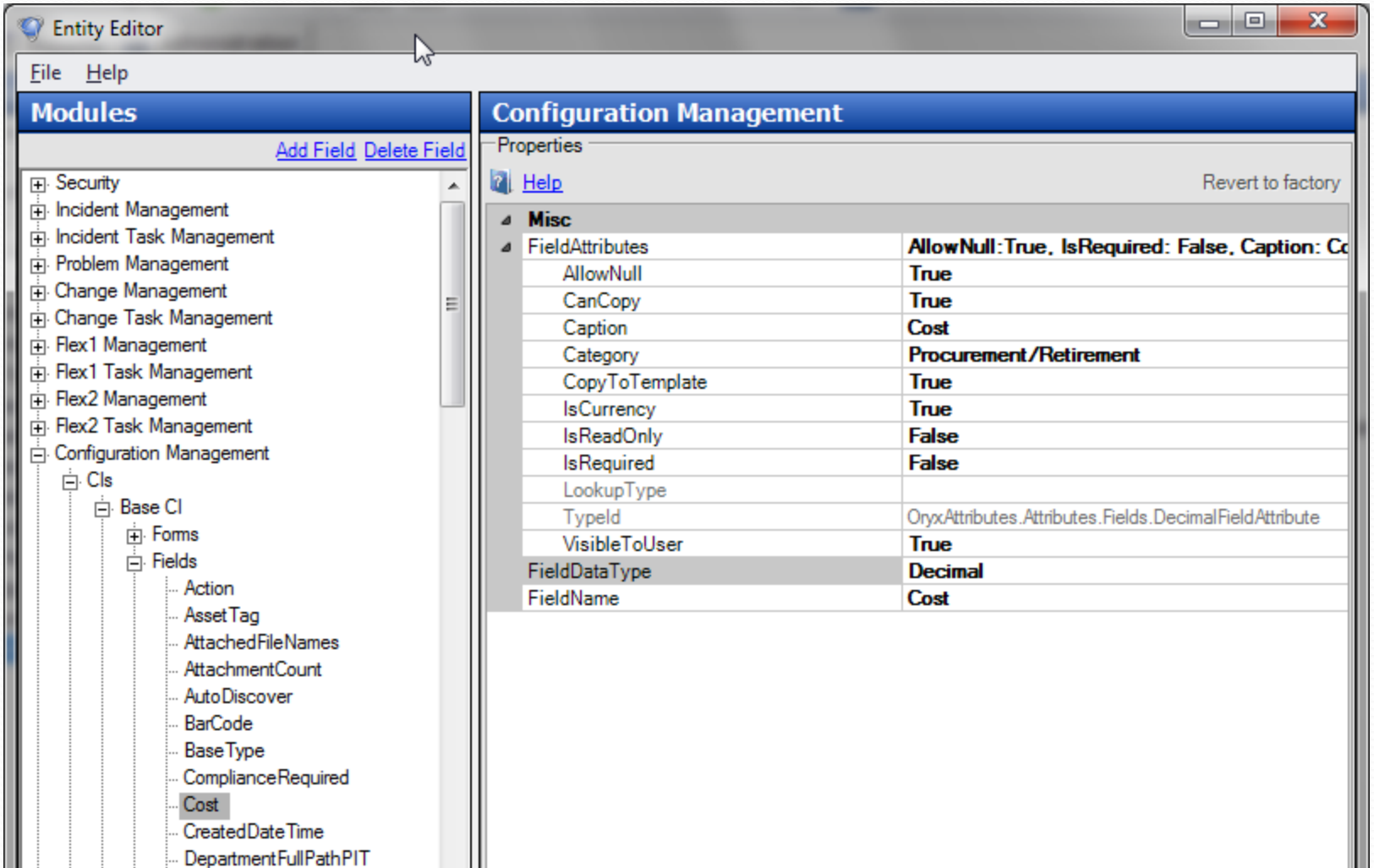
- Object Model
- Business Logic
- Security Model
- User Actions
- UI Customization
- Reporting
- Integration Interfaces

All of the above, deliverable in a **Packages**

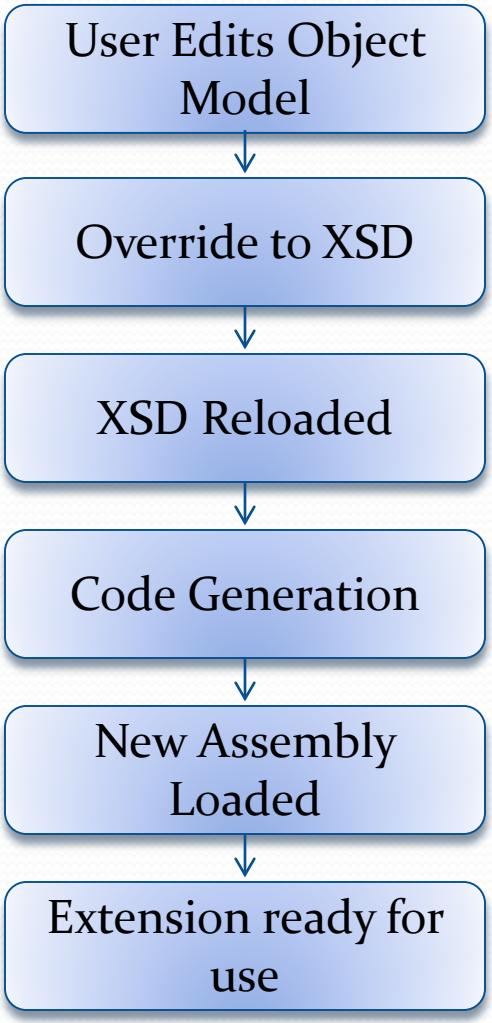
View of the Externalized What/How



Example – Object Model Edit



Object Model Extension

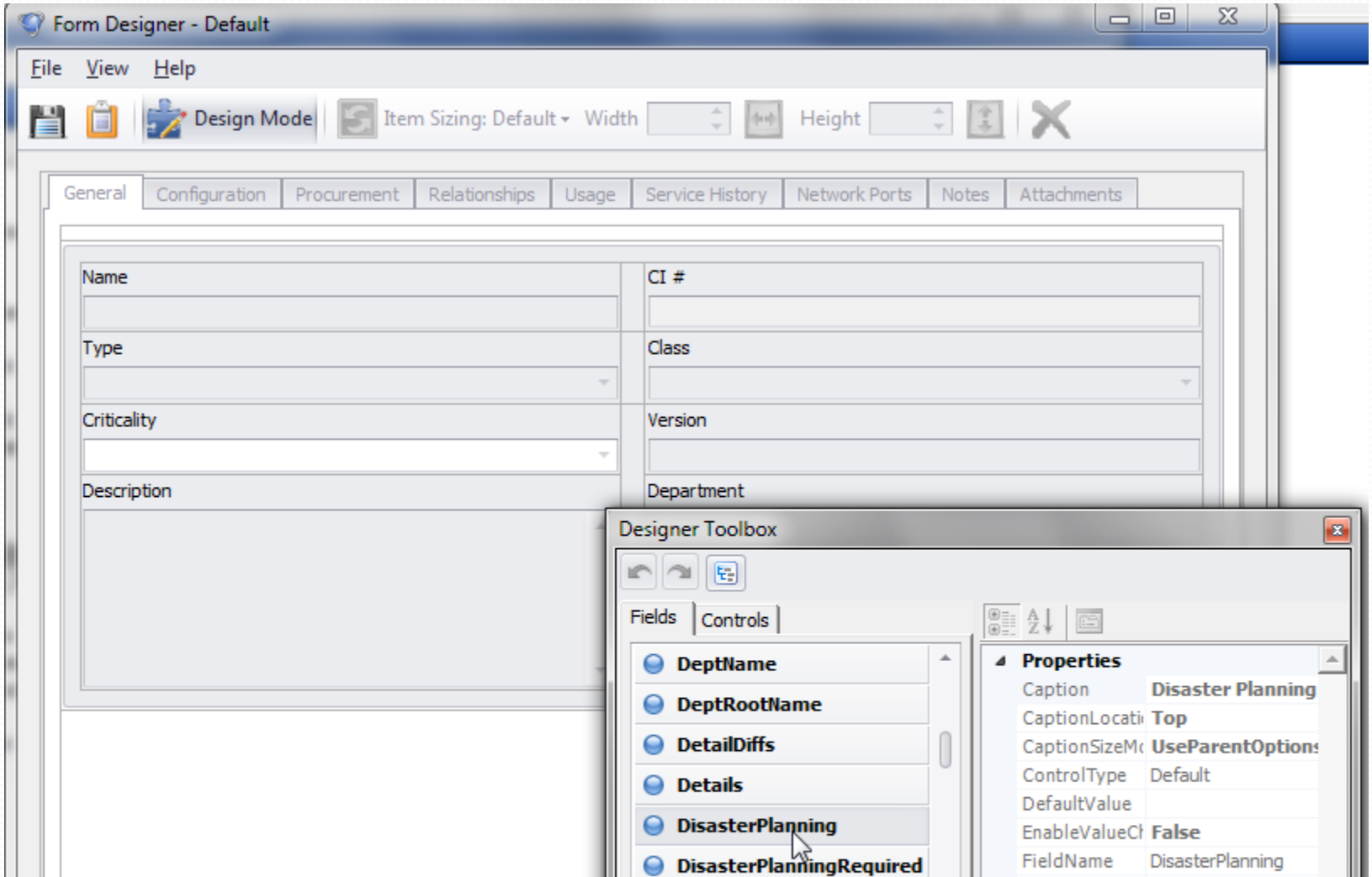


Override:

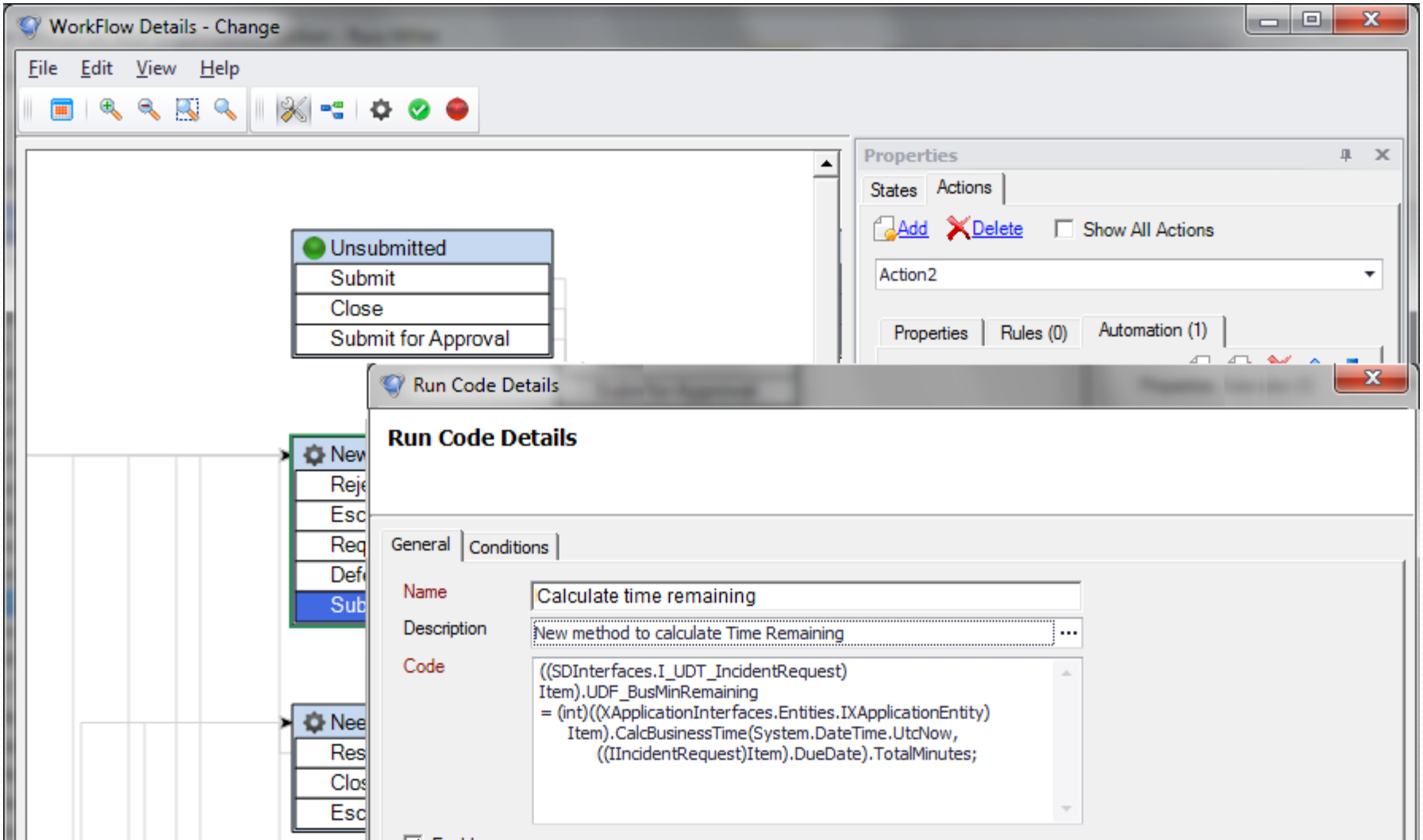
```
<xs:element name="Costs" VisibleToUsers="True"
                         VisibleForSearch="True"
                         IsRequired="False"
                        default="" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:decimal">
        <xs:maxLength value="5.7" />
      </xs:restriction>
    </xs:simpleType>
</xs:element>
```

Override approach makes system upgrades possible without losing extensions.

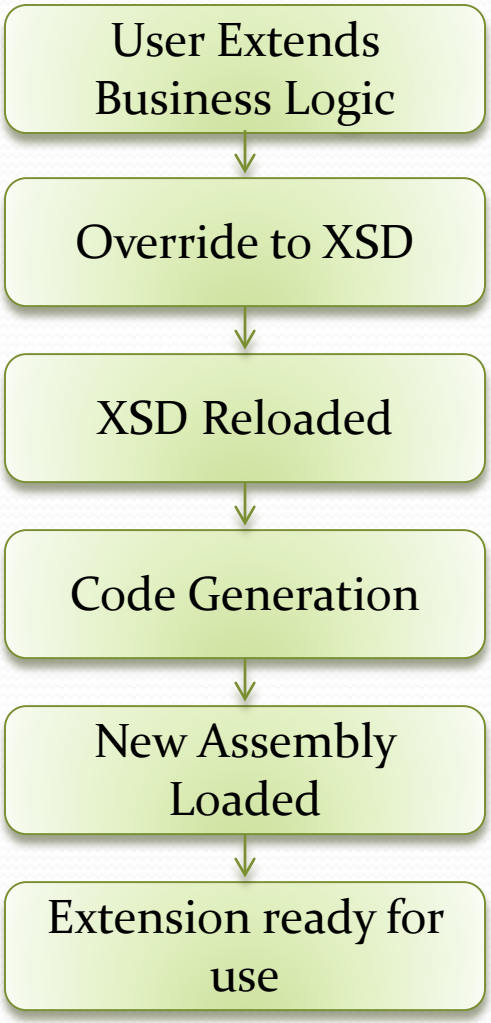
D&D From Object Model to Forms



Imparting the “How”



Object Model Extension



Override:

```
<action name="CalculateTimeRemaining" type="Run Code" id="2" Provider="InLineCode"
      description="New Method for calculating time remaining" enabled="true">
   <arguments>
      <argument name="Code“
        <codeSegment>
            <![CDATA[((SDInterfaces.Entities.I_UDT_IncidentRequest_Extended)Item)
                .UDF_BusMinRemaining
                  = (int)((XApplicationInterfaces.Entities.IXApplicationEntity)Item)
                     .CalcBusinessTime(System.DateTime.UtcNow,
                  ((IIncidentRequest)Item).DueDate).TotalMinutes;]]>
        </codeSegment>
      </argument>
   </arguments>
</action>
```

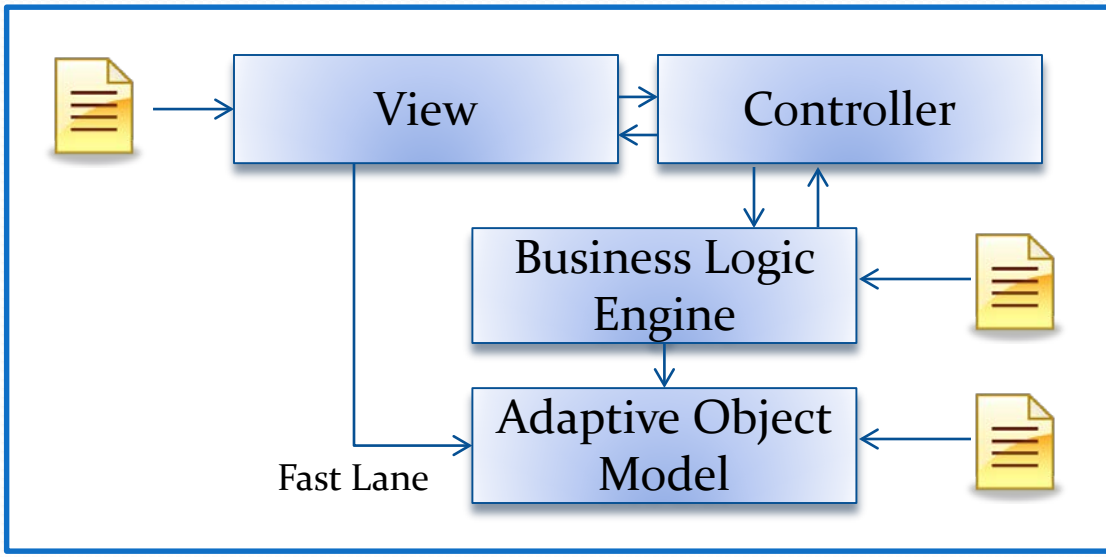
Again: Override approach makes system upgrades possible without losing extensions.

Generic Views and Controllers

Model View Controller

But the View and Controller are “dumb”, very generic.

No hard knowledge of domain above the Business Logic Engine



Separating What from How

- As recognized by Reenskaug and Coplien, very critical to separate “What this system is” from “what it does or how it does it”
- They need to live separate but parallel lives
- Externalized meta-model clearly separates out the “what it is”
- Externalized meta-logic separates out “what it does”
- Looking toward more explicit application of DCI going forward

Future Directions

- Currently uses SQL database, considering NoSQL (polyglot?)
- Currently more pluggable code harder for customers to craft than it should be—DCI exposed for customer?
- Open source community involvement

Conclusion

- Enterprise customers demanding increasing adaptability/extensibility/flexibility
- They expect to adapt at runtime (without an expensive consultant)
- They expect this with no down time (even when upgrading)
- Don't bake in the "what" and "how"—at any level
- Separate the "what" and "how"
- **Where's the industry wide answer?** (open source?)

Questions?

@RussMiller123 Russ-Miller.com