



Lisp and .NET

Using Lisp to Generate .NET
Business Applications



Focus on Productivity

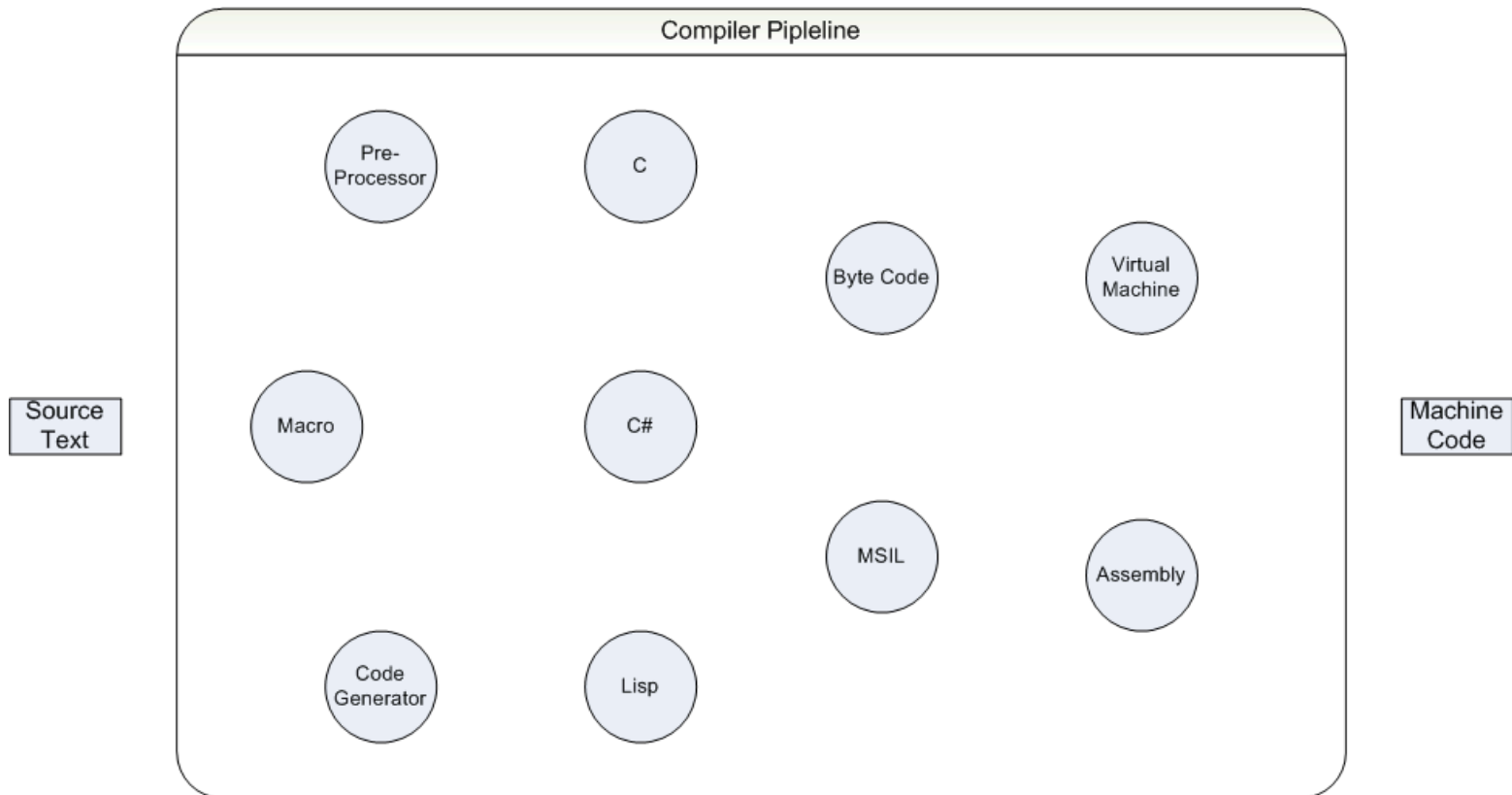
- Comac is a fulfillment and digital print services company for marketing materials
- My job is to build business applications
- ROI is the key
- Lisp contributed greatly



Our Solution

- Lisp Macros for C#
- Generative Programming
- Domain Specific Language
- Metadata Driven
- Patterns are Key

Compiler Pipeline Model





Why This Solution?

- Lisp Advantages and Disadvantages
- .NET Advantages and Disadvantages
- Lisp.NET?



Why Build DSL with Lisp?

- Functional, Higher Order
- Map, Filter and Accumulate
- Eval
- Development Environment

Why C#, .NET?

- Hundreds of great components (ROI)
- Huge library (for business apps)
- Comprehensive infrastructure
 - Authentication, Authorization
 - Message Queue (Asynchronous, Reliable)
 - Remote deployment



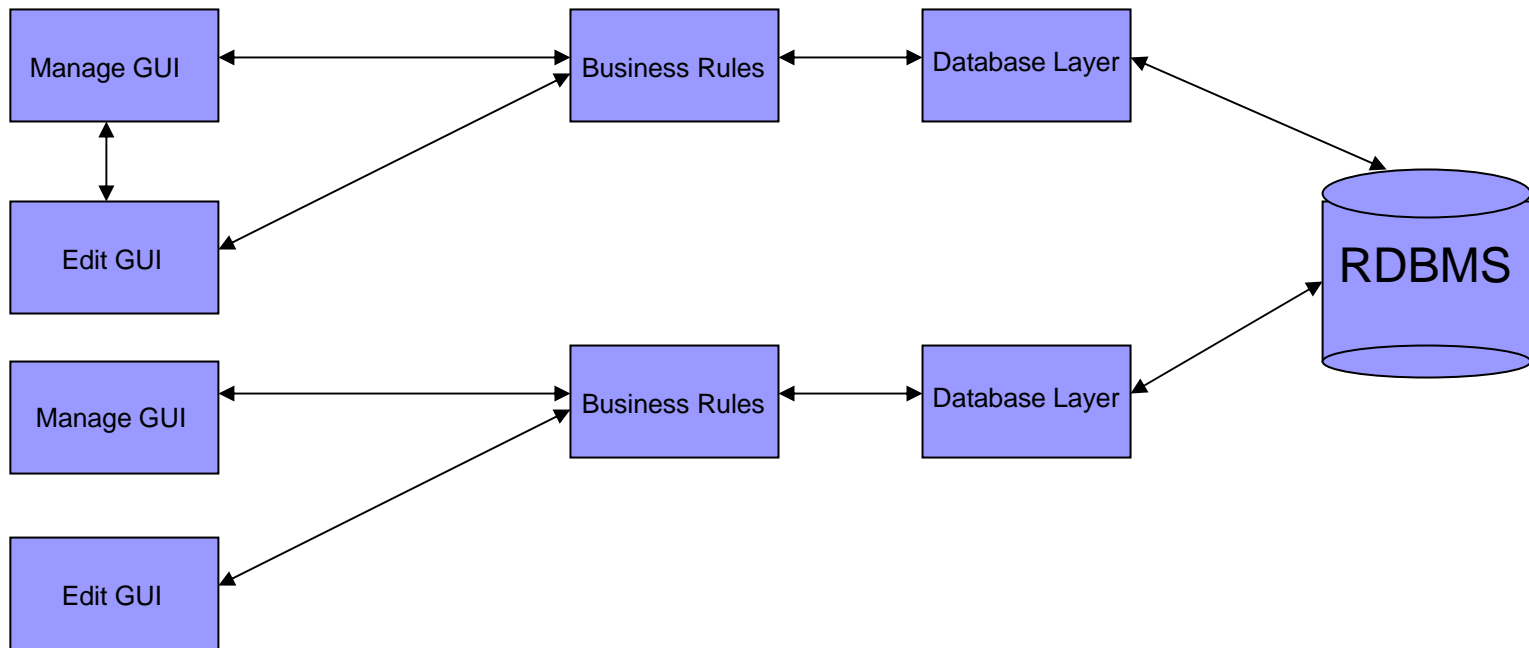
Lisp and .NET Today

- RDNZL – Edi Weitz
- FOIL – Rich Hickey, Eric Thorsen
- LSharp – Rob Blackwell
- Common Larceny – Will Clinger
- ...

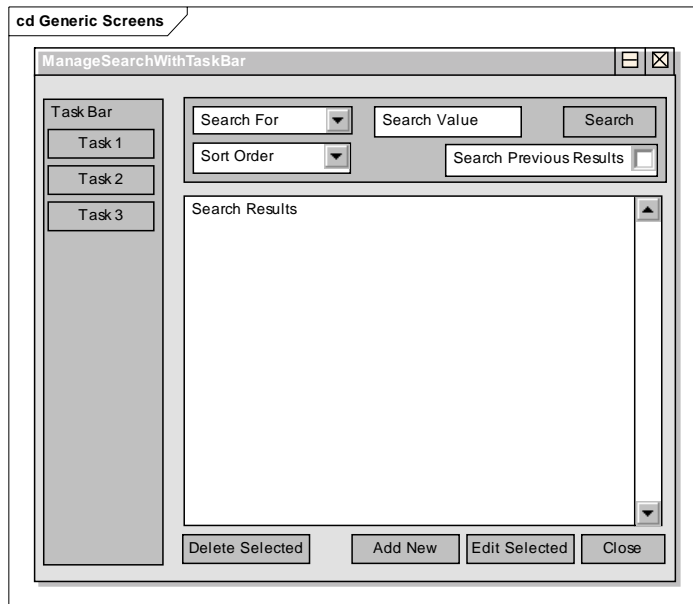
Demo

- Building the Booklet Manager
- 400 lines of DSL -> 25,000 lines of C#
- It runs, so ready for user feedback

Works Best with Repeating Patterns



GUI Example



```
(multiple-value-call (db-put!-c *db*)
 (add-relation "Billable_Function"
  :attrs (mapcar-apply #'create-attr '(
   ("Billable_Function_Id" int :is-pk t)
   ("Description" char :dom-len 30 :lookup-display t)
   ("Category" varchar :dom-len 25 :nullable t)
   ("Sub_Category" varchar :dom-len 50 :nullable t)
   ("Unit_Measure" int)
   ("Is_Taxable" smallint :is-bool t)
   ("Is_Software" smallint :is-bool t)
   ("Is_Broker" smallint :is-bool t)
   ("Is_Commissionable" smallint :is-bool t)
   ("Listing_Order" int)
   ("Oracle_Exist_Id" int)
   ("Oracle_New_Id" int)
   ("Definition" longvar :dom-len 8000)))
 :indexes (mapcar-apply #'create-ix `(
  ("XIE1Billable_Function" :ix-attrs
   ,(mapcar-apply #'create-ix-attr '(("Category" :pos 1))))
  ("XIE2Billable_Function" :ix-attrs
   ,(mapcar-apply #'create-ix-attr '(("Description" :pos 1))))
  ("XIE3Billable_Function" :ix-attrs
   ,(mapcar-apply #'create-ix-attr '(("Listing_Order" :pos 1 ))))))))
```



Domain Specific Language

- Business Application Builder Language
 - Metadata + Parameters
 - Templates
 - Code Snippets

Results

- 250 Database Tables, 600 Screens
 - 24 Person Months
- 5 Years Experience
 - Active Server Pages then VB6 then .NET
 - Evolvable
 - Maintainable
 - Refactorable
- Works on Business Applications – Patterns
- Others more problematic

Conclusion

- Lisp Macro Concept -> C#/.NET
- Advantages of Lisp
- Advantages of .NET
- Domain Specific Language
- Great Productivity!



Future Ideas

- Business rules metadata
- IDE with visualizations
- Database designer, round tripping