Dynamic Objects for Service Oriented Architecture (SOA) Programming

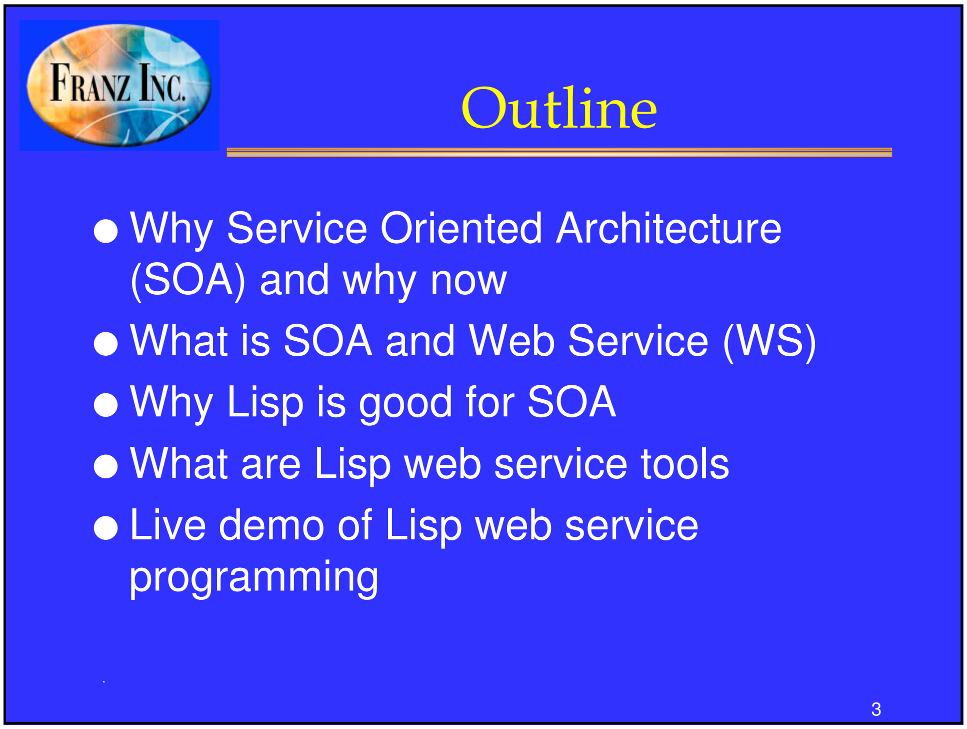
By

Sheng-Chuan Wu & Charley Cox

Franz Inc.

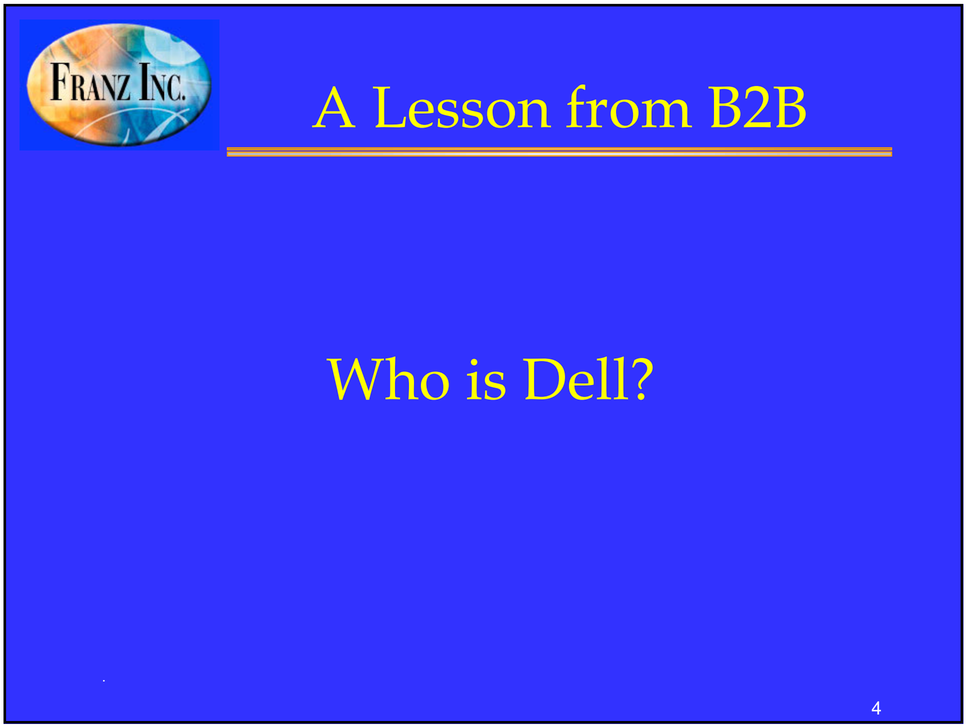
Who is Franz?

- Founded in 1984
- Founders out of UC Berkeley
- More than 16 years experience developing and marketing Common Lisp and CLOS object systems
- Creator of Allegro Common Lisp development environment and tools



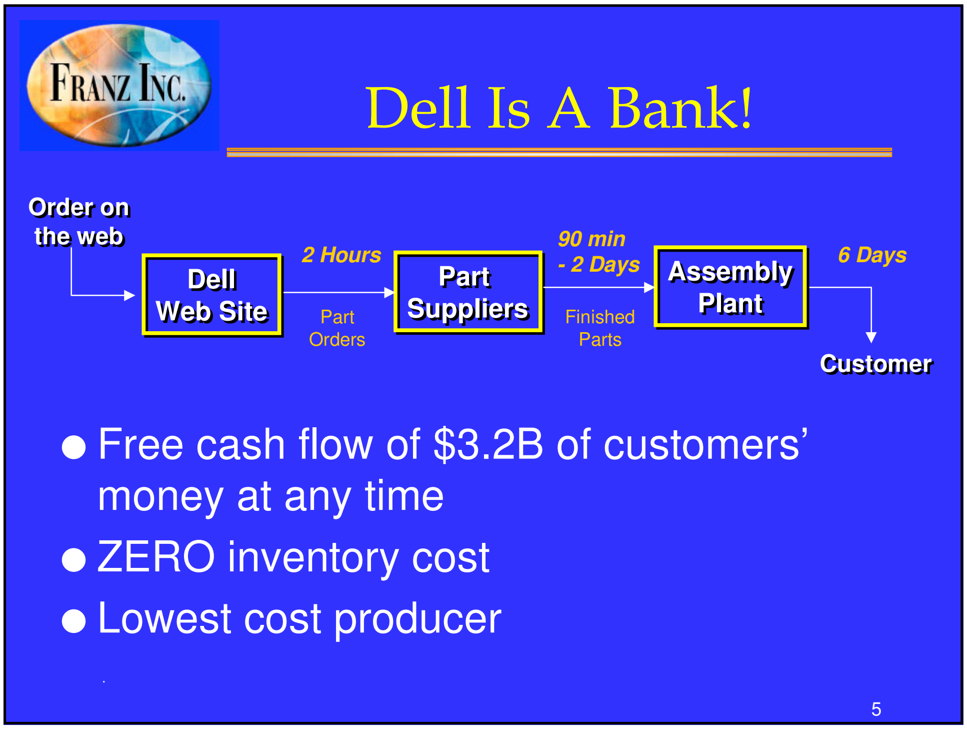
Outline

- Why Service Oriented Architecture (SOA) and why now
- What is SOA and Web Service (WS)
- Why Lisp is good for SOA
- What are Lisp web service tools
- Live demo of Lisp web service programming



A Lesson from B2B

Who is Dell?



Dell Is A Bank!

Order on the web

Dell Web Site

2 Hours

Part Orders

Part Suppliers

90 min - 2 Days

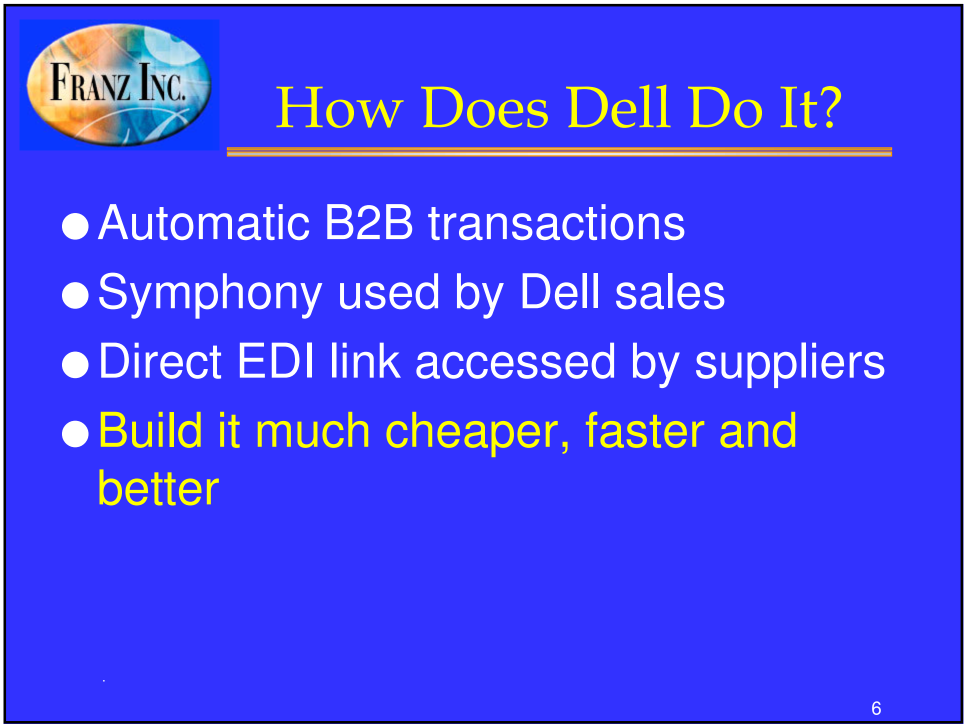
Finished Parts

Assembly Plant

6 Days

Customer

- Free cash flow of $3.2B of customers' money at any time
- ZERO inventory cost
- Lowest cost producer

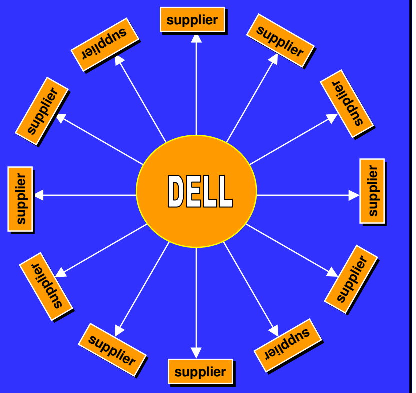


How Does Dell Do It?

- Automatic B2B transactions
- Symphony used by Dell sales
- Direct EDI link accessed by suppliers
- Build it much cheaper, faster and better

But, It's Very Expensive

- Only 5% of US companies use full B2B
- Proprietary EDI
- Hub and Spoke B2B
- $40K per EDI adapter
- Fixed, not portable nor scalable
- "Should I embrace B2B now?"



Central Business Dilemmas on Technology

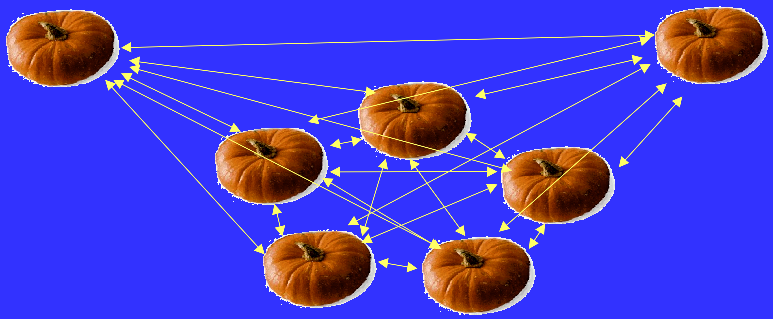
- When to embrace a new technology to improve business performance?
- Which technology to embrace?
- How much longer to continue with the current technology?

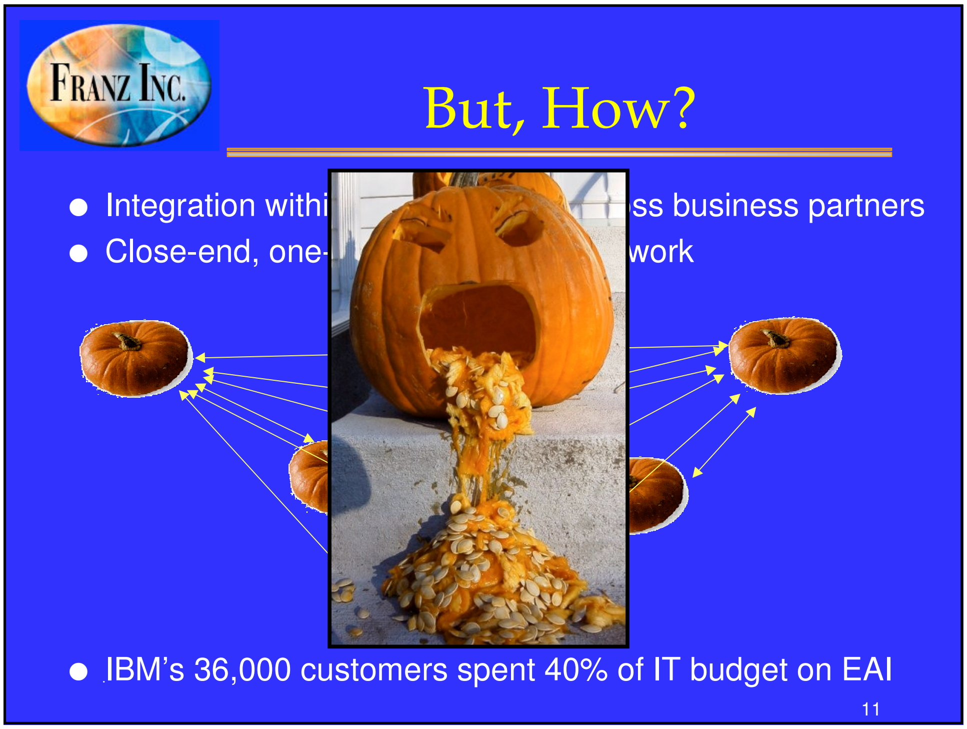
The Perfect *Storm*

- **Technology**: Internet
- **Market**: Globalization and outsourcing
- **Wal-Mart**: Economy of scale
- Companies have no choice but to truly embrace B2B e-Commerce NOW!

But, How?

- Integration within enterprise and across business partners
- Close-end, one-to-one system won’t work

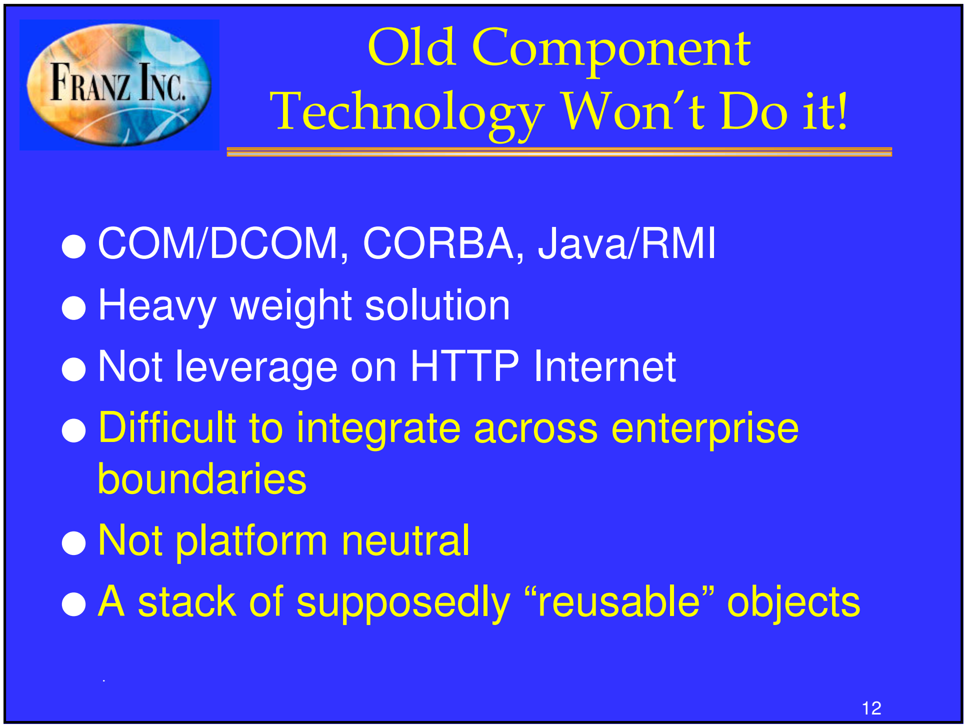




But, How?

- Integration withi ss business partners
- Close-end, one- work

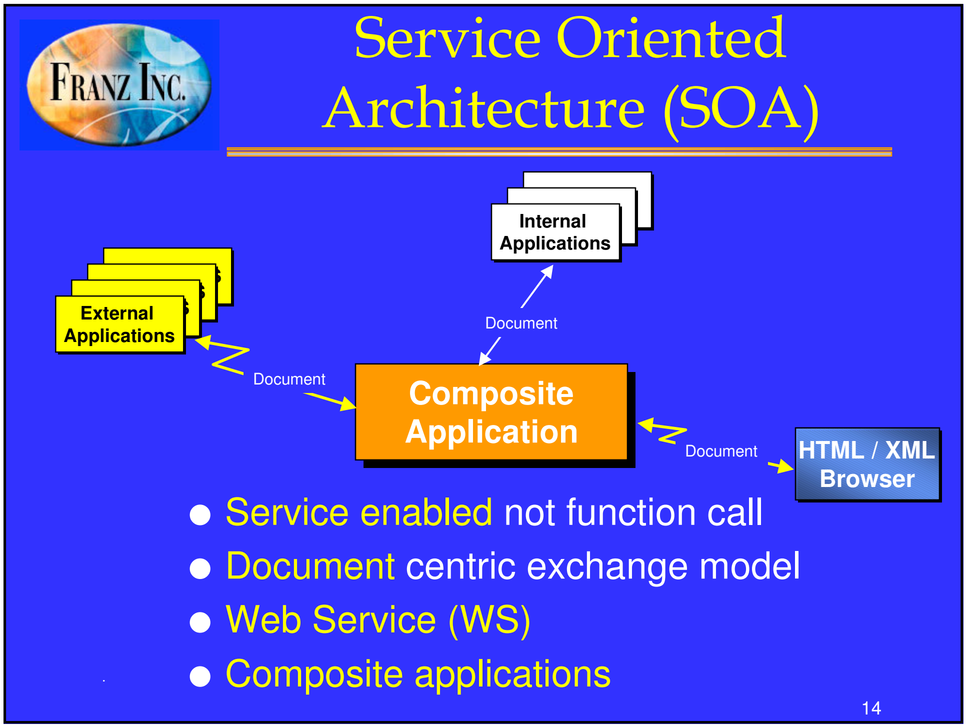
- IBM’s 36,000 customers spent 40% of IT budget on EAI



Old Component Technology Won’t Do it!

- COM/DCOM, CORBA, Java/RMI
- Heavy weight solution
- Not leverage on HTTP Internet
- Difficult to integrate across enterprise boundaries
- Not platform neutral
- A stack of supposedly “reusable” objects

Service Oriented Architecture (SOA) and Web Service (WS) may help us get there!



Service Oriented Architecture (SOA)

Internal Applications

Document

External Applications

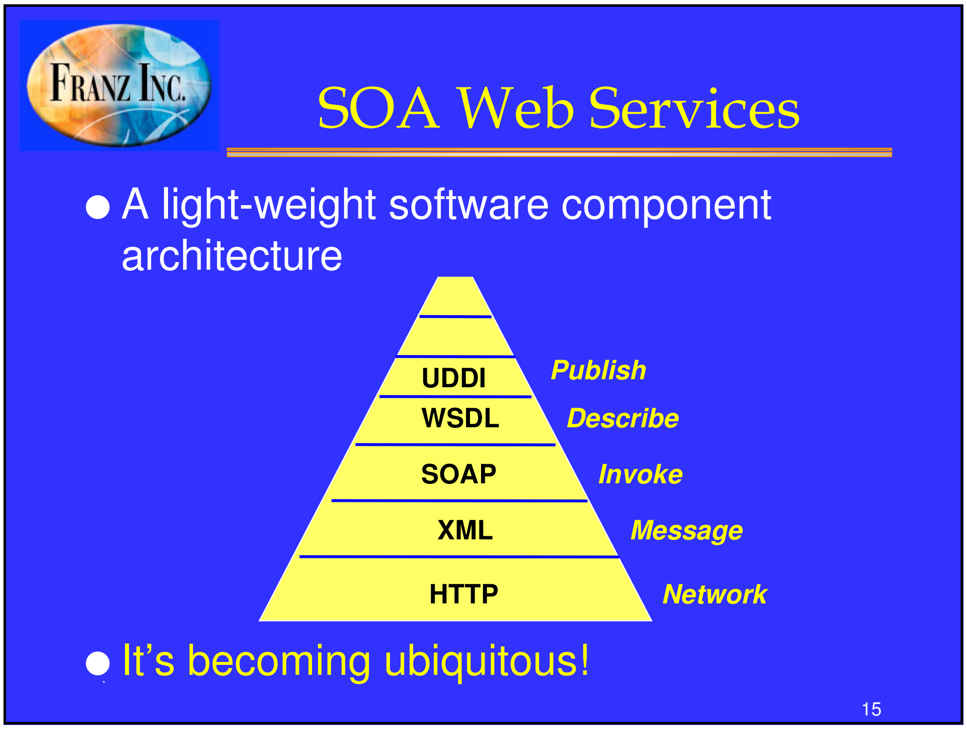
Document

Composite Application

Document

HTML / XML Browser

- Service enabled not function call
- Document centric exchange model
- Web Service (WS)
- Composite applications



FRANZ INC.

SOA Web Services

- A light-weight software component architecture

UDDI — *Publish*

WSDL — *Describe*

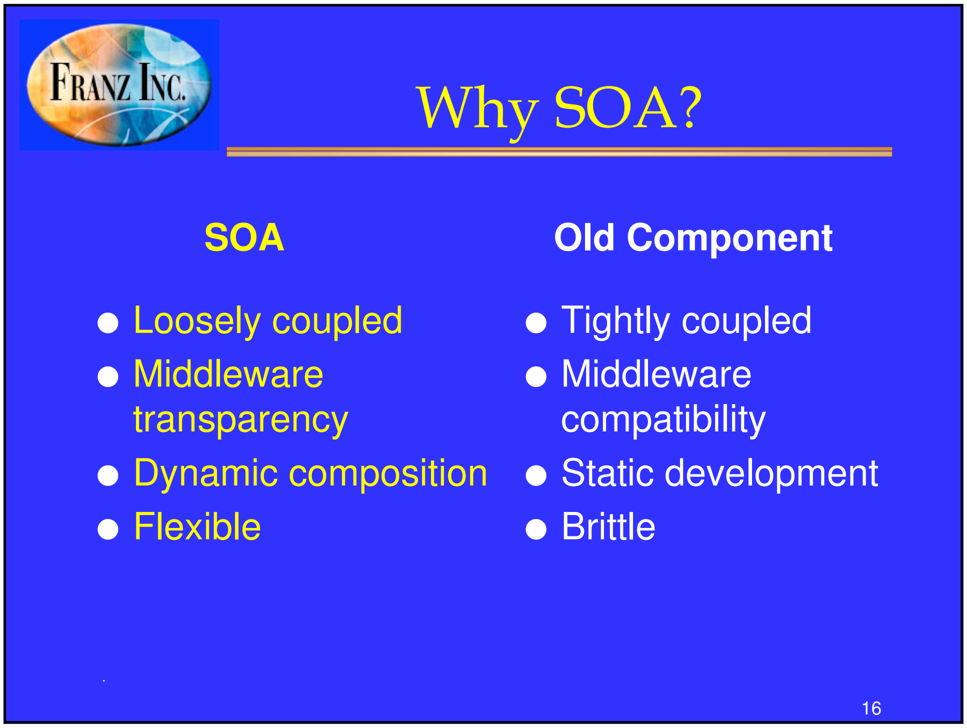
SOAP — *Invoke*

XML — *Message*

HTTP — *Network*

- It's becoming ubiquitous!

15

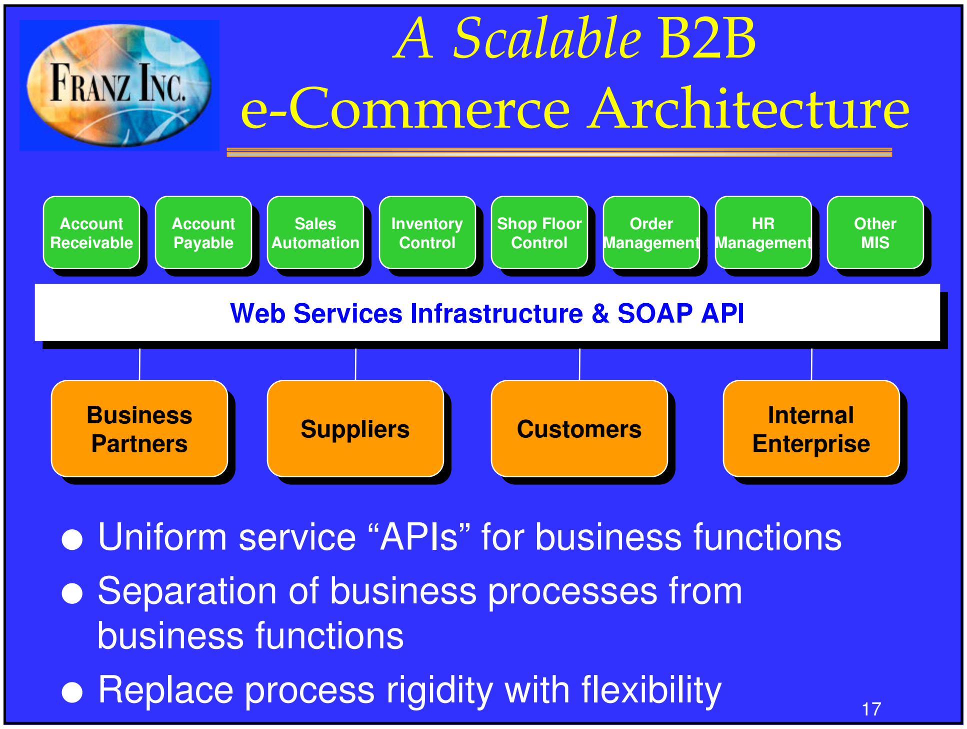


FRANZ INC.

Why SOA?

SOA	Old Component
Loosely coupled	Tightly coupled
Middleware transparency	Middleware compatibility
Dynamic composition	Static development
Flexible	Brittle

16



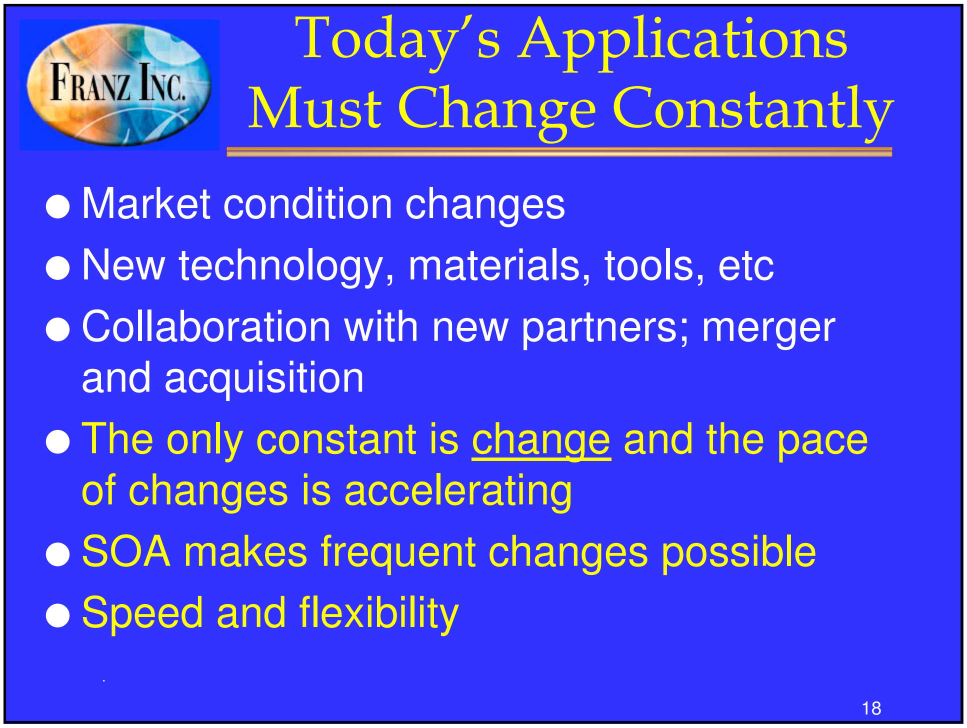
A Scalable B2B e-Commerce Architecture

Account Receivable | Account Payable | Sales Automation | Inventory Control | Shop Floor Control | Order Management | HR Management | Other MIS

Web Services Infrastructure & SOAP API

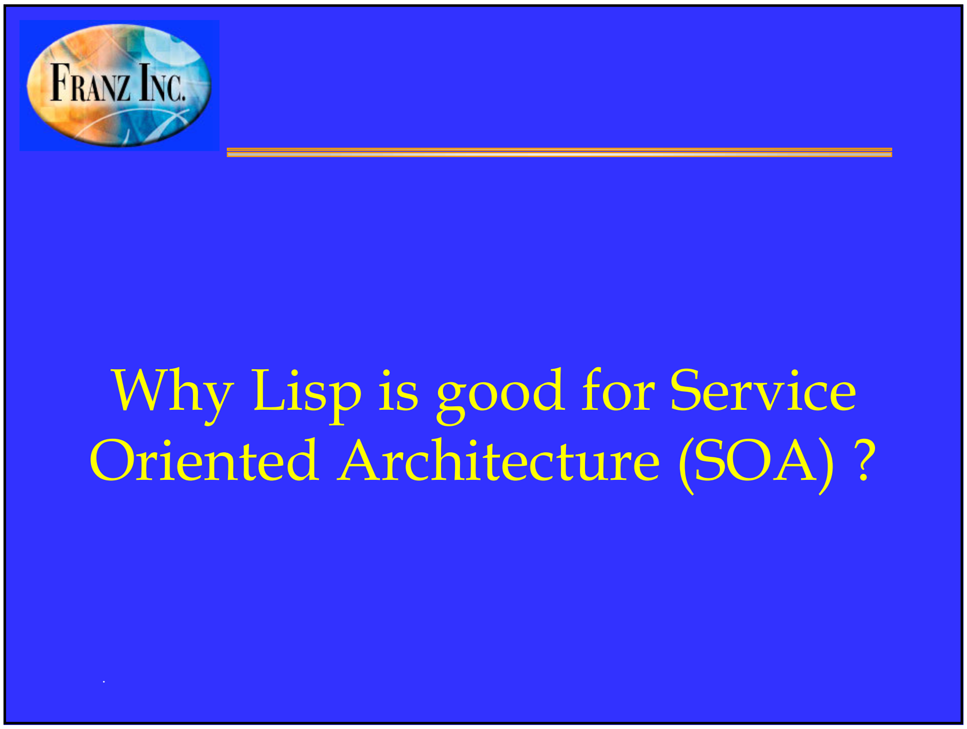
Business Partners | Suppliers | Customers | Internal Enterprise

- Uniform service “APIs” for business functions
- Separation of business processes from business functions
- Replace process rigidity with flexibility

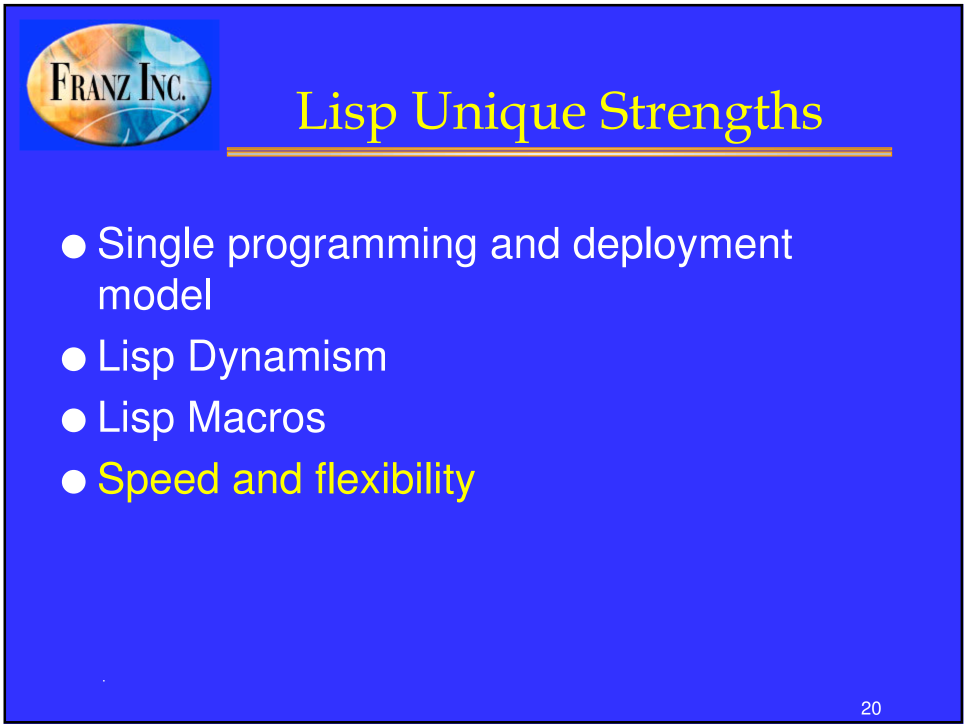


Today’s Applications Must Change Constantly

- Market condition changes
- New technology, materials, tools, etc
- Collaboration with new partners; merger and acquisition
- The only constant is change and the pace of changes is accelerating
- SOA makes frequent changes possible
- Speed and flexibility

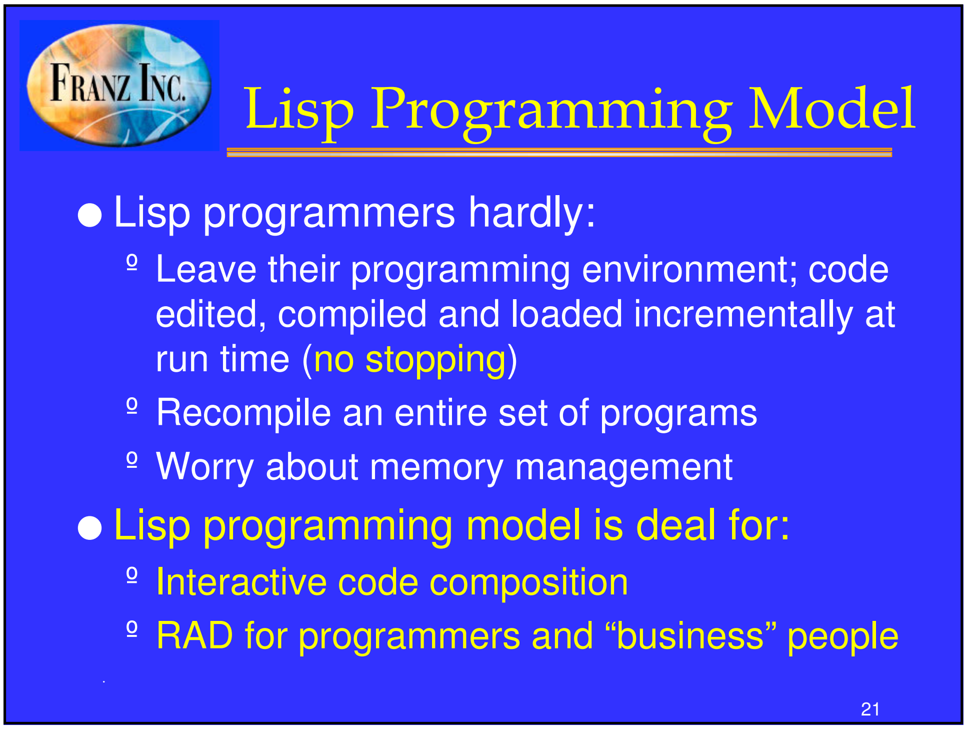


Why Lisp is good for Service Oriented Architecture (SOA) ?



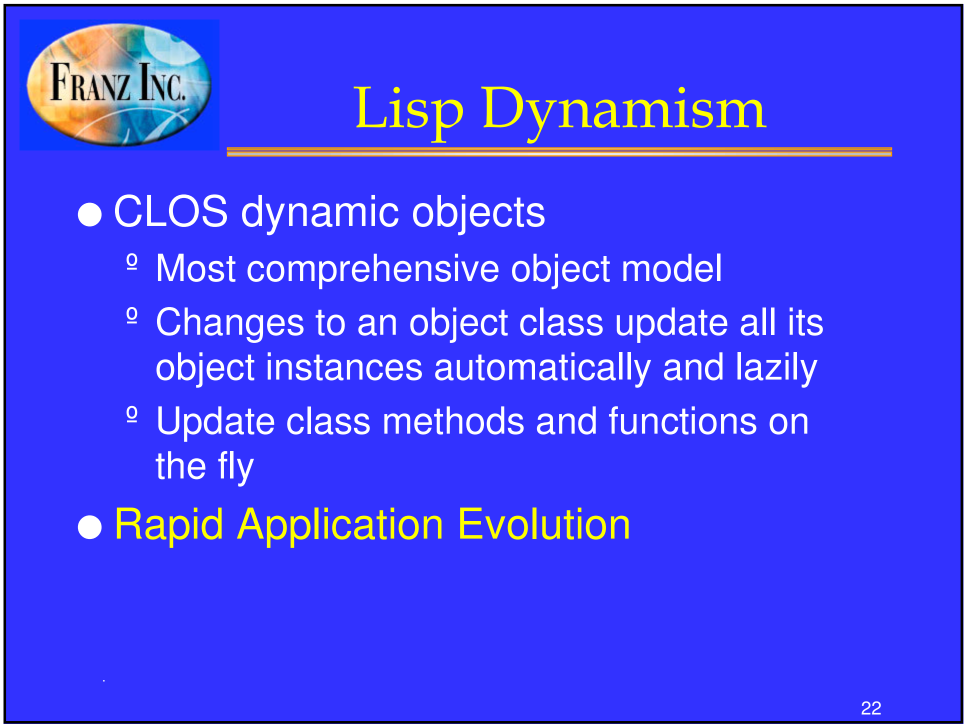
Lisp Unique Strengths

- Single programming and deployment model
- Lisp Dynamism
- Lisp Macros
- Speed and flexibility



Lisp Programming Model

- Lisp programmers hardly:
  - Leave their programming environment; code edited, compiled and loaded incrementally at run time (no stopping)
  - Recompile an entire set of programs
  - Worry about memory management
- Lisp programming model is deal for:
  - Interactive code composition
  - RAD for programmers and “business” people



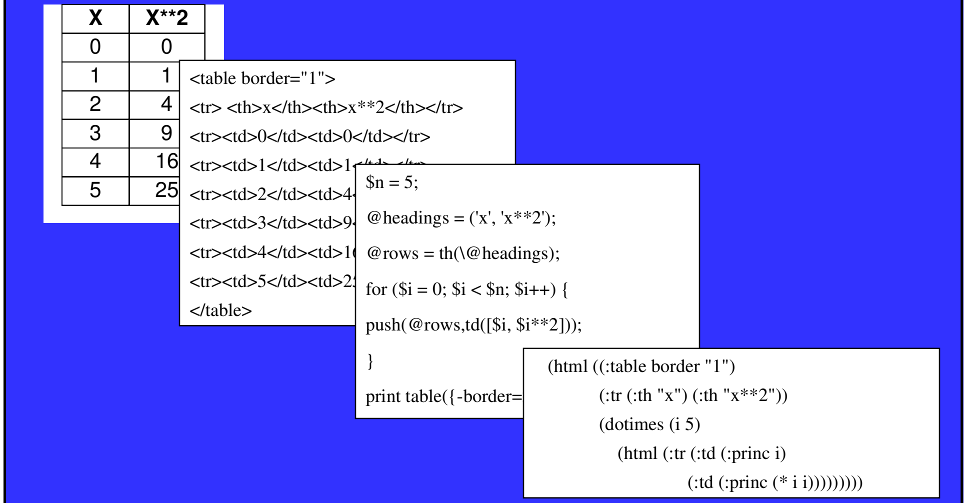
Lisp Dynamism

- CLOS dynamic objects
  - Most comprehensive object model
  - Changes to an object class update all its object instances automatically and lazily
  - Update class methods and functions on the fly
- Rapid Application Evolution

Lisp Macro

- Both Lisp code and data are s-expression
  - S-Expression ➔ atom [number, character, symbol, string, array, etc.]
  - S-Expression ➔ (S-Expression *)
- Code transformation and generation
- Syntax extension for specific domain (DSL)
- Up to 10x code base reduction and 3 to 5x productivity improvement
- Unparalleled speed and flexibility!

An HTML Macro Example

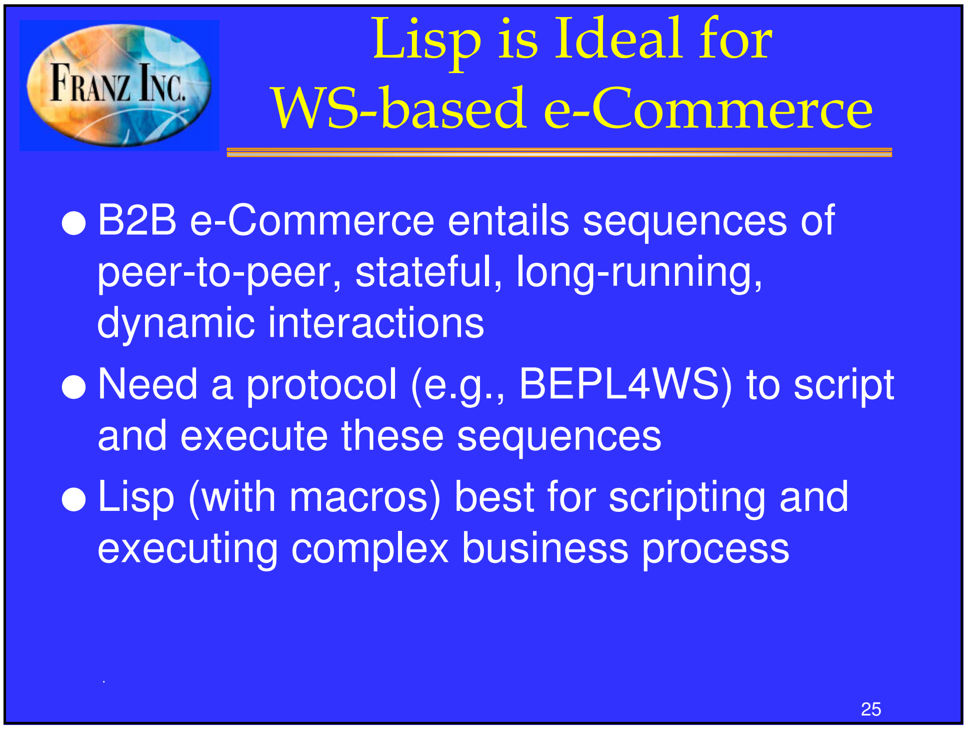


X	X**2
0	0
1	1
2	4
3	9
4	16
5	25

```
<table border="1">
<tr> <th>x</th><th>x**2</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1
<tr><td>2</td><td>4
<tr><td>3</td><td>9
<tr><td>4</td><td>1
<tr><td>5</td><td>2
</table>
```

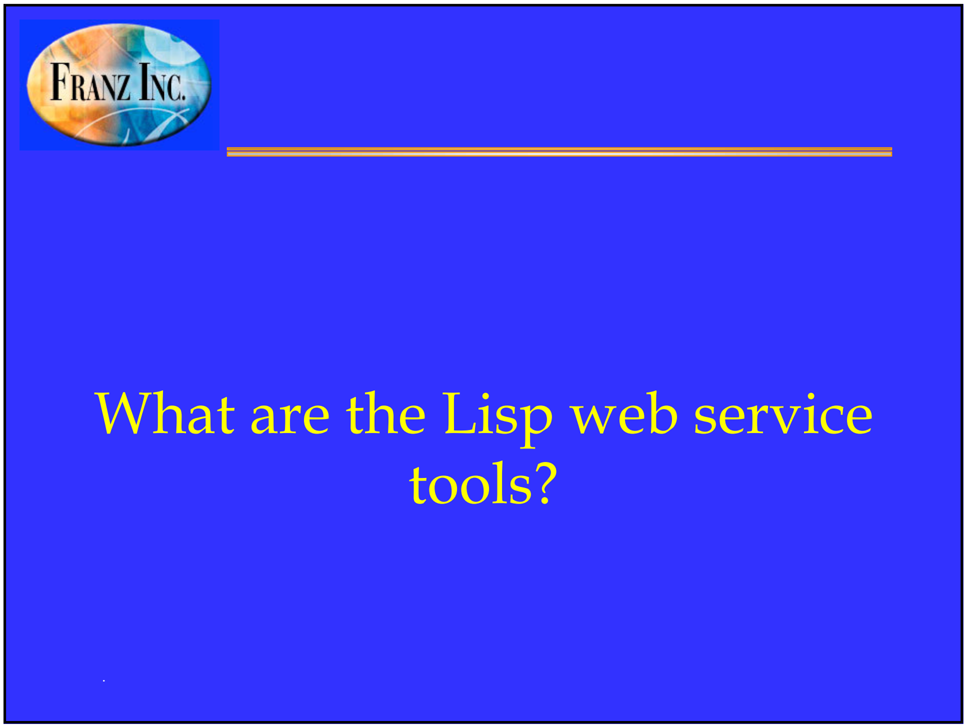
```
$n = 5;
@headings = ('x', 'x**2');
@rows = th(\@headings);
for ($i = 0; $i < $n; $i++) {
push(@rows,td([$i, $i**2]));
}
print table({-border=
```

```
(html ((:table border "1")
        (:tr (:th "x") (:th "x**2"))
        (dotimes (i 5)
          (html (:tr (:td (:princ i)
                      (:td (:princ (* i i)))))))))
```

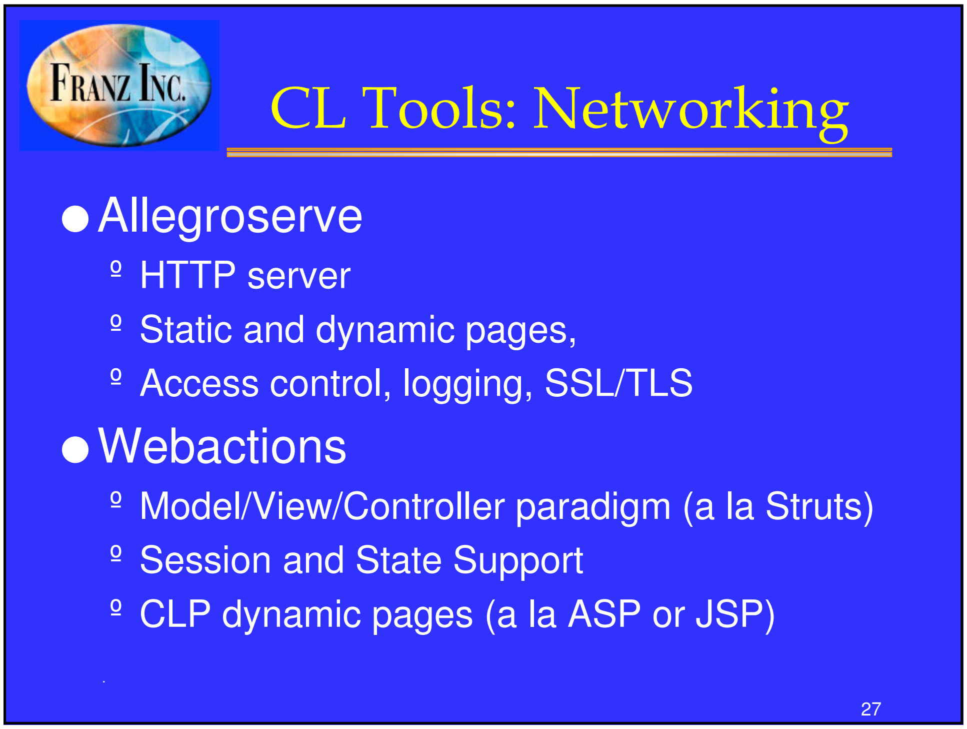


Lisp is Ideal for WS-based e-Commerce

- B2B e-Commerce entails sequences of peer-to-peer, stateful, long-running, dynamic interactions
- Need a protocol (e.g., BEPL4WS) to script and execute these sequences
- Lisp (with macros) best for scripting and executing complex business process



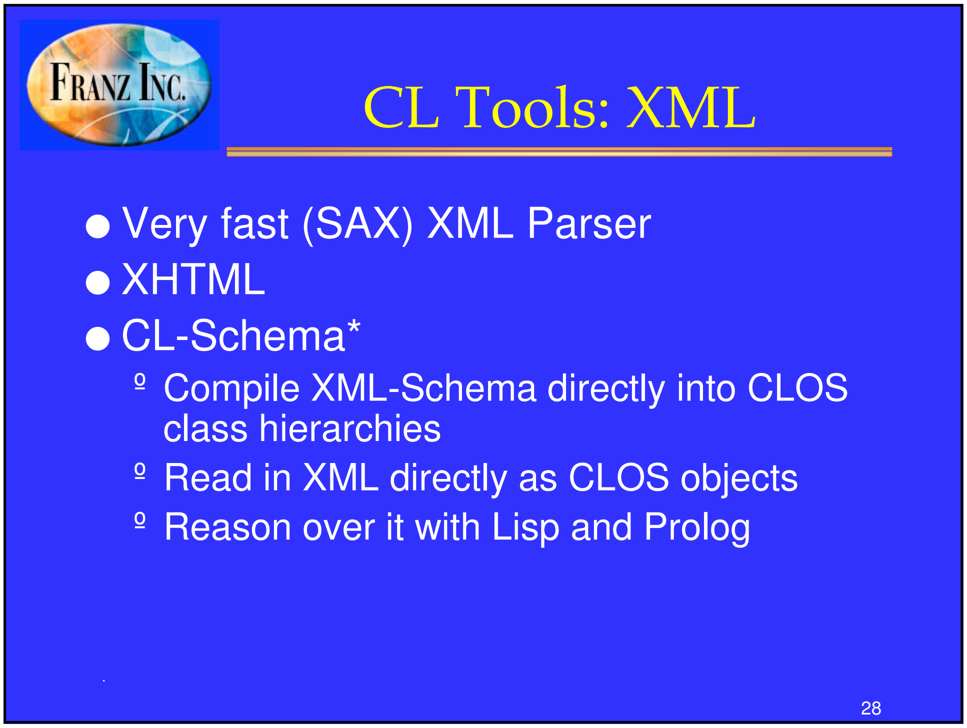
What are the Lisp web service tools?



CL Tools: Networking

- Allegroserve
  - º HTTP server
  - º Static and dynamic pages,
  - º Access control, logging, SSL/TLS
- Webactions
  - º Model/View/Controller paradigm (a la Struts)
  - º Session and State Support
  - º CLP dynamic pages (a la ASP or JSP)

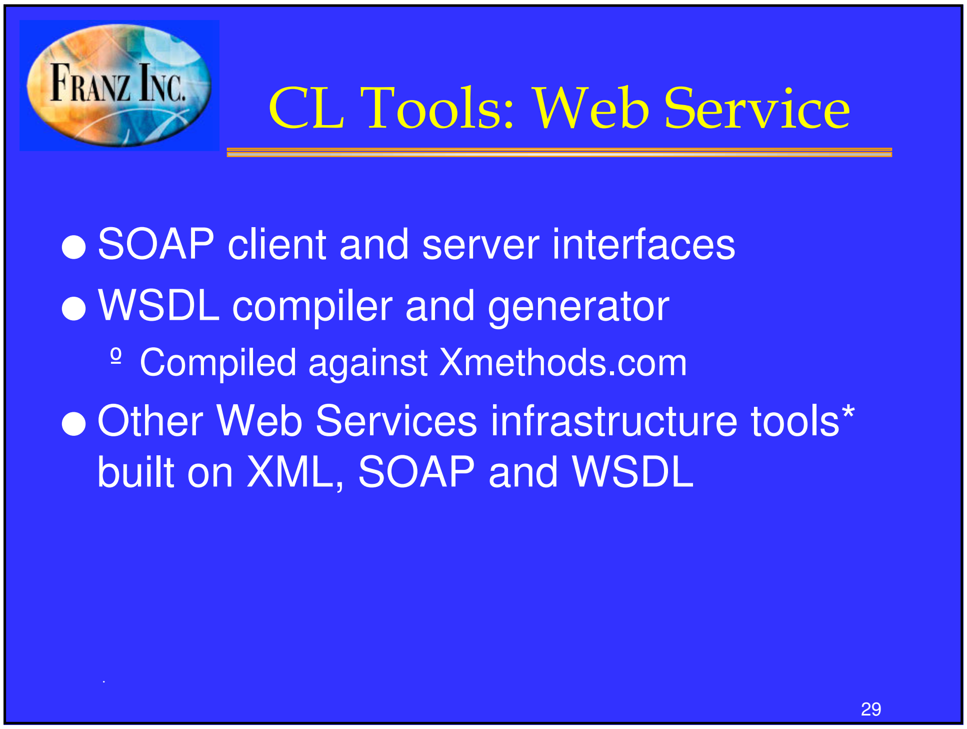
27



CL Tools: XML

- Very fast (SAX) XML Parser
- XHTML
- CL-Schema*
  - º Compile XML-Schema directly into CLOS class hierarchies
  - º Read in XML directly as CLOS objects
  - º Reason over it with Lisp and Prolog

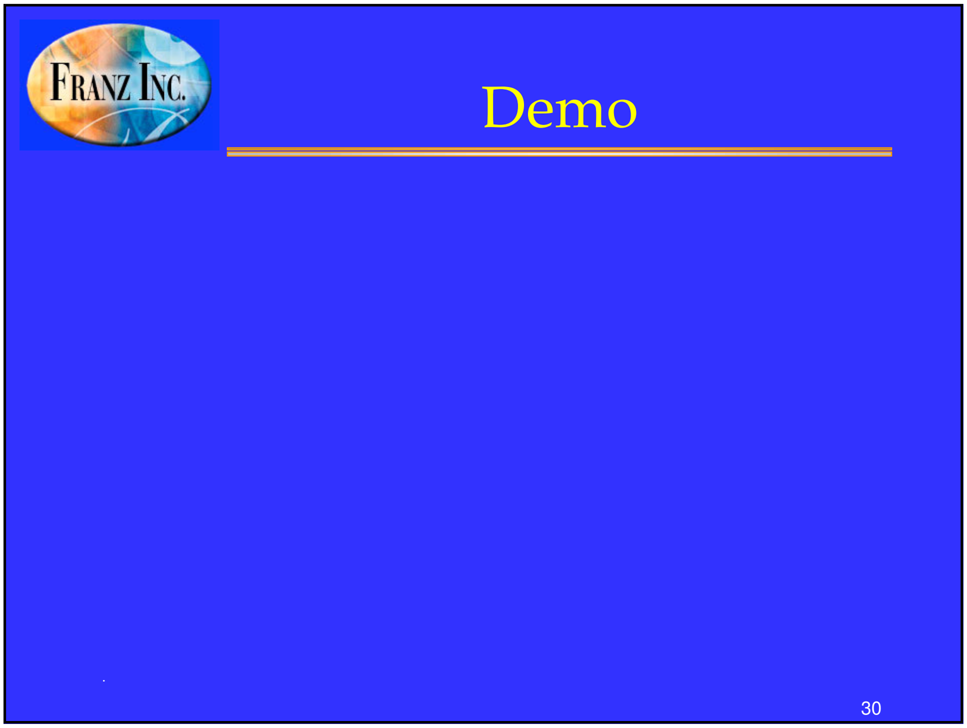
28



CL Tools: Web Service

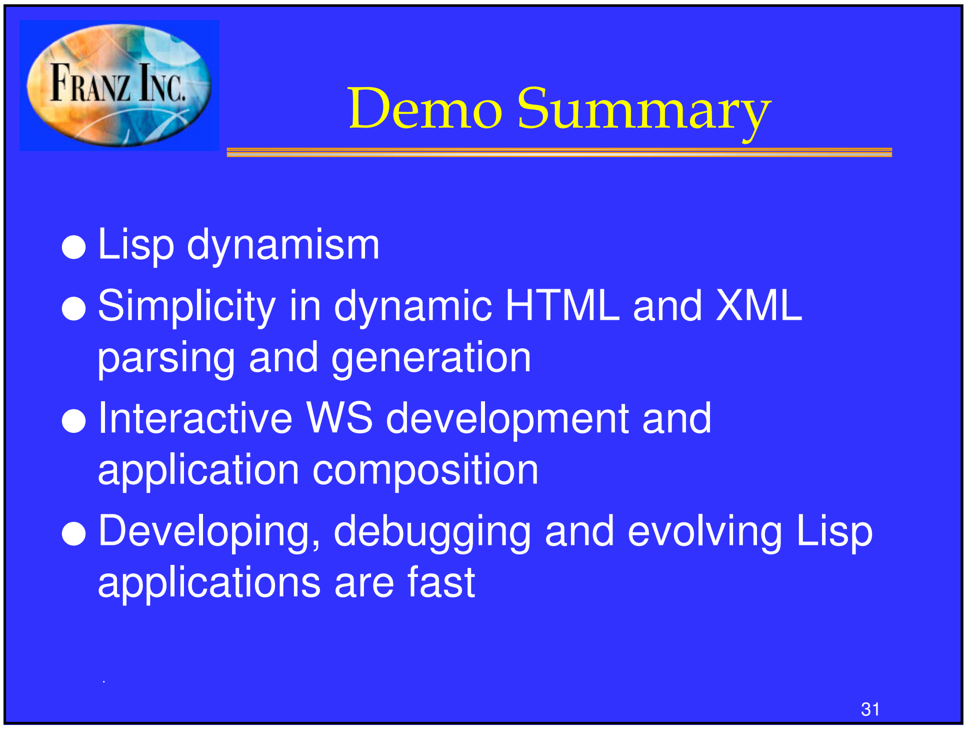
- SOAP client and server interfaces
- WSDL compiler and generator
  - Compiled against Xmethods.com
- Other Web Services infrastructure tools* built on XML, SOAP and WSDL

29



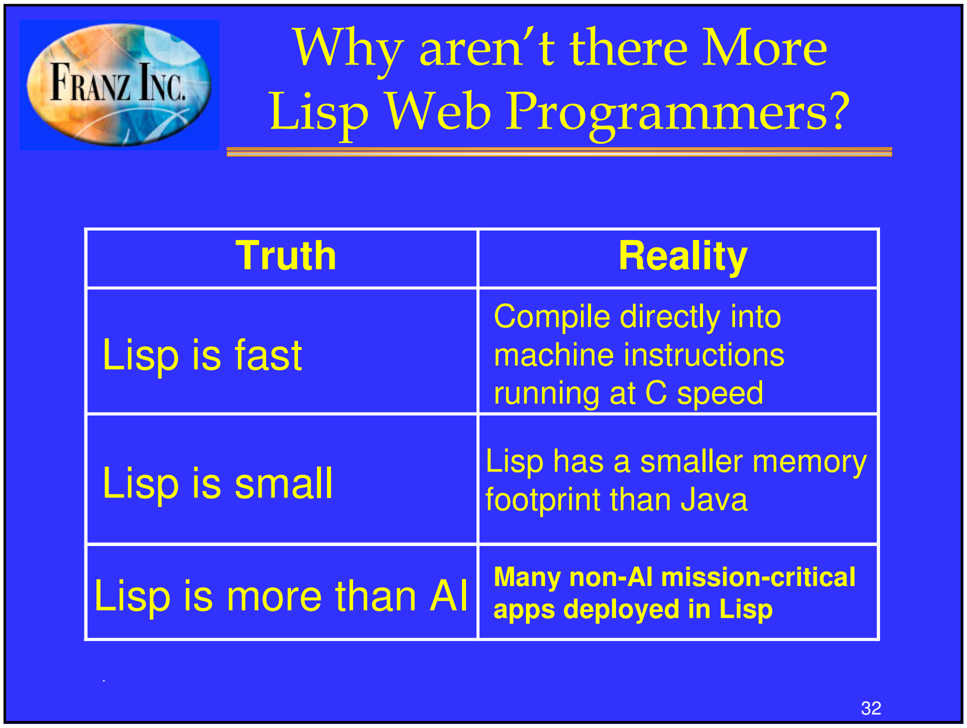
Demo

30



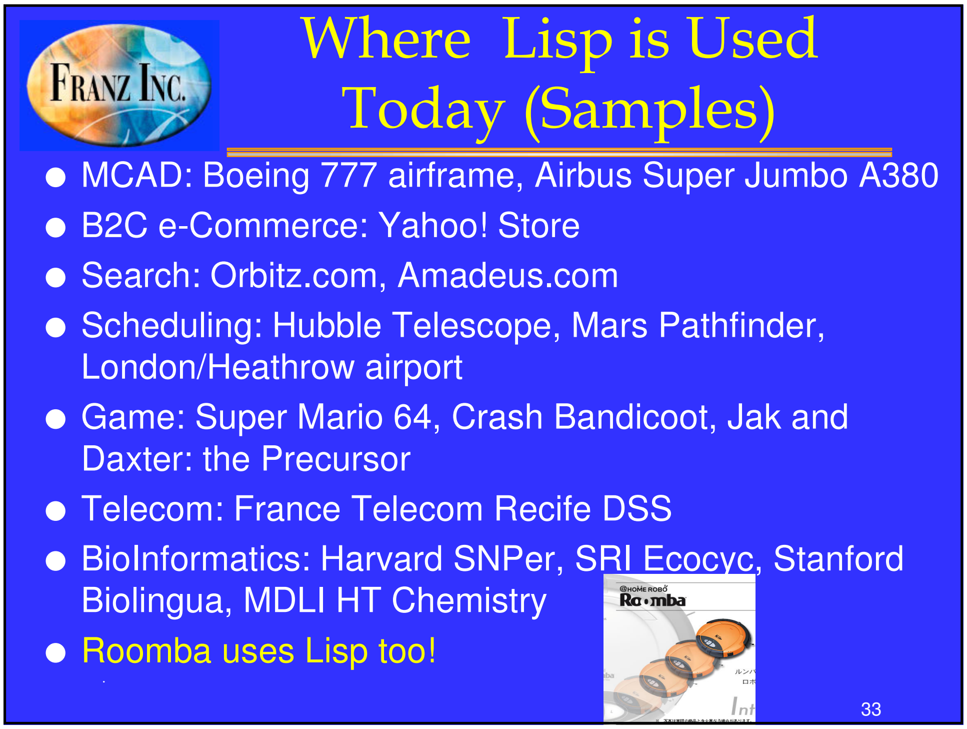
Demo Summary

- Lisp dynamism
- Simplicity in dynamic HTML and XML parsing and generation
- Interactive WS development and application composition
- Developing, debugging and evolving Lisp applications are fast



Why aren't there More Lisp Web Programmers?

Truth	Reality
Lisp is fast	Compile directly into machine instructions running at C speed
Lisp is small	Lisp has a smaller memory footprint than Java
Lisp is more than AI	**Many non-AI mission-critical apps deployed in Lisp**

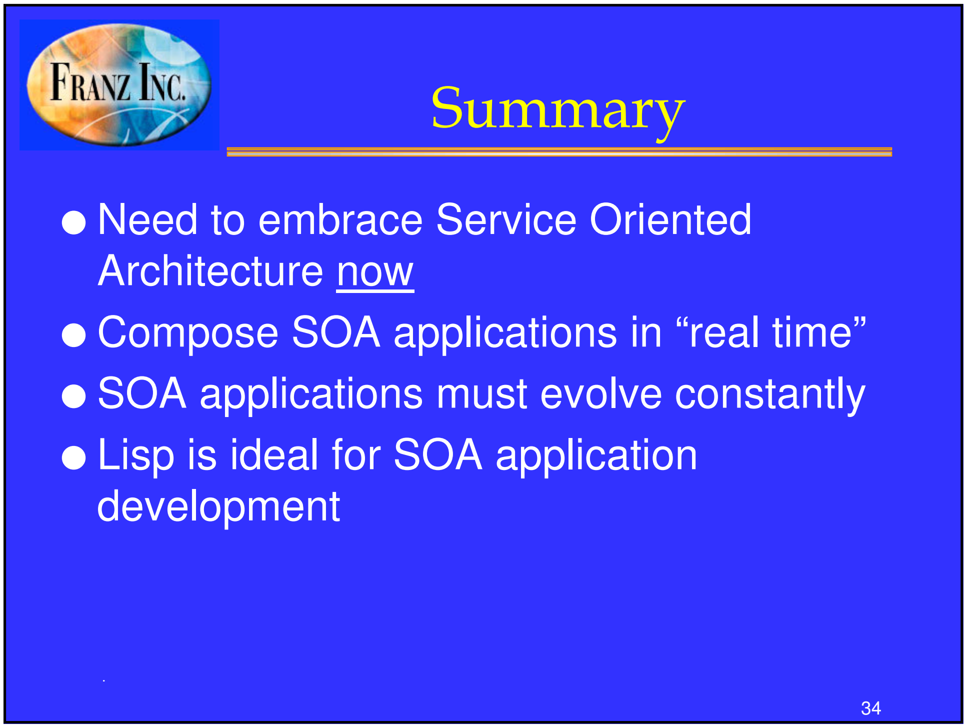


FRANZ INC.

Where Lisp is Used Today (Samples)

- MCAD: Boeing 777 airframe, Airbus Super Jumbo A380
- B2C e-Commerce: Yahoo! Store
- Search: Orbitz.com, Amadeus.com
- Scheduling: Hubble Telescope, Mars Pathfinder, London/Heathrow airport
- Game: Super Mario 64, Crash Bandicoot, Jak and Daxter: the Precursor
- Telecom: France Telecom Recife DSS
- BioInformatics: Harvard SNPer, SRI Ecocyc, Stanford Biolingua, MDLI HT Chemistry
- Roomba uses Lisp too!

33

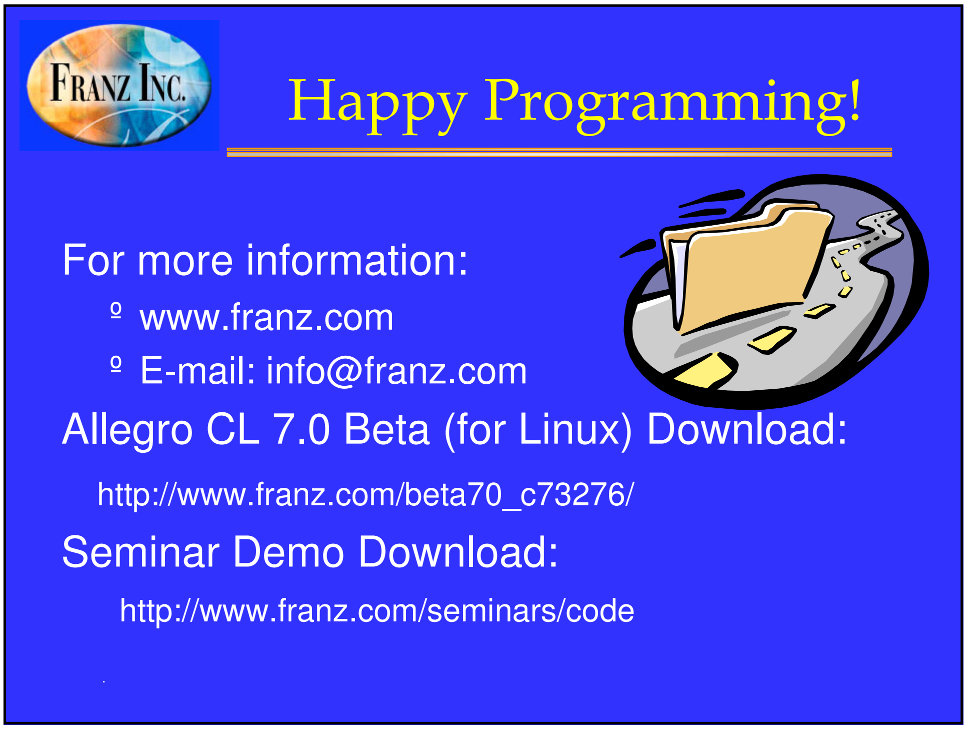


FRANZ INC.

Summary

- Need to embrace Service Oriented Architecture now
- Compose SOA applications in “real time”
- SOA applications must evolve constantly
- Lisp is ideal for SOA application development

34



Happy Programming!

For more information:

º www.franz.com

º E-mail: info@franz.com

Allegro CL 7.0 Beta (for Linux) Download:

http://www.franz.com/beta70_c73276/

Seminar Demo Download:

http://www.franz.com/seminars/code