

Next Generation IT - Life After Jurassic Middleware

Dave Thomas

Bedarra Research Labs, Object Mentor
Carleton University
Queensland University of Technology
www.davethomas.net
dave@bedarra.com

The Journey to Post Modern IT Circa 2010..

The Challenge

Middleware – Life in The Tar Bit
Next Generation Application Challenges
Business Agility – The Driver for Change
Lean and Agile – Towards Agility

BUT !?!

The Solution

Next Generation IT
Pervasive Computing Power – The Hardware Enabler
Cloud Computing – The Infrastructure
Higher Order Service Oriented Computing – The Software Enabler
Do It Yourself Programming – The Empowerment

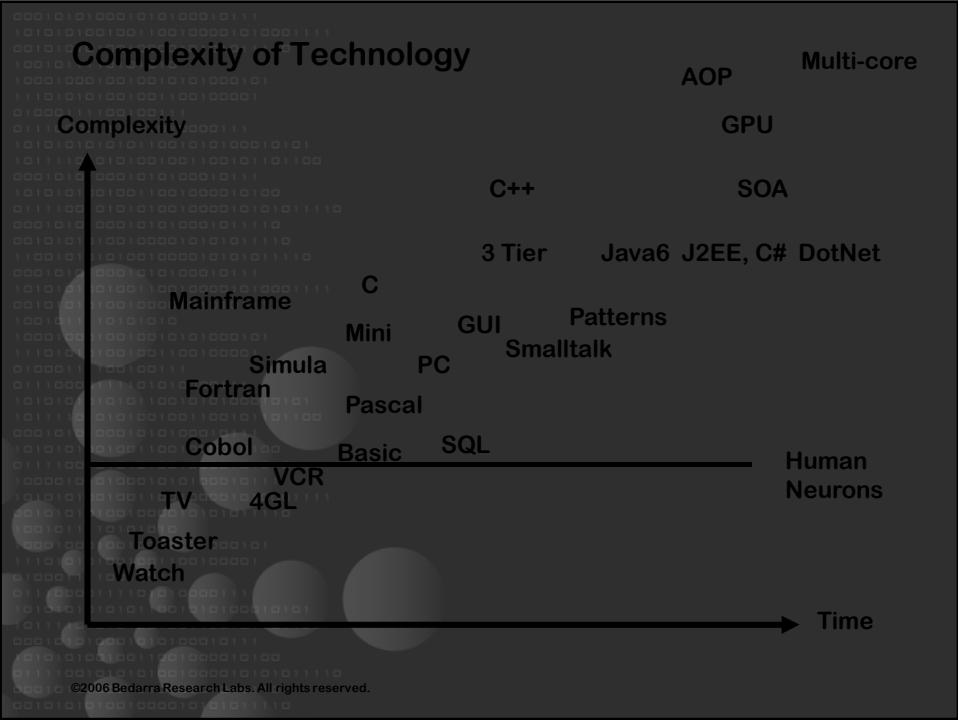
Discussion

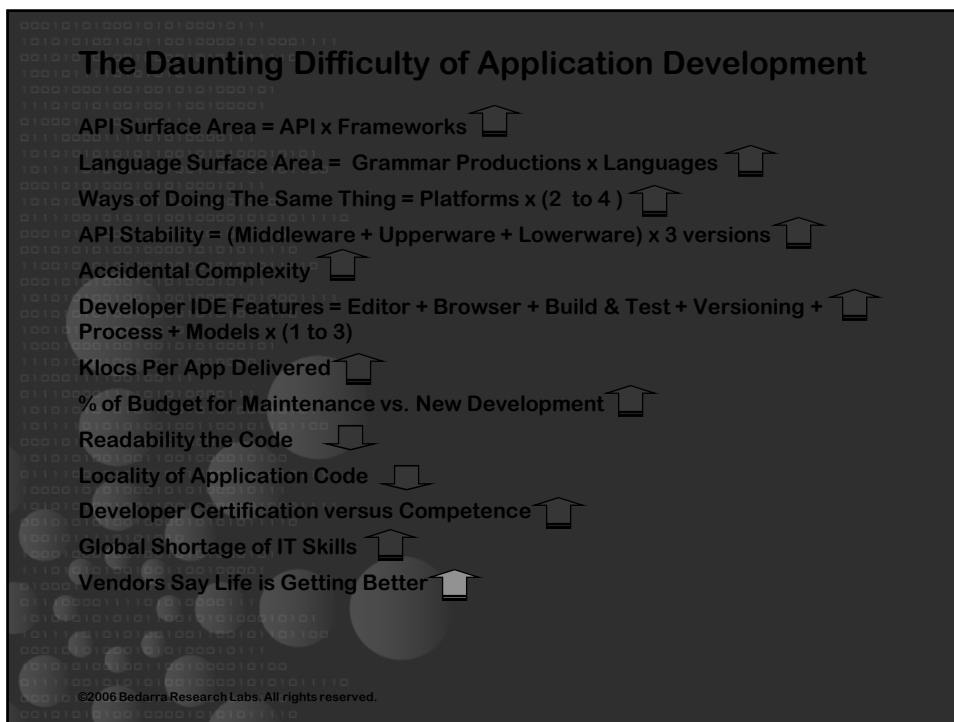
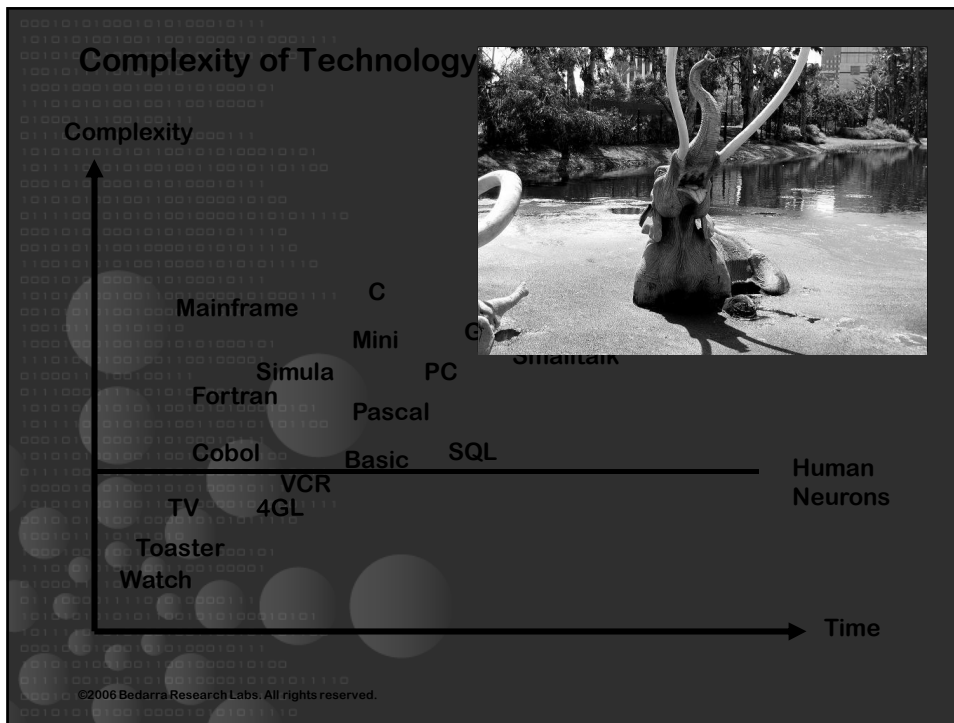
©2006 Bedarra Research Labs. All rights reserved.

Jurassic Middleware – Life in The Tar Bits

1. Prehistoric Software As A Service – Mainframe Dinosaurs, Simple Verbose Cobol, PL/I, Fortran and Cryptic but powerful APL
2. The Age of Complexity – Client-Server, C/C++, SQL, PSQL, TSQL
3. The Age of Naïve Simplicity – Spreadsheets and Enterprise Application
4. The Age of Absurd Complexity – Objects, Middleware, XML, BPM, SOA ...
5. Post Modern Computing – Life after Middleware?

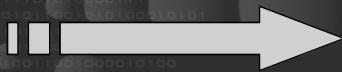
©2006 Bedarra Research Labs. All rights reserved.





The Escalating Costs of Ownership

- Software Tiers
- Hardware Tiers
- Software Stack
- Software Tools – IDEs, Build, SCM, SQA, Modeling, Performance
- Installation and Upgrades
- Vendor Interoperability
- Enterprise Open Source Version Management
- Recruited and Retaining Top Talent
- Development Maintenance as a % of IT Budget
- Platform, Framework, Tool churn rate
- License Complexity and Costs
- Vendor Lock
- Vendors say things are getting better



Software As A Service (SaaS)

©2006 Bedarra Research Labs. All rights reserved.

The Challenges of Next Generation Applications

Real-time Business

- Development in Real-time
- Execution in Real-time
- Deployment in Real-time (zero install)
- Global Applications Deployed Directly To Any Device
- Massive Amounts Of Data
 - Raw Processing versus Aggregation
 - 3D Data – semi-structured numeric, text, voice, video
 - Event Streams

©2006 Bedarra Research Labs. All rights reserved.

The Challenges of Next Generation Applications

Collaborative Applications

- Eliminate artificial technical or organizational barriers
- Cross Functional
- Cross Organizational Virtual Teams
- Selective sharing of data with partners, competitors and customers
- Leverage Service Providers
- Mass Customization

©2006 Bedarra Research Labs. All rights reserved.

Business Agility – The Driver

Agility enables the business to respond quickly to customers, partners and the competitors.

Agility is a strategic business hence IT capability.

Agility implies the ability to rapidly build and configure tailored solutions which span internal and external systems.

Want to exploit the price/performance of new hardware technology.

Agility is defined by companies that your CEO envies.

Progressive Insurance

Google

Amazon, Yahoo

Sales Force

Facebook, MySpace, LinkedIn

Apple



©2006 Bedarra Research Labs. All rights reserved.

Enterprise IT Best Practices

- 3 Tier Architecture
- Thin/Web Client
- J2EE or Dot Net OO Middleware
- ORM to Relational DB and File Systems
- OO Languages and Tools
- Scripting
- Agile Development

©2006 Bedarra Research Labs. All rights reserved.

Enterprise IT Best Practices

- 3 Tier Architecture
- ORM to Relational DB/Legacy Files
- J2EE or Dot Net OO Middleware
- OO Languages and Tools
- Thin/Web Client
- SOA and BPM
- Scripting
- Agile Development



©2006 Bedarra Research Labs. All rights reserved.

BUT We Can't Get There From Here?!

Lean and Agile => Improve Predictability and Quality, but still we have limited Agility!

Both Business IT and Software Product Vendors are:

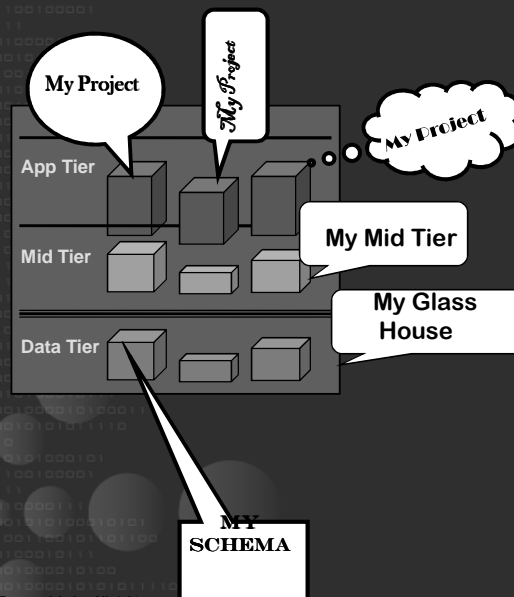
- Frustrated with their lack of agility in both development and deployment of applications and services.
- Finding that their best people, equipped with the best practices, tools and middleware can barely keep pace.
- Concerned with their ability to meet the demands of Next Generation Applications

Both are coming to realize that the companies they want to emulate are taking a different road with respect to infrastructure, development and delivery which provides them substantially reduced operating costs and increased agility.

This realization is the primary driver for what we call the Next Generation IT.

©2006 Bedarra Research Labs. All rights reserved.

Barriers To IT Agility – Techno-cultures and Territories



©2006 Bedarra Research Labs. All rights reserved.

Pervasive Computing Power – The Hardware Enabler

Processors, Memory, Bandwidth and Storage are "free"

- 100s to 100000s of processors ("Googleplex")
- Oodles of Memory and Gaggles of Disk Storage
- High Speed Wired and Wireless Bandwidth
- Multiprocessing, Multi Cores and System On Chip
- Special purpose processors e.g. GPUs
- Dynamically Reconfigurable Systems

©2006 Bedarra Research Labs. All rights reserved.

Pervasive Computing Power – The Hardware Enabler

Pervasive Connectivity

- Internet Everywhere
- Always on but Occasionally Disconnected Devices

Enhanced Input and Output

- Audio and Video Input
- Two handed input
- Smart Materials
- Environment, Location Awareness

©2006 Bedarra Research Labs. All rights reserved.

Global Distributed Data Centers – Tomorrow's Glass House

“Hey! You! Get On To My Cloud!”*

- Google, Amazon, SalesForce
- Dell Cloud Computing Division
- IBM On Demand and Blue Cloud
- Microsoft MSN Live Cloud

IT Glass House => Distributed Super Computer

- Low Cost, Low Power Components
- Fault Tolerance through failover and replication
- Non Stop Operation through HW and SW upgrades of both systems software and applications
- Embedded Security versus Simple Access Control
- Automatic Computing
- Dynamic Resource Management

* Apologies to the Rolling Stones

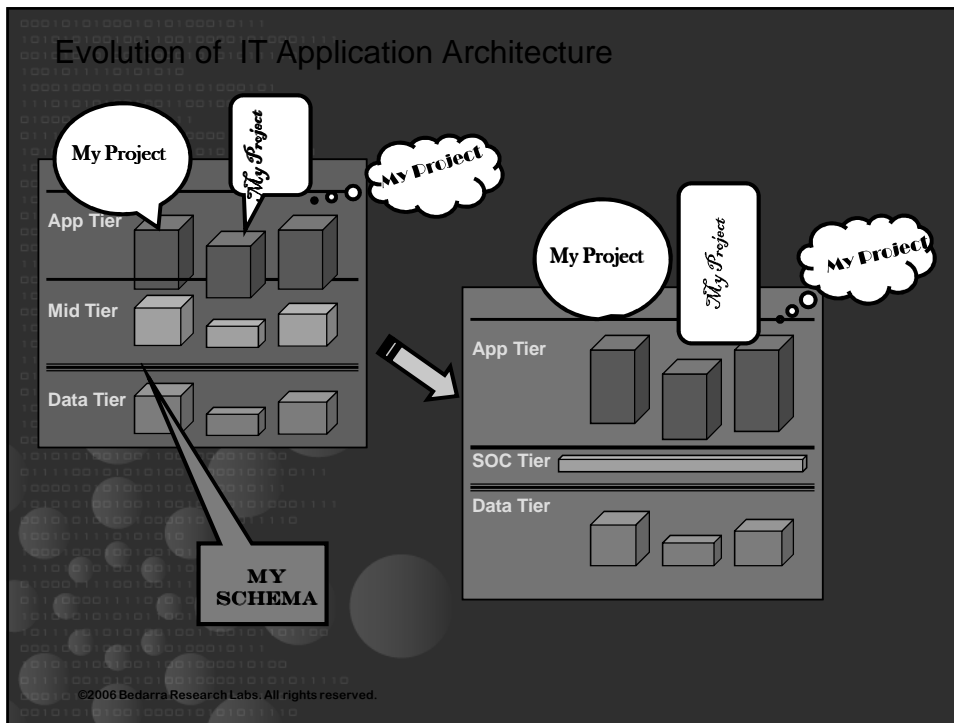
©2006 Bedarra Research Labs. All rights reserved.

Service Oriented Computing Infrastructure The Software Enabler

The Emergence of A Simpler Application Infrastructure

- Examples - On Demand, Software As A Service such as Amazon S3, EC2, SimpleDB, Google App Engine, Sales Force
- Simpler limited “thin” service API (< 50) closer to underlying platform which provides support for scalable, distributed, secure computing
- Independence on mainstream vendor Underware and Middleware
- Application Development Benefits
 - Small Service API (thin to none class library & frameworks)
 - Limited Choice Reduces Decisions and Support
 - Leverages Other Apps through Services
 - Total App Responsibility from envisioning to production i.e. App Team carries the beeper

©2006 Bedarra Research Labs. All rights reserved.



Most Applications are Still Essentially CRUD!

Modern Application = CRUD + Compute + Interact

- CRUD against federated data sources
- Computation against a huge heterogeneous data sets
- Display based on an known information model

If one ignores objects, heterogeneous data sources and fancy UI it is essentially still a simple 4GL problem

```

BEGIN MyApplication
  SELECT what user and/or application needs from
        WHERE it is stored
  THEN
    Perform more Filtering and Calculations
  THEN
    UPDATE appropriate things WHERE it is needed
  THEN
    Display what is needed
END MyApplication

```

©2006 Bedarra Research Labs. All rights reserved.

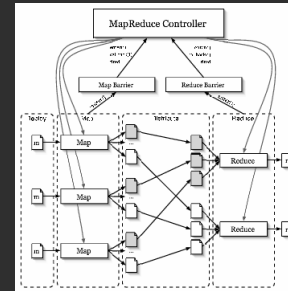
Super CRUD – Functional Programming for the Masses The Leverage

Examples

- Google Map Reduce (f(g(x))
(Apache Hadoop, Connection Machine Lisp)
- Microsoft LINQ
(inspired by Haskell and Duck Types)

Benefits

- Functional architecture reduces API surface area and enables rapid application development
- Allow developers to think in terms of simple collections independent of their shape and representation
- Enables implicit fault tolerant data parallelism and distribution (i.e. move the function to the data instead of moving the data to the function)
- Transactional Shared Memory enables simpler programming for “state full sinners”



©2006 Bedarra Research Labs. All rights reserved.

Application Experiences using High Order Languages

Examples

- Relational Programming - SQL
- Vector Programming – APL, NIAL, J
- Functional Programming – Scheme, Haskell, F#
- Set Programming – SETL, Kleisli, XQuery
- Dynamic Object Programming – Smalltalk, CLOS, Ruby
- Data Flow – Pipes and Filters, Symbol
- Logic Programming – Prolog, Datalog
- Constraint Programming – ThingLab
- Reactive Programming – Erlang
- Text Processing – SNOBOL/Icon, Omnimark

©2006 Bedarra Research Labs. All rights reserved.

Do It Ourselves Programming – The Empowerment Domain Oriented Programming and DSLs

Business Driven Development

- Next Generation Query and Update Language - Beyond SQL
 - Collection Programming - Relations, Sets, Dictionaries, Lists and Arrays
- Business Teams with domain experts and developers embedded in the team
- Use of Domain Specific Language
- Scripting, Assembly and By Demonstration Programming

©2006 Bedarra Research Labs. All rights reserved.

Do It Ourselves Programming – The Empowerment

Examples of Business Programming by Knowledgeable Business Users

- 4GLs – Synon, Natural, Mapper, ZIM, Cool Gen, Access
- Programming By Example – QBE/OBE/SBA, Tinker
- Rule Programming – Business Rules, Expert Systems, Decision Tables, State Tables, StageCast
- Spreadsheets – Excel, The CIA Analyst, AgentSheets, DabbleDB
- Mathematical Programming - Matlab, Mathematica, Maple
- Visual Languages – Prograph, Labview
- Pattern Matching – AWK, Perl

©2006 Bedarra Research Labs. All rights reserved.

Do It Ourselves Programming – The Empowerment

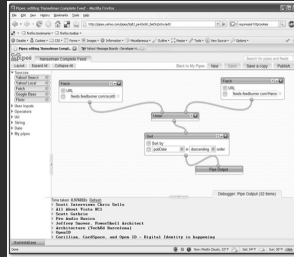
Business Driven Development

- Enterprise Mashups – The Real SOA?
- Applications Assembled from Services
- Examples

IBM QEDWiki

Yahoo Pipes

DabbleDB



Enterprise
Spreadsheets

©2006 Bedarra Research Labs. All rights reserved.

©2006 Bedarra Research Labs. All rights reserved.

Summary

Challenges – Middleware, Vendor Lock, Skills Shortages, Expense

- Current middleware cannot provide the agility nor provide cost effective scalable commodity infrastructure.
- Current programming technology is too complex and too inefficient to leverage next generation infrastructures
- We are facing an acute shortage of skilled application developers
- Hence we need to consider simpler alternatives

Solutions – Cloud Computing enabled Domain Oriented Programming

- Simple Services enable Agility and Leverage Scalable Commodity Technology
- Functional Services enable rapid application develop and enable the service infrastructure to handle concurrency
- Domain Oriented Programming enable domain specific service development
- Business Programming enable business teams with embedded developers to deliver applications

©2006 Bedarra Research Labs. All rights reserved.

References

"Enabling Application Agility - Software as A Service, Cloud Computing and Dynamic Languages", in *Journal of Object Technology*, vol. 7 no. 4 May-June 2008, pp. 29-32
http://www.jot.fm/issues/issue_2008_05/column3/

©2006 Bedarra Research Labs. All rights reserved.