Delivering Excellence in Software Engineering

Development of Software Products with Simultaneous Support of J2EE and .NET

Aliaksei Kazlouski,

EPAM Systems

for Software Engineering Conference (Russia), 2005

Agenda

- About EPAM and LogicLibrary
- LogicLibrary Logidex product overview
- Porting Logidex from J2EE to .NET
- Supporting .NET and J2EE versions in parallel

About EPAM Systems

EPAM at a Glance

- US Corporation founded in Princeton, 1993
- No.1 on “Top 5 to Watch in Central and Eastern Europe” (“Offshore 100”, 2005)
- No.3 on global “Top 10 Specialty Application Development Leaders” (“Offshore 100”, 2005)
- Over 1300 employees
- Geo-diverse delivery: US, Russia, Hungary, Belarus
- First CMMI Level 4 assessment in Europe
- Successful projects deployed in over 30 countries
- 6,5+ million hours of software development experience

http://www.epam.com

EPAM Selected Clients

Global business leaders such as Reuters, CareFirst BCBS, Empire BCBS, Colgate-Palmolive, Halliburton, London Stock Exchange, SBLI, British Telecom, AeroMexico, Schlumberger.
Global technology leaders such as SAP, Microsoft, BEA Systems, Hyperion, Sun Microsystems.

InfoWorld

Leveraging a global advantage

By Jon Udell, April 18, 2003

"EPAM equips teams with the best tools, trains them to use and refine best practices in multiple disciplines, and deploy them in engagements that are, for their clients, highly strategic..."

<epam>

About LogicLibrary

LogicLibrary

- Founded in 2000
- Headquartered in Pittsburgh, PA
- Delivers **Logidex**, a patent-pending enterprise application development tool that manages software development assets and related artifacts
- Key partnerships with all major application development vendors, including Microsoft, IBM, Serena, Borland and SUN

www.logiclibrary.com

About EPAM and LogicLibrary

EPAM and LogicLibrary

- In 2002 LogicLibrary selected EPAM to partially outsource Logidex development (co-working with LogicLibrary’s Dev team)
- EPAM started from: J2EE UI development, QA automation – functional and performance testing
- Scope extended to: Porting Logidex to .NET, Add-ins development for IDEs (Eclipse, WSAD, VS.NET, JBuilder, SAP NetWeaver DevStudio)

Logidex Product Overview

LogicLibrary Logidex

What it is:

- Catalog of software development asset (SDA) metadata

Why it’s different:

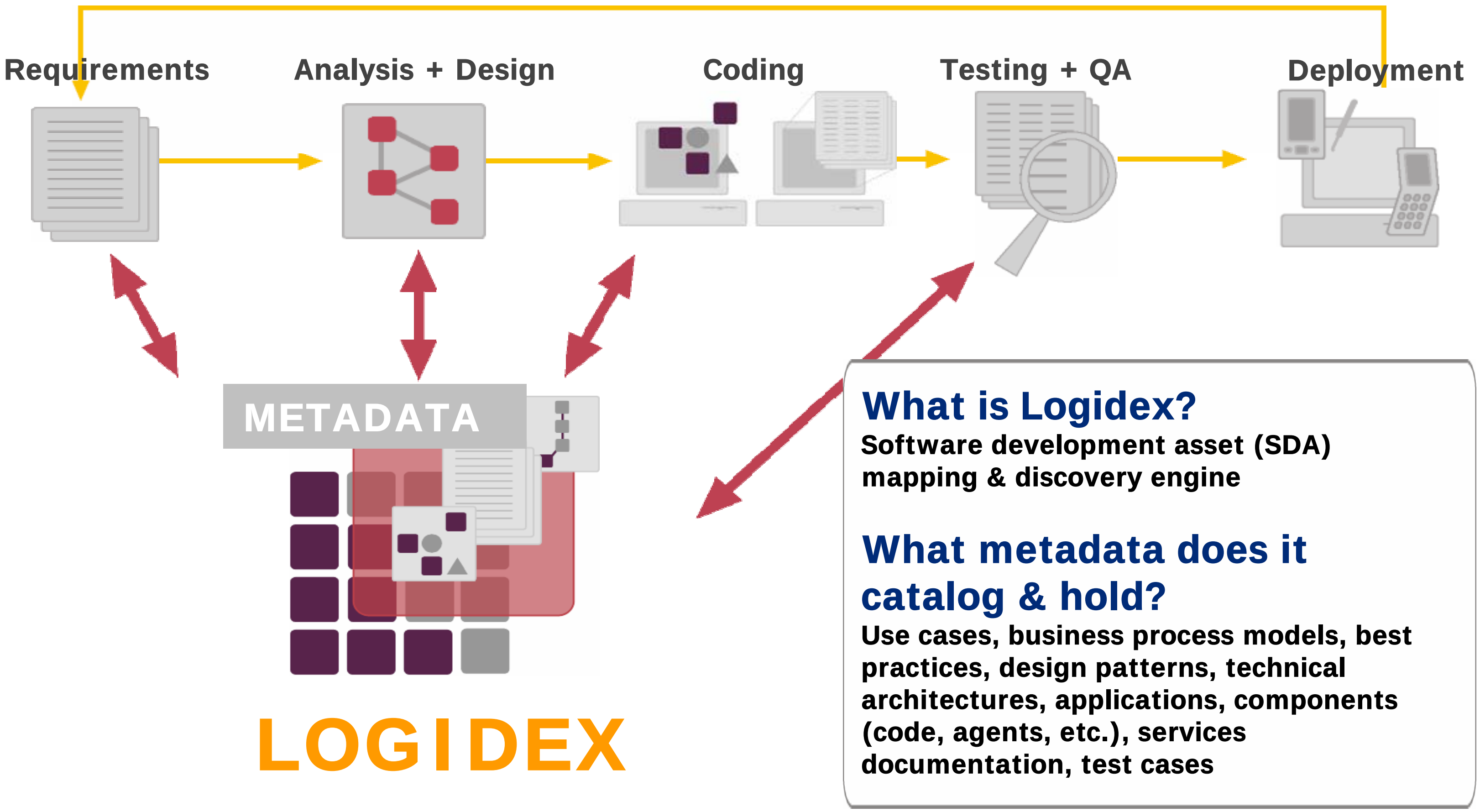
- Powerful mapping, discovery and collaboration engine for identifying and using SDAs in all types of software development and integration

Clear Value Proposition:

- Logidex enables faster and less costly consolidation, migration and/or integration of enterprise applications

Logidex Product Overview

Application Development Lifecycle: With Logidex



What is Logidex?

Software development asset (SDA) mapping & discovery engine

What metadata does it catalog & hold?

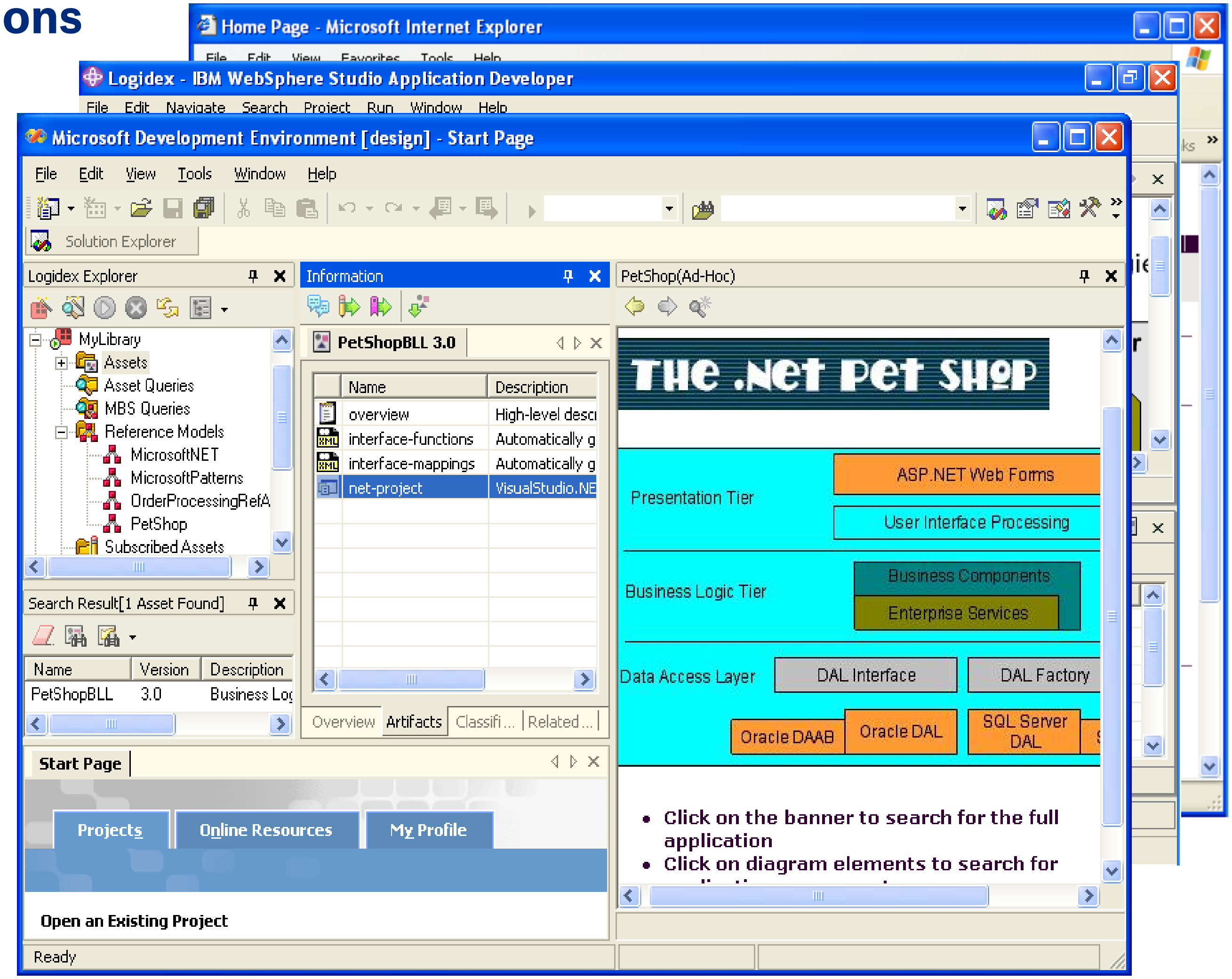
Use cases, business process models, best practices, design patterns, technical architectures, applications, components (code, agents, etc.), services documentation, test cases

Logidex Product Overview

User Interface Options

Web-Browser

Integrated with IDE (Eclipse, VS.NET, ...)



Name	Description
overview	High-level desc
interface-functions	Automatically g
interface-mappings	Automatically g
net-project	VisualStudio.NE

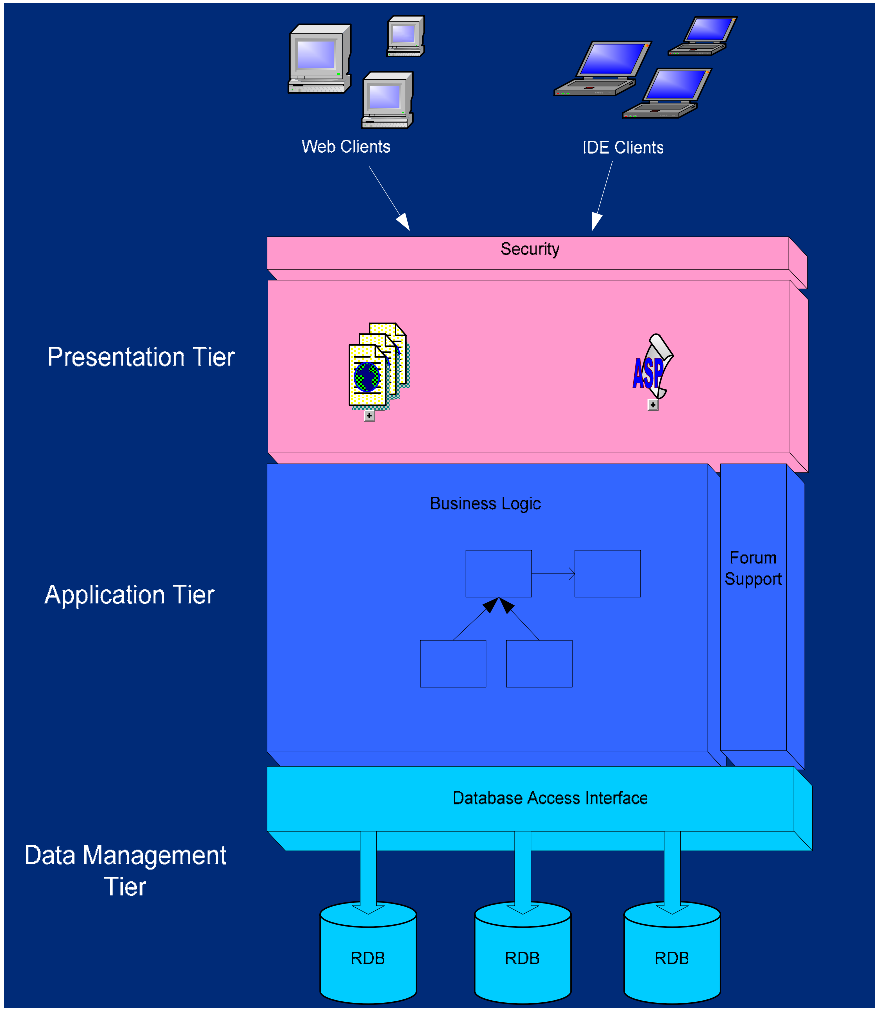
Search Result[1 Asset Found]

Name	Version	Description
PetShopBLL	3.0	Business Log

- Click on the banner to search for the full application
- Click on diagram elements to search for

Logidex Product Overview

Logidex J2EE Architecture



Presentation Tier

- Thin-client: JSPs/Servlets within Struts MVC framework
- Rich-client: IDE plug-in connected via WebServices

Application Tier

- Stateless EJB SessionBeans
- Java Entity classes and lightweight persistence framework

Data Tier

- Persistence framework connection to RDB via JDBC

Imbedded Components

- Discussion forums
- Authentication
- Email

Porting J2EE Solution to .NET

Porting Objectives

Primary objective**: deliver fully native Microsoft-based Logidex implementation**

- .NET-based server application
- SQL Server 2000-based RDB
- IE-based thin client
- Visual Studio add-in

Secondary objective**: preserve the code base as a single source to the extent possible**

Porting J2EE Solution to .NET

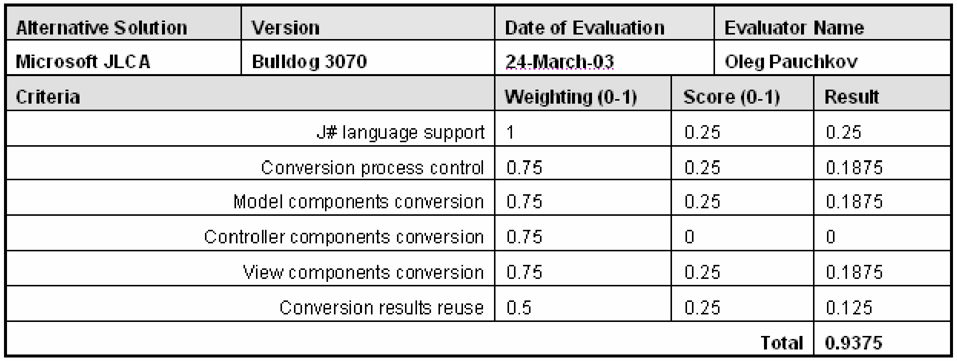
Porting Options

Option 1: Java Language Conversion Assistant (JLCA) tool

- Pro: semi-automated conversion tool
- Con: must be run on a regular basis to maintain product consistency

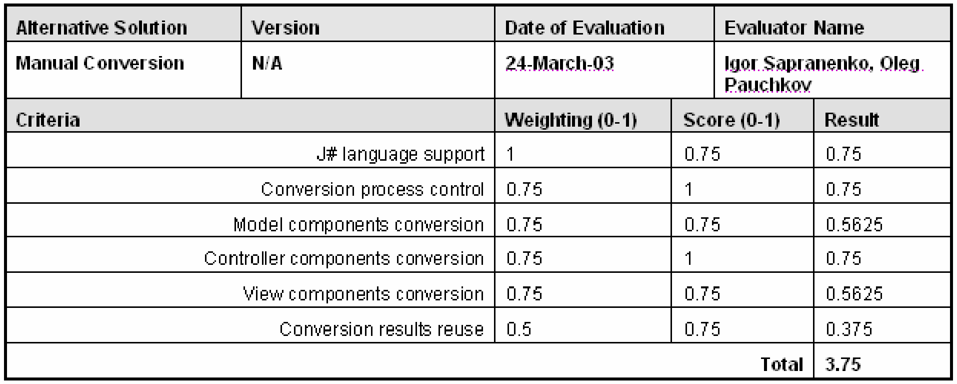
Option 2: J# on .NET

- Pro: single-source opportunity for major portions of the product
- Con: language compatibility restricted to Java 1.1.4 equivalency



Alternative Solution	Version	Date of Evaluation	Evaluator Name
Microsoft JLCA	Bulldog 3070	24-March-03	Oleg Pauchkov

Criteria	Weighting (0-1)	Score (0-1)	Result
J# language support	1	0.25	0.25
Conversion process control	0.75	0.25	0.1875
Model components conversion	0.75	0.25	0.1875
Controller components conversion	0.75	0	0
View components conversion	0.75	0.25	0.1875
Conversion results reuse	0.5	0.25	0.125
		Total	0.9375



Alternative Solution	Version	Date of Evaluation	Evaluator Name
Manual Conversion	N/A	24-March-03	Igor Sapranenko, Oleg Pauchkov

Criteria	Weighting (0-1)	Score (0-1)	Result
J# language support	1	0.75	0.75
Conversion process control	0.75	1	0.75
Model components conversion	0.75	0.75	0.5625
Controller components conversion	0.75	1	0.75
View components conversion	0.75	0.75	0.5625
Conversion results reuse	0.5	0.75	0.375
		Total	3.75

Our choice: J# on .NET

Porting J2EE Solution to .NET

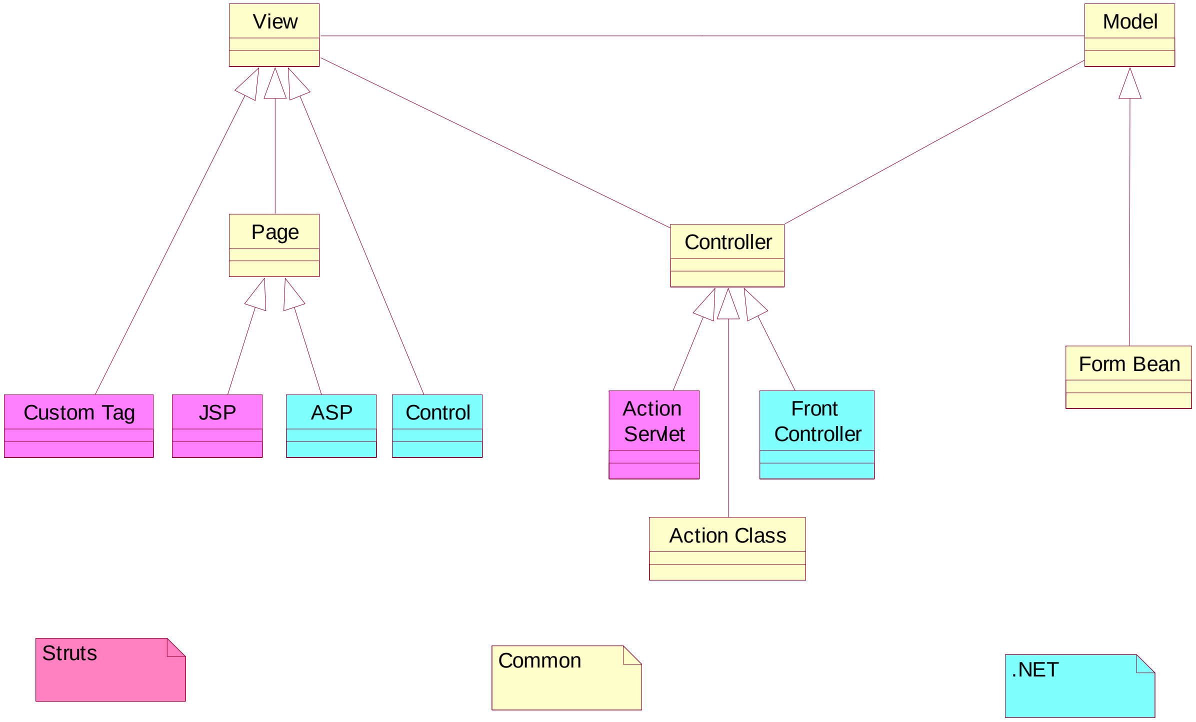
Porting Details

- Presentation Tier
- Application Tier
- Services
- Data Tier
- Porting Conclusions

Porting J2EE Solution to .NET: Presentation Tier

Porting Strategy:

- Implement Struts MVC pattern using .NET Framework
- Convert JSPs and Custom Tags to ASPs and .NET Controls
- Form Beans and Action Classes directly recompile as J#



Porting J2EE Solution to .NET: Presentation Tier

Converting JSPs and CustomTags to ASPs and Actions

- **Separating page presentation from page logic is key to enabling easy porting**
  - CustomTags contain page logic and are reimplemented as .NET Controls
- **JSPs then become mostly HTML-based and are easy to port**
  - Especially if page layout details are maintained via Cascading Style Sheets (CSS)

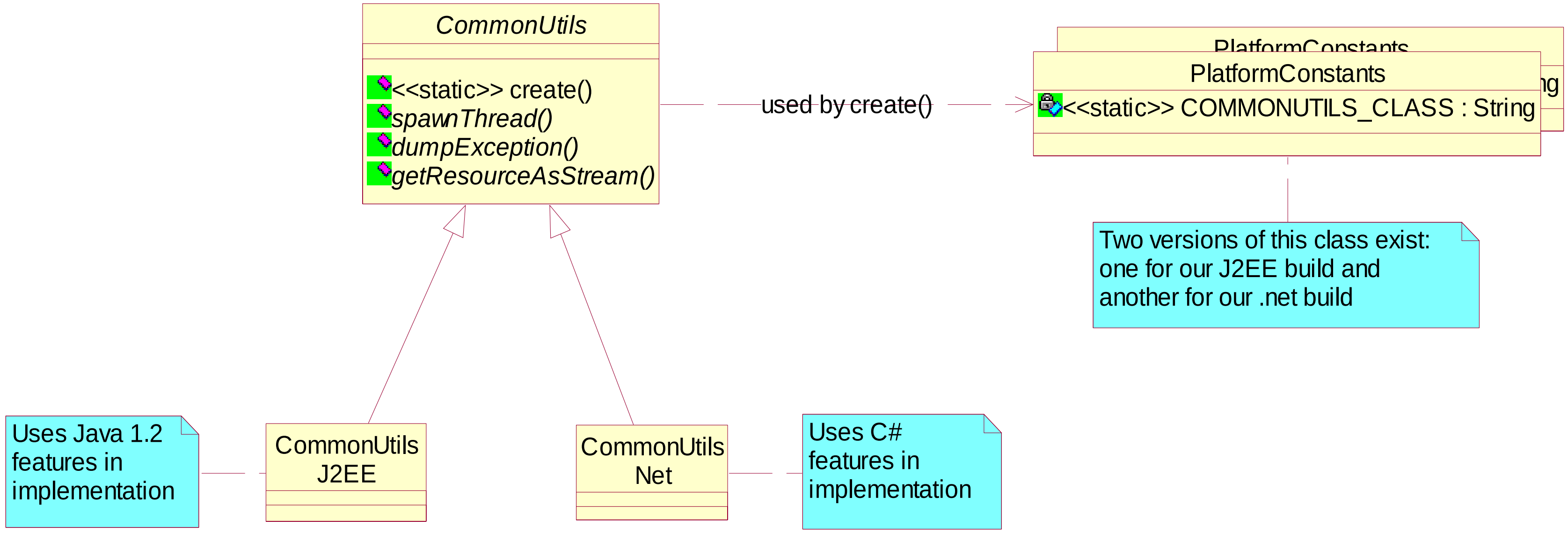
Porting J2EE Solution to .NET: Application Tier

Porting Strategy:

- **Step 1: Compile Java code as J# to identify unsupported features**
  - Very few specialized Java-package dependencies typically occur within application logic
- **Step 2: Where mandatory dependencies exist, implement Abstract Factory pattern to isolate Java code from common code base**
  - Suitable for both imbedded components and specialized Java packages
- **Step 3: Convert EJB SessionBeans to COM+ ServicedComponents**
  - COM+ ServicedComponents preserve transactional semantics supported by SessionBeans

Porting J2EE Solution to .NET: Application Tier

Abstract Factory Example: CommonUtils class:



CommonUtils

<<static>> create()
spawnThread()
dumpException()
getResourceAsStream()

used by create()

PlatformConstants

<<static>> COMMONUTILS_CLASS : String

Two versions of this class exist: one for our J2EE build and another for our .net build

Uses Java 1.2 features in implementation

CommonUtils J2EE

CommonUtils Net

Uses C# features in implementation

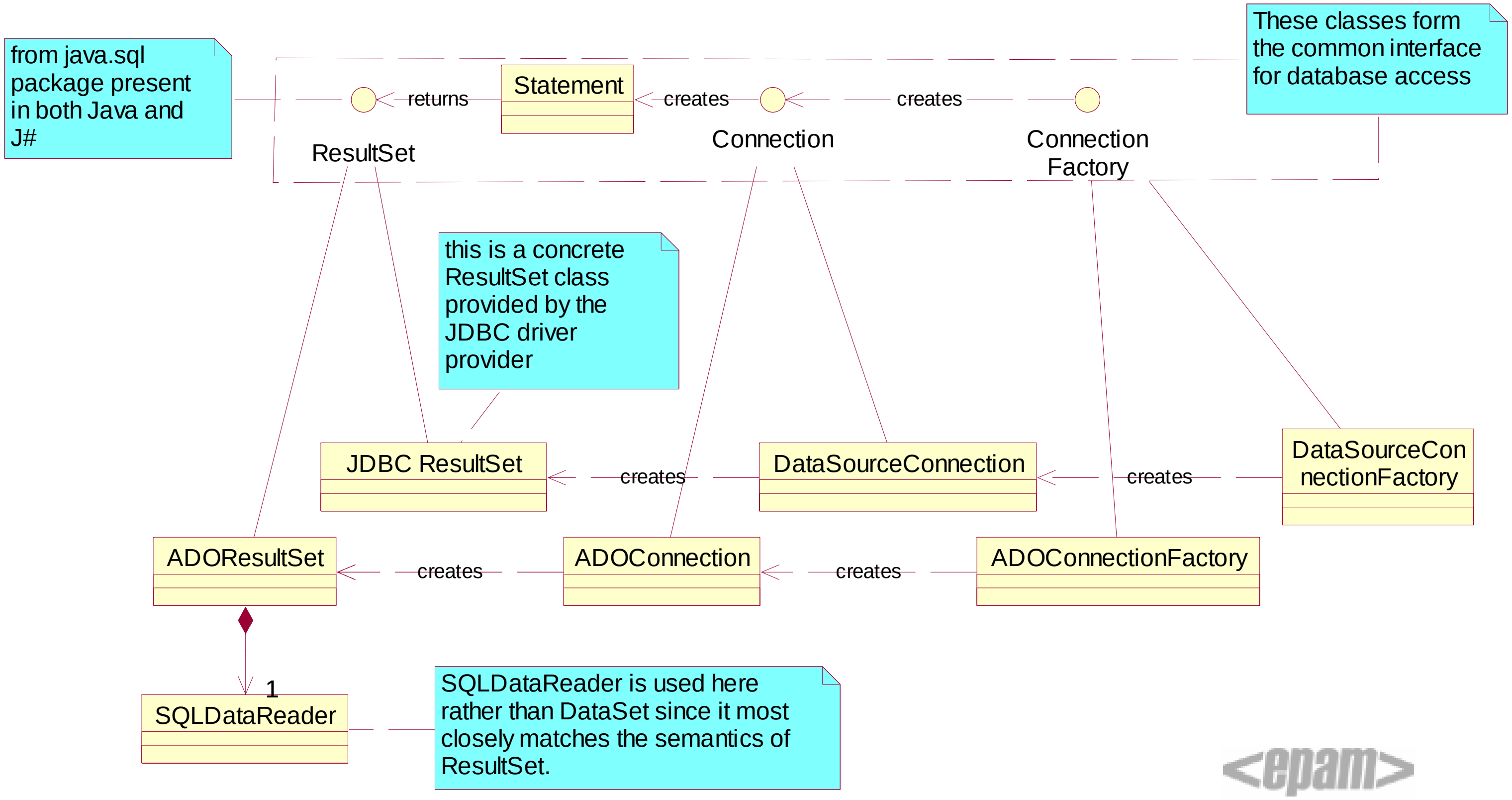
Porting J2EE Solution to .NET: Services

- Integration with IDEs is dependent upon SOAP APIs
- Logidex components are defined to expose service operations
- J2EE and .NET toolsets provide SOAP generators
  - Including strongly-typed client stubs
- Abstract Factory / Adapter patterns are used to provide single-source SOAP client classes
  - Encapsulating the tool-generated stubs
  - Enables some level of single-source code within IDE add-ins adapter implementations
- WS-I standard compliance is followed to reach cross-platform (J2EE <-> .NET) WebServices interoperability

Porting J2EE Solution to .NET: Data Tier

Porting Strategy:

- Define abstraction layer for RDB connections, statements, and result sets
- Use the Abstract Factory pattern (again!) to instantiate JDBC or ADO.NET- specific implementations



from java.sql package present in both Java and J#
These classes form the common interface for database access
Statement
returns
creates
creates
ResultSet
Connection
Connection Factory
this is a concrete ResultSet class provided by the JDBC driver provider
JDBC ResultSet
creates
DataSourceConnection
creates
DataSourceConnectionFactory
ADOResultSet
creates
ADOConnection
creates
ADOConnectionFactory
1
SQLDataReader
SQLDataReader is used here rather than DataSet since it most closely matches the semantics of ResultSet.

Porting J2EE Solution to .NET: Conclusions

Conclusion #1: Architecture is important!

- 3-tier architecture provides good isolation
- Component-based design supports loose coupling
- Web services provide a good encapsulation layer for client integration

Porting J2EE Solution to .NET: Conclusions

Conclusion #2: Patterns are important!

- **Model-View-Controller** pattern: isolates presentation from application code
  - Struts framework reimplemented in .NET technology
- **Abstract Factory** pattern: isolates dual-source code from single-source code base
- **Adapter** pattern: isolates platform-specific framework constructs from application code

Porting J2EE Solution to .NET: Conclusions

Conclusion #3: J# is a good choice to enable J2EE and .NET platforms support in parallel

Some more details in: .NET Developer's Journal

- Serving Multiple Platforms: Refactoring Logidex for .NET,
  Tim Graser and Paul Tamminga,

http://dotnet.sys-con.com/read/43660.htm

Microsoft-published Case Study for porting approach using J#

- http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=15272

Supporting .NET and J2EE in Parallel

Processes which are sensitive to bi-platform support

- Product planning
- Analysis & Design
- Implementation
- Building
- Testing
- Delivery & Support

Supporting .NET and J2EE in Parallel

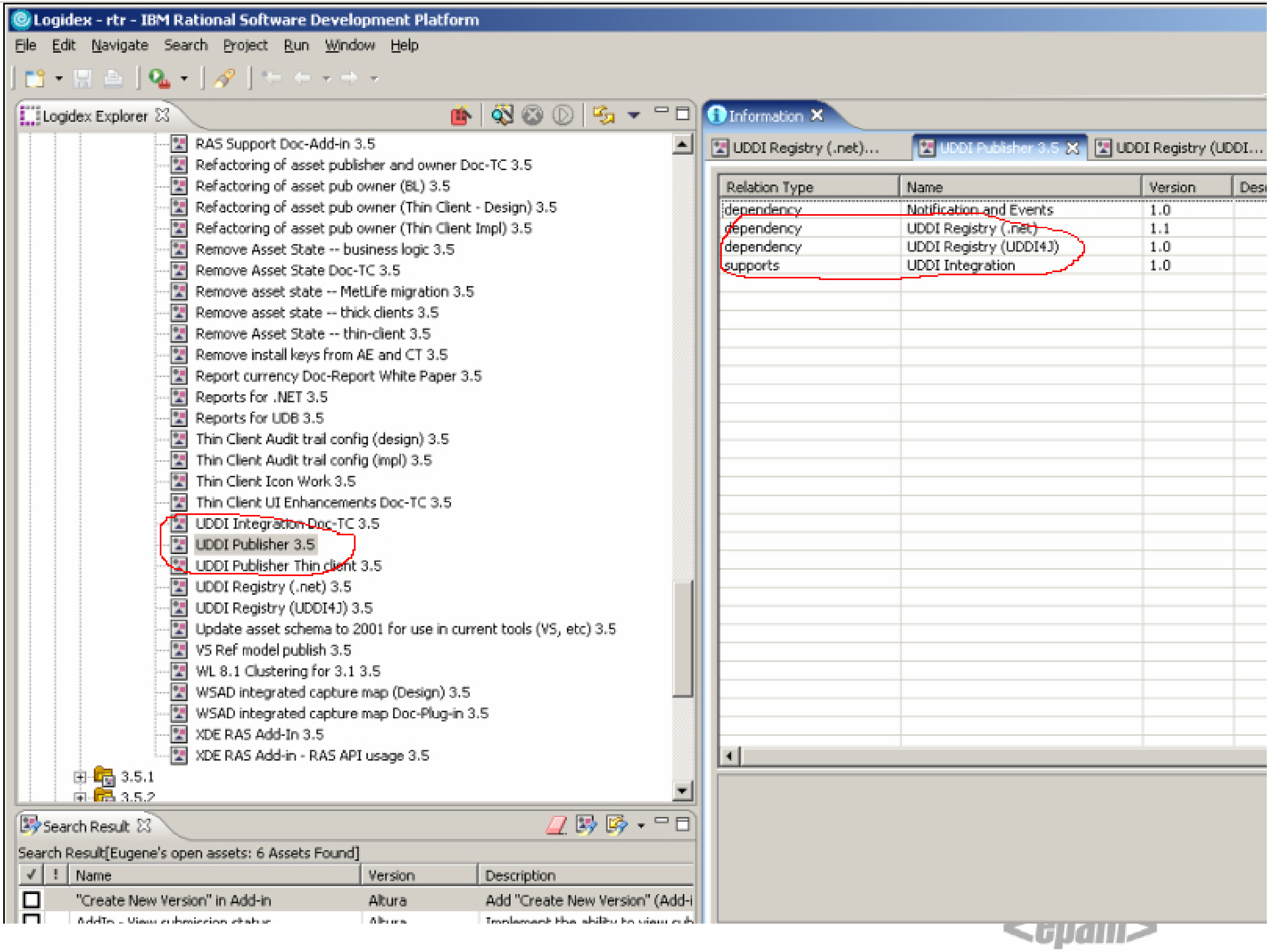
Logidex itself is used to support Logidex development

- An asset describes a product feature or a requirement
- Asset classifiers contain other important information: estimated effort, owners, target releases, etc.
- All related assets (blueprints, specs, findings) are stored within the same repository
- Asset relationships are used to show dependency and relationship information between features and/or requirements
- State of the art search sub system allows to find any asset quickly and easily

Supporting .NET and J2EE in Parallel

Proc

- Pa
- 2
- O
  im



- RAS Support Doc-Add-in 3.5
- Refactoring of asset publisher and owner Doc-TC 3.5
- Refactoring of asset pub owner (BL) 3.5
- Refactoring of asset pub owner (Thin Client - Design) 3.5
- Refactoring of asset pub owner (Thin Client Impl) 3.5
- Remove Asset State -- business logic 3.5
- Remove Asset State Doc-TC 3.5
- Remove asset state -- MetLife migration 3.5
- Remove asset state -- thick clients 3.5
- Remove Asset State -- thin-client 3.5
- Remove install keys from AE and CT 3.5
- Report currency Doc-Report White Paper 3.5
- Reports for .NET 3.5
- Reports for UDB 3.5
- Thin Client Audit trail config (design) 3.5
- Thin Client Audit trail config (impl) 3.5
- Thin Client Icon Work 3.5
- Thin Client UI Enhancements Doc-TC 3.5
- UDDI Integration Doc-TC 3.5
- UDDI Publisher 3.5
- UDDI Publisher Thin client 3.5
- UDDI Registry (.net) 3.5
- UDDI Registry (UDDI4J) 3.5
- Update asset schema to 2001 for use in current tools (VS, etc) 3.5
- VS Ref model publish 3.5
- WL 8.1 Clustering for 3.1 3.5
- WSAD integrated capture map (Design) 3.5
- WSAD integrated capture map Doc-Plug-in 3.5
- XDE RAS Add-In 3.5
- XDE RAS Add-in - RAS API usage 3.5

3.5.1

3.5.2

Relation Type	Name	Version	Desc
dependency	Notification and Events	1.0	
dependency	UDDI Registry (.net)	1.1	
dependency	UDDI Registry (UDDI4J)	1.0	
supports	UDDI Integration	1.0	

Search Result[Eugene's open assets: 6 Assets Found]

✓	!	Name	Version	Description
□		"Create New Version" in Add-in	Altura	Add "Create New Version" (Add-i
□		AddIn - View submission status	Altura	Implement the ability to view sub

Supporting .NET and J2EE in Parallel

Product building:

- Perforce is used as SCM tool
- Folders structure is designed in the way to make code management transparent for J2EE and .NET developers
- Automated build procedure allows the team to generate J2EE and .NET build packages taking common and platform specific sources

Product testing:

- UnitTests are reused for J2EE and .NET platform (JUnit, HttpUnit)
- Automated Functional and Performance testing scripts are re-used

Supporting .NET and J2EE in Parallel

Supported platforms:

J2EE

- **Application Servers**
  - BEA WebLogic
  - IBM WebSphere
- **Operating Systems**
  - IBM AIX
  - Sun Solaris
  - Linux
- **RDB**
  - Oracle
  - IBM UDB

.NET

- **Application Servers**
  - IIS
  - .NET f/w 1.0, 1.1, 2.0
- **Operating Systems**
  - MS Windows 2000
  - MS Windows 2003
- **RDB**
  - MS SQL Server 2000
  - Oracle

IDE

- Eclipse
- IBM WSAD
- IBM RAD
- Borland JBuilder
- MS Visual Studio .NET
- SAP NetWeaver DS

Results

.NET and J2EE versions of Logidex share over 80% of the common code base

- Validated the core Logidex architecture
- Allows a small development team to build product versions for both platforms in parallel

You can try Logidex at http://lab.msdn.microsoft.com/logidex

- Preloaded PAG patterns and Application Blocks
- Graphical search integrated into Visual Studio

Delivering Excellence in Software Engineering

Development of Software Products with Simultaneous Support of J2EE and .NET

For more information, please contact:
Aliaksei Kazlouski
Senior Software Engineering Manager
EPAM Systems, Inc.
29, Khoruzhey str.
Minsk, Belarus
Phone: +375-17-210-1662
Fax: +375-17-210-1168
http://www.epam.com