

# Solutions Development: Trends, Technology, and Tooling

*“Nobody in football should be called a genius. A genius is a guy like Norman Einstein.”*

Football commentator and former player Joe Theismann

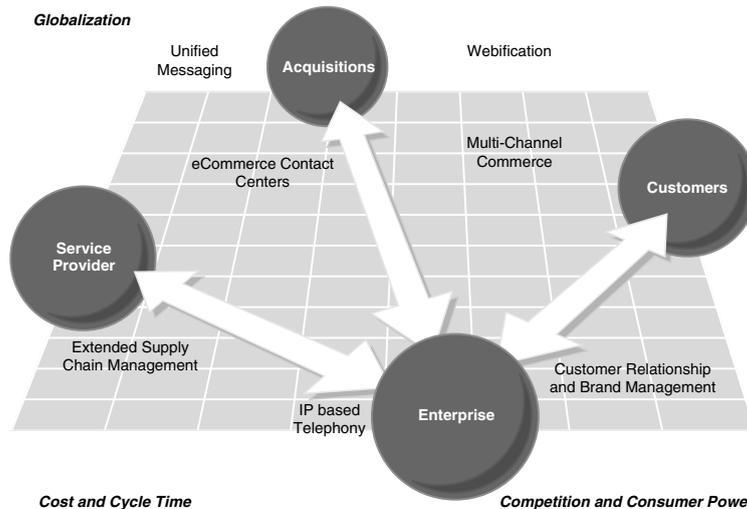
## 3.1 Trends

When we originally conceived this book, this chapter was going to be about integrated development environments (IDEs) and other development tooling as they relate to DB2. Along the way we realized that this approach was far too shortsighted. After all, this book is about integration. So in this chapter we decided to put the integrated business solutions environment into a larger context.

Today’s integrated business solutions environment must address distributed systems that are decidedly more complex than those typically found a decade ago. Many businesses have undergone globalization fueled in part by Web technology that has broken down traditional barriers to commerce. Some businesses have gained competitive advantage by leveraging distributed solutions technology to quickly integrate newly acquired assets, to reach customers faster, and to incorporate new searches available through the Web. These distributed systems (see Figure 3.1) can be characterized as *time-driven* “application islands” that often require Web-based enablement (“webification”).

The business challenges presented by complex, distributed environments are daunting and include a number of costs:

- *Integration costs* required to seamlessly integrate new business units (e.g., software application packages) with longstanding legacy systems



**Figure 3.1** Distributed Systems Characterized by Application Islands and Web-Based (“Webification”) Enablement

- *Systems management costs* associated with managing disparate systems and heterogeneous islands
- *Compliance costs* involved with government regulations and industry standards
- *Market development costs* driven by the need to respond quickly and effectively in new markets

Of the four types, integration costs are greatest. To remain competitive, information technology (IT) executives are spending large portions of their budgets on integrating systems and associated processes to address challenging and often complex environments.

To understand and develop solutions in this environment we don’t need to be geniuses. We need development tools and technologies that are integrated with the software that is part and parcel of our solutions environment. We also need decomposition methods to break down the complex environment into chunks that can be easily programmed using our tools.

The solution to this problem has been the introduction of *integrated* solutions development environments consisting of both general purpose IDEs and specialized development tools. As we will see throughout this book, integration problems come in different shapes and sizes. Choosing the right tool for the task at hand is critical to success.

Along with the development tooling, an IT architecture methodology is required to reduce complexity and to encourage reuse. One representative methodology consists of the “IBM Patterns for e-Business.” The patterns are a group of reusable assets that can be used to accelerate the process of developing integrated, Web-enabled solutions. These reusable assets may take on many forms.

- *Business patterns* identify the interactions among users, businesses, and data. Business patterns are used to create simple, end-to-end, solution-oriented business applications.
- *Integration patterns* connect other business patterns to create applications with advanced functionality.
- *Composite patterns* are combinations of business patterns and integration patterns that have themselves become commonly used types of business applications. Composite patterns are advanced business applications.
- *Custom designs* are similar to composite patterns in that they combine business and integration patterns to form an advanced, end-to-end solution. However, custom designs are not implemented to the extent of composite patterns but are developed to solve the business problems of one specific company, or perhaps several enterprises with similar problems.
- *Application and runtime patterns* are driven by the customer's requirements and the shape of applications and the supporting runtimes needed to build the business application.
- *Product mappings* inhabit the solution. Throughout this chapter we introduce representative products that may be used for solutions development.
- *Guidelines* are concepts for the design, development, deployment, and management of solution applications.

System architects and solution providers can use business patterns to reduce complex business environments into simple architectural components. Detailed exploration of the pattern methodology is left to the reader.<sup>1</sup>

We use selected pattern examples to help explain integration in its various forms. We also discuss the kinds of tools (including a generic IDE and many specialized tools) that are needed to support complex, integrated solution development environments.

## 3.2 Integration

What does integration really mean? Does it mean integration of IT infrastructure,<sup>2</sup> applications, systems, or platforms? In this chapter we introduce four categories of integration and relate them to relational database systems. Complex, distributed integration environments require one or more of the integration categories presented in Table 3.1. Each category has a perspective and an integration level associated with it that determine development paradigm and focus. We've also listed usage examples for each category as well as tooling examples and typical tooling users.

---

1. To learn more about the patterns, see [www-106.ibm.com/developerworks/patterns/](http://www-106.ibm.com/developerworks/patterns/).

2. Aberdeen Group, "e-Business Infrastructure Integration: Practical Approaches," November 2001, [www-3.ibm.com/e-business/doc/content/feature/offers/infraintegration.pdf](http://www-3.ibm.com/e-business/doc/content/feature/offers/infraintegration.pdf).

**Table 3.1** Integration Categories

Category	Perspective	Integration Level	Usage Example	Tooling Example	Tooling Users
Data integration	Functional viewpoint	Data	Directory synchronization, data synchronization, <sup>3</sup> scalar and table user-defined functions (UDFs)	IBM Directory Integrator, <sup>4</sup> DB2 Develop. Center	DBA developers, systems administrators
Application integration	Functional viewpoint	Application	Creation of stored procedures and UDFs, <sup>5</sup> generic integration of DB2 <sup>6,7</sup>	DB2 Develop. Center, WebSphere Studio Application Developer <sup>8</sup>	DBA developers
Business process integration	Business viewpoint	Business process	Model, analyze, monitor, and optimize business processes <sup>9,10</sup>	WebSphere Business Integration Family <sup>11-15</sup>	Business analysts, developers
Information integration	Business intelligence viewpoint	Information	Integration of diverse forms of data <sup>16</sup>	DataProgorator <sup>17</sup> DataJoiner, <sup>18</sup> Enterprise Information Portal <sup>19</sup>	Information analysts, developers

3. R. Cutlip and N. Joshi, "Supreme Integration Solutions with IBM Directory Integrator," [www-106.ibm.com/developerworks/library/i-directint/](http://www-106.ibm.com/developerworks/library/i-directint/).

4. IBM Directory Integrator, [www-3.ibm.com/software/tivoli/products/directory-integrator/](http://www-3.ibm.com/software/tivoli/products/directory-integrator/).

5. A. Al-Azzawe, "DB2 Development Center: The Next Generation AD Tooling for DB2," [www7b.software.ibm.com/dmdd/library/techarticle/0207alazzawe/0207alazzawe.html](http://www7b.software.ibm.com/dmdd/library/techarticle/0207alazzawe/0207alazzawe.html).

6. Download: Web Services Consumer, WSAD plug-in to extend ability of DB2 to invoke Web services from Structured Query Language (SQL), [www7b.software.ibm.com/dmdd/zones/webservices/wsconsumer/index.html](http://www7b.software.ibm.com/dmdd/zones/webservices/wsconsumer/index.html).

7. K. S. Bhogal, J. Puli, and C. Laxminarayan, "Writing a Web-Based Table Editor GUI for DB2," [www7b.software.ibm.com/dmdd/library/techarticle/0211bhopuli/0211bhopuli.html](http://www7b.software.ibm.com/dmdd/library/techarticle/0211bhopuli/0211bhopuli.html).

8. WebSphere Software Platform, [www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/studio&S\\_TACT=102BBW01&S\\_CMP=campaign](http://www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/studio&S_TACT=102BBW01&S_CMP=campaign).

9. IBM CrossWorlds Success Story, [www-3.ibm.com/software/integration/cw/success/multinational.html](http://www-3.ibm.com/software/integration/cw/success/multinational.html).

10. WebSphere Business Integration Solution for E2open, [www-3.ibm.com/software/success/cssdb.nsf/CS/MMAA-5DNL7F?OpenDocument&Site=default](http://www-3.ibm.com/software/success/cssdb.nsf/CS/MMAA-5DNL7F?OpenDocument&Site=default).

11. WebSphere Business Integration Server, [www-3.ibm.com/software/ts/wbi/](http://www-3.ibm.com/software/ts/wbi/).

12. WebSphere Business Integration Adapters, [www-3.ibm.com/software/integration/wbiadapters/](http://www-3.ibm.com/software/integration/wbiadapters/).

13. WebSphere MQ Workflow, [www-3.ibm.com/software/ts/mqseries/workflow/](http://www-3.ibm.com/software/ts/mqseries/workflow/).

14. WebSphere Business Integration Modeler, [www-3.ibm.com/software/integration/holosofx/](http://www-3.ibm.com/software/integration/holosofx/).

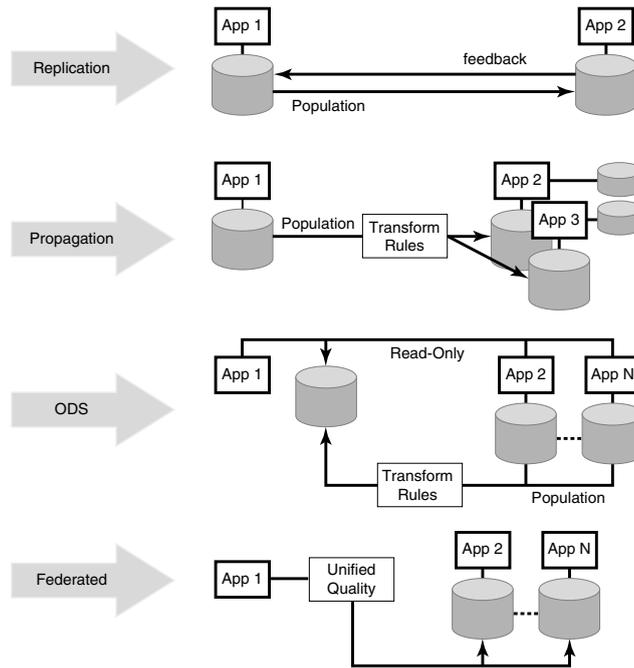
15. Rational Zone: Technical Resources for using Rational and WebSphere Software, [www7b.software.ibm.com/wsdd/zones/rational/](http://www7b.software.ibm.com/wsdd/zones/rational/).

16. The Principal Financial Group Creates Mutually Satisfying CRM System, [www-3.ibm.com/software/success/cssdb.nsf/CS/NAVO-4VPVEK?OpenDocument&Site=software](http://www-3.ibm.com/software/success/cssdb.nsf/CS/NAVO-4VPVEK?OpenDocument&Site=software).

17. DB2 Universal Database for Linux, UNIX and Windows, IBM DataPropagator, [www-3.ibm.com/software/data/dpropr/index.html](http://www-3.ibm.com/software/data/dpropr/index.html).

18. IBM DataJoiner, [www-3.ibm.com/software/data/datajoiner/](http://www-3.ibm.com/software/data/datajoiner/).

19. DB2 Information Integration for Content (formerly Enterprise Information Portal), [www-3.ibm.com/software/data/eip/](http://www-3.ibm.com/software/data/eip/).



**Figure 3.2** Data Integration Patterns: Replication, Propagation, Operational Data Store (ODS), and Federated Flows

### 3.2.1 Data Integration

Data integration is what first comes to our minds when we think of integration in terms of database systems and their contents. Here we broadly define data integration as the circular intermingling of database systems and their contents regardless of location or platform. Typically, data integration occurs *within* an enterprise, where we have the maximum amount of control. Using the e-business architectural patterns approach we can deconstruct data integration into its elemental forms, as shown in Figure 3.2.

Data integration may take several forms.

- *Data replication flows* deal with the interchange of distributed data within a homogeneous environment.
- *Data propagation flows* deal with the interchange of distributed data within heterogeneous environments necessitating data transformation.
- *ODS (operational data store) flows* use functions that access or read-only extracts of operational data stores.
- *Federated flows* typically employ *database middleware* to provide access to heterogeneous, distributed data. The target data sources may include structured as well as unstructured data.

Often replication and propagation can be accomplished with the same simple tooling. Whatever tool is used, it needs to include data movement and transformation capabilities, and perhaps the ability to construct multitarget data joins suitable for SQL calls. The tool of choice for DB2 is DataPropagator, which provides high-performance data movement between transactional data stores. In our Pervasive Solution presented in Chapter 7, we chose DataPropagator to synchronize data between our local and remote databases. Since both the tool and the approach have been around for quite some time we don't address them in detail. For those interested in learning more about replication strategies for pervasive solutions we suggest you consult "IBM Replication Solutions for Pervasive Computing with DB2 Everyplace and DB2 Satellite Edition."<sup>20</sup>

DataPropagator is delivered as a component of DB2 Universal Database (UDB) along with a companion product DB2 DataJoiner<sup>21</sup> on UNIX and Windows. DB2 DataJoiner facilitates SQL access and table joins across multiple data sources, relational and nonrelational regardless of location (i.e., federated data access).

Federation has become an increasingly common approach to data integration. Its popularity has fueled the emergence of "loosely coupled" integration technologies in the form of Web services and message-oriented middleware (MOM) as well as proliferation of data sources within and beyond the enterprise. Strictly defined, database federation is an approach to data integration where *database middleware* provides uniform access to a number of heterogeneous database systems. It may be used in conjunction with replication or synchronization but is not dependent on either and in many cases may obviate the movement of data.

Data integration through federation has the following advantages:

- It can access multiple data sources both within and beyond the enterprise, where moving data is problematic.
- It can leverage loose coupling technologies of MOM as well as open technologies such as Web services.
- It is dynamic, responding to environmental changes.
- Tooling is available to simplify the process of data integration through federation.

Federation can be divided into three separate categories<sup>22</sup> based on level of functional complexity:

- *Scalar UDFs* produce a single-valued (scalar) result. A prime example is the *db2mg.mqsend()*<sup>23</sup> UDF shipped with DB2, which permits exchange of data between DB2 and the WebSphere MQs.

---

20. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246217.pdf>.

21. For more information on DataJoiner see [www-3.ibm.com/software/data/datajoiner/](http://www-3.ibm.com/software/data/datajoiner/).

22. L. M. Hass, E. T. Lin, and M. A. Roth, "Data Integration through Database Federation," *IBM Systems Journal*, vol. 41, no. 4, 2002.

23. For more on WebSphere MQ Series consult Chapters 1 and 4.

- *Table UDFs* produce a table as output and may appear wherever a table can be referenced in SQL. This permits application of SQL to a much larger set of data than simply tables and views.
- *Wrappers* constitute the third class of federation style and generally involve an entire data source or a large subset of data. Wrappers may also integrate both the function (application) and the data into a single unit of federation. In these cases the line between data integration and application integration begins to blur.

A common method of creating wrappers is through the use of Web service technologies for federation within and beyond the enterprise. For example, the use of an XML wrapper can facilitate the integration of XML data sources into DB2<sup>24</sup> for integration through data movement as well as federated access. DB2 version 8 Federated Web Services access provides a cost-effective wrapper approach for data integration through federation. Using a single SQL query, DB2 can access information provided by Web services without requiring application generation. The approach is built upon the DB2 federated database infrastructure, which is often leveraged for Web-based application integration.<sup>25</sup>

Another technology used in support federated data access through wrappers is MOM. IBM WebSphere MQ is a MOM technology that when used in conjunction with DB2 is frequently applied to application integration.<sup>26,27</sup>

It can also be used solely as an integration layer for federated data integration. It provides a number of functions to allow SQL to include messaging operations. It further allows message operations and database operations to be combined into a single unit of work.

Because the use of wrappers involves entire data sources it can allow for the following types of integration.

- Multiple server integration
- Multiple data set integration
- Multiple operation integration

More advanced forms of wrappers—that is, those closely bound to the relational database engine—also allow for transactional integrity and query optimization.

---

24. For tooling to support integration of XML source data into DB2 see “XML Wrapper Generator,” IBM alphaWorks, [www.alphaworks.ibm.com/tech/wrappergenerator](http://www.alphaworks.ibm.com/tech/wrappergenerator).

25. C. M. Saracco, “Building Web Components that Access Federated Data,” [www7b.software.ibm.com/dmdd/library/techarticle/0209saracco/0209saracco.html](http://www7b.software.ibm.com/dmdd/library/techarticle/0209saracco/0209saracco.html).

26. D. Wolfson, “Using MQSeries from DB2 Applications,” [www7b.software.ibm.com/dmdd/library/techarticle/wolfson/0108wolfson.html](http://www7b.software.ibm.com/dmdd/library/techarticle/wolfson/0108wolfson.html).

27. M. Tong and D. Wolfson, “DB2 MQ XML Functions: Using MQSeries and XML Extender from DB2 Applications,” [www7b.software.ibm.com/dmdd/library/techarticle/wolfson/0201wolfson.html](http://www7b.software.ibm.com/dmdd/library/techarticle/wolfson/0201wolfson.html).

Data integration development is often performed by a wide variety of personnel, including database administrators, system and application developers, and administrative personnel. What form or level of data integration they choose ultimately drives the choice of tooling. If we think of data integration in terms of federation, UDFs are limited to scalar or tabular results and do so with transactional support. In these cases development of UDFs as well as stored procedure to facilitate data integration can be quickly accomplished using the DB2 Development Center.<sup>28</sup> The Development Center is the first DB2-specific graphic user interface (GUI) tool that supports the development of UDFs. For the purposes of federated data integration, the DB2 Development Center supports both SQL scalar and table functions on UNIX, Linux, and Windows platforms.

There are a number of tools available that can effect data integration, including federated data access through wrappers. These tools are characterized by easy-to-grasp development paradigms, a rich set of predefined connectors, and very simple, often “linear” execution models. IBM Directory Integrator is a prime example of this class of integration development tools.

Directory Integrator<sup>29</sup> is a development tool whose primary focus is on delivery of enterprise directory solutions. Although it had its technological beginning in addressing synchronization among identity directories such as Lightweight Directory Access Protocol (LDAP), it can also be used to effect data synchronization (replication or propagation) among multiple data stores including DB2. It can also be used to implement flows to operational data stores (ODS).<sup>30</sup>

The Directory Integrator development paradigm manifests itself in *AssemblyLines*. The AssemblyLine describes the origin, transformation, and destination of data in an easily comprehensible way. Integration elements within the Directory Integrator development paradigm include data sources, data flows, events, and attribute mapping transformation and are used to create the assemblies, or chains.

- Data Sources represent a wide variety of systems, data sources, and repositories including DB2, IBM Secureway Directory, XML files, Simple Object Access Protocol (SOAP) documents, and email.
- Events are usually associated with how data sources interact. When a data flow is triggered by a conditional change, events are invoked.
- Data flows represent the unique message being passed from one data source to another.
- Attribute mapping with transformation involves determining which field or attributes are to be passed by the data flow to the data source. The representation of the information is also determined.
- Connectors along the AssemblyLine execution flow provide bidirectional access to a number of devices, systems, applications, and platforms.

---

28. A. Al-Azzawe, “DB2 Development Center: The Next-Generation AD Tooling for DB2,” [www7b.software.ibm.com/dmdd/library/techarticle/0207alazzawe/0207alazzawe.html](http://www7b.software.ibm.com/dmdd/library/techarticle/0207alazzawe/0207alazzawe.html).

29. IBM Directory Integrator, [www-3.ibm.com/software/tivoli/products/directory-integrator/](http://www-3.ibm.com/software/tivoli/products/directory-integrator/).

30. R. Cutlip and N. Joshi, “Supreme Integration Solutions with IBM Directory Integrator,” [www-106.ibm.com/developerworks/library/i-directint](http://www-106.ibm.com/developerworks/library/i-directint).

- Parsers facilitate the interpretation and translation of byte stream data into structured information objects.
- Hooks provide the ability to execute actions under very specific conditions or at specific points along the AssemblyLine.

Once an AssemblyLine is constructed with these integration elements it can be automatically triggered. Event handlers detect predefined event conditions and apply the Directory Integrator execution model [Event + Condition → Action]. On invocation, event parsers may provide interpretation and transformation functions to the defined data flow. Prebuilt connectors delivered with Directory Integrator provide access to a number of well-known data sources including DB2, LDAP, and Microsoft Active Directory. The appeal of this kind of tool lies in its simple development paradigm and straightforward execution model. The AssemblyLines are based on a linear flow construction of subcomponents that facilitates incremental construction and test. With this kind of tool we have a form of data integration based on wrapper federation. In terms of federation, the AssemblyLine is the wrapper that can access a number of heterogeneous data sources. Scripting languages such as Java and JavaScript can be used to code and enforce transactional control in the form of business rules.

Data integration is focused on integrating database systems and their content, yet all of the development tools discussed here may also be used for application integration.

### 3.2.2 Application Integration

Application integration is what most people think of when they refer to integration. In the context of relational database systems, *application integration* allows applications that perform similar or complimentary functions to communicate with one another. These applications often access data sources that have or are undergoing synchronization, replication, or federation. As we've already pointed out, much of the tooling that applies to data integration (including the DB2 Development Center with its ability to generate UDFs and stored procedures) is used for application integration. Integration technologies including XML, messaging, and Web services also equally apply to application integration.

The characteristics of application integration are as follows:

- Application-to-application or application-to-data integration
- Application types:
  - Potentially complex, including transactional logic and perhaps requiring considerable custom programming efforts
  - Frequently intertwined with enabling technologies for distributed solutions (XML, messaging, Web services), including an application Web server
- Developer types: Tend to be general application programmers who may be less familiar with data technology than those involved with data integration
- Tooling types: May be specialized database application tools but tend to be sophisticated IDEs

So we conclude that application integration environments can be quite complex, with a myriad of technologies and systems involved. The developers of these systems may not be as familiar with database technology and how to efficiently exploit it as database professionals involved with data integration.

The best approach to application integration from a development perspective, particularly where the applications interface with database systems, is the use of a sophisticated, open, and extensible IDE. With this approach, every developer isn't forced to become a database expert, since the technologies used for sophisticated integration with database systems are fully integrated and available within the IDE.

The capabilities and functions required in an IDE include the following:

- A flexible development paradigm to support a number of different types of development personnel
- Simplified Java and Java 2 Enterprise Edition (J2EE) development (Enterprise JavaBeans [EJBs], servlets, JavaServer Pages [JSPs])
- An extensible development platform to support ISV (independent software vendor) plug-ins
- An integrated environment that can exploit the use of database-centric integration technologies such as DADX-based Web services<sup>31</sup> and SQL Java [SQLJ]<sup>32</sup>

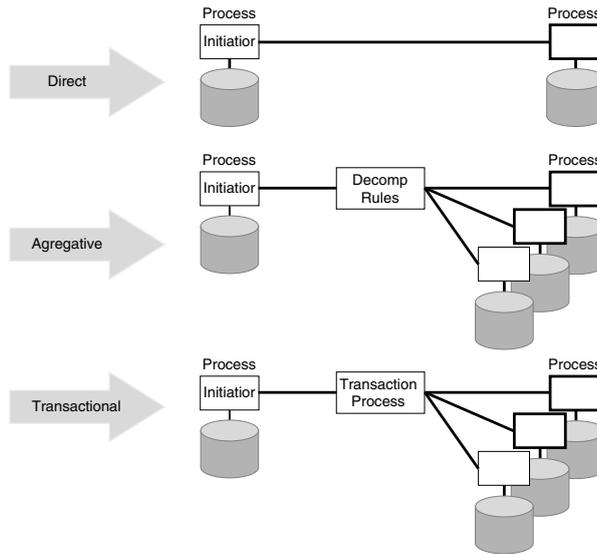
Although a sophisticated IDE is a “must have,” it alone is not enough to ensure successful development of distributed solutions. A best practice approach from an information technology (IT) architecture perspective is to provide tooling and a methodology that can reduce the complexity of both the application and data flows. We need to foster a common, systematic understanding of the deconstructed data and application flows that constitute our distributed solutions environment. Otherwise we risk miscommunication between the development, architecture, and management teams, which will result in disjointed and ineffective development and deployment.

Before we address these issues we want to briefly revisit the concept of patterns that we introduced at the beginning of this chapter. In terms of application integration, the types of patterns commonly encountered are represented in Figures 3-3 and 3-4. The simplest processes are the “direct” processes—that is, they interact directly with one another. With aggregative processes, processes from multiple providers are consolidated through the use of decomposition rules to satisfy the initiating application request. For transactional application process flows, a transactional process such as two-phase commit is used to ensure integrity of the interaction.

Figure 3.4 shows two processes commonly associated with application integration—brokered processes and managed processes. Brokered processes rely on durable messaging for reliable execution across complex environments. Leveraging messaging technology (e.g., MOM), brokered processing implements a loose coupling approach to application integration. The sec-

31. F. Chen, “Automating Web Service Creation,” *DB2 Magazine* (WSAD and DADX files), no. 2, 2002.

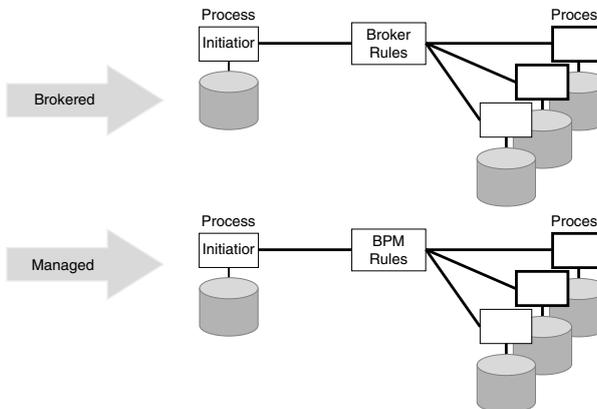
32. J. Fung and B. Searle, “SQLJ Tool Plug-in for WebSphere Studio Application Developer,” [www7b.software.ibm.com/wsdd/library/techarticles/0210\\_fung/fung.html](http://www7b.software.ibm.com/wsdd/library/techarticles/0210_fung/fung.html).



**Figure 3.3** Application Integration Patterns: Direct, Agregative, and Transactional Processes

ond flow represents managed processes, which are extensions of both the brokered and transactional patterns. Business process management (BPM) flows are characterized by persistence of execution and complex interactions.

Deconstruction of applications into the integration patterns shown in the application flows of Figures 3-3 and 3-4 is the first step in the patterns methodology. It provides a common understanding that leads to guidelines for the design, development, deployment, and management of applications.



**Figure 3.4** Application Integration Patterns: Brokered and Managed Processes

In the next section we introduce a representative IDE in the form of WebSphere Studio Application Developer (WSAD).<sup>33</sup> It has the features and functions needed to build integrated, distributed application and data integration flows.

### 3.2.2.1 Development Tooling in Application Integration

One can certainly write a Java application, any Java application, with nothing more than a text editor. WSAD is easy to use because it fully supports the J2EE component model, facilitating construction of complex yet portable applications, and also because it provides tooling for technologies important to business development (e.g., XML and Web services) as well as programming elements traditionally found in the enterprise environment.

Given the dynamic nature of the business environment the IDE also needs to be flexible, open, and extensible. It needs to grow as technologies grow. It needs to offer some level of integration with deployment environments. It also needs to support those functions we tend to take for granted in an IDE, such as team development and version control.

As we will discover in Chapter 7, the IDE also incorporates the functions of Web Services Object Runtime Framework (WORF) used to provide Web services access to DB2. WSAD contains a J2EE-compliant, single-server version of WebSphere Application Server that is primed and ready for unit testing of newly created applications.

WSAD is built on an open source framework (i.e., Eclipse-compatible tooling)<sup>34</sup> and includes application program interfaces (APIs) that permit other vendors to plug-in their tools. This translates into the kind of flexibility and extensibility required by the ever-changing business environment. The basis of this open source integration technology is the WebSphere Studio Workbench, which is the foundation of WSAD. The workbench provides APIs, frameworks, and other building blocks that allow independent software vendors to write tools as plug-ins to the workbench.

Figure 3.5 shows the basic architecture of the workbench. At the platform core are common services such as workplace synchronization and debugging. Also provided within the core are common frameworks for user interface (UI) frameworks, widget toolkits, and source editing. Below the core is a layer for configuration and version management that interacts with a number of repository technologies, including concurrent version system (CVS). CVS is an open standard for version control systems.

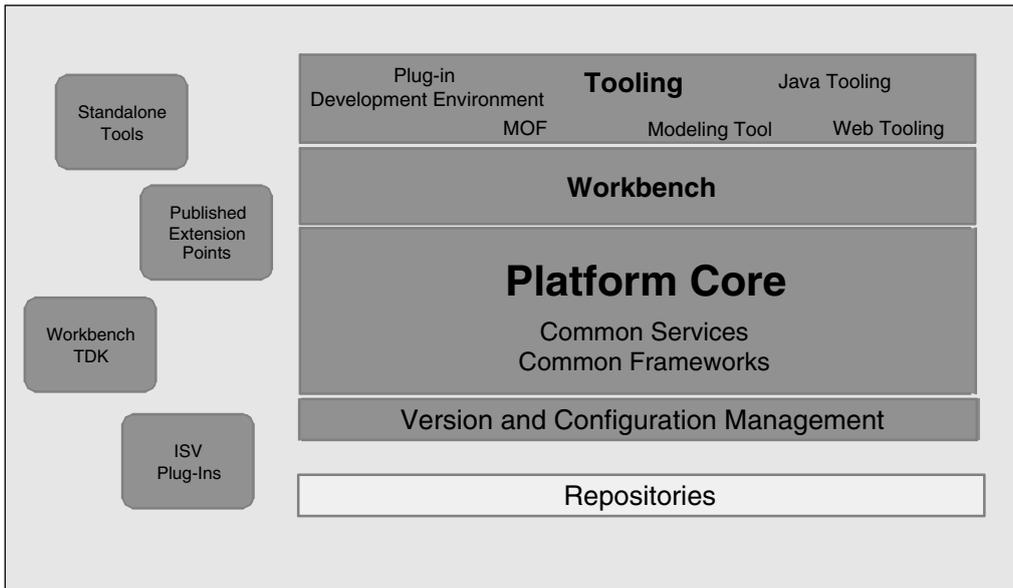
Above the workbench layer in Figure 3.5 is the tooling layer, which contains Web tooling, modeling tooling, and frameworks such as MOF (meta object facility).<sup>35</sup> Also shown on the left of the graphic are support standalone tools. Workbench tool development kit and published extension point vendors can use the development kit provided with the published extension points to create new plug-ins.

---

33. To learn more about WSAD see [www-3.ibm.com/software/ad/studioappdev/](http://www-3.ibm.com/software/ad/studioappdev/).

34. To learn more about the Eclipse consortium and the open source initiative consult the Resources section.

35. To learn more about MOF see [www.omg.org/cwm/](http://www.omg.org/cwm/).



**Figure 3.5** WebSphere Studio Workbench

For the application developer, WSAD brings with it two important concepts—*projects* and *perspectives*. Projects are used for organizing resources related to a particular development effort. A number of project types (including Java, Web, and Enterprise Archive [EAR]) are used in our solution. EAR projects are used to develop J2EE applications and often include some mix of Web, EJB, and client modules

Depending on role and task orientation, WSAD, unlike other development tools we've presented, permits the developer to adopt different perspectives (see Table 3.2). There are predefined perspectives for Java-focused development. The Java perspective, for instance, provides tooling and editors that facilitate Java development. The data perspective is tailored to facilitate relational database design. There are also Web, data, J2EE, XML, and other development-focused perspectives. One developer may use several of the available perspectives or create custom ones to build a particular application solution.

### 3.2.3 Business Process Integration

Business process integration is the third integration category. It is characterized by *development* at the business process level and *exchange* of business process objects (which may be combinations of both data and application logic). Typically, business processes might include purchase orders destined for supply chain management (SCM) systems or ATP (availability-to-promise) business processes integrated with customer relationship management (CRM) systems. The application integration of the business process management flow in Figure 3.4 represents the level of abstraction found in business process integration.

**Table 3.2** WSAD Projects and Perspectives

	Project						
	EAR	EJB	Web	Java	Server	All	Other
<b>Perspective</b>							
J2EE	*						
Web			*				
Java				*			
Server					*		
Debug						*	
Data		*	*	*			*
XML		*	*	*			*
Team	*	*	*	*	*	*	*
J2EE + Java		*					

The development paradigm and the tooling that supports it must be process focused to allow business analysts and traditional developers to construct business process applications. Often these business process applications are used to facilitate B2B interchange or to integrate disparate OEM packages (i.e., “application islands”) quickly within an enterprise.

The WebSphere Business Integration (WBI) Server<sup>36</sup> has a number of business process focus components, including the CrossWorlds development tooling. The WBI development tooling<sup>37</sup> includes a unified modeling language (UML)–based process design tool as well as tools for business object design and transformation map building. Apart from some code generated for WBI connectors that is in the C language, the underlying code generated by the WBI tooling is in Java and can be easily modified or exported. Although the business process development paradigm and tooling are important, the real strength of the business process level integration in WBI is reuse. WBI delivers a number of preconfigured and reusable business process components including collaboration templates (application stubs), maps (transformation maps), and connectors, including those for Java Database Connectivity (JDBC) and DB2.

Business process integration is a logical response—a natural outgrowth of application integration in the highly distributed, complex environment of integrated solutions development. In

36. To learn more about WBI consult [www-3.ibm.com/software/ts/wbi/](http://www-3.ibm.com/software/ts/wbi/).

37. For more on CrossWorlds tooling see [www-3.ibm.com/software/integration/wbitools/](http://www-3.ibm.com/software/integration/wbitools/).

many cases business process integration is the only approach that can give timely, cost-effective results by integrating disparate “application islands.” Business process integration makes use of many integration technologies already discussed, especially Web services, to make the level of abstraction required by distributed business processes practicable.

### 3.2.4 Information Integration

Information integration, the last integration category, has only recently begun to take shape.<sup>38</sup> The intent of information integration is to bring together well-structured schema-based data and poorly structured or wholly unstructured data. It uses much of the same integration technology and products that we have already discussed.<sup>39</sup> It also includes tools typically applied to data warehousing and data mining efforts, including portals.<sup>40</sup>

Like business process integration, it takes a business view of integration but does so at the information level. From our perspective, information integration is really best defined by three features—federation, heterogeneity, and intelligence.

The growth in technologies supporting federation and heterogeneity allow extraction of intelligence (otherwise known as information) from what are otherwise disparate assemblages of data. Advances in query processing technology such as SQLX and XQuery, text search, and mining capabilities as well as content management technologies will also enable information integration.

## 3.3 Summary

In this chapter we introduced the complex distributed solutions environment. We presented the four major forms of integration—data, application, business process, and information integration. We also showed how the various forms of integration can be related to DB2 in terms of both development tooling and integration technologies.

Choice of a particular development tool is often a function of the level of integration, use cases, and tool users. There are no hard and fast rules, but most environments include both specialized development tools that tend to be very simple to use and powerful, general purpose IDEs that can address the entire scope of development. We also used the “patterns” to illustrate the various forms of integration and to show the value of decomposition to simplify development and foster reuse.

---

38. “Information Integration,” *IBM Systems Journal*, vol. 41, no. 4, 2002, [researchweb.watson.ibm.com/journal/sj41-4.html](http://researchweb.watson.ibm.com/journal/sj41-4.html).

39. For more on Information Integration see [www-3.ibm.com/software/data/integration/solution/](http://www-3.ibm.com/software/data/integration/solution/).

40. For more on DB2 Information Integrator for Content (formerly Enterprise Information Portal) see [www-3.ibm.com/software/data/eip/](http://www-3.ibm.com/software/data/eip/).