



# Process-Centric Application Development with Compuware's UNIFACE Eight

*An IDC White Paper*

*Analyst: Rikki Kirzner*

## **The Web Created a Crisis in Application Development**

The unparalleled success of the Web and the relentless pressure on businesses to deliver Web-based solutions in record time has forever changed the way applications need to be created. Rapid deployment of robust, high-performance applications is essential for companies competing in today's global economy. More importantly, companies require the ability to differentiate products and services from their competition. This need arose out of the explosive growth of the Web, which has created a crisis in the development community along with a compelling business requirement to find the tools that would help developers work smarter and faster.

There is increased pressure on companies to respond quickly to competitors' products, prices, and services or risk losing the customer. To make matters worse, competitors can see what a company's strategy and pricing are as soon as the company posts its products to the Web site. Customers are demanding more services and better support from companies. This puts pressure on businesses to keep up with changing customer demands and provides a requirement for companies to deliver more services, price adjustments, special promotions, and so on. To respond to both the customer and competitors, companies have an increasing requirement for more flexible and sophisticated applications that can be created in shorter timeframes than traditional coding methodologies have allowed. Companies must also have a way to make modifications to existing applications quickly enough to maintain differentiation and competitive advantage.

Success in ebusiness entails around-the-clock, follow-the-sun computing involving real-time processing of transactions and the need to service and respond to millions of users simultaneously. Success in ebusiness also includes being able to handle effectively and seamlessly a myriad of Web-based and legacy applications and technologies. Some of these applications and technologies include online billing and inventory systems, electronic catalogs, Web standards, network protocols, online

*24 x 7 computing thrives on rapid change and accelerated application development cycles.*

tracking of goods and services, and the underlying hardware and software systems that support all of these applications and capabilities. This type of 24 x 7 computing thrives on rapid change and accelerated application development cycles.

Furthermore, it requires many different skill sets to develop and support sophisticated applications in today's large computing environments. IT professionals must be trained to understand the complexities of the application and its business logic, the communication protocols of the networks, the technologies of different heterogeneous platforms that comprise a distributed environment, the interfaces to the hardware, software, and applications, and the complexities of the system software. Rarely does one person have all the skill sets required to produce an application from concept to deployment. Traditionally, the creation and/or modification of any large, network-ready, transactional, distributed application system requires large teams of engineers with the above-mentioned skills. Consequently, these development projects are fraught with potential problems and are at high risk for introducing major design errors and bugs, which are often difficult, if not impossible, to extract once they are embedded in the code. Finding and retaining employees with all these skills is growing more difficult. As a result, this type of undertaking always takes far longer than expected. As time goes on, the integration costs, maintenance, and training costs required for keeping these systems running are escalating.

*Traditional development tools and environments cannot produce applications or handle the requirements to support diverse platforms and technologies fast enough to keep up with customer demands and competitive pressure.*

Therefore, it is important for companies to have tools and environments that allow them to create and evolve business processes and applications in response to rapid changes in industry and customer demand. Traditional development tools and environments cannot produce applications or handle the requirements to support diverse platforms and technologies fast enough to keep up with customer demands and competitive pressure.

Moreover, the traditional technology-based approach to software development is time consuming. Only a small percentage of the time required to develop applications is spent on building the logic and business processes. The vast majority of the engineering effort is dedicated to developing infrastructure, low-level processes, hardware, software, technology interfaces, and system drivers. And the time-honored methods of developing applications not only often fail to meet performance, cost, service, and support requirements of most vertical industries but will also not be able to meet the demands of Internet users in the 21st century, particularly for application differentiation.

Copyright © 2000 IDC. **Reproduction without written permission is completely forbidden.**

External Publication of IDC Information and Data—Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

*Printed on  
recycled  
materials*



*An inability to create component-based applications that are not flexible enough to respond rapidly to changes in technology trends and user requirements is now a recipe for disaster.*

Traditional application development approaches often create business process logic within a monolithic and/or proprietary development environment. While most vendors “publish” the API to these environments, this reduces an application’s flexibility to accommodate changes. Applications developed in this manner frequently become enterprise “stovepipes” — standalone systems or elements that don’t integrate well with other networked systems. Enterprise stovepipes are difficult and expensive to modify, enhance, and maintain. An inability to create component-based applications that are not flexible enough to respond rapidly to changes in technology trends and user requirements is now a recipe for disaster. Locating and modifying business logic that is not component-based is time consuming, risky, and expensive. Haphazard changes made to fragile code in these types of environments can often result in cascading failures throughout the computing environment. And to complicate matters further, in most large organizations, critical support service applications such as customer support systems often do not access the same databases, making service and support more difficult. As a result, many of today’s sophisticated applications are often unable to easily accommodate new requirements of businesses, particularly as they evolve, expand, or merge with other corporations.

### **A Process-Centric Approach for Developing Applications**

Simply put, a process-centric approach to application development occurs with the business analyst(s) and the software engineer(s) working together on two different phases of developing the application. The business analyst defines the business processes, the required tasks that have to be performed, the business rules, and the task assignments within the organization. The software engineer defines the data model, the business logic, and may also mine the current IT infrastructure to find reusable business components or applications. When all relevant components have been identified, created, or generated, the software engineer connects the relevant component signatures to the tasks that the business analyst has defined. Once these steps have been completed, the processes are linked and synchronized in the process model. In this step, the definition of the required data fields that the process requires in order to exchange parameters with tasks are established so that it is possible for information to be passed from one task to another.

*Using a process-centric approach, the business analyst works from the top down and software engineers work from the bottom up to design the application.*

Using a process-centric approach, the business analyst works from the top down and software engineers work from the bottom up to design the application. At runtime, business managers can change the process model. Deployment engineers may exchange old components for newer versions at deployment time. Table 1 depicts the steps of this role-based approach to development and deployment.

Developing and deploying applications to run today’s businesses is a complex and demanding task. Applications must be more adaptable, more integrated, and more flexible. Since traditional development methodology fails to deliver these types of applications in time to meet

**Table 1  
Role-Based Development and Deployment**

Role	Development	Deployment
Business analyst	Design business process models Define organizational model Specify B2B message exchanges	
Software engineer	Build data model Implement business logic Implement server-side presentation logic Generate components Build custom components Integrate existing components and applications	
Web designer	Design Web pages Implement client-side presentation logic	
Business manager		Reassign tasks Monitor process execution Change business processes
Deployment engineer		Specify component implementations Configure middleware Assign execution locations
Webmaster		Manage content

Source: IDC, 2000

*A new approach to application development now exists that is process-centric and creates component-based applications that can be deployed in a manner that is independent of the underlying infrastructure.*

*Crucial decisions can be made later in the design process and changed at any point prior to physical deployment. An application designed in this manner has the added advantage of being able to accommodate one or more best-of-breed technology choices.*

the demands of today's businesses, a more efficient approach to developing applications is required. A new approach to application development now exists that is process-centric and creates component-based applications that can be deployed in a manner that is independent of the underlying infrastructure. Business processes and business components can be logically and physically developed and deployed in a loosely coupled manner. This new paradigm allows developers to spend the majority of their development efforts working on creating sophisticated business logic specifically suited to solving business problems rather than worrying about issues of interoperability. Compuware UNIFACE takes care of the "system-level plumbing" so that the technical details are automatically handled by the framework servers. Therefore, additional enhancements can be modeled and added easily without having to make extensive modifications to the infrastructure as well as the business processes.

A developer should not have to decide on the particular database, operating environment, or component models until deployment time. This type of development makes the application flexible enough to be easily deployed across any environment. Crucial decisions can be made later in the design process and changed at any point prior to

*The resulting applications can be developed quickly and are easy to maintain or modify. Companies would be able to respond to changing economic and competitive climates within a few months or less, instead of a year or more.*

*UNIFACE Eight is a business-to-business (B2B) ecommerce application development environment that enables organizations to model, develop, and deploy sophisticated B2B solutions within and across an organization.*

*UNIFACE accelerates application delivery by hiding technical complexities from developers.*

physical deployment. An application designed in this manner has the added advantage of being able to accommodate one or more best-of-breed technology choices. Correspondingly, being able to reuse the logic and business rules, along with most of the tested and proven code of the application by developers, systems integrators, or value-added retailers, increases profitability and decreases time to market significantly.

Finally, development is faster with this new methodology because there is a server technology and development system that makes it easy to generate applications directly from object-oriented or component models. The resulting applications can be developed quickly and are easy to maintain or modify. Companies would be able to respond to changing economic and competitive climates within a few months or less, instead of a year or more.

There have been many previous efforts in recent years aimed at addressing distributed computing and component-based development needs, including middleware, object-oriented programming, componentization, application servers, Java, and unified modeling language (UML). Despite the value that each of these technological developments has brought to application development, the unification of these technologies now enables a significant advance in how applications are developed. One such approach is presented to the developer in Compuware's newest release of its UNIFACE solution.

### **Compuware's UNIFACE Eight**

Compuware's UNIFACE Eight is a comprehensive ebusiness solution for creating enterprise business applications and services. It provides out-of-the-box functionality for business process automation, application development, technology management, and end-to-end integration.

UNIFACE Eight is a business-to-business (B2B) ecommerce application development environment that enables organizations to model, develop, and deploy sophisticated B2B solutions within and across an organization. It facilitates the creation of those applications and their business processes that link customers, partners, and suppliers. UNIFACE Eight also integrates the business processes with the underlying IS systems that support these processes. As part of its B2B solution, UNIFACE Eight features a robust deployment infrastructure that offers 24 x 7 availability, transaction management, security, scalability, and data integrity.

The applications produced using UNIFACE Eight are technology-independent and run on all mainstream architectures, from the Web to multitier client/server architectures. UNIFACE Eight supports a wide range of operating systems, databases, graphical user interfaces (GUIs), and networks.

UNIFACE accelerates application delivery by hiding technical complexities from developers. Additionally, UNIFACE offers support for change management through a repository; importing and exporting repository

data; data conversion; loading model definitions from existing data sources and modeling environments; and many team development requirements, including support for version control authorization and personalization of the development environment.

UNIFACE also provides advanced technology to maintain and manage applications throughout their life cycle, including a component architecture, a rich language and sophisticated debugger, impact analysis tools, and comprehensive repository reporting capabilities.

*UNIFACE Eight applications are generated directly from models of the business logic of the application. Developers are required to do only a minimal amount of traditional coding.*

### **Compuware B2B Applications Are Modeled, Not Coded**

Compuware has advanced the art of application development to a new level of productivity with its UNIFACE Eight product. Through the use of sophisticated business process automation technology, UNIFACE Eight breaks the mold of traditional application development. The business processes and logic of UNIFACE Eight applications are not coded using traditional 3GL languages. Instead, the UNIFACE application development environment offers a model-driven, component-based approach that allows developers to implement business rules that are inherited by application components. Almost all the effort of developing a UNIFACE Eight application goes into creating and revising a high-level design or model of the business processes. Application components and data can be modeled using Rational Rose. In addition, the structure of databases can be extracted and used as a model as well. UNIFACE Eight applications are generated directly from models of the business logic of the application. Developers are required to do only a minimal amount of traditional coding. The UNIFACE Eight PROC language is an event-driven, declarative-style language that provides an extensive, hierarchical set of triggers that are activated under well-defined circumstances. These triggers contain elements of the PROC code that convey a specific portion of business logic. These triggers are then selectively inherited by the various UNIFACE components when they are created or generated. The PROC language also provides access to a vast array of built-in UNIFACE functionality. This built-in functionality, which has been implemented in the UNIFACE deployment engines and connectors, is optimized for each platform that supports the UNIFACE runtime environment. In this way, specific aspects for handling and processing the technology are simplified for developers, shielding them from the underlying complexities of the technology. For instance, calling an operation in another component is always done using the same PROC statement (e.g., “activate MyComp.MyOper”). UNIFACE knows how the “MyComp” component must be invoked and will select the appropriate technology to complete the call (e.g., selecting native UNIFACE, CORBA, EJB, COM, and COBOL).

*Built-in functionality, which has been implemented in the UNIFACE deployment engines and connectors, is optimized for each platform that supports the UNIFACE runtime environment.*

With UNIFACE Eight, when the IT professional is satisfied that the business application models have been created correctly, the model is translated directly into executable components that can then be

*With UNIFACE Eight, when the IT professional is satisfied that the business application models have been created correctly, the model is translated directly into executable components that can then be automatically assembled into an application that runs on a specific application server or in any runtime environment.*

*The use of an open, modeled design approach allows a developer to determine the essential structure of an application and/or software system without having to first select the specific technology infrastructure and interfaces.*

*A modeling approach supports the creation of open systems and can therefore accommodate a variety of proprietary and legacy technologies.*

*By building abstract, component-based frameworks, developers can create both an application infrastructure and processes to improve time to market for crucial integrated applications. They can reduce costly delays by doing rapid prototyping of their systems and by using an iterative approach to designing enterprise-class application systems.*

automatically assembled into an application that runs on a specific application server or in any runtime environment. UNIFACE creates components by assembling both generated fragments of code and user-defined fragments of code into a component. When that component is compiled, a stream of symbolic code is generated that can be instantiated and executed as a component inside a UNIFACE runtime environment.

However, the code that is produced is highly optimized to support the modeled application, and the applications can be developed quickly and are easy to maintain. Since they are model-driven, UNIFACE applications can accommodate sweeping changes in business and technology.

### ***Advantages of Modeled Application Development***

The use of an open, modeled design approach allows a developer to determine the essential structure of an application and/or software system without having to first select the specific technology infrastructure and interfaces. To this end, Compuware supports most major technology standards and will continue to support new standards as they emerge. For instance, when developers tightly couple the logic of their applications (e.g., billing systems or customer support systems) to the infrastructure upon which it executes, then that application becomes bound to that infrastructure and cannot be easily changed or moved. On the other hand, if all the code is abstracted from the idiosyncrasies of individual technologies, corporations and systems integrators can reuse their existing or new applications and easily integrate them or move them to another architecture.

A modeling approach employs a degree of abstraction that facilitates network elements and technologies so they become plug-in interfaces. This process supports the creation of open systems and can therefore accommodate a variety of proprietary and legacy technologies. For instance, existing C and C++ routines can also be invoked directly from the model operations. This is important to developers writing algorithms for specific functions or those that are done more efficiently in straight-line procedural hand coding.

A modeling approach also removes the burden of hiring or making developers technology experts. Developers do not need to be trained in the low-level technology standards and interfaces in order to exploit these within their applications. Finally, by building abstract, component-based frameworks, developers can create both an application infrastructure and processes to improve time to market for crucial integrated applications. They can reduce costly delays by doing rapid prototyping of their systems and by using an iterative approach to designing enterprise-class application systems. Developers can also build and deliver more personalized services and thereby protect their customer base from moving to the competition because of changes to pricing or the ability to provide better competitive services and

solutions. Additional features and/or changes to the application can be implemented in a fraction of the time it took using more conventional coding.

To facilitate developer productivity and workgroup support, UNIFACE Eight provides the tools and architecture for collaborative development, change and configuration management, and version control.

### ***How UNIFACE Eight Supports B2B Process Development***

*Programmers and/or IT professionals are responsible for creating and/or assembling the components that construct the runtime environment on which the business logic or tasks of the business application will run.*

The approach that encompasses these critical areas is fundamentally different than that used in traditional programming methodologies. Each type of developer has a separate role in creating the application, but there is a shared understanding of the processes. Business analysts define the process logic of the business application. They are not concerned with the system code, system-level drivers, system resource management, technology interfaces, and other coding efforts. Programmers and/or IT professionals are responsible for creating and/or assembling the components that construct the runtime environment on which the business logic or tasks of the business application will run. In essence, both the business analysts and the system programmers need to concentrate and work only in their own areas of expertise to produce a complete B2B solution using UNIFACE Eight. And because UNIFACE Eight provides many of the prebuilt components for much of the underlying system routines that are required to support the system infrastructure, there is significant reusability of system-level components and less effort required in creating or debugging system-level routines.

*In UNIFACE Eight, the models are completely independent of the underlying system level infrastructure. Details regarding target environments, such as hardware and software platforms, databases, middleware, and connectors are specified in separate workbenches in the UNIFACE environment.*

Component development is the phase in which models are specified to define the underlying data structures, relationships, and relevant business logic. Components are then built, which inherit the model definitions. In UNIFACE Eight, the models are completely independent of the underlying system-level infrastructure. Details regarding target environments, such as hardware and software platforms, databases, middleware, and connectors are specified in separate workbenches in the UNIFACE environment. Through this process of separating the business logic from the underlying system code, UNIFACE Eight eliminates the requirement for developers to create low-level drivers and interfaces to specific operating systems.

UNIFACE Eight components are capable of generating HTML and XML following the data structures that are specified by the developer. When generating HTML, for instance, UNIFACE automatically incorporates JavaScript as well as user-defined HTML. Macromedia's Dreamweaver environment has been integrated into UNIFACE to further enhance the appearance of HTML pages. To aid in the construction of XML messages that adhere to industry standards, UNIFACE provides prebuilt components that create BizTalk-compliant messages.



*Used together, these tools and components can speed up the development time for integrating applications with standard systems, technologies, databases, and communication protocols. It also makes the process of modifications, enhancements, and maintenance much easier.*

*UNIFACE Eight incorporates a layered approach to make the process of creating, modifying, and maintaining business logic easier for the developer.*

UNIFACE Eight provides the development framework, servers, editors, and a collection of prebuilt components for key application tasks or actions, interfaces, and drivers that address developer requirements for each of these areas of development. Used together, these tools and components can speed up the development time for integrating applications with standard systems, technologies, databases, and communication protocols. It also makes the process of modifications, enhancements, and maintenance much easier.

There are four critical areas that need to be considered in the use of UNIFACE Eight to develop B2B ecommerce solutions:

- Business process automation
- Application development
- Technology management
- End-to-end integration

UNIFACE Eight functionality and support for these four areas are summarized below.

#### *Business Process Automation*

Business process automation involves the actual design and creation of the business process logic. Here, development focuses on defining the flow and structure of the business processes and the associated logic. This phase entails defining the organizational structure and the employees in the organization, the business tasks that need to be performed (such as real-time, online processing or batch processing) and the required interfaces to the application components. It also entails the use of UML and other modeling languages to create the structure and flow of the actual business logic of the application. UNIFACE Eight incorporates a layered approach to make the process of creating, modifying, and maintaining business logic easier for the developer.

- The process logic layer contains the business rules that define how a business analyst or a business manager would view their business. This includes the business decisions that need to be made, the order in which the rules are to be executed, the decision trees, and the actual tasks that need to be carried out. This layer determines what application components (that are defined in the lower layers) need to be invoked, when they are to be invoked, and whether the required tasks are to be performed automatically or manually.
- The presentation layer contains all the components that implement the interactive user interface.
- The task logic layer contains the components that fulfill a specific task such as processing a customer's order, checking credit status, and so on.

- The data logic layer is completely managed by UNIFACE. It contains all logic relevant to ensuring integrity, consistency, and persistence of data. UNIFACE uses the developer's data model, including any contained business logic, to derive its internal logic.

### *Application Development*

Application development is the phase of creating the software code and components used to create the remainder of the applications and integrate them with the hardware, software, database, middleware, and technology standards of the environment in which they will run.

*UNIFACE Eight provides the tools to develop components as well as prebuilt components to support team collaboration, change, and configuration management.*

UNIFACE Eight provides the tools to develop components as well as prebuilt components to support team collaboration, change, and configuration management. UNIFACE aids developers in the rapid development and implementation of ebusiness applications by managing the creation of HTML, Javascript, DTDs, XML, BizTalk, and any other messaging standards. It also includes a Dreamweaver interface. Compuware UNIFACE will support SOAP for remote component execution via the Internet.

After a model has been populated with entities, fields, relationships, business logic, and so on, UNIFACE Eight is capable of generating Entity Services for all entities in the model and Session Services for all entities with all their relationships in the model. For every entity, a Session Service is created that is aware of all relevant relationships with all other entities. UNIFACE Eight also generates Server Pages that contain all necessary controls to create, read, update, and save any data in the database, including the controls required to jump to related Server Pages (e.g., to switch from a customer's request for a product to the customer's account information). Consequently, the model generated produces a complete application ready to maintain all pertinent data for the business transactions. Furthermore, the application can be quickly modified or enhanced to maintain a company's competitive edge and allow the company to quickly differentiate its application.

### *Technology Management*

*UNIFACE Eight delivers a robust, B2B ecommerce application deployment infrastructure.*

Technology management is the ability of the UNIFACE Eight tools to help developers create, target, and manage applications in host-based, two-tier, multitier, or Web-based architectures. UNIFACE Eight delivers a robust, B2B ecommerce application deployment infrastructure. Applications can be deployed on virtually any platform, regardless of the development platform, from Windows NT to a mainframe.

*UNIFACE will easily support new platforms and accommodate new technologies as they become mainstream.*

UNIFACE Eight promotes and enhances the concept of true portability. An application written in UNIFACE is easily deployed on any platform and since Compuware is always adding support for new environments and new technologies, applications written in UNIFACE

will easily support new platforms and accommodate new technologies as they become mainstream.

### ***End-to-End Integration***

End-to-end integration is the support provided by UNIFACE Eight for integrating the business application logic with the client, server, and network architectures, operating systems, and third-party business processing systems.

*UNIFACE Eight eliminates the requirement for developers to create low-level drivers and interfaces to operating systems.*

UNIFACE Eight eliminates the requirement for developers to create low-level drivers and interfaces to operating systems, such as UNIX, AS/400, S/390, databases, other industry systems such as SAP, as well as application servers (e.g., WebSphere, iPlanet, WebLogic, and Domino). Developers can choose to create these components from scratch, or they can use Compuware's prebuilt components.

UNIFACE Eight includes an extensive set of prebuilt connectors and adapters for databases such as Oracle, Sybase, Informix, SQL Server, and DB2 and mainframe data structures, including VSAM and IMS. UNIFACE Eight also offers connectors for component models such as COM, CORBA, and Enterprise JavaBeans; legacy components including CICS and IMS/DC; and messaging. UNIFACE Eight supports HTTP servers such as Microsoft IIS, Apache, Lotus Domino, WebSphere, and WebLogic.

### ***Developing Applications Using UNIFACE Eight***

*UNIFACE Eight brings a higher level of abstraction to the development process. This yields greater reuse, extensibility, and simplicity in building complex applications.*

After the business processes have been modeled in UNIFACE, developers can use these process definitions to build or assemble the underlying application components. The first step for developers building components is to define an application model that contains the entities, relationships, and business logic relevant to that level. Where existing components are to be used, developers will load their interfaces into UNIFACE to bring a higher level of abstraction to the development process. This yields greater reuse, extensibility, and simplicity in building complex applications.

By carefully defining the application model (i.e., specifying such areas as the user interface logic, data model, task logic, and general business constraint logic), the resultant construction process is greatly simplified. Building the application model helps the developer define the application as a series of components, each with its own name, parameters, and processing. This logical design is leveraged in the construction phase to generate components, so the physical application is derived from the application model. In addition, the design allows for modification of existing component creation, addition of new components, and the invocation of existing components, regardless of their origin.

Development and maintenance are streamlined as the model can be modified, resulting in the regeneration of components that depend on that part of the model. Developers can modify the generated

*UNIFACE Eight enables the developer to define the detailed relationships necessary to build complex applications.*

UNIFACE components or implement their own. Developing the application model using UNIFACE Eight tools is a straightforward process, since UNIFACE Eight provides sophisticated support for defining fields, keys, relationships, interfaces, events, and triggers. UNIFACE Eight enables the developer to define the detailed relationships necessary to build complex applications. Defining the application using a modeled approach helps the developer understand, enhance, and resolve application issues related to completeness, integrity, reuse, componentization, and interoperability.

UNIFACE Eight provides a set of application service modules for handling many of the application processing tasks. These are a variety of generic services that can be used by developers for transaction processing, transaction management, distribution, fault management, external interfaces, and many other core processing services. Consequently, application developers can focus on designing and developing modules that address the business problems alone.

### *The Editors*

UNIFACE Eight provides a range of editing tools to help organizations create and manage the components used to automate and refine their business applications.

- The UNIFACE Process Model Editor allows developers and business analysts to define and modify business processes and tasks graphically. This allows developers to easily see which business tasks are associated with application components.
- The UNIFACE Organization Model Editor allows organizational models to be defined and modified. Hierarchical structures are represented graphically, and organizational units are grouped as departments or business units. Developers can describe a set of roles and behaviors for each department or division in an organization. In this way, job tasks associated with job titles can be identified and associated with individual business tasks. This gives developers the ability to more efficiently incorporate the business structure within the application logic.
- The UNIFACE B2B Integration Editor simplifies the integration of enterprise applications at a B2B messaging level. It helps developers create and integrate BizTalk, Open Application Group Interface Specification (OAGIS), and user-defined XML messages.
- The UNIFACE Component Editor helps developers build and maintain application components, including UNIFACE Forms, Server Pages, Services, Reports, Session Services, Entity Services, Document Type Definitions (DTDs), and XML. The UNIFACE Component Editor also generates DTDs from UNIFACE application components, which may in turn be derived from the model.

- The UNIFACE Component Integration Editor allows developers to rapidly integrate almost any technology without needing to develop low-level application programming interfaces. The UNIFACE Component Integration Editor is a graphical tool for integrating UNIFACE components with non-UNIFACE components, including JavaBeans, C++ programs, CICS/COBOL transactions, stored procedures, and complete applications and subsystems. Using drag and drop technology, application developers combine these components and applications without having to know how individual components work or how they are implemented. The UNIFACE Component Integration Editor displays detailed component interaction diagrams to help developers visualize, understand, and manage large applications. Comprehensive search and browse capabilities simplify application manageability and maintenance.
- The UNIFACE Component Signature Editor is used to edit non-UNIFACE components. The interaction of individual application components is made possible by component signatures. UNIFACE automatically creates and manages its own component signatures.
- The UNIFACE Web Front-End Editor incorporates Dreamweaver from Macromedia for professional Web page design and site management.
- The UNIFACE Business Object Model Editor accelerates the definition and modification of business objects, which consist of data structures and typical behavior.

### *UNIFACE Eight Repository*

Component-based development ultimately ensures the availability of a large number of previously developed and prepackaged components at the developers' disposal. This suggests that the careful management of these components must be addressed. Repositories ensure that the vast number of components available can be located, accessed, used, and managed.

UNIFACE manages metadata by providing a repository. All objects, including the application model, are stored once within the repository with reusable definitions and property inheritance across the hierarchy. This feature extends to both UNIFACE proprietary objects as well as non-UNIFACE-based objects (such as those based on CORBA, DCOM, or a 3GL), which are also definable and stored in this repository. All objects are encapsulated with a UNIFACE signature that defines the following:

- Operational information
- Parameters
- Component implementation (optional)

All components must be defined in the repository before application assembly can be done. Once complete, all components are defined to the developer in common terms for easy application assembly through the Component Integration Editor.

*When the modeled application is complete, the components are assembled into an application.*

### **Realizing the Model**

As was mentioned earlier, component construction is the first step in the process to implement the application model. The construction process generates components that have well-defined names and interfaces (parameters and I/O). These components can also have other generated features, such as fields and navigation, if they are forms. Although the developer remains responsible for building custom business processes either through the UNIFACE PROC language or a more traditional coding language, careful planning should allow a much higher level of reuse over time for any custom code that is developed. When the modeled application is complete, the components are assembled into an application. The process of building an application from diverse components and linking these components through their interfaces into an application is known as application assembly and integration.

The creation of a UNIFACE Eight application is essentially done in the following generalized manner:

1. Define a business process containing the individual tasks that need to be performed. In this stage, it is not yet important whether the task will be batch-oriented, interactive, or fully manual.
2. Investigate the existing IT infrastructure to locate application components that are candidates for reuse. The availability of legacy components may influence the way tasks are defined in your processes.
3. For components that need to be newly developed, define a data model.
4. Place as much of the business logic as possible in the relevant triggers in the model.
5. Allow UNIFACE Eight to automatically generate Entity Services, Session Services, and Server Pages (which are described below).
  - UNIFACE Entity Services are noninteractive components that contain business functions and behaviors. Services capture a distinct, logically grouped set of functions that together make up a business object. UNIFACE Entity Services have the unique capability of being machine-independent, database-independent and location-independent. Because code is not physically moved around the network when repartitioning, developers can locate it to enhance performance.

- UNIFACE Session Services contain all business task-specific behaviors for a hierarchical data structure. Hierarchical data structures are defined in the business object model and consist of entities and their relationships. Business task-specific behavior is implemented using component operations. Communication with UNIFACE Session Services is stateless, and data is interchanged as disconnected XML record sets. UNIFACE Session Services can delegate specific business object behavior to UNIFACE Entity Services.
  - UNIFACE Server Pages are components that dynamically generate HTML screens, allowing users to input data, update databases, and call other components. UNIFACE Server Pages execute on a server, while the generated HTML pages run in a browser on the client. These features reduce administration overheads, providing a highly scalable solution for B2B e-commerce applications.
7. Make any necessary modifications required to tailor the application and/or user interfaces to the required specifications of the business.
  8. Connect the tasks in the process models to the signatures (interfaces) of the application components. Define the relevant process variables to host the data passed via the task parameters. Note that the task parameters are always identical to the parameters of the connected component (via the signature).
  9. Deploy the processes.

UNIFACE Eight translates the high-level model directly to a compact, executable application by generating code for a known runtime environment — the UNIFACE Business Process Server. The UNIFACE Business Process Server is already performance tuned, so a developer only has to concentrate on performance tuning the business logic portion of the application.

UNIFACE Eight also provides a range of tools to help organizations automate and refine their business applications. It features a wide array of task-specific servers, consoles, and a router to help developers create the components and applications and manage the components and development tasks throughout the development process. Table 2 summarizes the deployment infrastructure detailed above.

Table 3 lists the group of servers and tools that complete the entire UNIFACE Eight development framework. When used together, these components, systems, and tools give developers the ability to rapidly create, deploy, and modify B2B applications.

**Table 2**  
**The Deployment Infrastructure**

Functional Area	Development	Deployment
Process model	Process Model Editor	UNIFACE Process Server Process Console (the stripped-down version of the Process Editor for the business manager) Task Console (the task lists of the various employees in the organization)
Organization model	Organization Editor	Resource Console (the super task list for the business manager)
Interface management	Signature Editor and Component Integration Editor	UNIFACE Application Server and Remote Application Invocation (connects to CORBA, COM, EJB, etc.)
Component construction	Component Editor	UNIFACE Virtual Machine, including UNIFACE Clients and UNIFACE (Web) Application Servers
B2B messaging	B2B Integration Editor	UNIFACE B2B Message Server
Data modeling	Business Object Model Editor	UNIFACE Application Server and Remote Data Access

Source: IDC, 2000

### Analysis and Conclusion

The key to controlling development and application support costs depends not only on efficient coding of business processes but on being able to continuously and quickly modify and enhance them. The ability to automate the creation, modification, and enhancement of critical processes across the entire supply and demand chain improves an organization's efficiency and productivity and reduces the costs of doing business.

*UNIFACE Eight is model-driven so that developers can create applications that are capable of adapting to sweeping changes in process, structure, and logic.*

UNIFACE Eight gives developers a range of tools to automate the process of building application components that implement the business processes. UNIFACE Eight is model-driven so that developers can create applications that are capable of adapting to sweeping changes in process, structure, and logic. Components may be reused, boosting productivity, quality, and development times. Development proceeds faster using UNIFACE Eight because there is a server technology and development system that simplifies the generation of applications directly from object-oriented or component models.

*Compuware has introduced a new development paradigm that is ideally suited to automating the process of developing B2B applications.*

With UNIFACE Eight, developers can build applications independently of the deployment platform, which means that they can focus on implementing business logic rather than worrying about specific operating systems, databases, or architectures. This capability further enhances productivity, quality, and development times. Compuware has created tools that tie together heterogeneous environments, making the data in disparate systems available to customers, partners, and suppliers. As a result, Compuware has introduced a new development paradigm that is ideally suited to automating the process of developing B2B applications.



**Table 3**  
**Servers, Consoles, and Tools**

Servers, Consoles, Tools	What It Does
UNIFACE Business Process Server	The UNIFACE Business Process Server ensures that every business process is executed correctly. The UNIFACE Business Process Server executes the modeled processes and manages the workflow and distribution of tasks to the appropriate participants in the business process. It is the server that is responsible for notifying end users of tasks that require their interaction. At deployment time, business managers can monitor the application using one of the three UNIFACE consoles.
UNIFACE Process Console	The UNIFACE Process Console allows business managers to optimize performance according to availability of resources. It lets managers change the execution flow of the business rules as well as the organizational model at runtime.
UNIFACE Task Console	The UNIFACE Task Console displays the tasks that individual end users need to perform. The UNIFACE Task Console also provides end users with a work list for starting individual business tasks and activities and provides support for remote users.
UNIFACE Resource Console	The UNIFACE Resource Console allows business managers to reassign tasks to other participants or departments to resolve bottlenecks. Tasks can be rescheduled or reassigned to other employees, alerts can be reallocated, and task priorities can be adjusted. The UNIFACE Resource Console lets business managers review the history of any process, regardless of its status (e.g., running, pending, or completed).
UNIFACE Task Server	The UNIFACE Task Server executes the application components and applications that do not require user interaction.
UNIFACE B2B Message Server	The UNIFACE B2B Message Server manages communication with external systems via the Internet. It connects each process with its counterpart external process. In addition, the UNIFACE B2B Message Server manages BizTalk, Ariba, Commerce One and user-defined XML messages and delivers their contents to the appropriate application components.
UNIFACE Virtual Machine	The UNIFACE Virtual Machine offloads heavy transaction processing from several individual systems to a powerful central server for fast processing of thousands of transactions. Several UNIFACE Application Servers can be combined to address extremely high-volume transaction processing.
UNIFACE Application Server	The UNIFACE Application Server is designed to deploy scalable enterprise applications on the Web and offers server-based transaction processing, remote application connectivity, and remote data access. Intelligent integration between mainstream Web servers, such as Apache, iPlanet, WebLogic, WebSphere, and IIS, and a pool of UNIFACE Application Servers provides a scalable Web application environment.
UNIFACE Router	The UNIFACE deployment infrastructure includes the UNIFACE Router, a multithreaded traffic agent that routes business process requests to the UNIFACE Business Process Server and application requests to the UNIFACE Application Server. UNIFACE Router manages resource usage for transaction processing, application invocation, and data access. It also balances the load over a small pool of servers so that server resources are shared optimally among users.

Source: IDC, 2000





## NORTH AMERICA

### Corporate Headquarters

5 Speen Street  
Framingham, MA 01701  
508-872-8200

### IDC Canada

36 Toronto Street, Suite 950  
Toronto, Ontario, Canada M5C2C5  
416-369-0033

### IDC Irvine

18831 Von Karmen Ave, Ste 200  
Irvine, CA 92612  
949-250-1960

### IDC Mtn. View

2131 Landings Drive  
Mountain View, CA 94043  
650-691-0500

### IDC New Jersey

120 Wood Ave South, Suite 509  
Iselin, NJ 08830  
732-632-9222

### IDC New York

2 Park Avenue  
Suite 1505  
New York, NY 10016  
212-726-0900

### IDC Texas

100 Congress Ave, Suite 2000  
Austin, TX 78701  
512-469-6333

### IDC Washington

8304 Professional Hill Drive  
Fairfax, VA 22031  
703-280-5161

## ASIA/PACIFIC

### IDC Asia/Pacific (Hong Kong)

12/Floor, St. John's Building  
33 Garden Road  
Central, Hong Kong  
852-2530-3831

### IDC Asia/Pacific (Singapore)

71 Bencoolen Street, #02-01  
Singapore 189643  
65-226-0330

### IDC Australia

Level 4, 76 Berry Street  
North Sydney, NSW 2060, Australia  
61-2-9922-5300

### IDC China

Room 611, Beijing Times Square,  
88 West Chang'an Avenue, Beijing,  
P.R. China, 100031  
86-10-8391-3456

### IDC (India) Limited

Cyber House  
B-35, Sector 32 - Institutional  
Gurgaon - 122002, Haryana, India  
91-124-6381673 to 80

### IDC Japan

10F The Itoyama Tower  
3-7-18, Mita Minato-ku  
Tokyo 108-0073, Japan  
81-3-5440-3400

### IDC Korea Ltd

Suite 704, Korea Trade Center  
159-1, Samsung-Dong, Kangnam-Ku  
Seoul, Korea 135-729  
82-2-55-14380

### IDC Malaysia

Suite 13-03, Level 13, Wisma KiaPeng  
No. 3, Jalan Kia Peng  
50450 Kuala Lumpur, Malaysia  
6-03-2163 3715

### IDC New Zealand

Level 7, 246 Queen Street  
Auckland, New Zealand  
64-9-309-8252

### IDC Philippines

7F, SEDCCO 1Bldg  
Rada Street Corner  
Legaspi Street  
Legaspi Village  
Makati City, Philippines  
632-894-4808

### IDC Taiwan Ltd.

10F, 31  
Jen-Ai Rd, Sec 4,  
Taipei 106, Taiwan, R.O.C.  
886-2-2731-7288

### IDC Thailand

27 Soi Charoen Nakorn 14  
Charoen Nakorn Road, Klongtongnai  
Klongsan Bangkok 10600, Thailand  
66-2-439-4591-2

### IDC Vietnam

37 Ton Duc Thang Street  
Unit 1606  
District-1 Hochiminh City Vietnam  
84-8-910-1235

## EUROPE, MIDDLE EAST, AND AFRICA

### IDC Austria

c/o Loisel, Spiel, Zach Consulting  
Mayerhofgasse 6  
A-1040 Vienna, Austria  
43-1-50-50-900

### IDC Benelux (Belgium)

29 Avenue Louis Gribaumont  
B-1150 Brussels, Belgium  
32-2-779-46-04

### IDC Benelux (The Netherlands)

A. Fokkerweg 1  
1059 CM Amsterdam  
The Netherlands  
31-20-669-2721

### IDC Central Europe (ECE)

Male Namesti 13  
Praha 1 110 00, Czech Republic  
420-2-2142-3140

### IDC Central Europe (Germany)

Nibelungenplatz 3, 11th Floor  
60318 Frankfurt, Germany  
49-69-90502-0

### IDC Central Europe (Switzerland)

Niederlassung Züerich  
WTC, Leutschenbachstrasse 95  
CH - 8050 Züerich  
Switzerland  
41-1-307-1000

### IDC Egypt

39 Iraq Street  
Mohandesseen, Cairo, Egypt  
20-2-336-7355

### IDC France

Immeuble La Fayette  
2, Place des Vosges, Cedex 65  
92051 Paris la Defense 5, France  
33-14-904-8000

### IDC Hungary

Bajcsy-Zsilinszky út. 57  
Building 3, Rooms 103-104  
H-1065 Budapest, Hungary  
36-1-153-0555/ext. 165, 166

### IDC Israel

4 Gershon Street  
Tel Aviv 67017, Israel

### IDC Italy

Viale Monza, 14  
20127 Milano, Italy  
390-2-284-571

### IDC Nigeria

House 2, 'C' Close  
403 Road, 4th Avenue  
New Extension, Festac Town  
Lagos, Nigeria  
234-1-883585

### IDC Nordic (Denmark)

Jagtrvej 169B  
DK-2100 Copenhagen, Denmark  
45-39-162222

### IDC Nordic (Finland)

Jarrumiehenkatu 2  
FIN-00520  
Helsinki, Finland  
358-9-8770-466

### IDC Nordic (Sweden)

Box 1096 Kistagången 21  
S-164 25 Kista, Sweden  
46-8-751-0415

### IDC Poland/ProMarket

Wrobla 43  
02-736 Warsaw, Poland  
48-22-644-4105

### IDC Portugal

Av. Antonio Serpa, 36 Piso 9  
1050-027 Lisbon  
Portugal  
351-21-796-5487

### IDC Russia

c/o PX Post, RDS 186  
Ulitsa Zorge 10  
Moscow 125525  
Russian Federation  
7-501-929-9959

### IDC South Africa

c/o BMI-TechKnowledge  
3rd Floor, 356 Rivonia Blvd.  
PO Box 4603, Rivonia, 2128  
South Africa  
27-11-803-6412

### IDC Spain

Ochandiano, 6  
Centro Empresarial El Plantio  
28023 Madrid  
34-91-7080007

### IDC Turkey

Tevfik Erdonmez Sok. 2/1 Gul Apt.  
Kat 9D; 46 Esentepe  
Istanbul, Turkey  
90-212-275-0995

### IDC U.K.

British Standards House  
389 Chiswick High Road  
London W4 4AE  
United Kingdom  
44-20-8987-7100

## LATIN AMERICA

### IDC Miami

Latin America Headquarters  
5301 Blue Lagoon Drive  
Suite 490  
Miami, FL 33126  
305-267-2616

### IDC Argentina

Trends Consulting  
Rivadavia 413, 4th Floor, Suite 6  
C1002AAC, Buenos Aires, Argentina  
54-11-4343-8899

### IDC Brasil

Alameda Ribeirão Preto, 130 cj 41  
01331-000 São Paulo  
SP Brazil  
55-11-253-7869

### International Data Corp. Chile

Luis Thayer Ojeda 166 Piso 12  
Providencia, Santiago 9, Chile  
56-2-231-0111

### IDC Colombia

Carrera 40 # 103-78  
Bogota, Colombia  
571-533-2326

### IDC Mexico

Select - IDC  
Av. Nuevo Leon No. 54 Desp. 501  
Col. Hipodromo, Condesa  
C.P. 06100 Mexico, D.F.  
52-5-256-1426

### IDC Venezuela

Calle Guaicapuro  
Edif. Torre Seguros Alianza  
Piso 6, Ofc. 6-D, El Rosal  
Caracas 1060, Venezuela  
58-2-951-3270

IDC delivers accurate, relevant, and high-impact data and insight on information technology to help organizations make sound business and technology decisions. IDC forecasts worldwide IT markets and adoption and technology trends, and analyzes IT products and vendors, using a combination of rigorous primary research and in-depth competitive analysis. IDC is committed to providing global research with local content through more than 500 analysts in more than 40 countries worldwide. IDC's customers comprise the world's leading IT suppliers, IT organizations, and the financial community. Additional information on IDC can be found on its Web site at <http://www.idc.com>.

IDC is a division of IDG, the world's leading IT media, research, and exposition company.