APPLICATION LINK ENABLING (ALE)

Technology Marketing
Bernd Killer

Business Integration
of Distributed Applications

Version 1.0/July 1995

APPLICATION LINK ENABLING (ALE)

Business Integration of Distributed Applications

With R/3 Release 3.0, SAP is launching its pioneering ALE (Application Link Enabling) initiative for distributed application systems. ALE permits efficient, reliable business communications, thus achieving a high degree of integration between technically separate systems. Distribution models and technologies for linking together applications, as well as tools for the design and operation of distributed applications, are made available in the form of an optimized architecture.

Within the scope of business reengineering, users are calling for cost-effective concepts for distributed application systems, which despite requiring a high degree of business integration can also be decoupled for local use. SAP's ALE initiative is making available business and technological innovations for business-process-driven communication between technically independent applications.

Frequently demanded distribution scenarios are preconfigured within the scope of R/3 Release 3.0 in the form of distribution models, thus providing a tested basis for customer-specific business solutions. For example, ALE supports business processes with distribution of accounting and logistics, sales and distribution, central and local profitability analysis, and central and local sales and operation planning (SOP).

ALE loosely couples distributed application systems. A configurable distribution model enables exchange of business messages and master data and comparison and adjustment of control information. When setting up and operating distributed application systems, this also ensures the manageability and usability of comprehensive scenarios: so-called distributable process units (e.g. "inventory management") yield internally consistent distribution.

The ALE technology is based on controlled, timely exchange of business messages. Synchronous and asynchronous communication mechanisms are the basis for demand-driven application integration, thus eliminating the need to link them via a central database.

The Needs of Modern Businesses

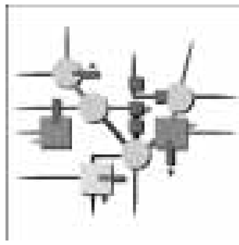
Changing market requirements and fiercer competition mean that corporate business processes are subject to unrelenting pressure to be changed and modified. The key factors driving these changes are global competition, ever-shorter production and product cycles, and falling profit margins.

Speed, flexibility, and simplicity are the crucial ingredients for the success of business reengineering. But the individual processes involved are not growing simpler: increasingly, they are being extended to tightly embrace multiple companies, customers, suppliers, and autonomous business units within the enterprise.

Users are therefore calling for economical, easy-to-use concepts for distributed applications, which—despite requiring highly integrated business application systems—can be decoupled to permit temporary stand-alone operation. Increasingly, integration into virtual production and service environments means distributing value-creation chains beyond corporate boundaries.

The move from sequential processing of functions to network-oriented process flow with electronic communications is primarily an organizational issue, not a technical one. In many cases, however, reengineering projects are unable to practically implement cost-effective IT architectures, not least because they lack a technologically reliable, high-performance approach to distributed utilization of information.

Rapid adjustment of business processes: When a company adjusts its product portfolio to focus better on its core competences, organizational measures are inevitably involved: in all cases, make-or-buy decisions must be made that can, for instance, generate differently structured supply relationships (e.g., a switch to single sourcing or JIT). The outsourcing vs. insourcing debate can prompt a partial or complete shift of production elsewhere, distributing it among various decentralized locations. Possibilities open up for selling or buying companies or striking strategic alliances.



ALE - the Business Initiative

ALE: a business approach for design, implementation, and operation of integrated distributed applications

Customer orientation: Customers' organizational and business requirements determine the roll-out of distributed applications.

Technological quality: Since the R/3 Reference Model is the starting point, business integration is methodologically and technically ensured. Applications control communications on the basis of cross-system business processes.

Reliable distribution: Distributable function types form distribution objects and ensure complete distribution of sub-processes. Filter mechanisms guarantee allocation of information to the correct areas. This eliminates uncontrolled dispatching of messages.

Efficient implementation: Process models (scenarios) are included in the new R/3 release, offering all the well-known benefits of this standard software (model-based, methodologically reliable, preconfigured, customizable, continually updated by fresh releases, and maintainable).

Greater independence: Autonomous business units responsible for their own activities administer their own data and applications while simultaneously being integrated into the overall company's coordinated IT architecture.

ALE: an economical solution for integrating distributed applications

Release independence: Implementation mechanisms guarantee transfer of messages while automatically adjusting the data to ensure compatibility with different release levels of applications. It used to be necessary to implement all local applications at the same time, but now this can proceed in stages.

Simple maintenance: Loosely coupled applications are integrated into a release-independent overall architecture. This also applies to system software upgrades (operating system, DBMS, etc.).

Tailored to needs: Communication takes place when and as required. So everything stays more up to date and the network load is balanced better. Strictly process-based views ensure that only genuinely needed data is transferred. Data is also compressed prior to transfer.

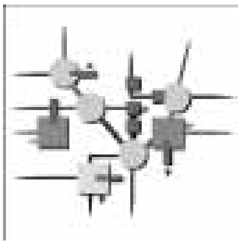
Reduced complexity: Graphical tools ensure manageable, consistent support of the distribution model for design, implementation, and operation of integrated distributed applications.

Greater flexibility: Greater independence from vendors and releases of distributed applications, plus the ability to integrate third-party systems.

Flexible design of processes: Internal corporate responsibilities tend to change faster than the IT infrastructure supporting them. Because of this, autonomous business units need to be designed on the basis of flexible structures, which must be quick to implement yet easy to adjust, well documented, and simple to maintain during their useful life. In addition, it must be possible for business objects to communicate with one another between units and thus also between applications.

Reduced complexity: Decoupled deployment of information systems is high on today's agenda: time-independent installation and operation of information systems have enormous potential for reducing complexity while enhancing security and boosting efficiency. By contrast, implementation and maintenance of distributed applications—to the extent that these are possible at all—are very labor-intensive.

Optimized use of resources: No matter how well thought-out, lean, or integrated business processes are, it is primarily economic considerations that determine their technological feasibility. The system architecture should permit handling of temporary bottlenecks; even better, these should be avoided in the first place, eliminating the need for costly repair work. When "legacy systems" are properly implemented and stable, they should be brought into the fold to permit their continued use, keeping the option open of integrating them into new distributed architectures, if this is expedient from a business standpoint.



Accepted procedure: Workers must identify with the system's structure and, espacially, with the business processes. In order for this to work, the processes must be manageable. They must be as easy as possible to use in order to encourage staff to acceptance responsibility for them. Structures and bussiness processes must remain dynamically adjustable, and stuff must experience them as such. User requirements thus focus on ease of use, customizing, and high-quality system support (e.g., workflows for handling errors and problems).

Architecture for Distributed Business Processes

Older Approaches Beset with Problems

The implementation and operation of enterprise-wide distributed application systems still rank among the greatest IT challenges. The complex solutions developed in the past do not entirely meet the diverse requirements of users. Yet the approach of using ”distributed databases” in connection with distributed application systems, which have been repeatedly brought into the game, is still impracticable.

The main reasons for this are as follows:

1. The same release levels of the operating system, DBMS, and applications have to be installed at all network nodes. Performing an upgrade on just one of the systems is very labor-intensive and makes all of the other applications temporarily unavailable.
2. Application systems perform consistency checks that are important for security reasons. No database system can completely assume this function. So-called stored procedures can replicate part of the application logic involved, but this involves a tradeoff as both instances must then be maintained. Another drawback is that the logic cannot be ported to the database systems of other vendors.
3. Local systems can be customized differently. But then the variants have to be reset for every data transfer.
4. It must also be possible to easily incorporate customer modifications into the distribution system.
5. It is still problematic to access large volumes of distributed data across large distances (because of high error rates, large network loads, and lengthy response times).
6. System administration is difficult (requiring, for example, recovery at different locations).

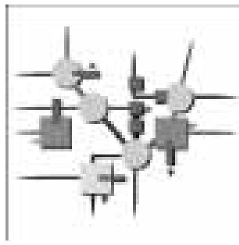
Other approaches—involving implementation and maintenance of a large number of interfaces for exchanging data in batch mode between heterogeneous application, or integration of distributed applications exclusively on the basis of the possibilities offered by Electronic Data Interchange (EDI)—turn out, under closer scrutiny, to also be inadequate, too inflexible, or impracticable in many cases.

Requirements of Modern Distribution Architectures

When distributing task areas, it is essential for the relevant cross-enterprise processes to have well-developed, functioning communications at their disposal. Fast and direct access to information is an absolute must: for example, a sales department may require information about the finished products the company’s plants have in stock.

Within the context of business process reengineering, the need for economically sensible concepts for distributed IT systems is growing dramatically. User requirements are mainly focusing on:

- Highly integrated business application systems as a prerequisite for optimally designing business processes
- Decoupling applications so that they can be operated locally and independently of the underlying technology
- Overcoming the limitations of existing integration concepts: rigid links, close (release) dependencies, high complexity, distributed data structures without semantics (no application logic), redundancy, and high maintenance requirements



Application Link Enabling (ALE)

Business is what drives technology: to enable competitive companies to meet these requirements, R/3 Release 3.0 introduces SAP's pioneering ALE initiative (Application Link Enabling) for distributed applications. ALE enables efficient and reliable business communications, thus achieving highly integrated business processes in technically independent systems. Distribution scenarios and communication technologies for linking together applications, as well as tools for designing and operating distributed applications, are made available within the scope of a harmonized architecture. Distributed applications can be defined and operated between R/3 systems, between R/3 and R/2 systems, and between R/3 systems and external applications.

Based on business distribution scenarios, applications exchange process-related messages. Messages and data that the user has characterized as necessary are distributed via defined channels. Local business units thus quickly receive the data they need. Consistent and complete distribution is ensured by linking related functions and data assigned to them to create distribution objects.

This gives users complete control over the processes involved in distributing "their" data. For the first time ever, the semantics of distribution—in other words, knowledge about the business-process-related utilization of distributed data—is available from within the application itself. This goes far beyond the capabilities of stored procedures, which have also been extensively used in connection with distribution databases (mainly for management of structural information).

ALE takes care of the "what" and "where to" of distributed business processes, while the database system continues to be responsible for "how". The task breakdown is based on the principle that each participating system component (application, database system) should assume the tasks it is able of along the lines of an optimized business distribution architecture.

SAP supports efficient distribution of R/3 application systems by furnishing a distribution reference model. This model contains scenarios for distribution variants that customers often ask for. They can be used to create customer distribution models.

To implement **integrated distributed systems**, you access an integrated graphical front end. The next step is to create a customer distribution model by specifying:

- The installation sites and tasks of R/3 systems
- Which applications are to run on which systems (R/3, R/2, external)
- Which master and movement data the applications should exchange
- Which control data the distributed systems needs to know

The following data types can be communicated, with the transmitting application coordinating the process at the business level:

Master data: e.g., material, customer, vendor, cost center, and the like. Master data is depicted in the reference model. As a rule, not all of the master data is distributed, but merely views of it, for instance sales views in the material master.

Transoptional data: e.g., customer order, purchase order, invoice, etc.; these are stored in the distribution reference model.

Control data: objects for system organization; customizing data like company code, languages, purchasing organizations, plants, user maintenance, etc.

Distributed Business Processes

Loosely coupled applications are based on a replicated and, consequently, redundant data instance, although with new synchronization rules. In contrast to distributed databases, with ALE only data that changes is distributed via the network ('you get what you need').

Changes to distributed data simultaneously trigger communications for asynchronous copying of these changes to the data records. This efficient synchronization of data distribution within the overall system keeps data more up to date than periodic updates in conventional distributed data storage schemes, and also avoicds placing a greater burdon on the network.

ALE Scenarios Meet a Greater Range of Requirements

The scenarios in R/3 Release 3.0 are based on actual experience, and meet most of our customers' requirements today. SAP has devised detailed descriptions of these scenarios and the associated business attributes (business processes), in addition to developing more scenarios.

The following example illustrates a hypothetical integrated scenario with three application systems communicating by means of ALE. Let's say that a corporate headquarters uses an R/3 system as an enterprise-wide reference system for master data and control tables for accounting, human resources, central purchasing, and sales planning, and as a global information system for logistics.

Change of scene: In a production facility, R/3 serves as a local SOP system for production planning and control (PPC). It also handles local purchasing and inventory management.

Another view, this time of a branch sales office: Here R/3 takes care of sales, distribution, inventory management, and purchasing of traded merchandise. Application Link Enabling (ALE) links together these distributed application systems (the basic configuration is, as already described, based on the individual distribution models) to enable exchange of business messages, change master data, and adjust control information.

ALE Distribution Reference Model

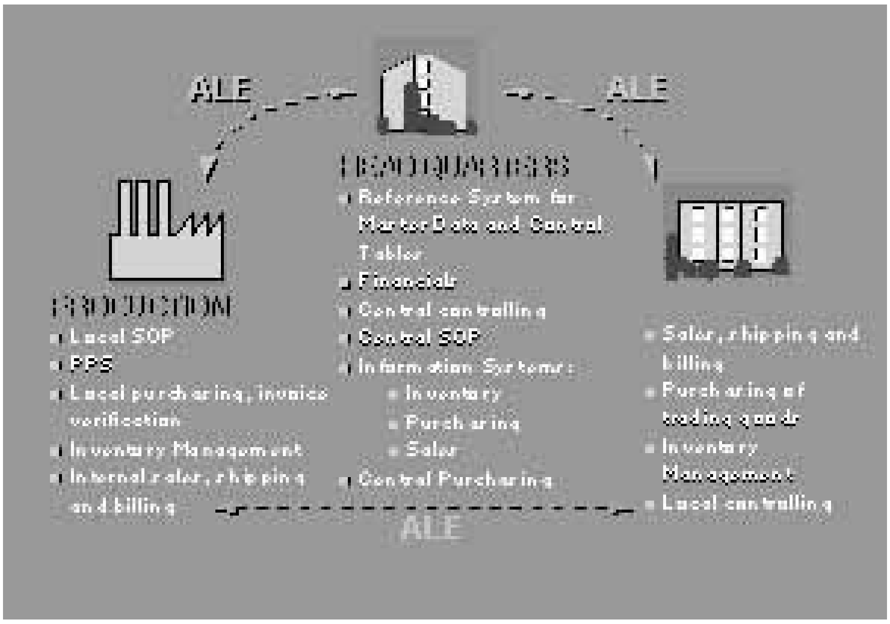
Possible distribution scenarios are stored in the distribution reference model. SAP's focus here was on ensuring that data can only be distributed if there is a good reason to do so. Consequently, functional types belonging together on the basis of business criteria are bundled into "distributable function types" like "inventory management" and "inventory control". In this way, the function types "goods receipt posting", "stock transfer", and the like are combined to form the distribution object "inventory management".

In this case, inventory management can run autonomously on a distributed system and exchange messages with other systems. The data and routines are kept consistent in every case.

Which data need to be transferred (control data, master data, transaction data) is also specified in the reference model. For example, if the inventory control module needs certain information (e.g., on goods movements) from the inventory management system in order to perform its processes, then the inventory management system passes this movement data to the inventory control system.

Also of importance in connection with integrated distributed application systems are so-called "filter object types". For instance, there can be more than one inventory management system in a network of distributed systems. Each inventory management system is responsible for one plant. Such allocation criteria are referred to as filter object types. For a given function type (e.g., inventory management), they stipulate the criteria for determining when multiple instances may exist within a distributed system.

Figure 1: Organizational units in a company



It is also generally possible to implement multiple, autonomously running, independent inventory control systems. The filter objects for this are then (as shown in the figure) plant, division, business area, or material. In the example shown, one inventory control system is responsible for plant X and plant Y, while another is only responsible for the top 100 materials.

To obtain detailed information about a distribution scenario, the distribution reference model is combined with the SAP process model. This contains, for example, information on which processes belong to a given function type, or where the interfaces linking two function types that exchange messages are located.

SAP maintains the reference model. The ALE distribution reference model is an element of the general reference model repository, and is supplied along with the R/3 System.

Scenarios: Numerous Distribution Possibilities

SAP discloses the available distribution scenarios in its distribution reference model, which is supplied as part of the standard system. R/3 Release 3.0, for example, supports the following:

- Linking of distributed sales organizations with a central sales and distribution department
- Distribution of profitability analysis between central and distributed units
- Linking of central financial systems and distributed logistics applications
- Local and central sales and operational planning
- Central management of purchasing contracts with local handling of procurement

From Reference Model to Customer Distribution Model

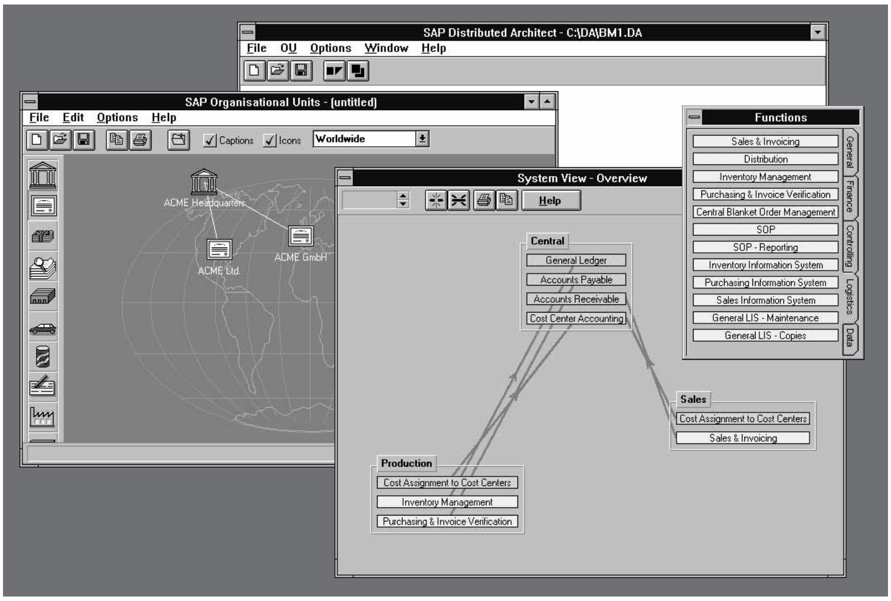
The function distribution scheme stored in the distribution reference model comes to life in customer distribution models, thus becoming tangible for users. Such customer models describe in detail how the functions are actually distributed. Accordingly, the customer model constitutes a specific manifestation of the (previously discussed) function types, distributable function types, message types, and filter object types of the reference model.

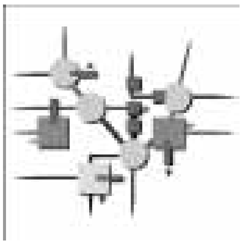
The customer distribution model also defines the control functions of the R/3 installation:

- Exactly how the data stored in the customer model describes the flow of messages between distributed logical systems
- How applications and functions of the ALE layer use the customer model
- How to identify the receiving system when dispatching a message

The user maintains the customer model. Definitions in the reference model support the individual steps involved in each case. Security mechanisms automatically reject definitions that violate the reference model. SAP provides a graphical PC tool for this, which lets you use ”drag and drop” techniques to graphically define organizational units, logical systems, and distribution scenarios along with their distributed systems and the usages to

Figure 2: Distribution architecture: drag & drop allocation of distribution objects to logical systems





be exchanged. Afterwards this is repieceted when cutomizing R/3. In accordance with the ALE conventions, customer models define the flow of messages between distributed applications running on different logical systems. Each system therefore requires data from the customer model. SAP has developed functions that take care of distributing the model data. They are based on a central maintenance system, from which the customer model is distributed to the other systems. This permits viewing of the model from any node.

In addition, all data that a given customer model uses to directly control the message flow can be made dependent on a validity date. This makes it feasible to make changes in advance and distribute them to the systems before they are due to take effect. They are later enabled at a predefined point in time. For example, data for a new logical system could be distributed but not activated until the beginning of the next fiscal year.

The principal advantages of this concept are:

- It supports creation of tailored customer models.
- The data of the customer model is distributed to all systems.
- The customer model controls data distribution. As a result, only the actually modeled data flows are executed.
- The model serves as a tool for configuring all scenarios. This means that there is no need to perform actions at multiple locations in the system; instead, modeling is done at one central location only.
- The model intercepts possible errors that can arise when configuring distributed systems.
- Users define a distribution customer model to specify how functions are actually distributed among the systems.

The ALE Architecture

Application Services

generate business messages and fill them with information. They support integrated workflow for processing of incoming messages.

The sending application initiates communications. It makes available business data and also defines the process for the receiving application.

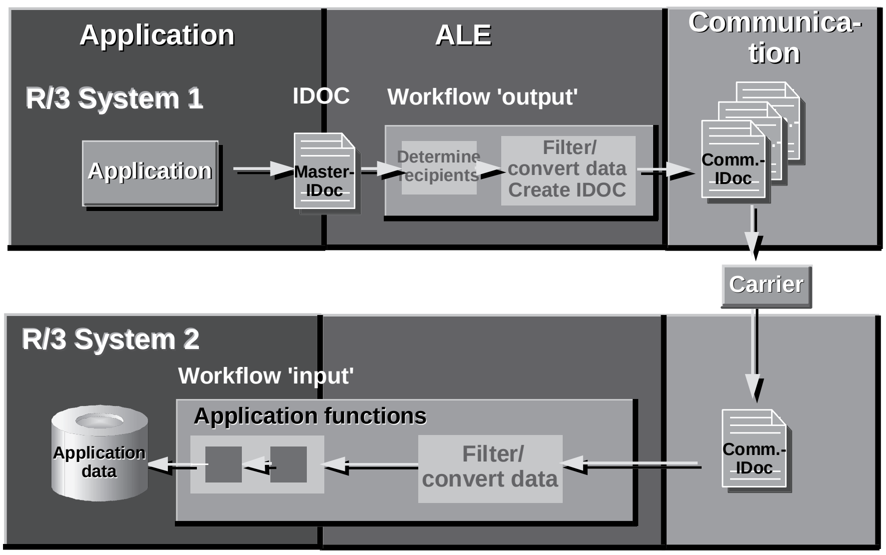
Processing Services

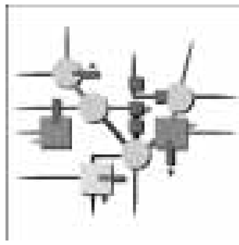
link the business level with the technical level. These include, for example, specification and checking of message recipients and filtering and conversion of messages. Links with R/2 or third-party systems are also possible.

Communication Services

ensure reliable transmission of data at the technical level. They support technical standards like X.400 as a standardized routing for e-mail, X.435 as an application-to-application extension of X.400, general messaging infrastructures based closely on e-mail systems, and software support for messaging (e.g., with Microsoft Exchange and Novell applications).

Figure 1: The principle of message-based application coupling





IDOCs as the Basis of ALE Distribution

Intermediate documents (IDOCs), well-known from the EDI interface, are the basis for smooth, reliable exchange of data in integrated distributed application systems. SAP devised additional IDOC and message types and extended their possible uses to take advantage of Application Link Enabling and meet these customer requirements:

- A documented, easily grasped structure, also for complex data (for instance, for integrating external systems or depicting packed fields)
- A compatible structure (no reprogramming of interfaces with each new release)
- An extendible structure (compatible extension within the scope of SAP developments, customer add-ons)

The IDOCs developed by SAP are data containers designed to ensure reliable, glitch-free exchange of messages between SAP systems or between SAP systems and third-party systems. IDOCs have a neutral data structure, independently of the specified application data. Consequently, third-party systems can also use them as a standard interface for data transfer. IDOCs consist of segments that may be hierarchically structured. Generally speaking, both SAP and customers may modify IDOCs.

ALE Layer Prepares IDOCs for Distribution

An application that has been configured for sending data to another system generates, in response to an appropriate event, a master IDOC containing the required data. Afterwards the IDOC is passed to the ALE layer, where it is prepared for transfer. Actual communication takes place in the communication layer.

After being transferred, the IDOC prompts processing of itself in the target system. The data is processed depending on the settings, which can range all the way to posting in a specific application—either automatically or with intermediate manual steps.

Processing of outbound and arriving IDOCs can be performed either individually or collectively. In the case of collective processing, several IDOCs are collected together into a packet and processed in a single step. The ALE layer contains serializing functions that automatically detect the sequence of data postings and execute them accordingly.

Communication with IDOCs between integrated distributed systems takes place asynchronously. In most cases, the data is immediately relayed after being posted in the application. Asynchronous communications means that the application does not depend on getting a reply from the target system. If a connection fails to be established—no matter what the reason—then the ALE layer initiates transmission at a later point in time and monitors it.

In synchronous mode, information is only queried. No data is distributed and posted. Synchronous requests are issued by directly calling a function in the target system.

Release Consistency Ensured

Compatibility of different versions is ensured by an extended concept with appropriate assignment of IDOCs to a message type and ALE functions for converting IDOCs between versions.

Moreover, customer extensions and enhancements are kept in a special dedicated data structure, thus making them release-independent. In SAP applications that write and receive IDOCs, customer exits are called at standardized locations to let the user fill and process added segments. These customer exits remain constant from release to release.

Customer extensions can—once they have been implemented—be further developed in step with enhancements to the standard SAP system. Customer IDOCs remain in the new IDOC version after changing releases. Consequently, a new SAP release can still write to and read from a previously defined structure.

Integration of Heterogeneous Systems

R/2-R/3 Links

Application Link Enabling provides far-reaching possibilities to link distributed R/2 and R/3 Systems to meet specific requirements. The R/3 Migration Tool ensures distribution of data from R/2 to R/3.

Distribution between R/2 and R/3 Systems is implemented by way of an ALE server (in the R/3 system). It is based on continual migration of objects from an R/2 System to an R/3 System. In the same way, the ALE server distributes data to one or more target systems, including R/3 Systems. The ALE server also converts data formats and transfers files between the host system and the R/3 instances.

Conversely, R/3 data to be distributed is addressed to the R/2 System. The ALE server takes care of normal processing of incoming data (filtering, conversion, comparison, etc.). Afterwards the incoming IDOC is relayed to the EDI interface of the R/2 System (Release 5.0f or later).

The individual steps are as follows: first the new port is written, then the IDOC is placed at the output, and then the function module for sending it is invoked. Transmission to the R/2 System takes place via CPI-C. In the R/2 System, it is recognized as an incoming IDOC, and another application module finally posts the IDOC in the R/2 System.

ALE Permits Links to Third-Party Systems

Besides the already mentioned ALE possibilities—like implementation and operation of technically autonomous systems, release independence of different applications, effective business process integration between diverse systems, utilization of standardized communication techniques and mechanisms, and avoidance of bottlenecks when processing large data volumes—one other aspect also possesses importance in connection with Application Link Enabling: linking of non-SAP applications.

The standard "IDOC" interface also permits connection of third-party systems to R/3. This is especially useful for coupling warehouse and merchandise management systems. But distribution scenarios are generally applicable to all third-party systems.

Data from external systems has to be converted into IDOC formats. Via a defined input port, the IDOC is then read into the R/3 system. In the opposite direction, the same sequence of steps is run through backwards.

Various software vendors offer migration tools for transferring data between R/3 and other systems.

ALE and EDI Supplement one Another

From the standpoint of users it makes no difference whether data communications are based on an EDI or ALE scenario, since ALE can also be superimposed on the EDI interface. Applications automatically detect whether data needs to be sent, and generate master IDOCs. Which connections are used for communication with IDOCs depends on the settings made for processing of outbound messages (ALE or EDI port? Do partner agreements exist?).

ALE generalizes EDI-based communications:

- The standard EDI protocols and formats are conventions that have been agreed upon industry-wide (e.g. ANSI.X12, EDIFACT), while ALE formats can be individually defined and modified at the enterprise level.
- ALE performs transmissions directly, by calling a function module (RFC) or a workflow in the remote system.
- In the case of EDI, an EDI subsystem (inserted between the sending and receiving systems) performs the actual transfer.