Chapter 2
Why Are Enterprise Applications So Diverse?

Abstract Today, even small businesses operate in different geographical locations and service different industries. This can create a number of challenges including those related to language, currencies, different regulatory requirements, and diverse industry expectations. For large organizations with a wider reach, the challenges are even greater. As organizations grow, they also need to keep track of huge amounts of information across different business areas. Modern enterprise applications need to be able to cope with these demands in a timely manner. In this chapter we provide a brief introduction to modern enterprise applications, describing selected tasks they perform and the business areas that they cover.

Enterprise applications are software systems that help organizations to run their businesses. They can add a degree of automation to the implementation of business processes as well as supporting tasks such as planning, data analysis, and data management. A key feature of an enterprise application is its ability to integrate and process data from different business areas, providing a holistic, real-time view of the entire enterprise.

Ideally, an enterprise application should be able to present all relevant information for a given context to the user in a timely manner, enabling effective decision making and allowing business departments to optimize their operations. This differentiates enterprise applications from other business software like spreadsheets, which are unable to pull data automatically from all the relevant data sources. Another factor that distinguishes enterprise applications from other types of software is that they are used exclusively in a business setting.

In this chapter, we describe the scope of an integrated enterprise application and resulting requirements (Sect. 2.1). Section 2.3 presents some selected enterprise application examples. The architecture of enterprise applications is discussed in Sect. 2.4. Resulting data access patterns are described in Sects. 2.5 and 2.6 closes this chapter.

H. Plattner and A. Zeier, *In-Memory Data Management*,
DOI: 10.1007/978-3-642-29575-1_2, © Springer-Verlag Berlin Heidelberg 2012

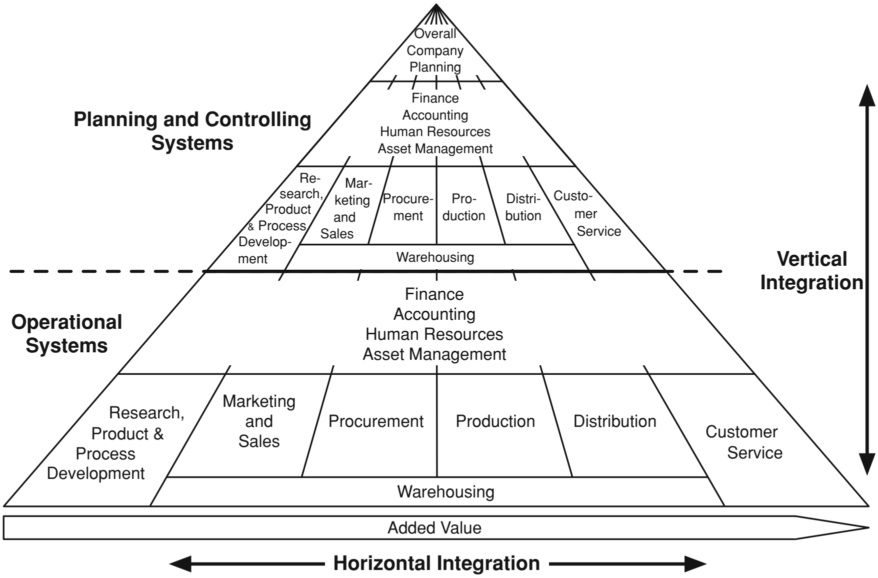

Fig. 2.1 Integrated enterprise information processing (based on [112] and [188])

2.1 Current Enterprise Applications

To get a sense of the variety of functionality that an integrated enterprise system has to cover, we present a view on an enterprise with regard to its divisions in Fig. 2.1. An enterprise system is in charge of integrating business processes that span multiple divisions (horizontal integration). It connects planning and controlling systems with operational systems (vertical integration).

Enterprise applications not only integrate information processing in homogeneous companies, but also cater to a wide range of different customer groups. For example, SAP Business Suite, the leading integrated Enterprise Resource Planning (ERP) package, is used within a large number of industries ranging from Aerospace and Defense, Telecommunications, and Banking to Industrial Machinery and Components Manufacturing. This leads to a situation of diverse and sometimes contradicting customer demands. For example, the product development and go-to-market lifecycle in the high-tech industry is less than six months, whereas in the chemical, railway, or oil and gas industry such a lifecycle can amount to decades [50, 161]. Companies in each of these diverse industries expects to enterprise system to be tailored for their requirements. To cope with this situation, successful enterprise applications have to allow parameterizations and the creation of industry-specific solutions. This includes giving customers the ability to determine how to query their data and enabling them to specify extensions to the underlying database tables if they need to include information that is not currently stored in the system. Another important customization feature in enterprise software is the ability to adapt existing business processes to customer-specific requirements.

International companies work across multiple time zones and rely on their software being available at all times. No downtime is acceptable. Enterprise applications must meet this requirement. This includes providing mechanisms for recovering from a wide range of failures, including power failures, data loss, and user input failures. Enterprise applications that are used in a number of different countries have to be internationalized to allow users to interact with them in their own language. They have to be compliant with all relevant national and international regulations and laws. This means that the software should be able to be easily updated because laws change on a regular basis, for example, with regard to income tax structures and tax rates. It is also often necessary for a company to maintain historical data for a number of years. The enterprise application should support this but should also make sure that the ever-growing data volume does not impact the processing of current data.

2.2 Examples of Enterprise Applications

We present below a subset of enterprise applications a company might use. We describe them in more detail throughout the rest of this section to provide the non-business reader with a sense of what enterprise applications do. For illustration, our company of reference has the following use cases:

- Forecast of product demand is needed for the following planning horizons. The forecast will be used as a basis for more detailed production planning (Demand Planning).
- Customers contact the company and order products (Sales Order Processing).
- Availability of the requested products is checked with every new order, and customers are informed if, and when, products can be shipped (Available-to-Promise).
- Some customers fall behind on payments for products received. Therefore, payments must be tracked and reminders generated for outstanding invoices (Dunning).
- The company performs a sales analysis each week (Sales Analysis).

Innovative enterprise applications that were not possible until now will be presented in Sect. 6.2.

Demand Planning

Demand planning is used to estimate future sales by combining several data sources. These include current and previous sales, new product introductions, product discontinuations, market forecasts, and other events with possible impact on buying behavior. The resulting outcome influences production planning, which determines the quantity and schedule for product manufacturing.

A demand planning application has the following characteristics: a single demand planning run for the company involves a large amount of data (up to 100 GB in larger companies). The data is created by hundreds of thousands of different products, their respective variations and configurations, and the fine-grained timely planning levels that allow planning on a daily or hourly basis.

The main operations on that data are aggregation and disaggregation because with every planning run, all current numbers are aggregated. Then, changes on a higher level in the planning hierarchy are applied, which are disaggregated to the finest granular level. The operation usually involves far more read operations than write operations, since all numbers are read, but only a fraction of them is actually changed. The underlying database system must be able to perform these operations with sub-second response time, as a human planner continuously works with the system in an interactive manner. Due to multi-user concurrency, isolated user contexts must be created to guarantee a consistent view on the data while planning. This acts as a multiplier for the overall storage and computing requirements.

We can see that this application has analytical aspects, but that it also requires real-time access to operational data. A combined transactional and analytical solution is required here.

Sales Order Processing

The main purpose of sales order processing is to capture customer orders. A sales order consists of a general section including customer name, billing address, and payment method. Details about the ordered products are stored as line items. Sales order processing involves read and write operations on transactional data. Read operations are required to check availability, item location and price. They are simple read operations, while write operations insert new sales orders and corresponding line items. More read operations than write operations occur and all transactions involve small amounts of data processed by highly predictable queries. Sales order processing has to have a fast response time, since clerks use the application interactively.

Available-to-Promise

The decision to accept an order for a product depends on the successful completion of an Available-to-Promise (ATP) check. The ATP application determines whether sufficient quantities of the product will be available in current and planned inventory levels for the requested delivery date [113]. To support flexible operations such as reordering where a high priority order overrides previously accepted low priority orders, ATP must aggregate across pending orders, rather than just materializing inventory levels. ATP encompasses read and write operations on large sets of data. Read operations dominate and are utilized to aggregate the respective time series. Write operations work on the fine-granular transactional level, such as to declare products as promised to customers. Variance in data types and records is small. The data volume grows only by adding new products, customers or locations.

Dunning

Dunning is the process of scanning through open invoices to identify those that are overdue, generating reminder notices for corresponding clients, and tracking which notices have been sent. It involves read operations on large amounts of transactional data. Either the read operations are batch-oriented, as the list of open items gets processed customer by customer, or the read operations are analytical queries, resulting in data aggregation, for example, when the overdue invoices for a selected customer group in a certain fiscal year must be determined. Read operations access very few attributes of the open items. Inserts are needed to keep track of issued payment reminders. Updates are needed to modify a customer's master data, such as the current dunning level, after a dunning run.

Sales Analysis

Sales analysis provides an overview of historical sales numbers. Usually, the analysis is used for decision making within an organization. The process involves read-only operations on large amounts of transactional and pre-aggregated data. For example, a sales manager might analyze sales volume by product group and country and compare it to previous years' numbers. Sales analysis comprises recent and historical data. The queries are multidimensional and usually scan and aggregate large amounts of data.

2.3 Enterprise Application Architecture

Current enterprise applications comprise client devices, application servers, and a persistency layer. Different enterprise application vendors implemented the business logic at different layers. For example, PeopleSoft preferred to concentrate the business logic at client devices whereas Oracle relies on stored procedures close to the database in the persistency layer to implement business functionality. SAP took a third option of having a thin client while all the business logic resides at scalable application servers. SAP's approach is shown schematically in Fig. 2.2.

In reality, the given figure would be more complicated, for example, as there is often redundancy added to the system at the database management system and application servers to increase performance. When this model was introduced with SAP R/3 about 20 years ago, the underlying rationale was that the database server was the limiting factor and as much load as possible should be shifted from the database server to multiple application servers. Usually multiple database management systems are supported by enterprise applications resulting in the situation that only

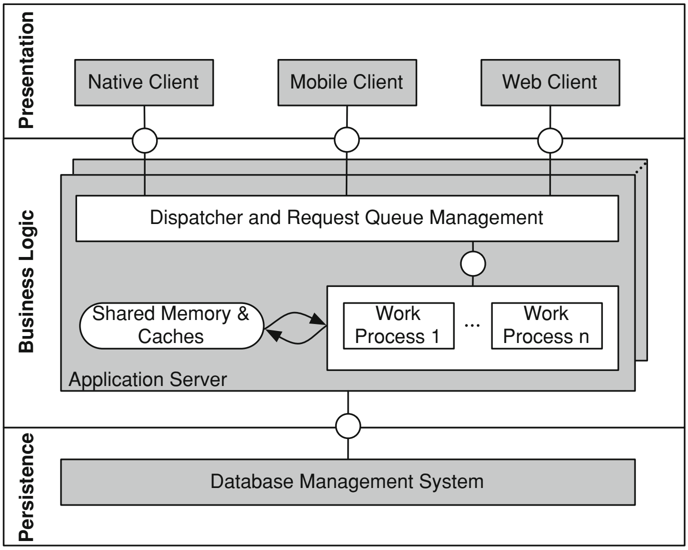

Fig. 2.2 Three-tier architecture of SAP R/3

the subset of all supported systems can be used in application development. This has led to a situation in which the application servers have become cumbersome and complex. One of the reasons is that application caches were introduced to store intermediate database results and all the computation is conducted in this tier to minimize database utilization.

2.4 Data Processing in Enterprise Applications

The data processed by an enterprise application comes in a number of different forms. Our customer analysis revealed that, for example, a medium-sized enterprise system today contains 100 GB of transactional data and 1 TB of read-only analytical data. Values for a large company are more than 35 TB of transactional data and 40 TB of read-only analytical data.

Processing this amount of data for planning and controlling purposes or to create reports required by management can have a significant impact on the performance of the overall system [124]. To alleviate these effects, most current enterprise applications separate these tasks into BI applications on the one hand and day-to-day transaction processing applications on the other hand. BI applications are characterized by read-intensive operations over large portions of the entire data in an enterprise system. This can comprise predefined reports run on a regular basis or complex ad-hoc queries—often requiring to aggregate large amounts of data. These queries can be in response to management questions formulated with regard to a current problem or an interesting aspect within a company, such as the current development of the company in a specific country or region of the world. These types of queries do not perform well on traditional transactional database schemas that are optimized for writing. This means that analyzing data stored in a number of different tables requires many tables to be joined, with a severe impact on performance.

BI applications tackle these problems by pre-aggregating data and storing the results in special read-optimized schemas. Analysis if this data is only possible along pre-selected dimensions [170]. In these read-optimized schemas certain attributes are duplicated across numerous tables to reduce the number of joins required. The pre-aggregated data is updated regularly by extracting data from the transactional system, transforming it according to the aggregation rules and the read-optimized schema, and then loading it into the BI application [93].

This separation of enterprise systems into those that perform OLTP and those that perform OLAP suggests a precise division of enterprise applications into transaction processing applications and analytical applications. This is far from the truth, since the previously discussed applications and others perform transaction processing and analytical tasks.

2.5 Data Access Patterns in Enterprise Applications

For the impact on the underlying in-memory data management system, it is relevant to distinguish between processing one single instance of an enterprise entity and the processing of attributes of a set of instances. A single instance is, for example, a sales order or a single customer. Set processing happens, for example, when all overdue invoices are read, or the top ten customers by sales volume are shown. Although transactional systems work on single instances of objects, most of the data in an integrated enterprise application is consumed by set processing [99].

Enterprise applications span operational systems as well as planning and controlling systems (Fig. 2.1). We can conclude that enterprise applications have a mixed workload [100].

2.6 Conclusion

In this chapter, it is evident that enterprise applications perform a very wide range of tasks and process large amounts of data. These data volumes are placing limits on the timeliness of data available for analysis, resulting in the need for separate analytical systems. This division of enterprise applications into two separate systems, which was conducted because of technical reasons, has a number of disadvantages. Therefore, some of the core themes that we focus on in this book are the technologies and techniques that can be employed to remove the need for separate analytical systems. These improvements allow enterprise application users real-time access to all the data in their enterprise system and to have a single source of truth.

Springer

http://www.springer.com/978-3-642-29574-4

In-Memory Data Management
Technology and Applications
Plattner, H.; Zeier, A.
2012, XXXIV, 267 p. 99 illus., Hardcover
ISBN: 978-3-642-29574-4