Chapter 2: ERP (Enterprise Resource Planning) Systems

1. Introduction

In today's highly competitive and fast-paced industrial environment, companies must coordinate a wide range of business processes such as production, inventory management, sales, purchasing, accounting, and human resources in an efficient and integrated manner. To achieve this, many organizations rely on **Enterprise Resource Planning (ERP)** systems.

An ERP system is a comprehensive, modular software solution designed to centralize and streamline a company's data and operations across all departments. By offering a unified platform, ERP systems enhance decision-making, improve productivity, and ensure consistency throughout the organization.

This chapter provides an overview of ERP systems, starting with their **definition**, **key functionalities**, and **main characteristics**. We will then explore the **different types of ERP systems**, from proprietary to open-source solutions. Special attention will be given to **ODOO**, a widely used open-source ERP platform, which we will examine in more detail to understand its architecture, modules, and practical applications in industrial settings.

By the end of this chapter, students will gain a solid understanding of how ERP systems function and why they are critical tools for modern industrial enterprises.

2. What is an ERP?

ERP stands for **Enterprise Resource Planning**, which is the English equivalent of the French term **PGI (Progiciel de Gestion Intégré)**.

An ERP is an integrated software solution designed to manage **all of a company's processes** by incorporating all its functions **(**Figure 2) such as **human resources management**, **accounting and financial control**, **decision support**, as well as **sales**, **distribution**, **procurement**, and **e-commerce**.

Figure 2 Different Modules in an ERP System

The main objectives of an ERP system are:

-To **network** all company functions through a **single, unified database**;

-To **centralize and harmonize** information flows across departments;

-To **increase the overall performance** and efficiency of the enterprise;

By providing real-time access to reliable data, ERP systems support better decision-making, improve coordination, and reduce redundancies.

3. Principle of ERP Systems

ERP systems are a **set of configurable software applications**, organized into **independent modules** that each manage a specific business function.

While these modules operate independently, they **share a single, centralized database [3]**, ensuring data consistency and real-time information flow across the entire organization (**Figure 3)**.

This modular and integrated architecture allows companies to adapt the ERP system to their specific needs while maintaining **coherence and unity of information**.

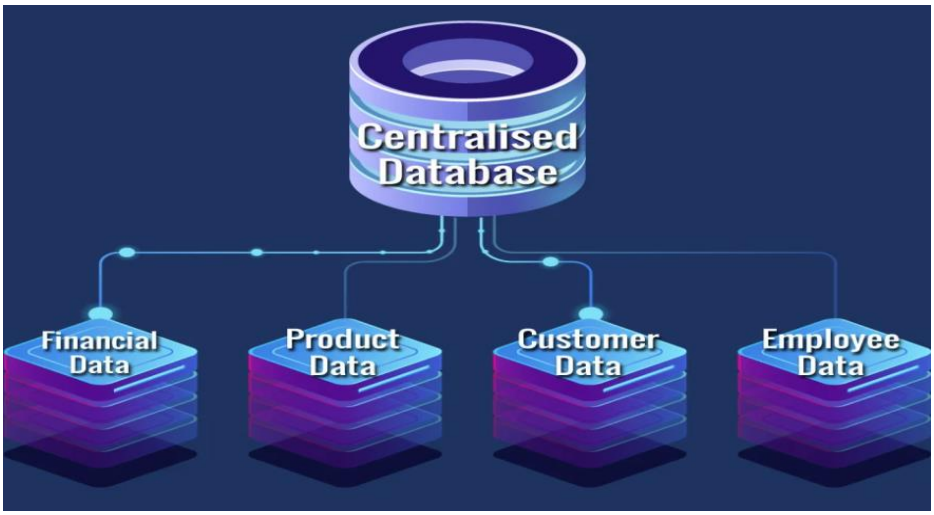

Figure 3 Centralised database architecture
***Source*: SOLIDitech. (2021, July 13). *An Introduction to Centralised Databases* [Image]. The SOLID Blog. https://blog.soliditech.com/blog/an-introduction-to-a-centralised-databases**

4. ERP Functionalities

ERP systems offer a wide range of functionalities that cover the core activities of an enterprise. These include, but are not limited to:

- **Marketing Management**
- **Human Resources Management**
- **Supply Chain Management**
- **Sales Management**
- **Customer Relationship Management (CRM)**
- **Financial and Accounting Management**
- **... and many other business processes**

Thanks to its modular structure, an ERP system allows each company to activate only the functionalities that correspond to its specific operational needs, while still benefiting from an integrated and unified information system.

5. Key Characteristics of an ERP System

ERP systems are defined by several essential characteristics that make them powerful tools for managing complex organizations:

-Use of a Single, Centralized Database→All modules share a common database, ensuring data consistency and real-time information access.

-Integrated Functionalities Across All Management Areas→From finance and HR to sales, logistics, and customer service, an ERP covers the full range of business functions.

High Configurability→ERP systems offer a large capacity for customization and parameterization to fit the specific needs of each organization.

Open and Scalable IT Architecture→Modern ERP systems are designed with open architecture, allowing for interoperability, scalability, and integration with other software tools.

Speed and Efficiency→By streamlining processes and eliminating redundant tasks, ERP systems improve productivity and response times.

Cost and Time Savings→Centralized data and process automation reduce operational costs and speed up workflows because time is money.

Decision Support Tool→ERPs provide analytical tools, dashboards, and reporting functions to assist managers in making informed decisions.

Traceability and Data Tracking→ERP systems ensure full traceability of operations and transactions, enhancing control, accountability, and quality assurance.

5.1 Use of a Single Database: Historical Evolution of Management Information Systems

The evolution of management information systems has gone through several key stages:

Independent Applications: In the early stages, each business function (such as accounting, HR, inventory, etc.) relied on its own software application **Figure 4**, often with a separate database. This led to redundant data entry, inconsistent information, and limited communication between departments.

Figure 4 Independent Applications

Integration into an ERP System: With the advent of ERP systems, these separate applications were replaced by a **modular system** based on a **shared, centralized database Figure5**.

This integration ensures data consistency, eliminates duplication, and facilitates real-time access to accurate information across all departments.

This transition marked a major step toward the **digital unification of business processes**, significantly improving coordination and decision-making within the organization.

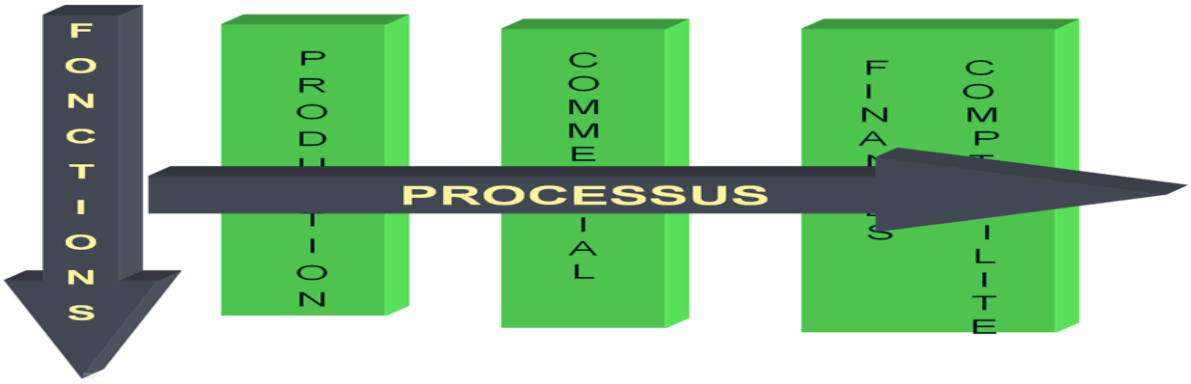

Figure 5 Integration into an ERP

5.2 Integration into an ERP System

In an ERP system, **each piece of data is entered only once**, at the moment the event that generates it occurs.

This data is then made **immediately available in real time** to all authorized users across the organization, through a **shared, common database** used by all modules.

The use of a **relational database** eliminates redundancies, ensures data consistency, and facilitates seamless information flow between departments.

5.3 ERP Modules Cover All Business Activities

ERP modules are designed to **cover the entire range of a company's activities**. The **richness of their functionalities**, combined with **integrated data**, allows for a **comprehensive and unified approach to management**.This integration facilitates better coordination between departments, improved decision-making, and an overall increase in organizational efficiency.

5.4 High Configurability

ERP systems offer a **high level of configurability**, allowing companies to tailor the system to their specific industry and operational practices.

-Database configuration can be adapted to the company’s business logic and workflows.

-User interface customization is available for each workstation, including:

-Access rights and permissions

-Personalized graphical environment

-Operation traceability and audit trails

Example:
The textile company Gilclaude needed to manage a product table containing 300 to 400 items, each with about twenty variants (size/color).

Thanks to advanced configuration, the company avoided the multiplication of item references by implementing a lot-based management system using generic product entries.

This kind of setup significantly simplifies inventory tracking and data maintenance, while keeping the system aligned with the company’s real-life practices.

5.5 Decision Support Tool

An ERP system provides **management control** with powerful tools to support strategic and operational decision-making. It enables:

-The use of **SQL query languages** from relational databases such as Oracle, SQL Server, or DB2

-Client/server access to the shared database, allowing multi-user and remote interactions

-The **creation of customized reports and dashboards**, tailored to users' specific needs

These capabilities make the ERP a valuable tool for analyzing performance indicators, monitoring operations, and supporting informed, data-driven decisions across the organization.

6. Types of ERP Systems

ERP systems can generally be classified into two main categories:

Proprietary ERP Systems: These are commercial solutions developed and distributed by software vendors. Their source code is closed, and they typically require the purchase of licenses and ongoing maintenance contracts (e.g., SAP, Oracle ERP, Microsoft Dynamics).

Open Source ERP Systems: These are ERP solutions whose source code is publicly available and modifiable. They offer greater flexibility and can be customized extensively, often at a lower cost (e.g., Odoo, ERPNext, Dolibarr).

Each type has its own advantages and drawbacks depending on the size, budget, and technical capacity of the organization (**Figure 6).**

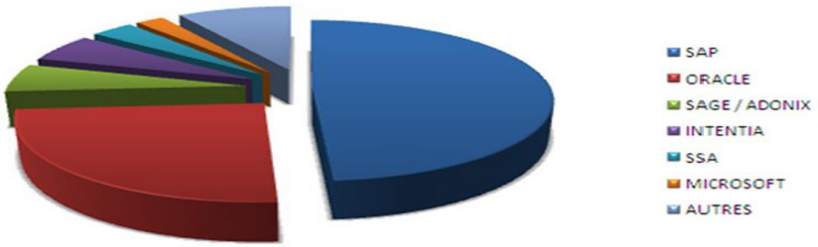

Figure 6 Distribution of leading ERP vendors in the market

7. The Odoo ERP System

Odoo, formerly known as **OpenERP** and **Tiny ERP**, is an open-source integrated management software that includes a large number of modules designed to simplify the overall management of a company.

With **over two million users worldwide**, Odoo is currently the **most popular open-source ERP system**.

Odoo offers two versions:

-Odoo Community Edition (Free) →This version is open-source and includes a wide range of essential applications for managing a business.

-Odoo Enterprise Edition (Paid) →This version includes all features of the Community edition, plus additional advanced applications and services such as:

Bank interfaces, Electronic signature, Integration with third-party platforms like **eBay** and **Amazon**

Thanks to its modular and user-friendly design, Odoo is widely adopted by companies of all sizes, from startups to large enterprises.

7.1 Odoo Architecture

Odoo is based on the **MVC architecture** (**Model-View-Controller**), a common design pattern in software development that separates concerns into three distinct layers:

Model

This layer manages the **data structure** and is linked to the **database**.Each object declared in Odoo corresponds to a **model**, which is mapped to a **table in PostgreSQL**.

View

This layer defines the **user interface**. Odoo uses **XML files** to design the views (forms, lists, dashboards, etc.) that users interact with.

Controller

This layer is handled by **Python classes** using Odoo's **ORM (Object-Relational Mapping)** system.Controllers manage the business logic and communication between the model and the view.

This architecture allows Odoo to be **modular, scalable, and easily customizable**, making it adaptable to a wide variety of business needs.

Figure 7 illustrates the MVC architecture of Odoo.

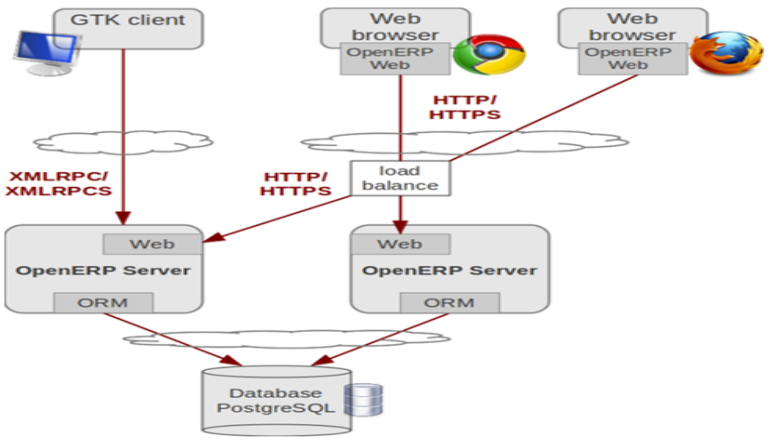

Figure 7 The MVC architecture

Modular Architecture of Odoo: Odoo's architecture allows for the development of software applications in a **modular way**, meaning that each module operates **independently** from the others while sharing a **common, single database (Figure 8)**.

This approach **eliminates multiple data entries** and **removes ambiguity** caused by duplicate or inconsistent data of the same kind, ensuring data consistency and integrity across the system.

Figure 8 Modular Architecture of Odoo

7.2 ORM: Object-Relational Mapping

Object-Relational Mapping (ORM) is a technique that establishes a link between a **class** (in object-oriented programming) and a **table** (in a relational database), as well as between each **attribute of the class** and a **field in the associated table**.

For example, the class Customer would be mapped to the table CUSTOMER, with the attributes associated as follows:

Customer ≈ CUSTOMER

Customer.customerId → CUSTOMER.CUSTOMER_ID

Customer.customerName → CUSTOMER.CUSTOMER_NAME

Customer.customerAddress → CUSTOMER.CUSTOMER_ADDRESS

This mapping allows developers to manipulate database records as Python objects, simplifying data access and logic implementation in systems like **Odoo**, which uses an ORM to manage models.

7.3 Languages and Technologies Used in Odoo

Python: is an interpreted, object-oriented programming language.(It is **interpreted**, meaning there is no explicit compilation phase.)

It was developed by **Guido van Rossum** at the **CWI (Centrum Wiskunde & Informatica)** in the Netherlands.

One of Python's key strengths is its **cross-platform compatibility** it runs on most common operating systems (Unix, Windows, etc.).

Python is also an **open-source language**, supported, developed, and widely adopted by a large global community of developers.

It is **widely used in web development** due to its simplicity and flexibility. Python's **clean and readable syntax** is very close to human language, which makes it **easy to learn and use**, even for beginners.

XML (eXtensible Markup Language): stands for **eXtensible Markup Language**. It is a computer language used to **store and structure textual data**.

XML is **self-descriptive**: the tags are **not predefined**, allowing developers to define their own tags based on the structure of the data.

It is a **text-based data format**, which means any XML file can be opened and read using a **simple text editor**.

An XML file consists of **tags** that are made up of **pure text**, organized in a hierarchical structure.

In Odoo, XML is widely used to define:**User interface views** (forms, lists, kanban, etc.), **menus and actions** and **access rights and workflows.**

HTML vs XML

HTML (HyperText Markup Language) →HTML is a markup language that uses **tags** to structure documents viewable on the **web**.

Limitation of HTML:HTML **mixes content and presentation**, which can make it difficult to manage or reuse the data separately from its display.

XML (eXtensible Markup Language) →Unlike HTML, XML provides a **strict separation between content and presentation**. It focuses on **storing and structuring data**, making it ideal for data exchange between systems.

XML Syntax

An XML document is considered **well-formed** if it follows these rules:

1) XML documents must begin with an **XML declaration**.
2) XML documents must have a **single root element**.
3) All XML elements must have a **closing tag**.
4) **Tags are case-sensitive**.
5) XML elements must be **properly nested** (no overlapping tags).
6) XML attribute values must always be **enclosed in quotation marks** (either single `'` or double `"` quotes).
7) **Entities must be used** for special characters (e.g., `&` for `&`, `<` for <, etc.).

Note: One should not confuse the notion of a **well-formed XML document** with that of a **valid XML document**. A **valid** XML document (**Figure 9)** is a well-formed document whose structure **complies with a predefined model**, described by a **DTD (Document Type Definition)** or an **XML Schema**.

DTD (Document Type Definition): defines the **grammar** or **structure rules** of the XML document.

```xml
<?xml version = "1.0" encoding = "ISO-8859-1" ?>

<!DOCTYPE bibliothèque SYSTEM "bibliothèque.dtd">

<bibliothèque>
  <livre catégorie = "BD">
    <titre>Gaston Lagafe</titre>
    <auteur>Franquin</auteur>
    <pages>61</pages>
  </livre>
  <livre catégorie = "Roman">
    ...
  </livre>
</bibliothèque>
```

Figure 9 Example of a valid xml file

DTD, XML Schema, XSL, and CSS:

A **DTD** (Document Type Definition) is either a **file** or a **part of an SGML or XML document** that describes the structure of the document. A DTD defines the **grammar** of the document listing the **elements (tags)**, **attributes**, their **content types**, and their **arrangement**.It can also include **a vocabulary of character entities**. ***Example***: a tag <age> may be required to contain a **positive number less than 105**.

An **XML Schema** is a **more sophisticated alternative** to DTDs.It provides a richer set of data types and constraints, and it is written in XML syntax, unlike DTDs.

XSL (eXtensible Stylesheet Language) refers to **stylesheet files** used to **transform** XML documents into other formats and to **format the output**.It enables the transformation of XML into HTML, PDF, or other structures.

CSS (Cascading Style Sheets) was created in 1996 to apply **style and formatting** to content. CSS can be used with XML to **visually format** data (e.g., using **colors**, fonts, spacing, etc.). Unlike XSL, which transforms content, CSS **styles the existing content** for display.

PostgreSQL

Is an **object-relational database management system (ORDBMS)**.It supports the use of **object-relational mapping (ORM)** to link classes and attributes in the application code to tables and fields in the database.

pgAdmin: a **graphical administration tool** for PostgreSQL, distributed under the **PostgreSQL license**.It allows users to **manage databases**, **execute SQL queries**, and **visualize structures** via a user-friendly interface.

7.4 Structure of an Odoo Module

All modules are located in the server/addons directory.

To create a new module, the following steps are required:

Create a subdirectory inside the server/addons folder.

Create a module description file: __manifest__.py (formerly __openerp__.py) → This manifest file contains the metadata and configuration of the module.

Create an __init__.py file to load the Python files and import objects.

Create the main Python file containing the model classes (business objects).

Create XML view files to define the user interface (forms, trees, etc.).

Additional directories and files may include:

wizard: Contains transient models used in wizard interfaces→ These are temporary interactive windows that help the user perform operations such as calculations or data entry. The data is deleted automatically after use.

workflows: Defines business processes and their sequential steps.

report: Contains **QWeb or XML templates** for generating PDF reports.

security: Defines **access rights, user groups, and security rules** through .csv or .xml files.

static: Stores **images, logos, stylesheets**, and static web assets.

i18n: Contains **translation files** (.po) to internationalize the module

8. Conclusion

The second chapter explored the crucial role of ERP (Enterprise Resource Planning) systems as integrated solutions for managing enterprise resources. These systems centralize and automate key processes such as inventory management, accounting, purchasing, production, and human resources management, thereby providing a coherent and real-time view of operations.

More specifically, the study of the Odoo ERP highlighted its modularity, open architecture, and flexibility, which make it a solution adaptable to the diverse needs of businesses, from small companies to large organizations. Thanks to its active community and numerous modules, Odoo facilitates the digitization of business processes while offering a user-friendly interface.

In summary, ERP systems play a fundamental role in the digital transformation of companies, and Odoo perfectly illustrates this trend with its accessible and comprehensive approach that combines customization and efficiency.

For further reading, one can refer to the work of Monk and Wagner, a major reference on ERP systems and their impact on modern enterprises [4].