# CEISAR Glossary Extract

## Principles

- Chosen words must be understandable by **Business** and IT Actors
- Definitions must be **short**. + possible extensions by Role
- **Upper letter** for concepts in glossary
- Use "-" when composed name
- No Homonyms
  - Service : Business-Service, IT-Service, Software-Service, by default means "Business Service"
  - Architecture: Architecture-Description or Architecture-Discipline, by default means "Architecture-Description"

## Action

Work executed by an Actor.
Recursive: an Action can be decomposed into Actions.
Can be a Process or a Function or an Activity.
Always named by a verb.

## Action-Model

Describes instructions given to an Actor so that he correctly executes the Action(s).

- For Manual-Actors, instructions are documentation (procedure, user guide).
- For Computer-Actors, instructions are software.
- For Assisted-Actors, instructions are documentation + software.

## Activity

Group of Functions of an Organized-Process executed by the same Actor at the same time

## Actor

**Human-Actor** or **Computer-Actor** (Automate) or **Assisted-Actor** (when Human-Actor and Computer-Actor are combined) executing an Activity.
An **Internal-Actor** (employee, consultant) works inside the Enterprise, while an **External-Actor** (customer, partner, provider) works outside the Enterprise but may use the same Enterprise Model.

## Agility and Reactivity

Agility is the ability to Transform fast and well. It allows to reduce the time between the rise of a new idea and its availability in the Enterprise Operations.
Reactivity is the ability to Operate fast and well.

## Architecture

2 meanings:

- Deliverable: **Architecture-Description** represents the structure (good or bad) of a Model (cf. norm IEEE 1471).
- How: **Architecture-Discipline** represents Transformation-Processes to Build the Architecture-Description

## Commodity-Solution

Their requirements are predictable. They do not allow differentiating from competitors.
Ex: Solutions for Accounting, Payroll…
Commodity Solutions were the first IT Solutions implemented in Enterprises.
Commodity Solutions are usually Built with a Contractual Approach. (See Evolving Solutions).
A Package offer generally exists for Commodity-Solutions because requirements are similar between Enterprises.

## Contractual-Approach

Build a Solution Model by defining all requirements before starting the IT Model.
This approach is adapted to Commodity Solutions.

### Cooperative-Approach

Construct a Solution-Model by successive Versions with a mixed team joining Business and IT competencies. Adapted to Evolving-Solutions. Also called **Agile-Approach**.

### Enterprise

An Enterprise is an agent which delivers Product to its Customer. The Product (Good or Service) can be economic or not (for example, cultural value).
Embraces not only capitalistic Enterprises but also governmental entity, university, research center, association…
An Enterprise can be a Legal Entity, a part of a Legal Entity, a network of Legal Entities.
A Group of Companies may represent a real Economic Entity with a unique decision center, without being described as one big Legal Entity. The Group teams represent an Enterprise, and each Company represents an Enterprise.

### Enterprise-Architecture

**Enterprise Architecture-Description** represents the Structure of the Enterprise Model
It generally is represented by Maps. which offer a Global View to better understand EA: Process Maps, Entity Maps, Function Maps, Solution Maps, Application Maps and Block Maps are the most commonly used. The most important is the Entity Map. CEISAR represents the Enterprise Architecture-Description according to a cube of 3 dimensions (see below):

- Complexity: isolates Enterprise Real World from Enterprise Model.
- Agility: isolates Enterprise Operations from Enterprise Transformations.
- Synergy: isolates Specific Elements from Shared Resources and Reusable Models.

**Enterprise Architecture-Discipline** represents the set of Transformation Processes to Build the Enterprise Model-Architecture.

### Entity

An Entity is a representation of a real world object ex: Mr Dupond, Contract of Mr Dupond, Account of Mr Dupond…
Entities which have same Attributes and same behavior (same Life cycle and same Functions) have the same Entity Model.
A **Business Entity** is necessary for Business, independently of the Organization of the Enterprise, such as Product or Customer or Contract or Account.
An **Organization Entity** is necessary for Organization purpose: Organization-Unit, Position, Profile, Right, Duty, Role… are all Organization-Entities

### Evolving-Solution

Their requirements evolve by the time. They often are Solutions to differentiate from Competitors.
Ex: Solutions for Front Office, Product Design, CRM, End to End Process, Business Intelligence…
Many Evolving-Solutions require specific developments which can be based on reusable Components.
Evolving-Solutions should be constructed with a Cooperative Approach.

### Foundation

Set of Reusable Models. Reusability perimeter can be the Solution, the Company, the Group, the same Business community, the country, or the world.
Reuse of Model is the way to create synergy and harmonize how the different Units of an Enterprise work.
**Operation Foundation** = Reusable Models for Operations which includes

- The Enterprise Model: Solution Map, Process Map, Function Map, Entity Map,
- Reusable Solution Model,
- Reusable Function Model,
- Reusable Entity Model,
- Reusable Process Pattern,

- Reusable Type,
- Reusable Human Actor Model,
- Reusable IT Actor Model

**Transformation Foundation** = Reusable Model for Transformation which includes

- Transformation Approach
- Transformation Tools like modeling tools

### **Function** (or Rule)

Action inside a Process. A Function may reuse other Functions.
A **Business Function** is independent from Organization chosen by the Enterprise.
Example: “check Information entry”, “compute price”, “send message to Customer”.
Not to be confused with **Organization Functions** which are added to implement an Organization.
The IT Model of a Function is sometimes called a **Software Service**.

### Information-Model (or Data-Model)

Defines Entities, their Relations and inheritance.
Describes how the Entities are identified, versioned, related to each other and detailed with Attributes.

### Model

Documentation, Software and Information which formalizes the complex real world.

### Operations

Groups all Processes which do not modify the Enterprise Model.

### Organization-Unit

Node of a hierarchical structure like a Direction, a Department, a Branch.
The smallest Organization-Unit is the **Position**. Ex of Position Role: "Salesman", "Assistant".
Human-Actors and Computer-Actors are assigned to Organization-Units.

### Process

An **Invariant-Process** is a chain of Business Functions (with no Organization Functions) triggered by an independent Business-Event and executed to deliver Process-Value to Process-Client.
Ex: Hire a new Employee, manage an order, sell a Product…
The Process-Client can be the Enterprise Customer, or other external Actors (External Partner, Provider, Government), or internal Actors (employees).

An **Organized-Process** is a set of Functions of an Invariant Process triggered by an Organization Event.
Example: the Invariant-Process “Manage a Customer order” can be implemented by 2 Organized Processes: “Capture the Customer Order” triggered by the “Customer request” and “Deliver Goods” triggered by “the truck is full”.
Not that the second Organized-Process executes several instances of the Invariant-Process “Manage a Customer order” as the truck delivers several orders.

### Reusable Model (or Reusable Component)

Model Element which is Reusable with different levels of granularity.
Example: different Companies may Reuse the same Solution Model.
Example: different Projects may Reuse the same Function Models (“Software Services”) or the same Information Model.
A Reusable Component can be a Black Component (interface is public, implementation is a black box) or a White Component (inheritance, types, patterns).

### Solution

The number of Action-Models (Process Models or Function Models) is huge; they must be grouped.
The “Solution Model” groups several Action-Models and includes the Information private to this group of Actions. Solutions may have different levels of granularity: a CRM Solution groups Processes, while a Pricing Solution groups Functions.

### Synergy or Mutualization

Definition of Shared Resources or Reusable Models grouped inside “Foundation”.
Company Synergy means Shared and Reusable elements between Business Units of the Company.
Group Synergy means Shared and Reusable elements between Companies of the Group

### Transformation

To change Enterprise Operations, Transformation must Model and Deploy a new Operation Model.
Transformation groups all Transformation-Actors with all Transformation-Processes they can execute and all Transformation-Information
There exist large Transformations like “merge companies”, “new product”, “new process” or small Transformations like “change a price”, or “create a security profile”.
As for Operations, Transformation-Actions are

- Invariant-Processes (ex: execute a Transformation project)
- Organized-Processes or “Activity” sometimes called “Phase” (ex: execute a Gap analysis, accept the Solution Model)
- Function sometimes called “Practice” (ex: update planning, test a new Function, develop a piece of code)

### **Transformation-Process** or **Transformation-Approach**

Process which Transforms the Enterprise. They are often summarized under the word ‘Methodologies”.
Ex: define an Enterprise road map, execute a Solution project, execute an Architecture project, deploy a new Solution, maintain Solutions. Transformation Processes are built with Transformation Functions (often called “Practices”).

### Transformation-tools

Tools to help Execution of Transformation Functions: Map tools, requirement tools, Process Modeling tools, design tools, development tools, programming language, quality check tools, teamwork tools, test tools, software configuration management, documentation tools, integration tools...

### View

Presentation of part of the Model adapted to the Actor. The same Model offers different views: one for the Business Analyst, one for the IT Developer, one for the Operation Actor, one for the Architect…