

Component

THS-02 “Thesaurus - the vocabulary”

CEISAR & Praxeme common glossary

Objective CEISAR and the Praxeme Institute present their common glossary, which results from an endeavor to simplify the methodological vocabulary and to share it between business actors and IT community. Their motive lies in the will for reducing the training effort and facilitating the use of their works.

Content Selection of terms related to enterprise transformation

Authors Jean-René LYON, Dominique VAUQUIER

Reviewers *Praxeme Institute* (see p. ii)

Version 1.0.0, le 6 March 2010

Configuration elements

The position of this module in the methodology

Situation in the documentation

“Thesaurus” is situated inside the Organum, which is the larger structure covering the Praxeme methodology. It comprises all resources related to terminology in the field of enterprise methodology.

Owners

CEISAR and Praxeme Institute

All suggestions or change requests are welcome (please address them to the authors).

Availability

This document is available on CEISAR and Praxeme websites and can be used if the conditions defined on the next page are respected. The sources (documents and figures) are available on demand. The most recent version is to be found on these websites.

Revision History

Version	Date	Author	Comment
0.0.3	December 2009	JRL	Selection of terms from CEISAR repository. Meetings with DVAU.
0.0.4	January 2010	DVAU	Review and shape
0.1.0	2010 Feb. 12	JRL & DVAU	Final version before submission to Praxeme community
1.0.0	2010 March 6	PL (CEISAR)	After English review
1.0.0			Current version of the document

This document has been reviewed by:

License

Conditions for using and distributing this material

Rights and responsibilities

This document is protected by a « [Creative Commons](#) » license, as described below. The term « creation » is applied to the document itself. The original authors are:

- Jean-René LYON, Dominique VAUQUIER, for the document
- CEISAR
- The association *Praxeme Institute*, for the entire Praxeme methodology.

We ask you to name one or the other, when you use a direct quotation or when you refer to the general principles of the Praxeme methodology.

This page is also available in the following languages :

[български](#) [Català](#) [Dansk](#) [Deutsch](#) [English](#) [English \(CA\)](#) [English \(GB\)](#) [Castellano Castellano \(AR\)](#) [Español \(CL\)](#) [Castellano \(MX\)](#) [Euskara](#) [Suomeksi](#) [français](#) [français \(CA\)](#) [Galego](#) [עברית](#) [hrvatski](#) [Magyar](#) [Italiano](#) [日本語](#) [한국어](#) [Melayu](#) [Nederlands](#) [polski](#) [Português](#) [svenska](#) [slovenski jezik](#) [简体中文](#) [華語 \(台灣\)](#)



You are free:

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor.



Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

Introduction

Principles

We aim to clarify the terminology, to make it a useful tool for communication and creativity in the context of enterprises and enterprise transformation. Therefore, we have endeavored to establish the simplest definitions possible.

The rules we apply in this glossary include:

- Chosen words must be understandable by Business and IT Actors.
- Definitions must be short.
- We allow possible extensions of a definition by Role or viewpoint.
- When the glossary provides the definition of a term, this term is always written starting with an upper case letter.
- In a phrase (e.g. “Architecture-Description”), the terms are linked with a hyphen (‘-’).
- We rule out homonyms.
- When a term conveys several meanings (e.g. Architecture as a Description and as a Discipline), we create as many items, using compound expressions¹.

¹ We have to be very cautious with some very common terms. An example is “service”: Business-Service, IT-Service, Software-Service...

Glossary

Action

Work executed by an Actor.

Comment

This notion is elementary, meaning that it is a core notion and that methodology contributes in no way to its definition. It is a starting point everyone can understand without debate.

Characteristics

Recursive: an Action can be broken down into other Actions.

Depending on its scale and scope, an Action can be considered as: a Process or a Function or an Activity...

Method recommendations

An Action should always be named by a verb.

Action-Model

Describes way of working or provides an Actor with directives, so that he/she correctly executes the Action(s).

Characteristics

- For Manual-Actors, instructions are documentation (procedure, user guide).
- For Automated-Actors, instructions are software.
- For Assisted-Actors, instructions combine documentation and software.

Activity

Any arrangement of Actions.



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



As a very common term, “Activity” conveys many meanings. In practice, it could lead to misunderstandings. The Praxeme metamodel distinguishes the connotations depending on the Level of representation where it applies. “Activity” as a term is reserved for the level of execution, so as to comply with the broader usage. When it comes to representation, a representation or a specification of an activity is called a Practice. As a result, we avoid confusion between the fact and its representation.

Actor

A being taking on or being able to take on an Action.

Typology

Human-Actor or **Computer-Actor** (Automat) or **Assisted-Actor** (when Human-Actor and Computer-Actor are combined) executing an Activity.

Characteristics

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

Agility is the ability to Transform fast and well.

Characteristics

It allows reduction of time between the emergence of a new idea and its availability in Enterprise Operations.

Related term

Reactivity.

Architecture

The term “architecture” relates either to something to be built or to the art of designing.

In common language, this term covers two different meanings, at least: a) a description of something to be built or already existing; b) a craft or discipline for designing things to be built. We will distinguish both meanings.



An Architecture-Description represents the structure of a Model.

An architecture may be good or bad.

A specific architecture is a collection of well-argued choices and building rules regarding an aspect of a system.

Some features of an architecture according to Praxeme methodology:

- A specific architecture is a model, the scale of which is global. It deals with the system as a whole.
- With respect to the limit of each discipline, a specific architecture considers only one aspect, as defined in the Enterprise System Topology. Therefore, we use the term to designate an “enterprise architecture” (upstream aspects), “logical architecture”, “hardware architecture”, “software architecture”, “technical architecture”, “physical architecture”.
- There is a continuum between the architecture of a system (global scale) and a model (local scale).



As a discipline, Architecture is the art of designing a construction.



Architecture-Discipline consists of Transformation-Processes to Build the Architecture-Description.

Characteristics

In the scope of the enterprise methodology, the construction is the enterprise itself or any of its inner systems. The main concern of architecture is structure, implying the qualities and behavior of the system in the long term.

Comment

This twin notions – description & discipline – correspond to the classical dichotomy: what & how. To make clear what is involved when dealing with architecture, we need to characterize its objects. Nowadays, we distinguish Enterprise Architecture, Business Architecture, IT Architecture, etc.

References

See IEEE 1471.

Aspect



Part of reality, which has been isolated for the sake of study, in accordance with its inner logic.

Comment

Roughly speaking, there are two ways to cope with complexity: the first is to break down the object into smaller parts, to find elements which are easy to understand and to manipulate; the second is to distinguish between sets of features that can be explained separately. The first way is known as the Cartesian approach and can apply as far as the system displays a unique and homogeneous nature. The second approach appears in system theory. Regarding the enterprise, the complexity stems from the merging of various concerns, the conflict between divergent interests and the cohabitation of extraneous cognitive universes. Therefore, we have to recognize the plurality of this reality.

Reality must be approached from several angles: these angles correspond to aspects in the Topology of the Enterprise System. An aspect is part of the System. This part reveals itself as a set of elements, closely related, focused on a specific concern. The elements differ from one aspect to another so that the aspects do not overlap.

Unlike the View, the aspect is a notion which does not refer to an actor considering the system (viewpoint), but which belongs entirely to the “instance”. This metaphysical standpoint can be easily challenged but we stick to it because it brings efficiency in representation and action.

Commodity-Solution



“Commodity” characterizes a solution which is the same for the different competitors and whose requirements are foreseeable before implementation

Characteristics They do not allow differentiation from competitors. Examples: 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

An approach that consists of building a Solution by defining all requirements before starting the IT Model.

Characteristics This approach is adapted to Commodity Solutions.

Cooperative-Approach

An approach that consists in building a Solution by successive Versions with a mixed team merging Business and IT competencies.

Characteristic Adapted to Evolving-Solutions. Also called **Agile-Approach**.

Discipline

A discipline aggregates know-how, procedures and methods, into a unit of skills based upon a generic knowledge.

Characteristics This notion is very close to the notions of art and crafts (art of engineering, art of architecture).

Quantitative limits in human intellectual and operational abilities determine the potential discipline.

Examples of disciplines: Enterprise Architecture, modeling, project management, strategic design, documentation management...

Enterprise

An Enterprise is an agent which delivers a Product to its Customer. The Product (Goods or Services) can be economic or not (for example, cultural value).

Characteristics The notion embraces not only profit-making Enterprises but also governmental entities, universities, research centers, associations...

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.



Praxeme plays on both meanings of “Enterprise”: a) as above b) “human endeavor, project, adventure, exploration...”². The generic definition that covers both meanings is: “By Enterprise, we mean any type of organized and willful entity”³. What is at stake is the domain for the application of the enterprise methodology (companies, networks, weapon systems, action systems...).

² See *The Enterprise Transformation Manifesto*, § 1.1 (www.enterprisetransformationmanifesto.org).

³ *Ibid.* § 2.

Enterprise-Architecture

Comment

Architecture that applies to the Enterprise as a whole.

As we did with “Architecture”, we have to distinguish two meanings: description and discipline.

Enterprise Architecture-Description

It represents the Structure of the Enterprise.

It is generally represented by Maps (high-level models), which offer a Global View to better understand the Enterprise or to design its changes: 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:

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



The Praxeme methodology proposes a template and guidelines for the General Architecture Dossier, a document that expresses the orientations and decisions for architecting the Enterprise System. It applies the methodological framework known as Enterprise System Topology.

Enterprise Architecture-Discipline

The discipline that thinks the Enterprise as a whole and makes the link from strategy to deployment.

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

References

See TOGAF.

Entity

Characteristics

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.

Typology

A **Business Entity** is necessary for Business, independently of the Organization of the Enterprise. Examples include: Product, Party, Contract or Account.

An **Organization Entity** is necessary for Organization purpose: Organization-Unit, Position, Profile, Right, Duty, Role... are all Organization-Entities



Praxeme uses “semantic” for Business and “pragmatic” for Organization, so that “Business” may still be used in the usual way, prior to the distinction between both Aspects. Instead of “entity”, Praxeme generally speaks of objects – business objects, real objects, administrative objects... Both expressions – “entity” or “object” refer to the same notion. Objects are represented and their properties expressed by means of classes.

Comment

While, for CEISAR, “entity” refers to an Instance, the software engineering tradition often uses the term for the description of instances, namely the Model.

Evolving-Solution



Characteristics

“Evolving” characterizes a solution which is not the same for the different competitors and whose requirements are not foreseeable before implementation.

The requirements of Evolving-Solutions evolve over time. They are often Solutions to

provide differentiation from Competitors. For example: 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. Competitive Solutions should be built with a Cooperative Approach.

Foundation

Set of Reusable Models.

Characteristics The Reusability perimeter can be the Solution, the Company, the Group, a Business community, a country, or the world!

Reuse of Model is the way to create synergy and to harmonize how the different Units of an Enterprise work.

Typology We distinguish two levels of foundations, depending on whether we are considering the Enterprise Operations or the Enterprise Transformation.

Operation Foundation = Reusable Models for Operations which include

- The Enterprise Model: Solution Map, Process Map, Function Map, Entity Map,
- Reusable Solution Models
- Reusable Function Models
- Reusable Entity Models
- Reusable Process Patterns
- Reusable Types
- Reusable Human Actor Models
- Reusable IT Actor Models

Transformation Foundation = Reusable Model for Transformation which includes

- Methodology
- Transformation Approach
- Transformation Tools such as modeling tools

Function

Action inside a Process.

Characteristics A Function may reuse other Functions.

Depending on the architecture style, the IT Model of a Function is sometimes called a Software Service.

Typology Our approach distinguishes between Business Functions and Organizational Functions.

Business Function A Business Function is independent from Organization chosen by the Enterprise.

Example: “check Information entry”, “compute price”, “send message to Customer”.

Organization Function Organization Functions are added to implement an Organization.



CEISAR considers “Function” and “Rule” as synonyms.



For Praxeme, Function and Rule have a different meaning; a Rule is a constraint on Action or Information while a Function represents all other elementary Actions. In fact, since as a term “function” is so ambiguous and polysemic, the methodology tends to avoid it. It does not appear in the metamodel.

Information-Model

Describes Entities, their Relations and inheritances.

Characteristics Describes how the Entities are identified, versioned, related to each other and detailed with Attributes. Other expression: Data-Model.



Instead of “information model” that connotes an approach restricted to data, Praxeme favors the expression “semantic model”. The semantic model aims to express the entire semantics of the field of study. In so doing, it considers information transformations and actions of the objects and concepts. The semantic model incorporates the information-model, but considering the transformations and actions may sometimes change the structure of the model. The methodology provides the rules for deriving the logical data model from the semantic model.

Invariant-Process



A process without any assumption or constraint related to an organization.

As opposed to Organized-Process.

Level of representation



- Pursuant to the OMG theory, Praxeme distinguishes:
- A level of execution – where we face real things and actions
 - A level of description – the place of the models that describe the real things
 - A level of expression – i.e. the metamodel articulating the categories of representation we use when modeling

In practice, the modeler is often faced with difficulties that stem from the mix of levels of representation. That is the reason why we have to make this notion clear.

Model



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



A model is a formal representation of a portion of reality.

Following this definition, a software solution can be seen as a model.

Operations

Comment

At the Enterprise level, “Operations” is a generic term for grouping all Processes which do not modify the Enterprise Model.

Operations are the normal life of the enterprise. In this glossary intended for decision-makers and enterprise architects, we insist on the operation-transformation couple.

Organization-Process or Organized-Process



A process including the organizational choices.

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”. The second Organized-Process executes several instances of the Invariant-Process “Manage a Customer order” as the truck delivers several orders.

Comment

Equivalent to Business-Process, insofar as the BPM practices do not isolate what is independent of the organization (see Invariant-Process).



In order to facilitate transformation and innovation, it is of paramount importance to free oneself from current habits, the usual way of working. The modeler has to be very careful not to be trapped by the implicit assumptions that dominate the organization. Rather than describing processes – even so-called conceptual processes – semantic modeling provides the modeler with another starting point, which is radically different. This helps to rethink the processes.

Organization-Unit **Node of a hierarchical structure like a Division, 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 **A Process is a collection of actions, aimed at a specified goal.**

Typology Our approach distinguishes between two kinds of processes, depending on whether or not they convey organizational assumptions. This distinction leads to dramatic consequences, as far as transformation is concerned.

Invariant-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).

Organized-Process 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 "Truck is full".

Note that the second Organized-Process executes several instances of the Invariant-Process "Manage a Customer order" as the truck delivers several orders.



Praxeme assumes this distinction in terms of aspects, namely: semantic aspect (independent from organization) and pragmatic aspect (the place for human activity).

But the enterprise methodology is reluctant when it comes to representing processes under the semantic aspect. Indeed, the mere breaking down of an activity always incurs the risk of articulating organizational choices or habits, more or less consciously. Instead, it encourages thinking in terms of lifecycles.

Reactivity **Reactivity is the ability to Operate fast and well.**

Related term Agility

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

In the context of Enterprise Architecture, a Solution groups and links together all elements required for addressing a problem or reaching a goal.

Characteristics

elements.

A Solution may include analysis, strategic formulation, Models – including software – means, actions, training, incentives... It generally combines many categories of such

Solution-Model



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 inherent to this group of Actions. Solutions may have different levels of granularity: a CRM Solution groups Processes, while a Pricing Solution groups Functions.

Synergy

The fact that two or more entities work together in a coordinated way, instead of on their own.



“Foundations”.

Synergy means two different things: pooling Resources (like Units or Computers or Information) or reusing Models when Transforming. Reusable Models are called

Company Synergy means Shared and Reused elements between the Business Units of the Company.

Group Synergy means Shared and Reused elements between the Companies of the Group

Related terms Mutualization

Transformation

The activity of changing the Enterprise Operation Models

Characteristics

In order to change Enterprise Operations, Transformation must Model and Deploy a new Operation Model. Transformation gathers together all Transformation-Actors with all Transformation-Processes they can execute and all Transformation-Information.

There exist large Transformations like “merge companies”, “launch a new product”, “implement a 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)
- Functions sometimes called “Practice” (ex: update planning, test a new Function, develop a piece of code)

Transformation-Process or Transformation-Approach

Process used for the purpose of Transforming the Enterprise.

Examples: define an Enterprise road map, execute a Solution project, execute an Architecture project, deploy a new Solution, maintain Solutions.

Related terms

Transformation-Processes are often grouped under the word “Methodology”.

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 a Model – or a set of Models – meant for a type of Actor and for a specific purpose.

Comment The content, form and vocabulary of the view adapt to the profile of the targeted actors.

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...



A partial set of information or knowledge about a reality, a view always refers to an actor or a given type of actors. This notion differs from that of aspect. Some views are made up of information, decisions, elements which come from various aspects.

Related terms Actor, Aspect, Model.