

Enterprise Modeling

White Paper

April 2008

1	Executive summary	4
2	Objectives	5
2.1	What is Enterprise Architecture?	5
2.2	Why is Enterprise Architecture so important?	6
2.3	CEISAR's ambition	7
2.4	A consistent language	7
3	The Enterprise Operations	9
3.1	The Enterprise.....	9
3.2	Resources and Organization Actors	9
3.2.1	Resources.....	9
3.2.2	Organization Actors.....	10
3.2.3	Organization Units.....	10
3.3	Which Actions?	11
3.4	The Process.....	11
3.5	The End to End Process.....	12
3.6	How to classify Operation Processes	13
3.6.1	Primary Processes	13
3.6.2	Resource Processes	14
3.6.3	Management Processes.....	14
3.7	Organized Process.....	15
3.8	Function and Activity	16
3.9	Information	17
3.10	Organization.....	18
3.11	Solutions	19
3.12	The Enterprise Model	20
3.12.1	The Global Model.....	20
3.12.2	The Detailed Model	20
4	Enterprise Transformation	22
4.1	What is transformation?.....	22
4.1.1	Main steps for a single Enterprise	23
4.1.2	Main steps for a Group of Companies	24
4.1.3	Construction Processes and Information	25
4.1.4	Deployment Processes and Information	25
4.2	Transformation complexity	25
5	Describing an Enterprise with the CEISAR cube	26
5.1	Consistency, Continuity and Metrics are key	26
5.2	Which Dimensions?.....	26
5.3	A Model for understanding Complexity.....	27
5.4	Operations and Transformations	28
5.5	Synergy.....	28
5.5.1	Group and Company levels	29
5.6	Summary.....	30
6	Describing each box.....	31
6.1	Operation Execution.....	31
6.2	Operations Model	32
6.3	Shared Operations Execution.....	33
6.3.1	Centralizing a Solution Unit	33
6.3.2	Centralizing IT Operations.....	33
6.3.3	Centralizing Master Data	33
6.3.4	Target Enterprise, Operating Enterprise and Modeling Enterprise.....	34
6.4	Reused Operations Model.....	35
6.4.1	Reuse of Global Model.....	35
6.4.2	Reuse of Actor Model.....	36
6.4.3	Reuse of Action Model	36
6.4.4	Reuse of Data Model.....	36
6.5	Transformation Execution.....	36
6.6	Transformation Model	37

6.6.1	Transformation Actor Model	37
6.6.2	Transformation Action Model.....	37
6.6.3	Transformation Data Model	38
6.7	Shared Transformation Execution	38
6.8	Reused Transformation Model	38
6.9	Share and Reuse summary	39
7	How the CEISAR model compares with existing models	41
7.1	MIT Sloan Center	41
7.2	TOGAF Architecture (http://www.opengroup.org/architecture/togaf8-doc/arch/)	43
7.3	Cobit Architecture.....	44
8	FAQ.....	45

1 Executive summary

EA is a **Global approach** which coordinates evolutions of independent domains like Transformation of Organization, Process Modeling, Master Data management, human resource management, Information Systems, and Transformation methodologies to provide a competitive advantage to the Enterprise.

Enterprise Architecture has two main parts:

- A static part: description of how the Enterprise works through the **Enterprise Model** which covers **Actors** (Organization with people and computers), **Actions** (Processes and Functions) and **Information**
- A dynamic part: **how to transform** the Enterprise, which means defining a Target Model in line with Enterprise Strategy and then moving to it.

EA has become key for the Enterprise: agility and time to market, productivity, capacity to better understand the Customer and extend its services, reduction of useless complexity, and synergies between Companies of the same Group, are all impacted by EA.

CEISAR delivers ongoing EA white papers on topics chosen by its Sponsors.

To guarantee the consistency of these white papers, CEISAR must use the same language to describe the Enterprise, its Processes, its Organization, its Information System, its EA Governance, its Methodologies.

This document describes this reusable language. It proposes definitions for:

- the **Enterprise** with different levels: Group, Company, Business Unit
- **Actions** like Processes and Functions
- **Actors** who execute the Actions: People can be Actors, but computers may also be Actors
- **Information** as master Data shared by different Processes
- **Organization** of the Enterprise

Then the Enterprise is described according to 3 main dimensions:

- **Complexity**: Real World **Execution** is separated from its **Model**; a **Global Model** helps understand how the Enterprise works thanks to global maps of Processes, Solutions, Functions, Services, Entities; a **Detailed Model** describes instructions given to Actors through documentation or software
- **Agility**: Processes are broken down into **Operation** Processes and **Transformation** Processes (like Projects)
- **Synergy**: we focus on **Shared and Reusable Elements** which are key to giving an efficient structure to Enterprise Architecture.

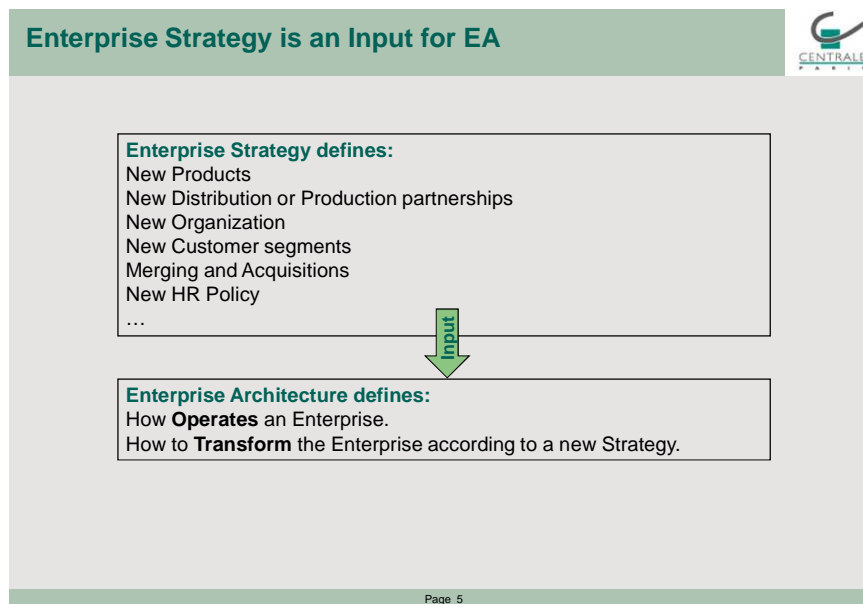
These 3 dimensions can be represented as an Enterprise Cube which helps to sort the many different EA elements: from description of EA Governance to Data Model or Transformation Tools.

Then, we conclude by mapping the CEISAR model with other existing Models like: Cobit, MIT Sloan Center, and TOGAF.

2 Objectives

2.1 What is Enterprise Architecture?

CEISAR must produce White Papers on Enterprise Architecture (“EA”).
But what is Enterprise Architecture?



Some standard definitions of Enterprise Architecture

ANSI/IEEE Std 1471-2000 definition : “Architecture = the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution”.

Jeanne W.Ross, Peter Weill and David C.Robertson

The Enterprise Architecture is the organizing logic for business processes and IT infrastructure, reflecting the integration and standardization requirements of the company’s operating model. The enterprise architecture provides a long-term view of a company’s processes, systems, and technologies so that individual projects can build capabilities, not just fulfill immediate needs.

TOGAF Architecture definition has 2 meanings:

1. A formal description of a system, or a detailed plan of the system at component level to guide its implementation.
2. The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.

Remark: for TOGAF, Enterprise Architecture means Approach and Structure.

Total definition

Enterprise Architecture is a continuous process which classifies, normalizes and ensures coherence and flexibility of the Enterprise System thanks to optimization of what could be shared or mutualized between Solutions.

Wikipedia definition: Enterprise Architecture is the description of the current and/or future structure and behavior of an organization's processes, information systems, **personnel** and organizational sub-units, aligned with the organization's core goals and strategic direction. Practitioners are called enterprise architects. Although often associated strictly with information technology, enterprise architecture relates more broadly to the practice of business optimization in that it addresses business architecture, performance management, organizational structure and process architecture as well.

These definitions are consistent. In summary, EA means 2 things:

- Description of Enterprise **Operations**: which Actors (persons and computers grouped inside an organization) execute which Actions (Process, Activities, and Functions) with which Information.
- Description of Enterprise **Transformation** aligned with Strategy, which means define a target and move to it.

EA brings together different topics like Business Processes, Organization, Information Systems and IT technologies.

The Resources who Operate and Transform the Enterprise are Persons and Computers (or a combination of a Person and a Computer). EA describes Configurations and Roles of **Computers** and competencies, Roles, Rights, Duties and assignment of **Persons**. **EA also** Transform Persons by defining training Processes and by offering assisted Processes which improve global Operations.

After one year Transformation, EA is not the same as Operation Processes and Solutions have been improved thanks to Transformation Processes, which, themselves, may also have been improved. So, EA must be versionned to reflect the successive “shots” taken periodically.

2.2 Why is Enterprise Architecture so important?

Traditionally, complexity has been associated with complex Products. To explain complexity to students, you show them a map of an Airbus broken down into its respective parts: it is a very complex map.

But describing or transforming Enterprise Operations is becoming ever more complex!

The number of Actors, providers, partners, products, processes, functions, software pieces, computers, handled by an Enterprise is huge and ever increasing. One of our sponsors explained that the Operating software running the Enterprise is made up of 500,000 pieces: how to describe it? How to ensure that a transformation of one of the Elements does not disturb relations with the other Elements? Another one handles several thousands Life Insurance products: how to explain to salespeople what they can sell? Another one wants to standardize Processes all over the world: how to train the tens of thousands of users? And as the pace of change accelerates: how to transform the Enterprise faster than competitors to take advantage of good ideas which can emerge anywhere in the world?

Enterprise Architecture is a **new discipline** which brings together Business Process definition, Organization, Information Systems, IT Technologies, and Human Resources. Today, Enterprises have realized that they must formalize how they work if they want to accelerate transformation.

It is set to become one of the major “technologies” of the near future, not only for Service Enterprises but also for Industry.

Which technology skills for the Enterprise?		
Types of Enterprises	Enterprises producing Goods* (30% of the economy) <i>(Industry, Construction, Energy)</i>	Enterprises producing Services (70% of the economy) <i>(Administration, Communication, Distribution, Finance, Software, Healthcare, Education, Transport, ...)</i>
Activities of the Enterprise		
Product Design Manufacturing	Enterprise Architecture Biology Chemistry Energy Fluids Mechanics Physics Embedded Systems.	Enterprise Architecture
Customer Management Sales network Delivery Reporting and Finance Resources:HR,IT,Facilities	Enterprise Architecture	Enterprise Architecture
Examples	Eads, Pharmaceutical Industry	Air France, Healthcare system

*includes Fluids: energy, water, physical information carrier




2.3 CEISAR's ambition

CEISAR's ambition is to offer training materials on Enterprise Architecture to cover the 2 key domains: How to Model an Enterprise and How to Transform an Enterprise.

Here is a list of the key Topics which will be progressively covered by the CEISAR.

(In bold: what is available on www.CEISAR.org on April 2008).

CEISAR Ambition: Enterprise Architecture



<h4>How to Model an Enterprise</h4>	<h4>How to Transform an Enterprise</h4>
<ul style="list-style-type: none">• Global description of an Enterprise<ul style="list-style-type: none">– The Enterprise Model– The Enterprise Organization– Enterprise Glossary• Describe Actions<ul style="list-style-type: none">– Model Processes– Describe shared Actions: SOA• Describe Informations<ul style="list-style-type: none">– Model Entities– Describe shared Informations: Master Data• Describe Resources<ul style="list-style-type: none">– Describe Organization Unit and Workers: Competencies, rights and duties– Describe IT infrastructure: Hardware, Software, Network– (Description of other Resources like Cash, Facilities is not a priority)	<ul style="list-style-type: none">• Define EA Governance• EA Road Map• How to simplify a Legacy System• Improve the Agility chain:<ul style="list-style-type: none">– Find shared Actions• The Solution Project: construction and deployment• The Architecture Project: construction and deployment• Define the Transformation Organization adapted to Enterprise strategy• Transform people<ul style="list-style-type: none">– How to train people to execute Transformation (Business and IT)– How to train people to accept Transformation (attitudes)• Consistent Transformation Tools

Page 7

To be consistent in all our white papers, we must propose an Enterprise Model which can be reused for all Topics: this is the goal of this document.

2.4 A consistent language

A new discipline as Enterprise Architecture requires a precise language as it exists in more mature disciplines.

As CEISAR focuses on Enterprise Architecture, it proposes a consistent language to describe how an Enterprise runs thanks to elements as various as:

- Enterprise **Processes**, including end to end processes which cross different organization structures
- The **Value** delivered by the Enterprise to its Process Clients: Customers, Partners, Providers or internal Clients.
- Enterprise **Information**
- **People** who work for the Enterprise, employees or not employees, with their competencies, their Role, their rights and their duties
- Structure of the **Organization**
- Inclusion of external people like Prospects, customers, partners, providers, when they become part of the game through integration into the **Extended Enterprise**
- **IT components**: computer, software, networks
- **Reusable elements** for different solutions and levels of synergy: reused Solution models (like reused software packages), reused data models, reused Functions, reused configurations

It is not enough to describe **Operations** Processes alone, it is also necessary to describe **Transformation** Processes (like Projects, methodology) which implement the Enterprise Strategy.

This language will be progressively refined by successive versions.

This **unique** Enterprise “Meta-Model” must deliver **different views** adapted to each population: global views for deciders, detailed views for people who operate and transform, business views for Business Analysts and IT views for IT designers.

This is not an easy task: how to be simple enough so that the model is understandable by all, and broad enough to cover all topics? Our goal is that **Business People** and **IT People can use the same simple language!**

Paul Valery was pessimistic: “Anything which is simple is wrong. Everything which is not is useless”.

But Albert Einstein was optimistic: “Everything should be made as simple as possible but not simpler”. Quite a challenge!

To help us we decided to focus on key business concerns. Working with our sponsors helped us identify these concerns which are:

- Understanding Enterprise **Complexity** which means we must split Real World execution from its Model.
- Increasing **Agility** which means splitting Operations Processes (the present) and Transformations Processes (the future).
- Finding the right **Synergy** level, the right balance between centralization (or mutualization) and decentralization (or subsidiarity): which means splitting specific elements and reused/shared elements.

Then we checked existing Models, such as:

- Archimate Modeling
- Atos Model
- Comet Business http://www.modelbased.net/comet/business_modelling/index.html
- Robert Winter model from University of St Galen
- Valtech Model
- John Zachman Model

Taking advantage of the many good ideas already published, and of the 3 key concerns (Complexity, Agility, Synergy) we propose a simple CEISAR meta-model based on the “**Enterprise Cube**” which focuses on:

- Understanding how an Enterprise works: distinction between real world and its model
- Understanding the Agility chain: distinction between Operations and Transformations
- Understanding what can be Mutualized

It will be progressively refined by successive versions.

3 The Enterprise Operations

Before presenting the Model we must define certain words which are useful for the Model like "Enterprise", "Action", "Information", "Resource", "Human Actor" and "IT Actor".

Some Enterprises use the same concepts with different terminology. This is not a major problem. The main objective is to **agree on concepts. If concepts are the same**, every Enterprise may **change the terminology** to better communicate internally.

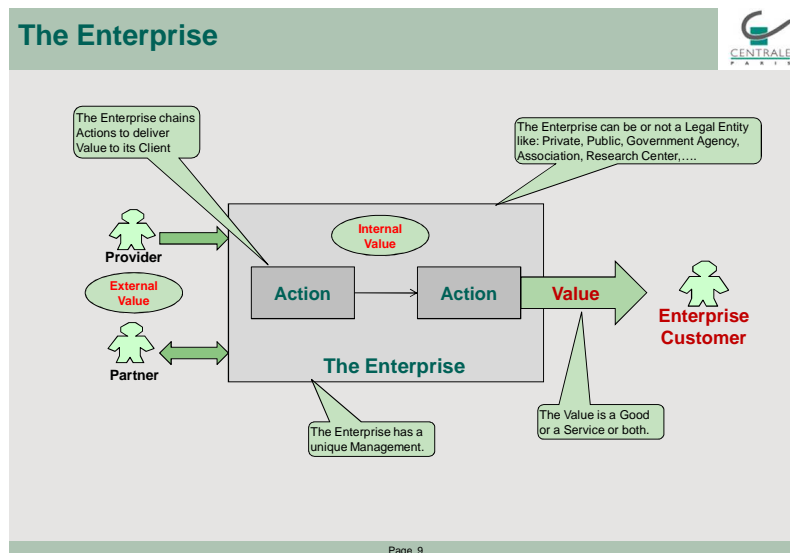
Enterprises may also **add concepts**: CEISAR does not intend to cover all EA details, but to deliver an Enterprise meta-model able to address the main EA concerns.

3.1 The Enterprise

An Enterprise is an agent which creates "value" for a Customer under the same management. The value can be economic or not (like cultural value).

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.



Customers, Prospects, Partners (such as distribution partners or production partners or experts), Providers, Government Units, and so on, are **Business Actors** involved in the Enterprise Business.

3.2 Resources and Organization Actors

3.2.1 Resources

To deliver Value to its Customer, the Enterprise uses **Resources**.

These Resources are:

- The **Organized Actors** who execute Actions for the Enterprise. They can be Persons or Computers or an association of both, which makes 3 Organized Actor categories:
 - The **Worker** using (non programmable) tools and unstructured information.
 - The **Automate** using a programmable Server accessing Structured Information and Network facilities.
 - The **Assisted Worker** comprising a Person and a programmable Work Station accessing Structured Information and Network facilities.
- The **Information**: it can be **structured information** which is interpreted by computers or **unstructured information** which can only be interpreted by the human brain. Structured information is stored by computers on digital supports(disk, DVD...). Unstructured information can be stored on digital supports (image, text, video) or other supports like paper.

- Offices or Factories in Locations: the **Facilities**
- **Capital**
- **Vehicles**
- **Non programmable machines and tools**
- ...

CEISAR will focus mainly on 2 resources: **Organization Actors** (people and IT) - because we need to describe who executes actions - and **Information**.

(The model could be enhanced in following versions, by including the Resource “Facilities”).

3.2.2 Organization Actors

Client Value is produced by **Organization Actors** who create the internal Value and by **Business Actors** (Providers and Partners like distribution partners or production partners or experts) who create external Value.

3.2.3 Organization Units

The Organization Actors (people and IT) are assigned to **Organization Units** (like Direction, Department, Division, Branch) which belong to a hierarchical structure.

We use the word “**Business Unit**” to define the upper Organization Unit only under Enterprise.

If a **Group** owns **Companies**, each Company is an Enterprise.

A Company is broken down into Business Units, a Group is broken down into Companies.

A question may arise: how to define whether a Unit is a **Company** or a **Business Unit**?

Traditionally, it was based on criteria such as **Geography** or **Size**:

- a Company is generally **local to a country**. But this criteria is not sufficient because it is not true anymore for some Groups which have built an Organization based on several worldwide Companies specialized according to Products.
- **Size**: a Company is bigger than a Business Unit. This is certainly true on average. But Business Units of large Groups may be bigger than Companies of average sized Groups.

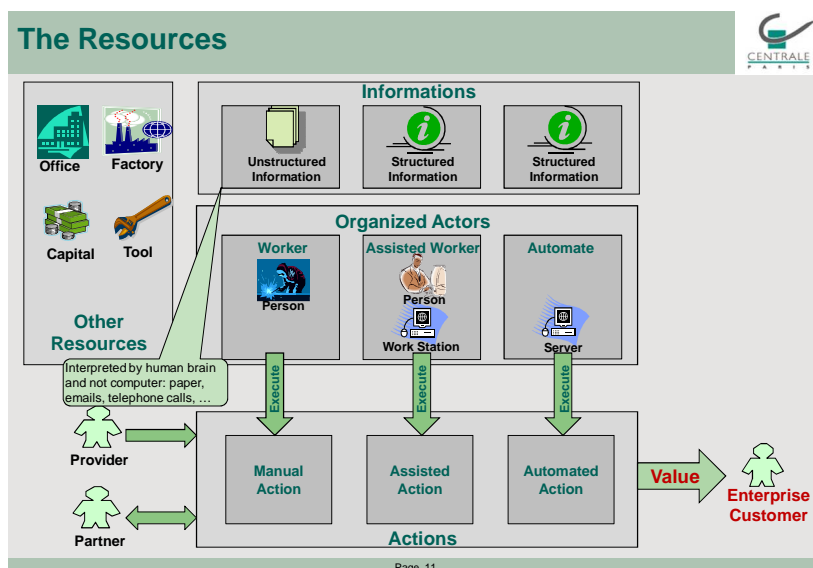
We require some other criteria: autonomy and Transformation capacity.

- **Autonomy**: a Company is more independent than a Business Unit, it **owns** its **Customers** and/or **Products** and is evaluated on its Profit.
- A Company **owns** its **Transformation Unit** (like an IT department)

The **specific** Activities of the Group: centralized cash management, Group HR management, relations with investors... are executed in headquarters which are considered as a **Company**.

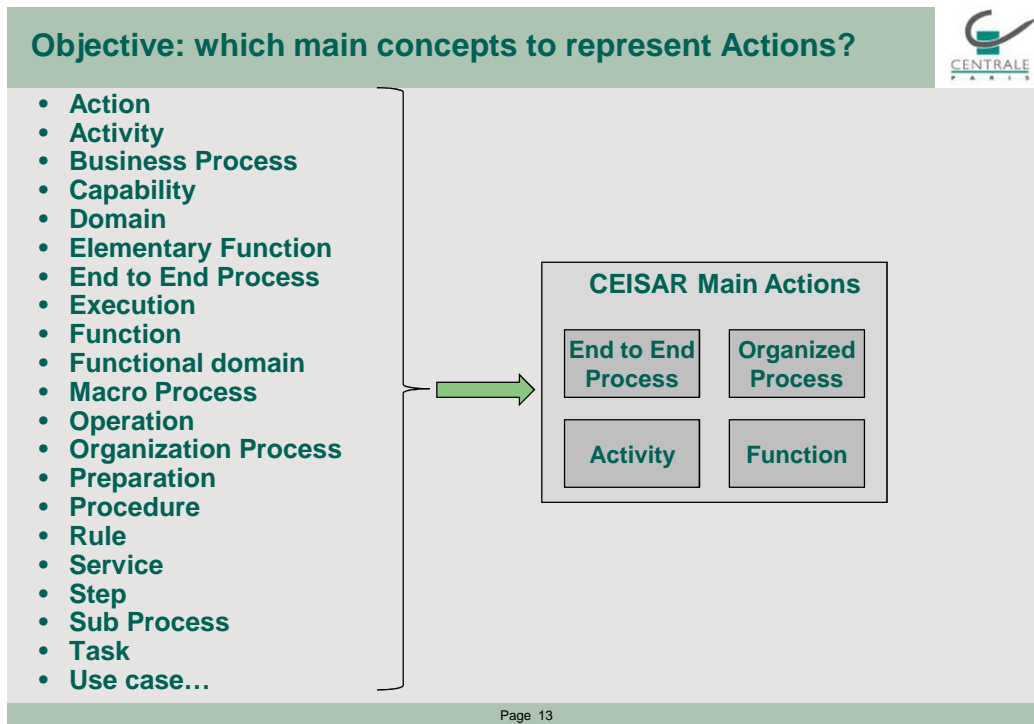
The **Shared** Activities executed by the Group for the Companies are **Group Units**.

Remark: a Unit which just has Automates and no Workers (Ex: a shared Data server) is also considered as an Organization Unit which is part of the overall Organization.



3.3 Which Actions?

An “Action” can be very large like “Launch a new Product” or small like “Compute a Price”. We need to define different levels. But among the currently used terms which are the most important for EA?



CEISAR proposes to use 4 basic concepts:

- End to End Process
- Organized Process
- Activity
- Function (sometimes called Rule)

This is an extensible Model: each Enterprise may add complementary concepts if it wishes. Now, let's explain each of the 4 terms.

3.4 The Process

We needed a starting point: as everything starts from the Enterprise Customer, we decided to call **Process** that chain of Actions triggered by an Independent Event which is the **Customer request** (see CEISAR white paper on Processes).

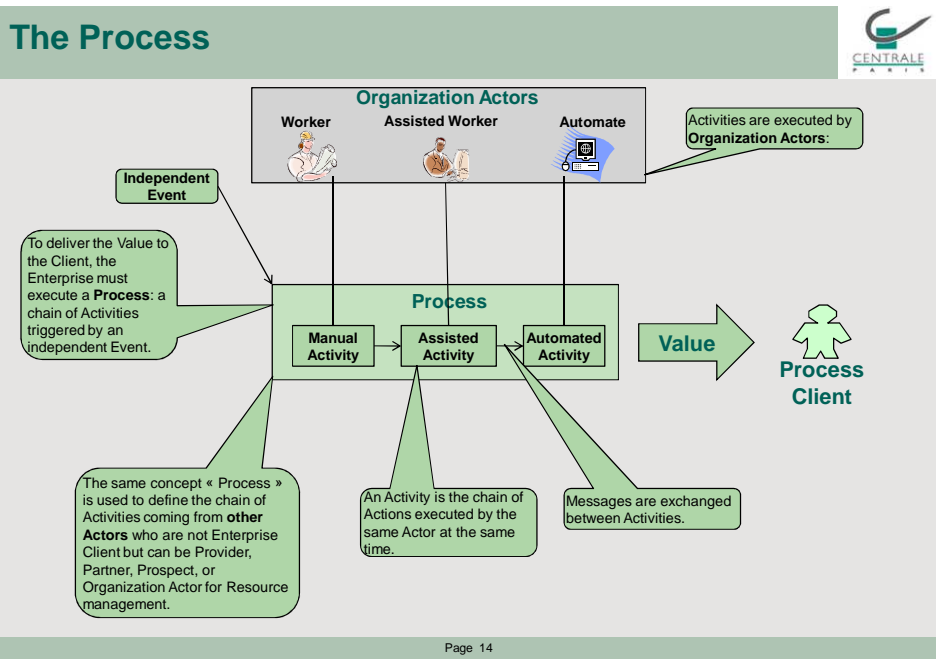
Examples of Customer requests: “Transport by plane”, or “Order Goods”, or “Subscribe an Insurance Contract”.

The “Process Client” is not always the Customer of the Enterprise:

- For the Process “hire a new Employee”, the Process Client is the Human Resource Unit
- For the Process “buy a computer” the Process Client is the IT Operations Unit
- For the Process “gather management data”, the Process Client is the management control Unit
- For the Process “manage a transformation Project”, the Process Client is the Transformation Unit
-

Note that the Client request is an **independent event**, which means that the Process is not the following step of a preceding Process.

Also note that an Action is defined by a **verb**: “Subscribe a Contract” and not “Subscription of a contract”.

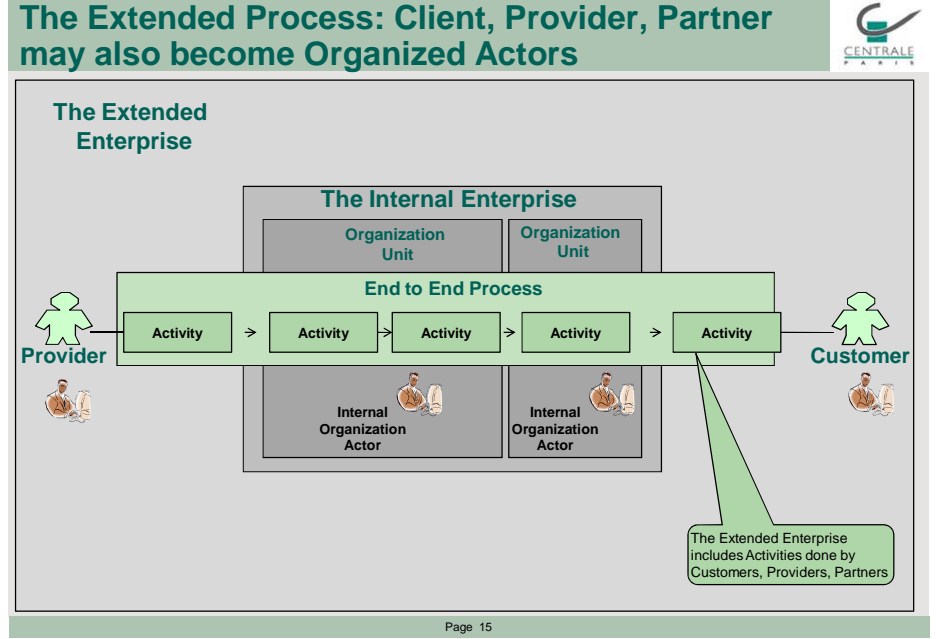


There are 2 kinds of Processes: End to End Process and Organization Process: let's define them.

3.5 The End to End Process

Processes may cross borders between different Units of the Enterprise. The Process "Order to Cash" may involve the sales Unit, the accounting Unit, the production Unit, the delivery Unit... It is an "**End to End Process**". It may also cross borders outside of the Enterprise. For example, when a Company sells its products through distribution networks which belong to other Enterprises, we must define End to End Processes, standardize Business Entity definitions and identifiers... which means that we must define the **Extended Enterprise**.

Thanks to IS, the Enterprise offers Business Actors like Partners, Providers, and Clients the chance to participate in Actions: Internet is a good way to connect them, but they can also exchange files. Thanks to this interaction, Business Actors save time, benefit from more open-for-business hours, and the Enterprise increases its productivity. Business Actors become **Organization Actors** as defined before.



3.6 How to classify Operation Processes

Classification of Processes offers a global view of Enterprises Actions which helps to define Organization, Roles, to align Business and IT...

This classification is today proper to each Enterprise.

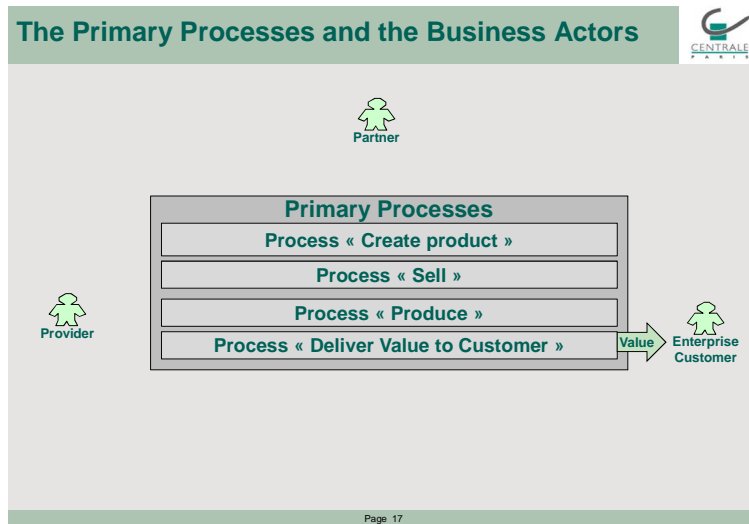
CEISAR only proposes the first level of classification which already exists in most Enterprises; it includes 3 categories of Processes:

- Primary Processes (expression used in Michael Porter Value Chain)
- Resources Processes
- Management Processes

3.6.1 Primary Processes

They define what must be done for the Enterprise Customer:

- **Create or modify a Product Model:** generally by strategic marketing
- **Sell** to Customer: done by the Distribution Units (some Distribution channels may be Partners) and the Operational marketing, who organizes advertising and marketing campaigns
- **Produce** Product instances
- Deliver Value to the customer: deliver **Goods** or deliver **Services** or both.



Service Enterprises like Banks, Insurance, Telecom, Transportation, have a key characteristic: the processes "Produce" and "Deliver Value" are the same because they cannot store Services in inventories. They represent 70% of added value in the developed countries, and this proportion is still increasing.

The table lists examples of service enterprises and their processes. A callout box points to the 'Produce' and 'Deliver Value' columns, stating: "When the Product is a Service, the Processes « Produce » and « Deliver Value » are the same." The CENTRALE logo is in the top right corner of the slide.

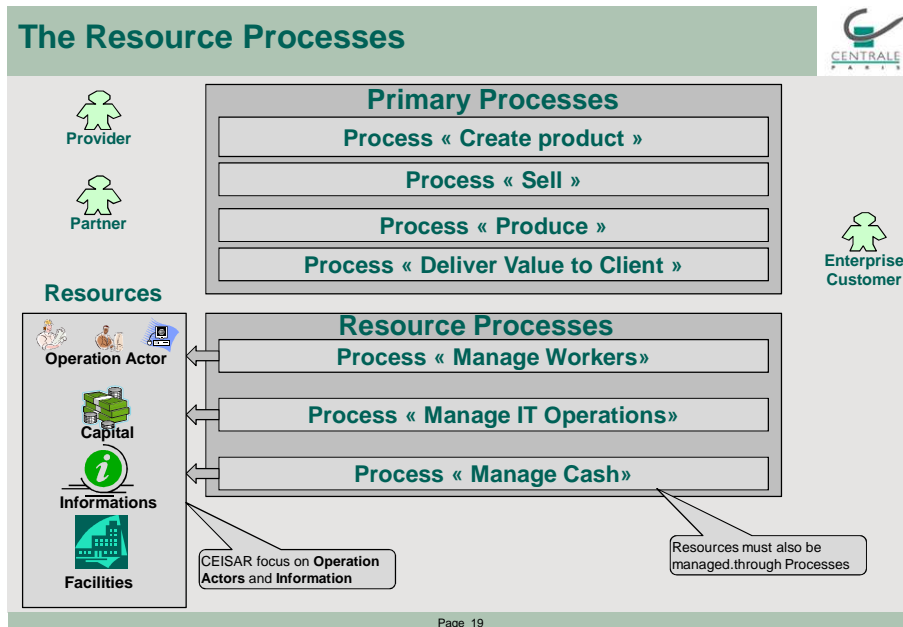
SERVICE Enterprise	Product	Sell	Deliver Product
Bank	Loan Product	Loan Contract	Transfer Money
Insurance	Car Insurance Product	Subscribe a Policy	Damage payment if Claim
Telecom	Telecom Offer	Open a Line	Transfer voice, data
Transport	Flight	Buy a Ticket	Transport the Person

3.6.2 Resource Processes

The Resource Processes - sometimes called "Support Processes" - allow us to manage Resources, from Employees to Facilities, Automates, Software, Information, Cash... It includes collaborative Solutions which help Actors to communicate.

Operating an IT Operations Center comprises a set of Processes like: Support Assisted Workers, Manage exceptions, Tune, Report quality level...

Surprisingly, when Enterprises formalize their Processes, they discover that Resource Processes are more numerous than Primary Processes!



3.6.3 Management Processes

Managers must:

- Make decisions
- Follow up implementation of Decisions
- Check results once the decision is implemented, and take remedial action

To decide, Managers must follow a Governance Process (defined in CEISAR Governance white paper).

To follow up implementation of Decisions, managers use Transformation Processes defined hereafter.

To check results, managers require Information, most specifically, aggregates.

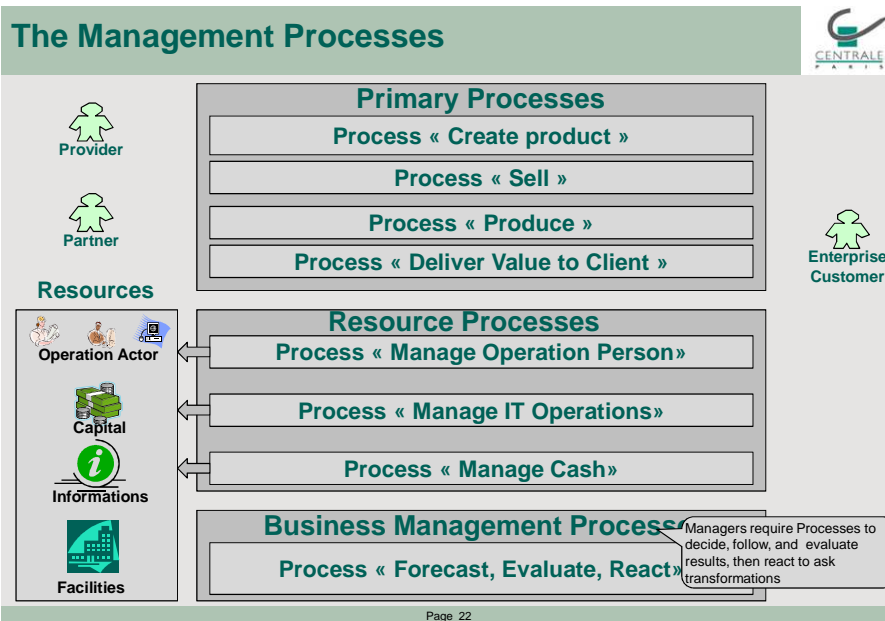
Aggregates must be computed and compared to budgets presented at decision time.

Aggregation is done according to several dimensions:

- Budget/real (with several successive versions)
- Time: instant (if stock) or period (if flow)
- Organization Units which represent territories (hierarchy)
- Measure: numbers, € or \$, % ...
- Product (hierarchy)
- Nature: like accounting nature (with composed aggregates)

Aggregates can be presented on manager boards according to presentation models.

Managers may also need to analyze Information on a non-regular basis. They require specific analysis tools to do so.

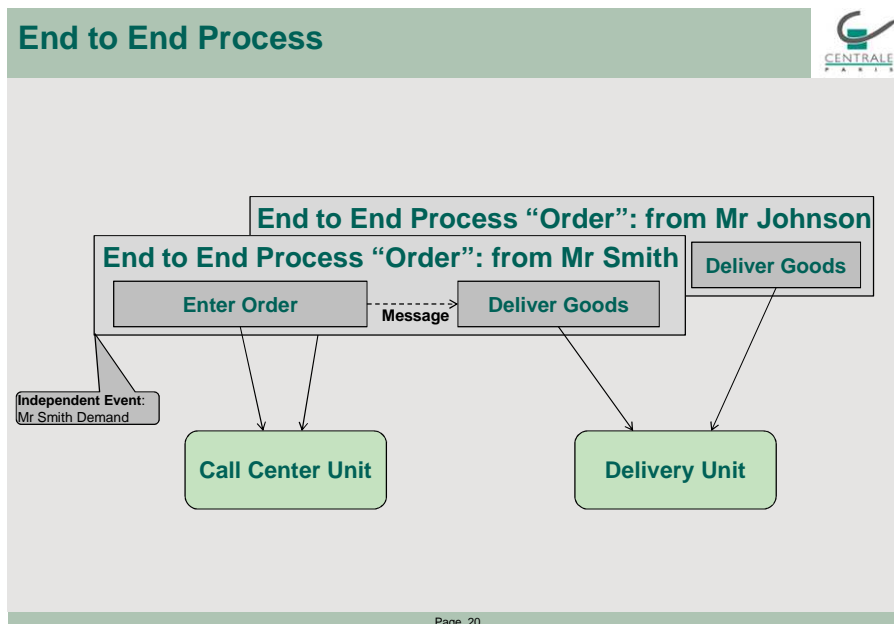


3.7 Organized Process

Optimization of Resources is a key goal of any Enterprise. It often implies that Enterprises must divide their End to End Process into parts called “Organized Processes”.

Let’s take an example:

The “Order” Process is an End to End Process where the Customer request triggers an Action “Enter order”, which is followed by an Action “Deliver goods”.



An Enterprise may decide to group deliveries to optimize Delivery costs.

It means that the End to End Process “Order” is broken down into 2 Organized Processes:

- One Organized Process “Enter order” which is triggered by the Customer request
- One Organization Process “Deliver several Goods” which is triggered by another event depending on the Resource optimization: “one delivery a week” or “deliver when truck is full”.

This break-down allows resource optimization.

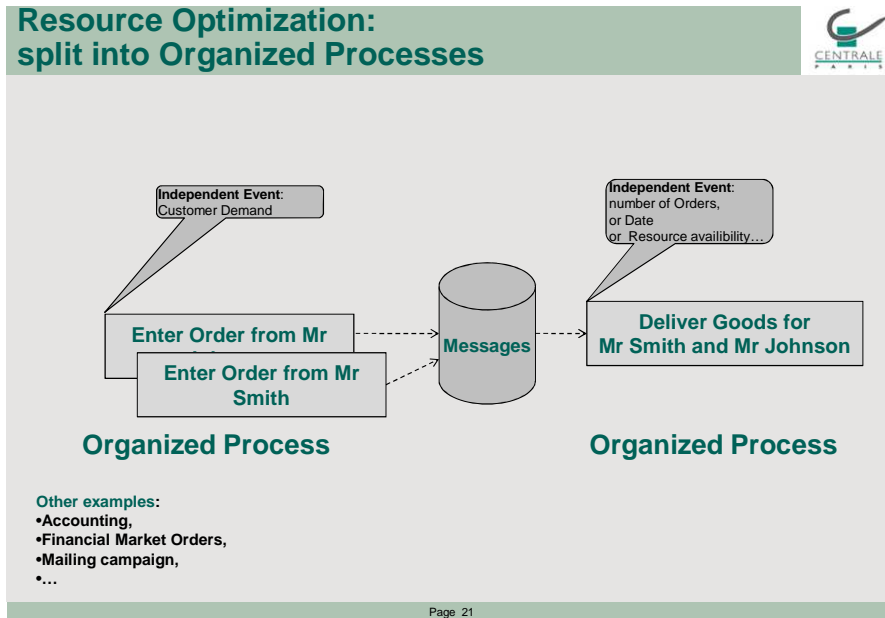
Enterprises currently do this:

- Accounting can be done at night by an independent Process

- Financial orders are grouped to be traded on financial markets
- Mailing campaign: letters are grouped to be sent

There is a big confusion between End to End Process (the one which is targeted to the client) and Organized Process: some Enterprises have lost the End to End Process vision and just focus on Organized Processes which is more visible because attached to an Organization Unit.

This is why some Enterprises have defined **end to end Process Pilots**: people who have a cross-unit responsibility.



Over time, Process Organization is improved by reorganizing and automating more and more Activities. For the same “**End to End Process**”, several combinations of “**Organized Processes**” may exist, each one being a different assignment of Activities to Actors.

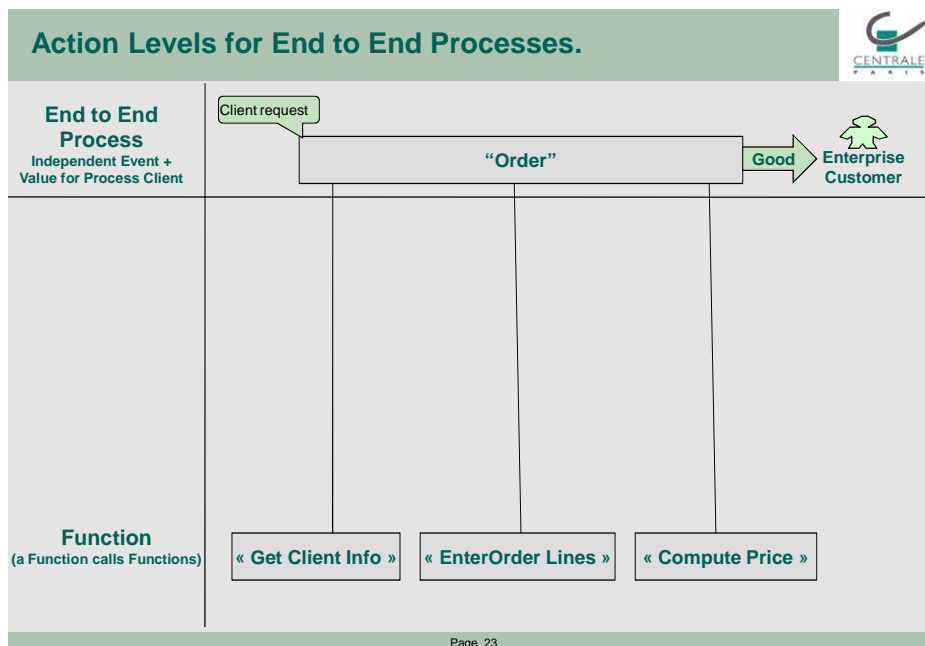
End to End Process defines **what** to do, while each Organization Process adds **who** does it and **when**.

3.8 Function and Activity

A Process is broken down into **Functions**.

Each Function may in turn call up other Functions.

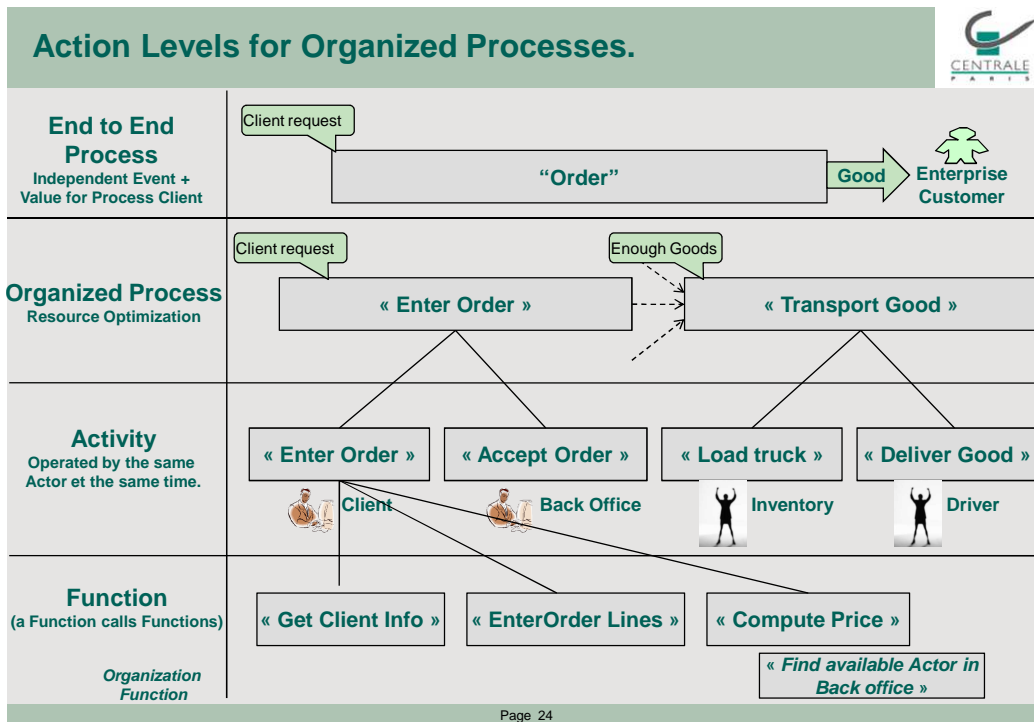
Functions can be classified: some people use the term “Rule” for Functions which represent constraints.



An **Activity** is the chain of Functions executed continuously by the same Organized Actor for the same Organized Process.

When defining Organized Processes, some **Organization Functions** must be added to the **Business Functions** already defined in the End to End Process. For example:

- Organization Function “Authorize a Worker”
- Organization Function “Find next Actor”
- Organization Function “What should I do, now?”



3.9 Information

To execute Actions, Actors require Information. This Information must be stored to be retrieved later on: this is called “persistence”.

Information is organized into **Entities** which are accessible thanks to an **identifier** which has a different value for each Entity Instance.

Information is classified into Entities like: Customer, Product, Contract, Delivered Service, Account, Entry, Organization Units, Position, Organization Actor, Worker, Employee, Automate, Building...

We suggest splitting them into 2 categories: **Primary Entities** (or Business Entities) used by Primary Processes, and **Resource Entities** (or Organization entities) used by Resource Processes.

Information on the same Entity may vary over time: this is why a **version** must be added to the identifier. It allows to maintain the history of values for the same Entity persistent.

Detailed Information is defined for each Entity.

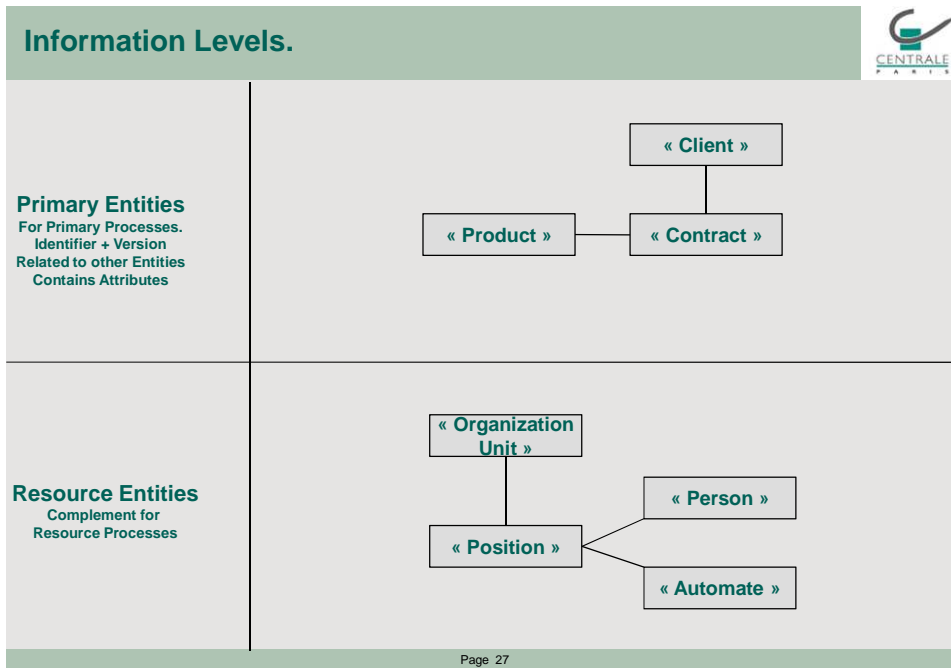
For example:

- The Entity “Person” has the **attributes** “Last Name”, “First name” and “Birth Date”.
- The Entity Instance “Mr Donald Smith” has **values** for Attributes: “Smith”, “Donald” and “3rd of October, 1980”

To standardize storage and presentations of values, each Attribute has a **Type**. For example the Type “Date” is reused for “birth date” and “Contract Subscription Date”.

Defining Types is important: the huge cost of software transformation for Y2K or Euro compliance is mainly due to bad definition of Types (see White Paper on Defining Entities).

Entities are interlinked through Relations.



Information and Actions are strongly related.

Entities and Processes for Operations

	Operation Entities	Operations Processes
Resource	Resource Entities: Organization Units, Roles Resources Actors + Rights/Duties Employees (+Contract,Account) Contractors (+Contract,Account) Partners (+Contract,Account) Providers (+Contract,Account) Automates: software,hardware,network Facilities:locations,building,equipement Cash External Informations	Resource Processes Select/update Workers. Select/update Automates. Manage facilities.
Primary	Primary Entities Product Model Product instance Client/prospect (+Contract, Account) Client Operations Client Payment	Primary Processes Create, Modify Product Model Produce,store,deliver Product Instances Create/modify Client, Contract, Account Manage Operations: deliver Value Manage Mktg campaigns Bill, receive payments
Mgt	Management Entities Decisions, Road Map, Aggregates by (Budget/real), (time), (product), (client), (unit), (nature)...	Management Processes Create/modify budget, Get results, react Free analysis

3.10 Organization

Organization Actors are assigned to an Organization Structure which is represented as a hierarchy of **Organization Units**: Headquarters, Direction, Division, Department, Branch, up the last level which is the **Position** to which an Organization Actor may be assigned.

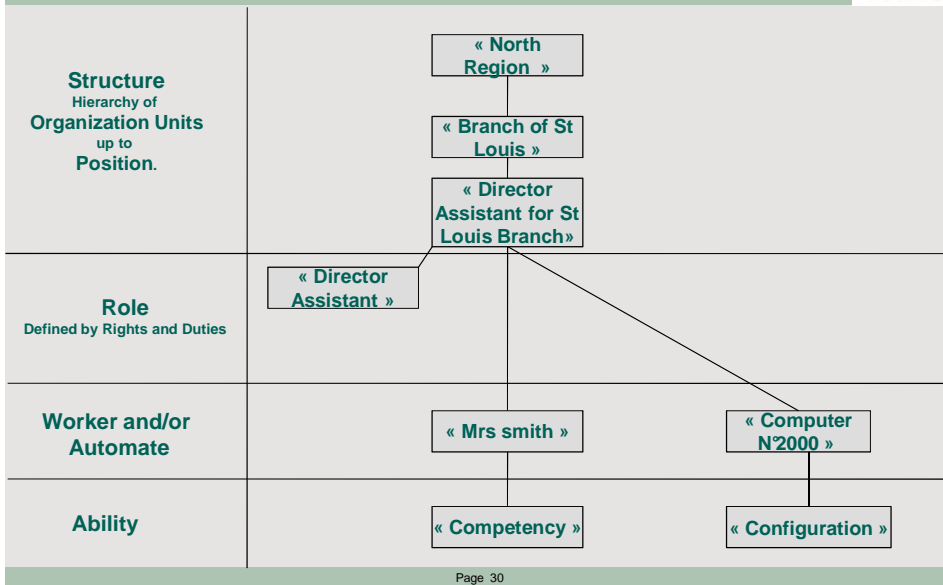
Each Organization Unit plays a **Role** defined as its **Rights** (what it can do) and **Duties** (what it must do). Assigned Workers and Automates have the ability to execute Actions.

For a Person, Ability is called "**Competency**".

For Computers, Ability is called "**Configuration**".

(see white paper on Entities for more details).

Organization Actor Levels.



3.11 Solutions

For a single Enterprise, the number of Organized Processes can be huge. For example, an Air Line company describes its Actions through 1,500 different main Processes.

The number of Activities is larger and the number of Functions even larger still!

To offer a Global Vision of Enterprise Operations, we must group these Actions into larger sets. We call a “**Solution**” a group of **Actions** which brings consistent Business value.

ex: HR Solution, CRM Solution...

So a **Solution Model** is associated with an Action Model, which generally is a **Software**.

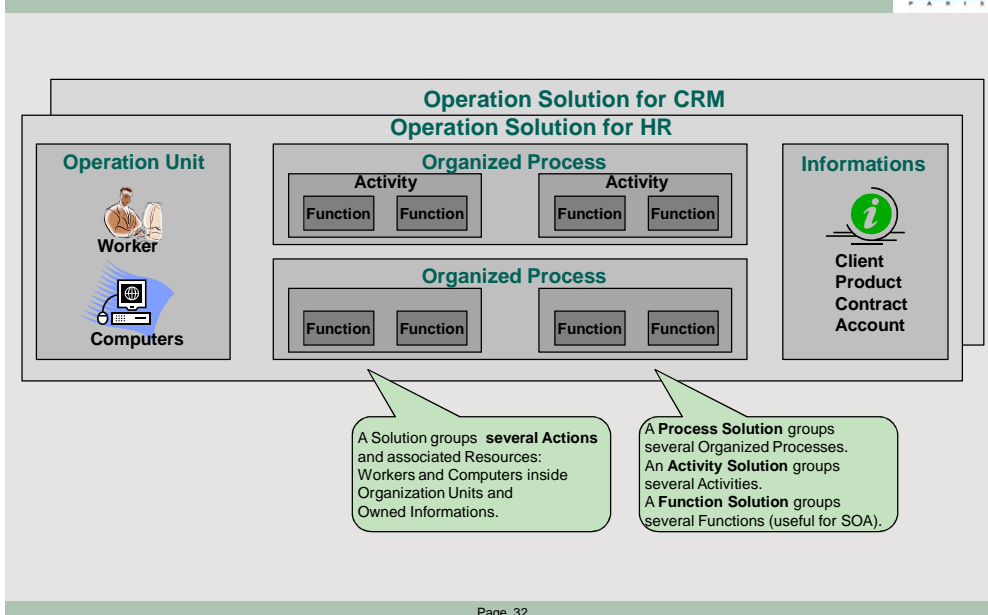
If several Enterprises use the same **Solution Model** each of them owns its **Operation Solution**.

The **Operation Solution** is an implementation of a Solution Model with its assigned Organization Actors (Workers and computers), and its owned Information.

Enterprise Operations is the sum of Operation Solutions.

A Solution Model can be implemented by an internal Software or by an external Package provider.

Solution=several Actions with their Resources



An Executed Solution may own or **Share** Resources (Workers, Computers, Information).
 A Solution Model may own or **Reuse** Functions.
 We need then to describe not only independent Solutions but also Shared and Reused Elements.

Remark: most of the time, a Solution is a set of Organized Processes. It can also be a set of Activities or a set of Functions. Example: if a Solution uses an external SOA Function, this Function is part of an **external Solution** with its Computers, Workers and Information.
 Solutions could be qualified as: “**Process Solution**”, “**Activity Solution**” or “**Function Solution**”.
 Solutions communicate by sending synchronous or asynchronous **messages**. The message Model is called the **Interface**, it represents how Solutions ought to communicate.

3.12 The Enterprise Model

The Model is the basis for Transformation: definition of a new Model and migration to this new Model.
 The “Model” has 2 meanings: Global Model and Detailed Model.

3.12.1 The Global Model

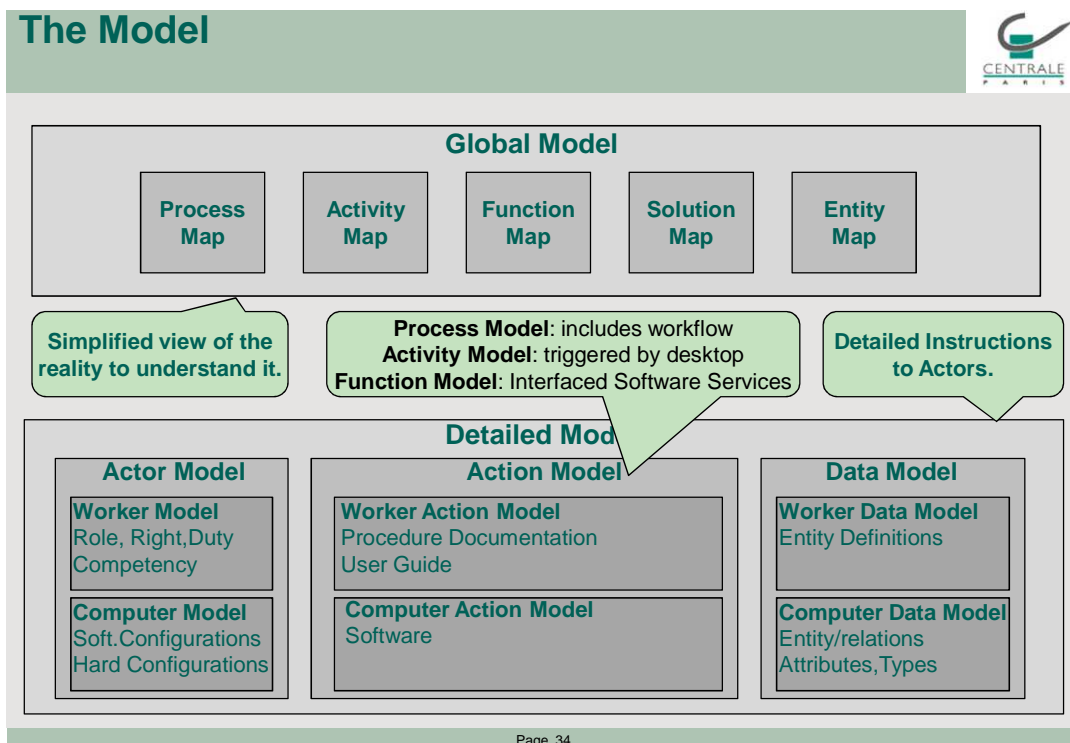
The **Global Model** helps us to **understand** how an Enterprise works by providing a simplified representation of reality. We then need Global “**Maps**” which formalize global views of the Enterprise: maps of Processes, Functions, Entities, Solutions.

3.12.2 The Detailed Model

If a Group of 10 companies wants each Company to manage itself, but to work in the same way, we say that the 10 companies reuse a single Model. So a Model also helps us to **standardize**. This standardization is achieved by reusing same detailed description.

The detailed Model includes:

- **Actor Model:** for Workers it defines Roles, Duties, Rights, and Competencies; for Computers it defines Roles and Configurations (hard and soft).
- **Action Model:** describes **Instructions** given to the Organization Actor so that he correctly executes Actions.
 - For Workers instructions are Documentation (Procedures, user guides)
 - For Computers, instructions are the **Software** itself
- **Data Model:** Describes Entities, Attributes, Types and relations between Entities



As we describe later on, Operations and Transformation are clearly defined: Transformation Processes change the Model and Operation Processes do not change the Model. So, it is very important to have a clear definition of what is a Model.

Examples of Transformation Processes which obviously change the Model:

- Make or buy a new Solution
- Change the Transformation Process
- Change the Data Model
- Change the IT technical standards

But it is not so obvious for other topics like:

- **Is a new Product changing the Model?** The answer is that it depends on your Product Model. In the Service industry, some Enterprises have built a “Product Factory” which allows the Business Professionals to directly create or update Products by assembling Services, creating pricing Rules or Eligibility Rules: for these Enterprises, creating or modifying the Product is part of Operations. For other Enterprises, it is part of Transformation. A powerful Product Model decreases drastically the number of Transformation Projects and improves time to market.
- **Is a new Process changing the Model?** The answer is that it depends on your Process Model. Some Enterprises have built a Workflow Modeling mechanism which allows the Organizer to directly update assignment of Activities to Actors. For these Enterprises, Process Modifications is part of Operations. For other Enterprises it is part of Transformation.
- **Is a new Branch creation changing the Model?** The answer is “no” as long as the Roles, Rights and Duties of this branch reuse the already existing Branch Model.
- **Is a new Organization Structure changing the Model?** If the new structure is just reassignment of Activities, without changing the Process Models, then this change is done by Operations.
- **Is maintenance changing the Model?** If the Software Model is changed, it is definitively Transformation, even if it is a very light change; it just reuses a light Transformation Process.

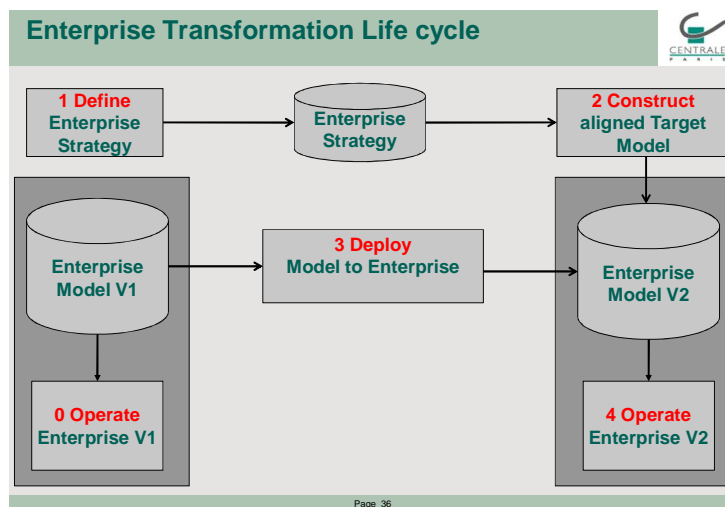
4 Enterprise Transformation

4.1 What is transformation?

As explained above, an Enterprise Operates using a Model.
 But the Enterprise is not static. It must move according to Strategic Objectives and Constraints.
 New Objectives mean a new, appropriate Target Model.

So Transformation entails:

1. Defining Enterprise Strategy evolutions: EA does not define this strategy, it is an input for EA
2. Constructing a new **Target Enterprise Model** in line with the strategy
3. **Deploying** this new Model inside the Enterprise
4. Once Deployment is done, the Enterprise may **Operate** according to the new Model



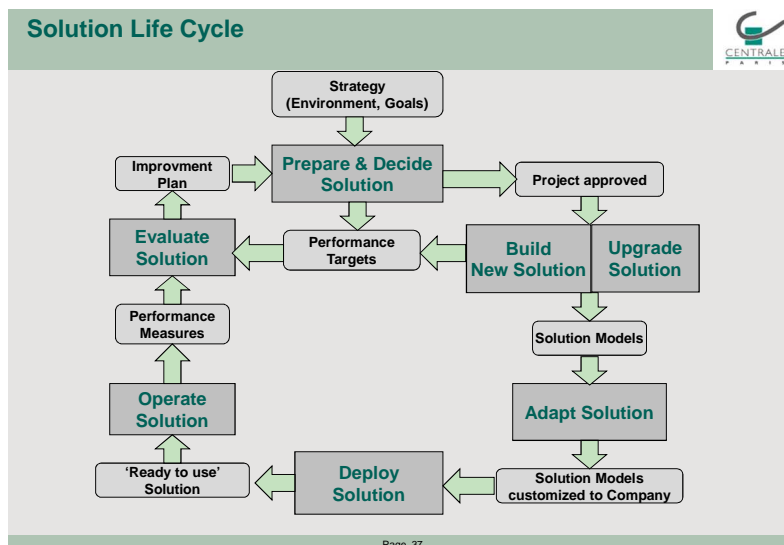
The Enterprise Model does not include Objectives. The Enterprise Model just explains how the Enterprise works at a given time. Objectives are part of Enterprise Strategy and justify why the Enterprise must define a new Model.

The **Transformation main Client** is Enterprise Operations (while the Operations main Client is the Enterprise Customer)

The **Transformation Provider** is a Package Solution provider, or a Tool provider

The **Transformation Partner** is the Consulting or Service Company which helps the Enterprise to build and Deploy

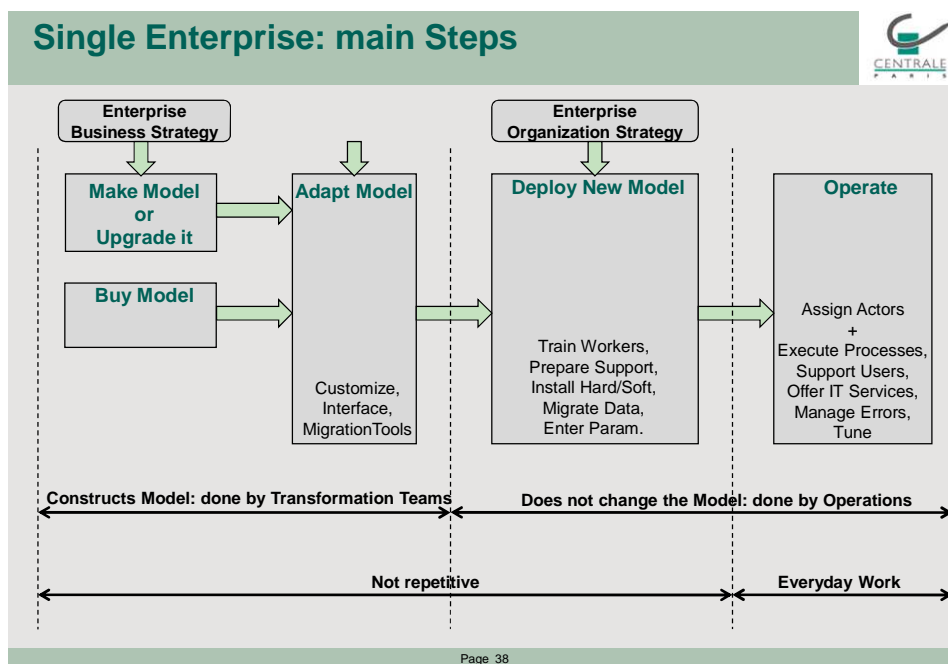
Another more detailed way of representing this life cycle is:



4.1.1 Main steps for a single Enterprise

The main steps are:

- **Make, Buy or Upgrade** the Model according to Business Strategy
- **Adapt** the Model to the Enterprise:
 - Customize the Model if the Model is bought (you buy a package)
 - Build Interfaces so that the new Model coexists with Legacy Model
 - Build Migration tools
- **Deploy** the new Model
 - Train Workers
 - Prepare the Support team which will help future Workers
 - Install new Software, and eventually new Hardware
 - Migrate Data
 - Enter Organization parameters and Rules: rights and duties useful for Workflow
- **Operate**
 - Operations includes assignment of Actors: Worker and Computers

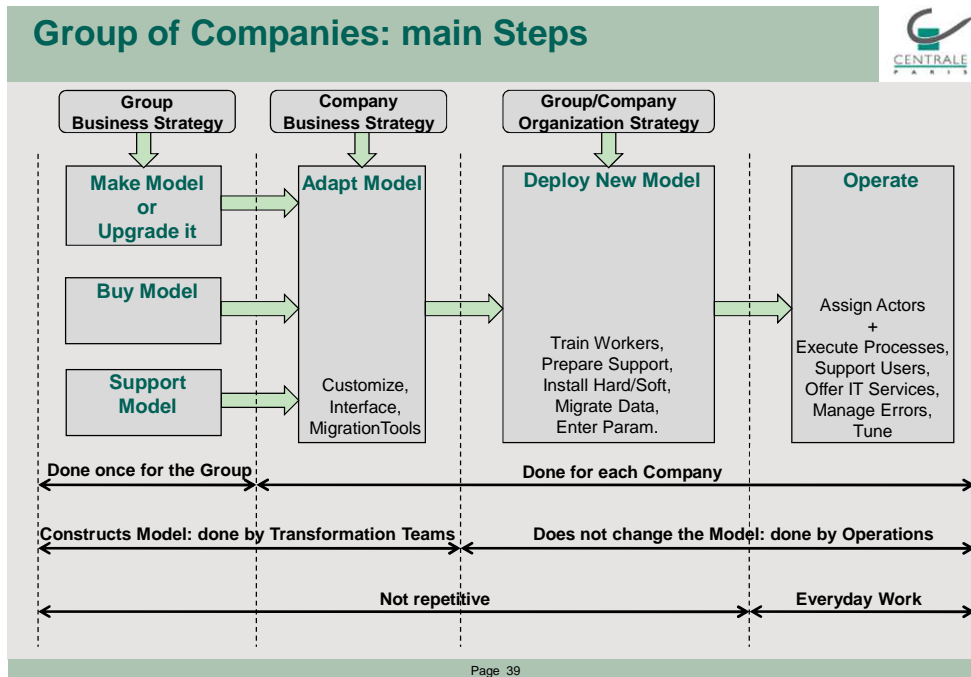


Whatever representation is chosen, some remarks can be made.

- **Strategy** is an input and not part of EA. Strategy is broken down into 2 sets of decisions:
 - **Business Strategy**: the model must support new Products, new Business Processes, new Partners, Providers or customers, but business strategy does not define **who** will do it
 - **Organization Strategy**: defines **who** (Organization Structure and Actors) operates the model. Ex: decision to open new Branches, to merge Back Offices, to outsource without changing the model
- Transformation can be divided into 2 parts:
 - **Constructing** the new Model
 - **Deploying** the new model
- Transformation Processes can be heavy (like new Solution Project) or simple (like change Price).
 - **Innovation** is part of an important Transformation
 - Any Model modification is Transformation. So, **Maintenance** is part of Transformation, not of Operations, even if it is done by dedicated teams, even if the maintenance teams use fast transformation Processes to correct a bug. Consistency of the model is under the responsibility of the Transformation team
 - If an Enterprise uses a **Rule Engine** to allow Business Professionals to directly modify a Product, this is also Transformation, because they modify the Product Model: they use a fast transformation Process without going through the long process of Classical Software transformation. But it is still Transformation (done as a “customization” step).

- Changing the Actor Model is also Transformation
 - Worker Model: categories of Roles are in the Model, they can be used by the Security System or the Workflow
 - Automate Model: software configuration models are in the Model
- Changing the Organization is not a change of model, it is not Transformation
 - Definition of a new Organization Chart without changing the Model is not Transformation
 - Assignment of Actors to Activities is not Transformation: the model does not change
- An Option: **Deployment** may be split into 2 parts: when Deployment is across multiple similar Organization Units (like a Branch distribution network), the deployment of the first Entity is under the responsibility of the Transformation team, while following deployments are under the responsibility of Operation teams.

4.1.2 Main steps for a Group of Companies



When a Group of Companies decides to share a Solution Model, the steps are similar except:

- Business Strategy is broken down into **Group strategy** and **Company Strategy**
 - The Group decides to Build and Support a Reusable Solution
 - Each Company decides to use the Group reusable Solution when appropriate: if the Group decides for the Company, there is no Company Strategy
- "Adapt Model" and "Deploy new Model" are done independently for each Company: a **Group Support team** must help Companies.

4.1.3 Construction Processes and Information

Entities and Processes for Construction		
	Entities for Construction	Processes for Construction
Resources	Construction Resources Entities Transformation Units, Roles Resources Actors + Rights/Duties Project Manager, Business Analysts, Developers, Architects Automates: Software, hardware Facilities External Informations	Construction Resource Processes Select/update, Automates, Tools, methodology... Subscribe Provider Contracts. Manage Transformation people like Architects, Business Analysts, Developers Define Transformation IT infrastructure
Primary	Construction Primary Entities Project Planning Maps Requirements Component Templates	Construction Primary Processes Define EA road Map Build or upgrade Solution Build or Upgrade Architecture Manage Maintenance
Mgt	Construction Management Entities Aggregates	Construction Management Processes Manage Transformation Indicators, Manage transformation budgets

Page 40

4.1.4 Deployment Processes and Information

Entities and Actions for Deployment		
	Entities for Deployment	Processes for Deployment
Resources	Deployment Resources Entities Deployment Units, Roles Resources Actors + Rights/Duties Teachers, people for Migration, parameters, ... Automates: Software, hardware, network Facilities	Deployment Resources Processes Select/update Deployment Workers and Automates
Primary	Deployment Primary Entities Training courses Workers to train Equipment Files of data to migrate	Deployment Primary Processes Deploy new IT infrastructure Deploy a Solution in an Enterprise Deploy Correction Migrate data Change Organization units Move and Train Users Enter parameters
Mgt	Deployment Mgt Entities Aggregates	Deployment Mgt Processes Manage Deployment Indicators

Page 41

4.2 Transformation complexity

Some simple Transformation Processes are automated (like deploy a bug correction, or change a Price). But Transformation Processes like “Manage a Project” are **more complex** than Operation Processes and are solely modeled through documentation called Methodology. There is no end to end Software Transformation Process with full retro-engineering.

Nevertheless, some sub-parts of the Transformation Process (Transformation Functions) are modeled and supported by Software Tools called « Design or Development tools ».

Construction Functions	Deployment Functions
Gather requirements	Migrate data
Design Process, Activity, Functions	Prepare support
Build Data Model	Train
Develop Software	Deliver Software, Hardware
Test	Enter Parameters
Manage Software Configuration	
Build Interface, migration tools	
Integrate	

5 Describing an Enterprise with the CEISAR cube

CEISAR delivers a simplified but consistent view of an Enterprise based on 3 dimensions.

5.1 Consistency, Continuity and Metrics are key

- **Consistency** means that the **same set of concepts** must be used for describing Organization Actors (Worker and Computers), Actions, and Information. This consistency must not prevent us from presenting **different views** of the Enterprise model, according to the profile of the reader.
- **Continuity** means alignment between Organization, Processes and Data, to **avoid discontinuity** between different approaches like Entity definitions, Process reengineering, Block cartography, Component approach, SOA Approach and MDM (Master Data management).
- **Metrics** means that Systems characteristics must be quantified (Complexity, Efficiency, Agility, Reliability...) and presented in Reports, to allow:
 - objective comparisons with competitors
 - objective comparisons between Companies of the same Group
 - progress monitoring over time

5.2 Which Dimensions?

But what are the key dimensions for describing the Enterprise?

All of following dimensions are interesting.

Real World	Model of the real World
Operations (the present)	Transformations (the future)
Specific (the Solutions)	Shared (the Foundations)
What it does (Value)	How it works (Organization)
Actions (Process, Activities, Functions)	Informations (Entities, Attributes, types)
Business	IT
Global view	Detailed view
Internal Activities	External Activities
Company subsidiarity	Group centralization

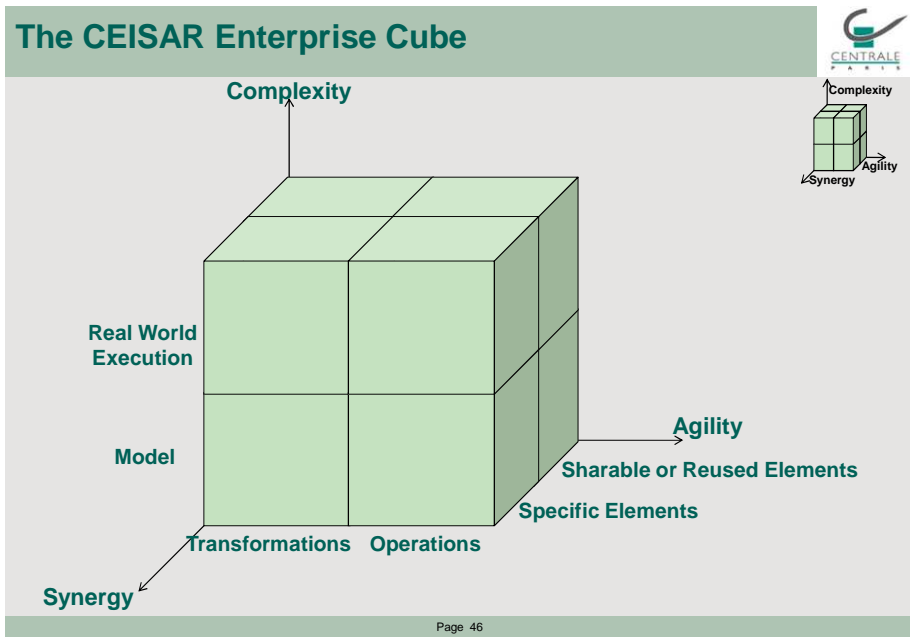
Yet it is impossible to represent a Model with 9 dimensions! The human brain may represent 3 dimensions, not more.

To select the really key dimensions, we must identify the main business concerns which are:

- Understanding Enterprise **Complexity** which means splitting Real World from its Model.
- Increasing **Agility** which means splitting Operations Processes (the present) and Transformations Processes (the future).
- Finding the right **Synergy** level, a good balance between Centralization (or mutualization) and decentralization (or subsidiarity), between specific elements and reused or shared elements.

And last, but not least, the **same** Model must cover **all** aspects of the Enterprise: different views will be delivered for each Role (Executive, Architect, Developer, Business Analyst).

Based on these objectives, CEISAR proposes to model the Enterprise as a cube.



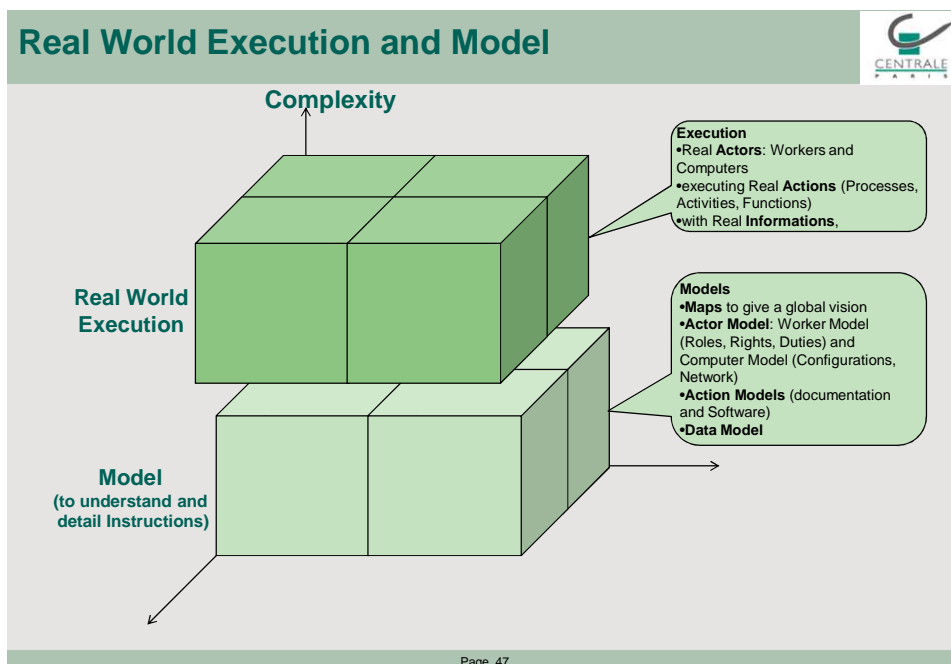
Two complementary dimensions will be used inside each of the 8 cubes:

- **Worker or Computer**: which helps us to understand that:
 - Worker Model (Role, Right, Duty, Competency) is not represented in the same way as Computer Model (Role, Duty, Software or hardware Configurations)
 - Instructions can be written inside documentation for Workers or inside Software for Computers
 - Entity Definition for Reused language between Workers is different from Reused Data Model understood by Computers
- **Actor, Action and Information** which is the baseline of Enterprise Model

5.3 A Model for understanding Complexity

As explained previously:

- The **Global Model** helps us to **understand** a complex reality by providing an understandable and simplified representation of that reality
- The **Detailed Model** describes **Instructions** for Workers (Procedure documentation) or Automates (**Software**)

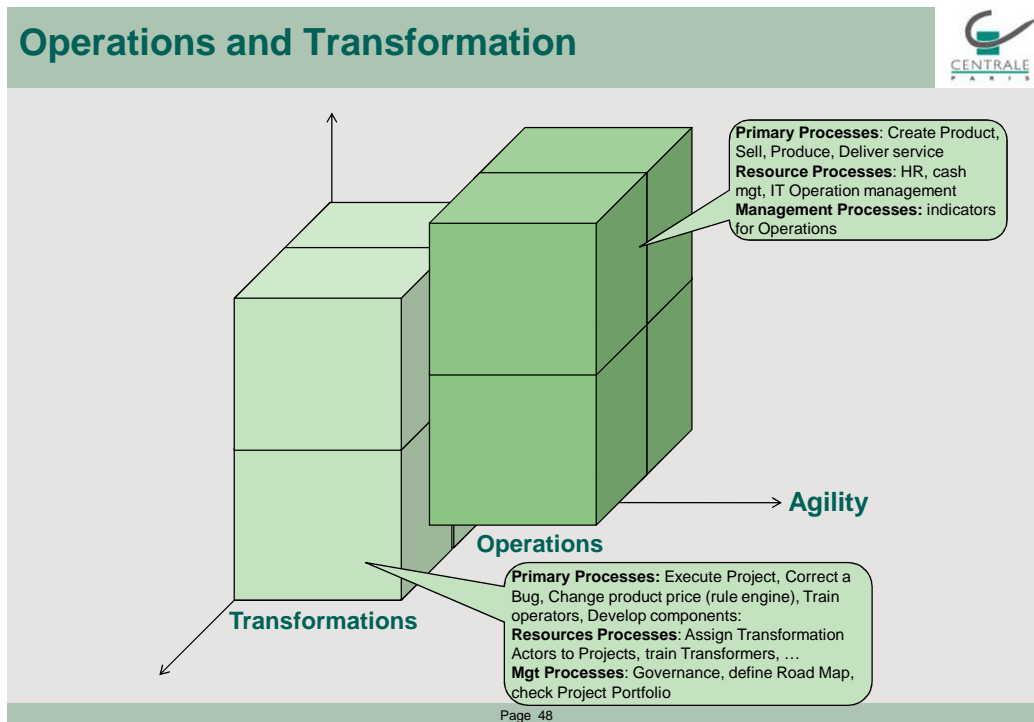


5.4 Operations and Transformations

The Operation Processes are sufficient as long as the Enterprise Operates according to the same Model. But if the Enterprise has to be Transformed, some Transformation Processes must also be defined to manage Actions such as: “Build an EA Road Map”, or a “Solution Project”, or a “Maintenance Process”.

Transformation Processes can be classified just like we did for Operation Processes:

- Primary Processes like “Execute a Project”
- Resource Processes like “hire a new Architect”
- Management Processes like “EA Governance”



5.5 Synergy

If nothing is shared or reused, each Solution is independent. The sum of the heterogeneous Solutions is the Enterprise System, but a sum of independent Solutions does not make for a good global System.

How to combat the legitimate selfishness of each Actor?

Synergy must be defined: which shared or reused elements?

We propose to use “Share” for Real World Execution and “Reuse” for the Model.

Different Business Units of an Enterprise may **Reuse**:

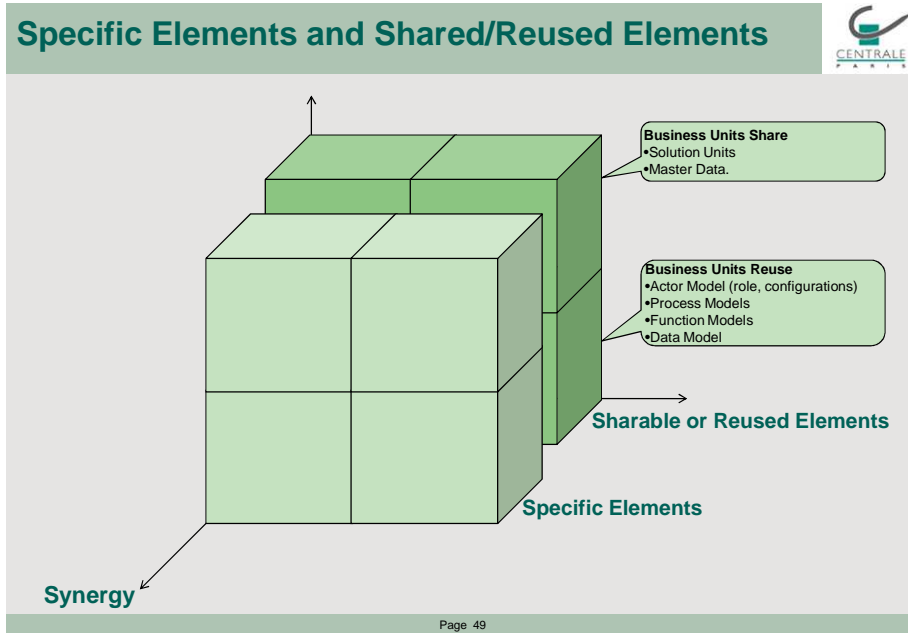
- a Global Model formalized with “Maps”
- the same Solution Model (ex: same accounting solution, same communication solution)
- the same Components to build Solution Models
- Data Model
- Transformation Methodology

Different Business Units of an Enterprise may **Share**:

- “an **Organization Unit**” which means **transferring execution of Actions** to this Organization Unit
 - transfer Call Center Actions to a centralized Call Center Unit
 - transfer IT Operation Actions to a centralized IT Operation Unit
 - transfer Transformation Actions to a centralized Transformation Unit
- **Master Data** like Customer Data.

Note that you can reuse an Action Model like a Process Model, but you **cannot share execution of an Action**, you can only transfer it.

Sharing is much more efficient if same Model is **reused**: this is obvious if you want to manage Master Data. But you can reuse without sharing: the same Solution model can be reused by different Units which Operate independently.



The set of sharable/reusable elements is often called Architecture.

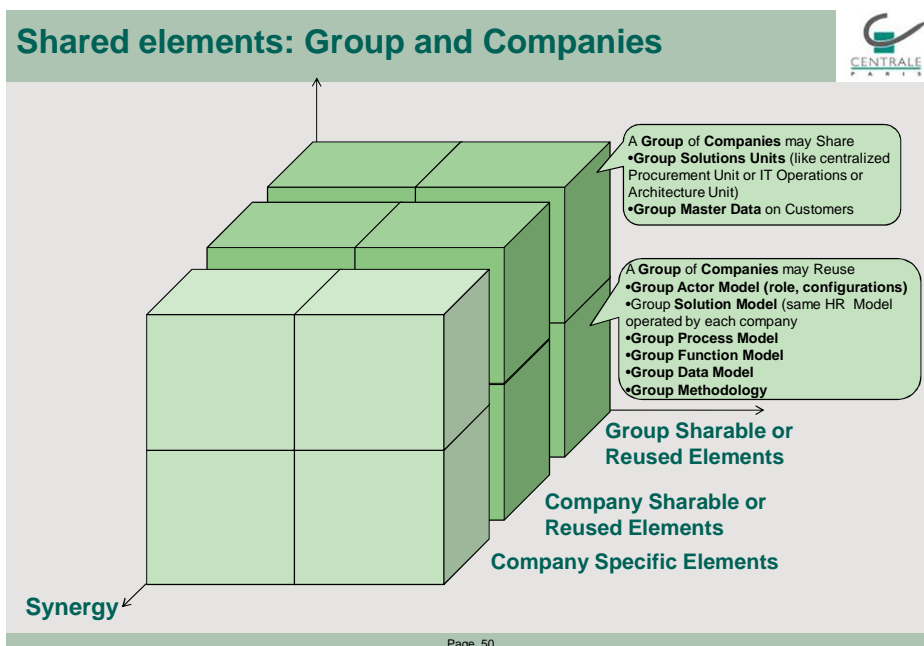
But “**Architecture**” is an ambiguous term:

- it means “**structure** of a System”: the break-down of a complex System into smaller elements
- it also means “**approach** and tools to deliver a structured system”

To be more open in our approach, we propose that Architecture is “**what allows or helps Sharing or Reuse: by extension, it also includes what is Shared or Reused**”.

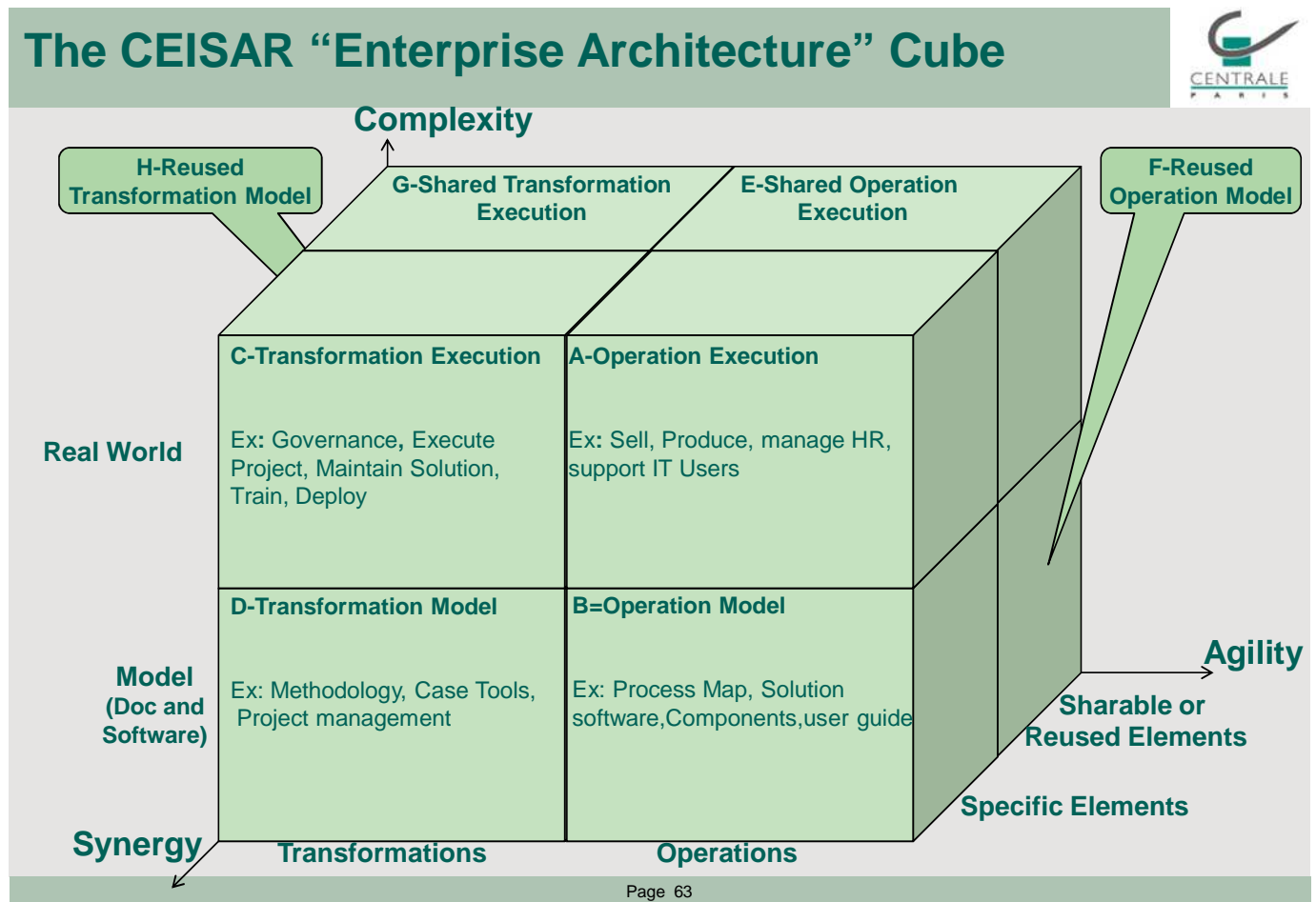
5.5.1 Group and Company levels

In a Group composed of several Companies, the Group may propose Group Shared Elements and the Company may enhance them by adding Company Shared Elements.



5.6 Summary

We have tried to summarize the Enterprise System in the following detailed Cube. To facilitate future reference, we identify each box with a letter.

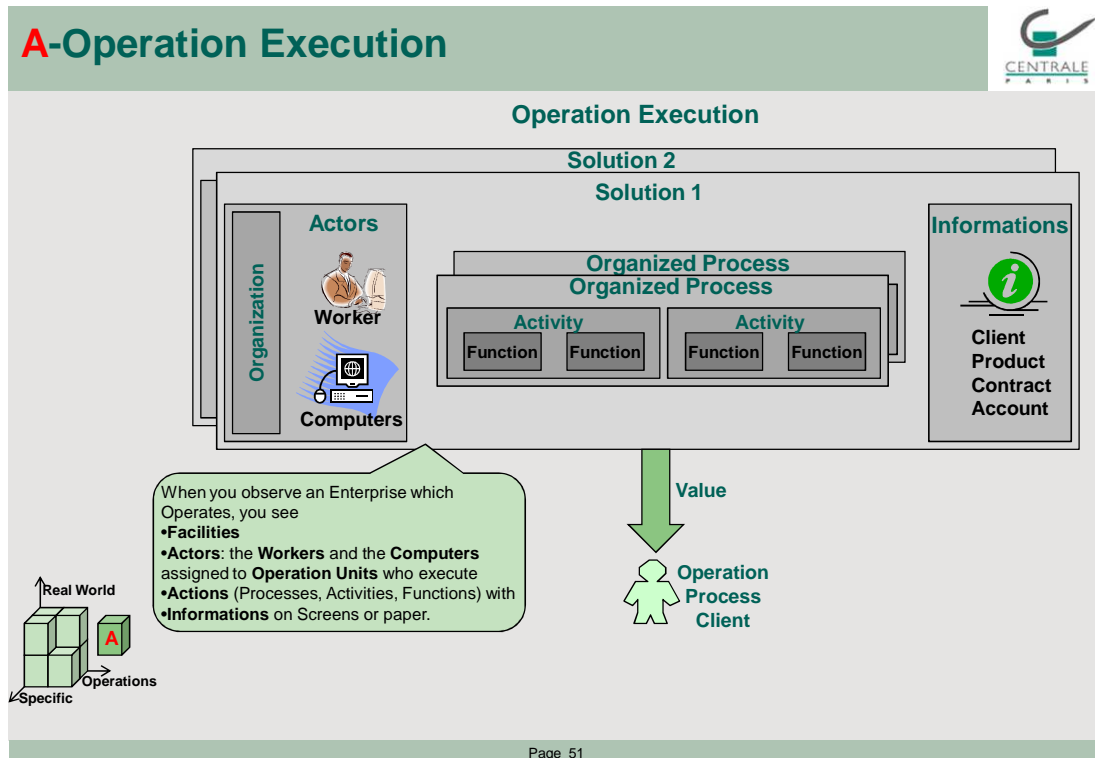


6 Describing each box

Now that we have defined the 3 dimensions (Complexity, Agility and Synergy) which form 8 separate cubes, we must define the contents of each cube.

This enables us to classify the different aspects of EA, and may help a Company not to overlook important topics when defining a new target.

6.1 Operation Execution



If you observe an Enterprise which Operates, you first see the facilities, you then see Actions executed by people who work: the **Workers**.

These Workers can be Employees of the Enterprise, or sub-contractors.

They are grouped into **Organization Units**.

All of them have access to paper-based information.

Many of them use **Work Stations** to execute **Actions** and to access **Information**.

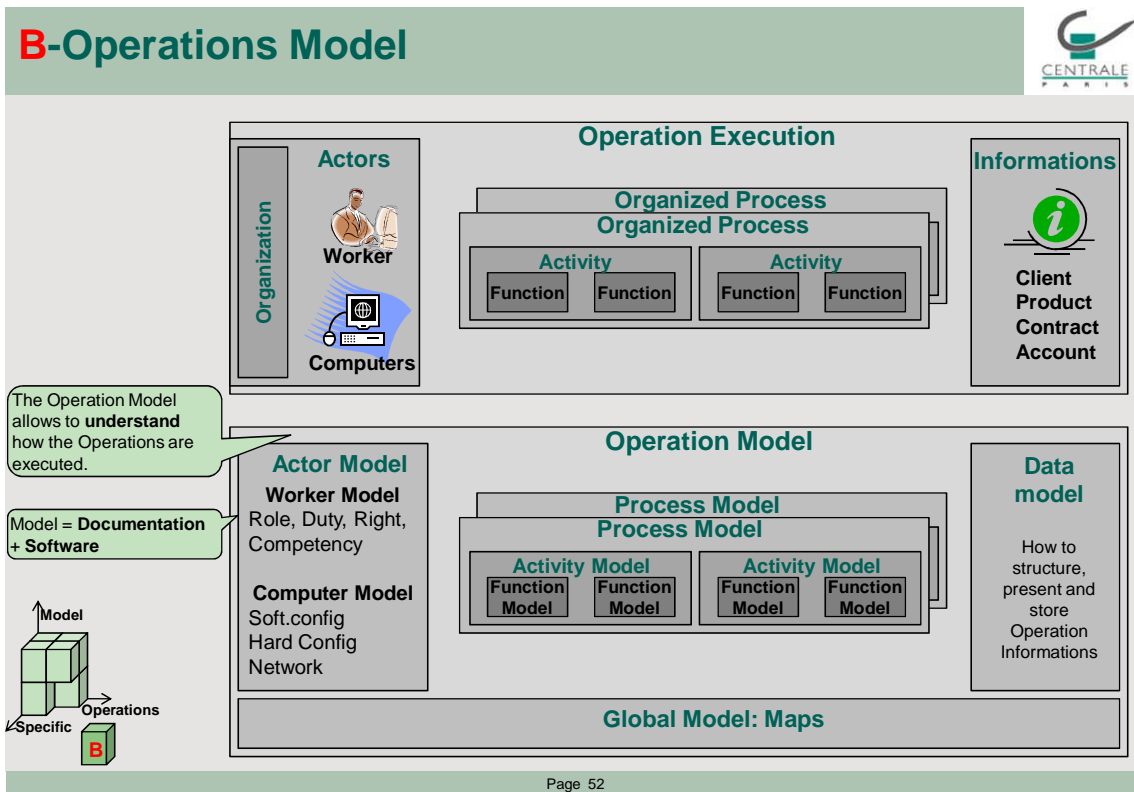
The consecutive suite of Actions executed by the same Actor on the same Process is called an **Activity** (as previously defined).

You notice that Activities are broken down into **Functions** (including **Data Access Functions**).

Part of the real world is not visible:

- You do not see the Software running, but if you suppress it, Operations quickly come to a halt
- You do not see Information stored in computers, but you see Information on paper or screens
- You do not see the Organizational Chart, but you understand that it is an Organized world with leaders and a team structure

6.2 Operations Model



When Enterprise Operations becomes too complex, it must be formalized into **Models**.

The **Global Model** helps us to understand how an Enterprise works. It is also the basis for Transformation: definition of a new Model and migration to this new Model.

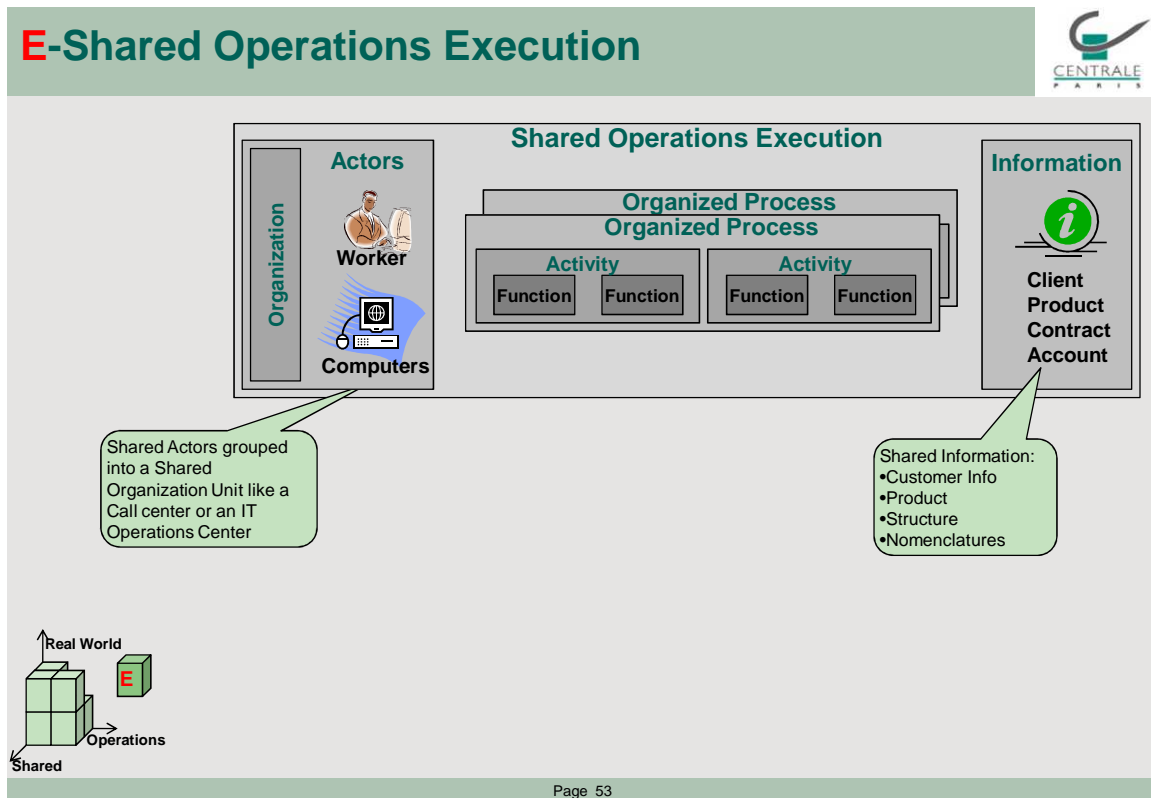
The **Detailed Model** describes:

- **Actor Model:**
 - For Units, defines Unit Roles
 - For Workers defines: Roles, Duties, Rights, and Competencies
 - For Computers (Work Stations or Servers) defines: Roles and Configurations (hard and soft)
- **Action Model:** Describes Instructions given to organization Actors so that they can correctly execute Actions.
 - For Workers: instructions are Documentation (Procedures, user guides)
 - For Computers: instructions are **Software**
- **Data Model:** includes documented definitions of Entities (what a Customer is, what a Product is...) and structured Data Model with Entity identifier, attributes, types and relations between Entities

As we have defined 3 types of Actions: Process, Activities, and Functions, the Computer Action Model can be divided into 3 layers:

- The **Process Model** which includes Workflow Functions for assigning Activities to Actors
- The **Activity Model** which calls Functions and is triggered by the desktop or the Workflow
- The **Function Model** modeled through Software Services callable through an Interface

6.3 Shared Operations Execution



To optimize Resources, the Business Units share Operations Elements.

- Business Units of an Enterprise may **share an Organization Unit**: for example, a back office or the Call center which works for several Solutions
- Business Units may also **share the IT Operations Units**, like IT Operations center or Worker Support Unit
- Business Units may share **Master Data** between Solutions: like Customer information

6.3.1 Centralizing a Solution Unit

A Group of several Companies may decide to create, at Group level, a Solution Unit which operates for several Companies. For example: procurement, HR, call center, specialized back office, etc.

Motivation

- economies of scale

Difficulty

- to build processes which run across different organizations

6.3.2 Centralizing IT Operations

A Group may decide to centralize its IT Operations Center and Worker support

Motivation:

- economies of scale
- to formalize contracts between IT Users and IT Operations: Service Level Agreements (SLA) are not easy to design and quantify, but SLA are more and more mandatory to assume the management of IT operations in respect of users needs

Difficulty

- it is more difficult if there are no standardized IT Operations approaches and tools in the different Companies of the Group

6.3.3 Centralizing Master Data

The Group may decide to centralize Group Data stored on a Single Server and Piloted by a single Group Unit.

Motivation:

- shared view of customer (global profitability, global risk, **equipment in Products**, behavior)
- shared view of Group and Company Organization, rights, duties
- shared definitions for nomenclatures
- ...

Difficulties

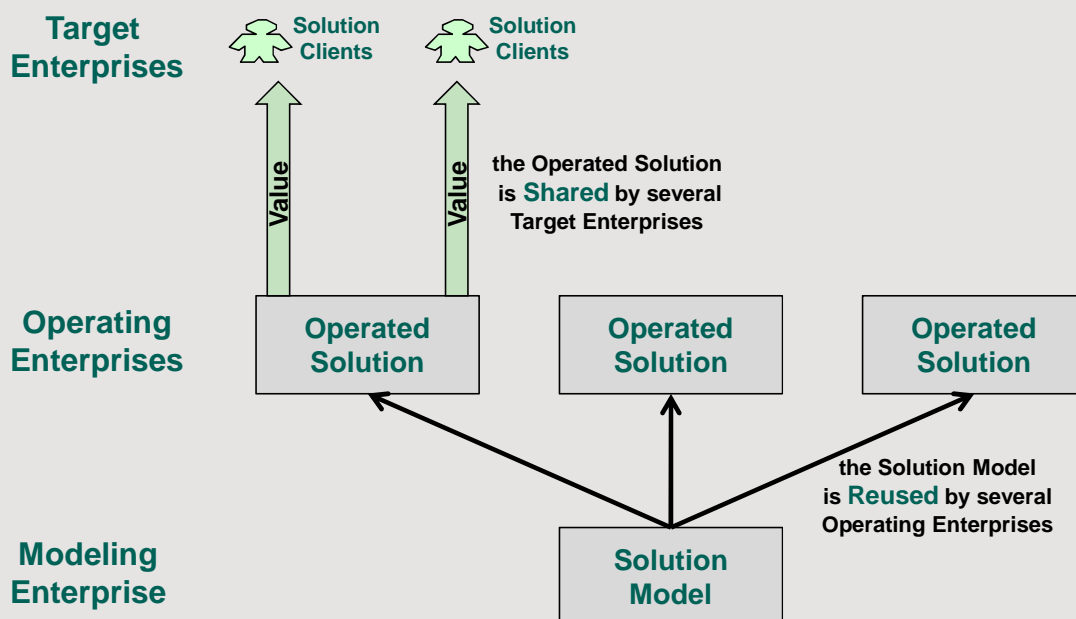
- it supposes sharing of Data Model (see below)
- in defining who is the **Data Model Owner** (the Transformation Worker who can modify the Data Model) and the **Data Operation Pilot** (the Operation Worker who can modify data or decide who can modify it)
- in managing synonyms (**yet that brings great value both for business and technical people**)

6.3.4 Target Enterprise, Operating Enterprise and Modeling Enterprise

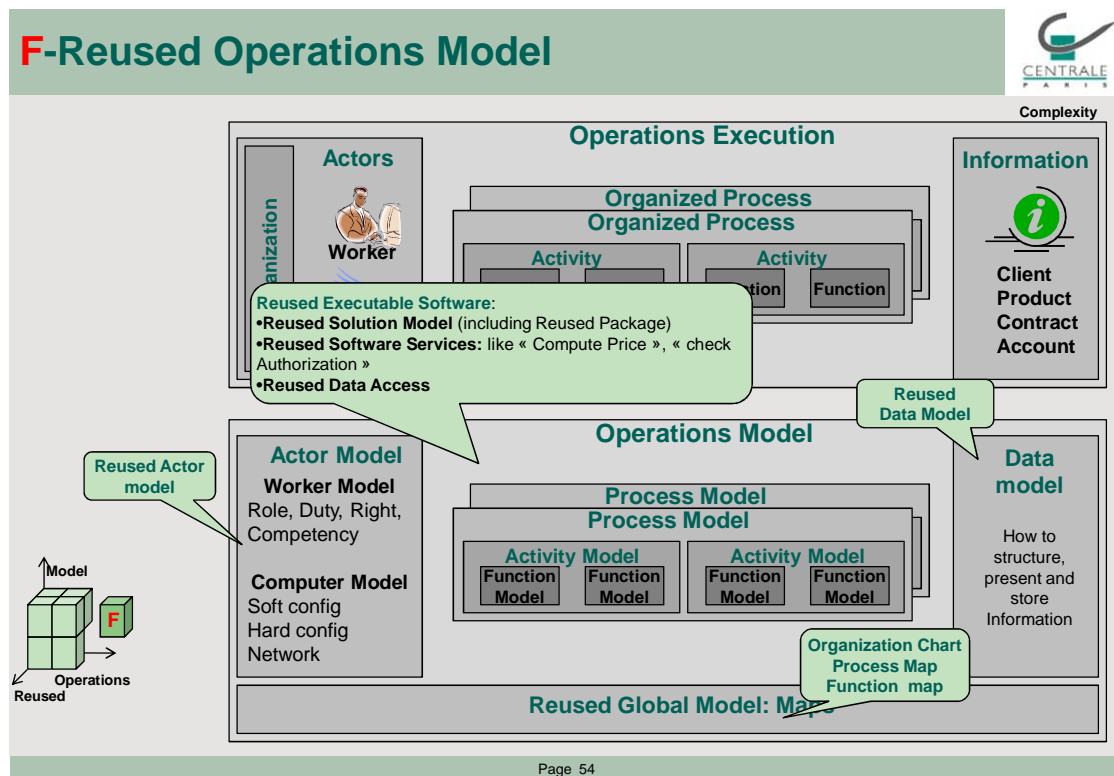
The same Enterprise delivers Value to its Customers by executing Operation Solutions according to an Operation Model. But there exist other scenarios which can be different for each Solution:

- **Outsourcing Operations:** an Enterprise (the Target Enterprise) may outsource Operations to another Enterprise (the Operating Enterprise)
- **Outsourcing Model:** an Enterprise (the target Enterprise) may Operate itself, but according to a Model delivered by another Enterprise (the **Modeling Enterprise**): it outsources the Model.
- **Outsourcing Operations and Model:** an Outsourced Operating Enterprise may Operate according to a Model delivered by another Enterprise (the Modeling Enterprise): 3 types of Enterprises must then be defined
 - The **Target enterprise** which owns the Customers
 - The **Operating Enterprise** which Operates
 - The **Modeling Enterprise** which Transforms

Shared and Reused Solutions



6.4 Reused Operations Model



Sharing Operations elements is much more efficient if the Enterprise stakeholders reuse the same Model. For example if an Enterprise shares its **IT Operations Center** between the different Solutions, it is much more efficient if IT Operation Processes are the same, or if Server Configurations are the same, **if they all belong to a Model.**

Another example: if an Enterprise shares its **back office Unit**, productivity will be higher if Solutions reuse the same User Interface, or the same security Functions, or the same master Data Model **which are part of a Model.**

An efficient way of ensuring that different Solutions make a good overall System, is that they **reuse same Elements.**

This is the case in the **Car industry**: with maturity, manufacturers understood that reusing parts to offer a new car model meant

- Easier design and fast time to market
- More reliability
- Lower costs

Today, the car market is dominated by those manufacturers who reuse most effectively. Reuse rate has become the key competitive advantage.

This will be the same with Enterprise Architecture: reusing components automatically generates good structure, which means less time and cost to deliver a new Solution, and more reliability for this Solution.

What Model Elements could an enterprise Reuse?

We follow the exact same structure that we defined above:

- Reuse of Global Model
- Reuse of detailed Model:
 - Reuse of Actor Model
 - Reuse of Action Model
 - Reuse of Data model

6.4.1 Reuse of Global Model

An efficient way to ensure that Solutions are well designed is that they **belong to a global map** where each Solution is in the right place and communicates well with other Solutions. Different Maps may exist:

- Process Map: Process classification, Process interactions

- Activity Map: to represent work to do for each Actor or Organization Unit
- Function and Service Map: component repository (useful for SOA)
- Block Map: hierarchic break-down of Blocks and interactions between Blocks
- Entity map: relations between Business Entities and Entity patterns
- Class map: Class repository

The difficulty is not only to **define** maps, but also to **align** the different maps, like:

- Aligning Entity Map and Process Map
- Aligning Process Map and Block Map

Remarks: some maps are not part of the Model, they are part of the Real World, for example:

- Organization Unit Structure
- Computer Network Map: location of Automates and communication links
- Data location map: where Data is located
- Software location map: where Software is located

6.4.2 Reuse of Actor Model

An Enterprise may Reuse:

- Worker Model: standard definitions of Roles, Rights, Duties, Competencies: to ease communication, training, worker mobility
- Computer Model: standard definitions of Configurations, which enable delivery of the same solutions

6.4.3 Reuse of Action Model

An Enterprise may Reuse:

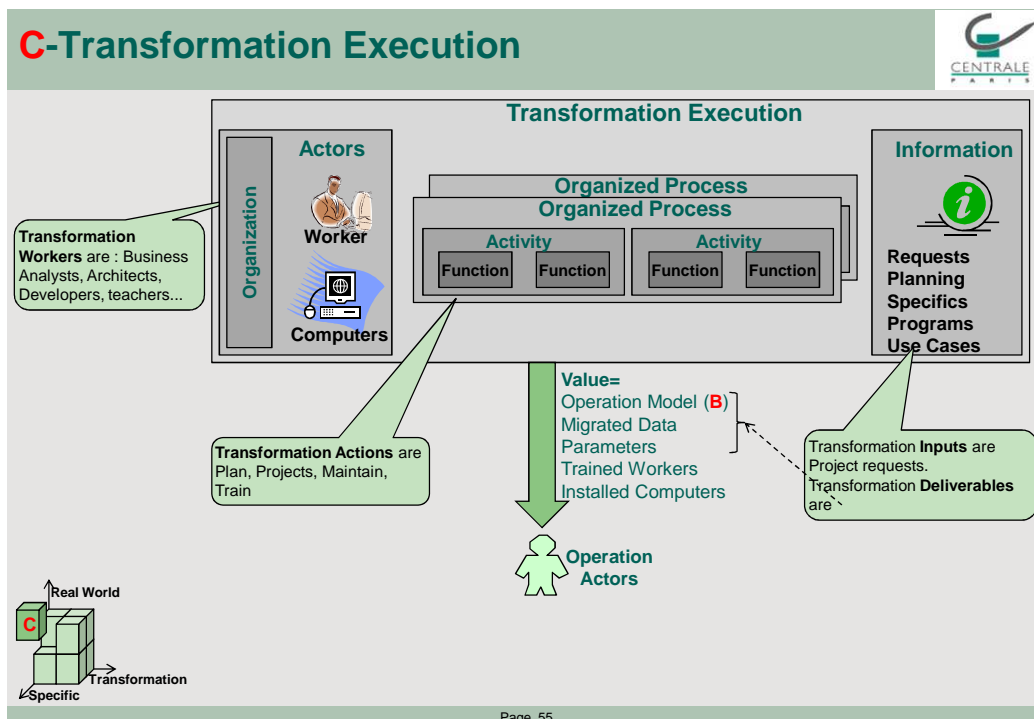
- **Solutions**
- **Functions** like Pricing Functions, Security Functions, User Interface Functions, Workflow Functions, Data Access Functions

6.4.4 Reuse of Data Model

An Enterprise may Reuse:

- Definition of Entities like Clients, Products, Organization Structure, with same identifiers
- Computer Data Model: which is a condition for sharing Master Data

6.5 Transformation Execution



To **build, adapt, deploy** an Enterprise model composed of Solutions and Shared Elements, the Enterprise requires a set of **Transformation Processes**.

The Transformation **Client** is "Operations".

The **Value** brought to Operations is:

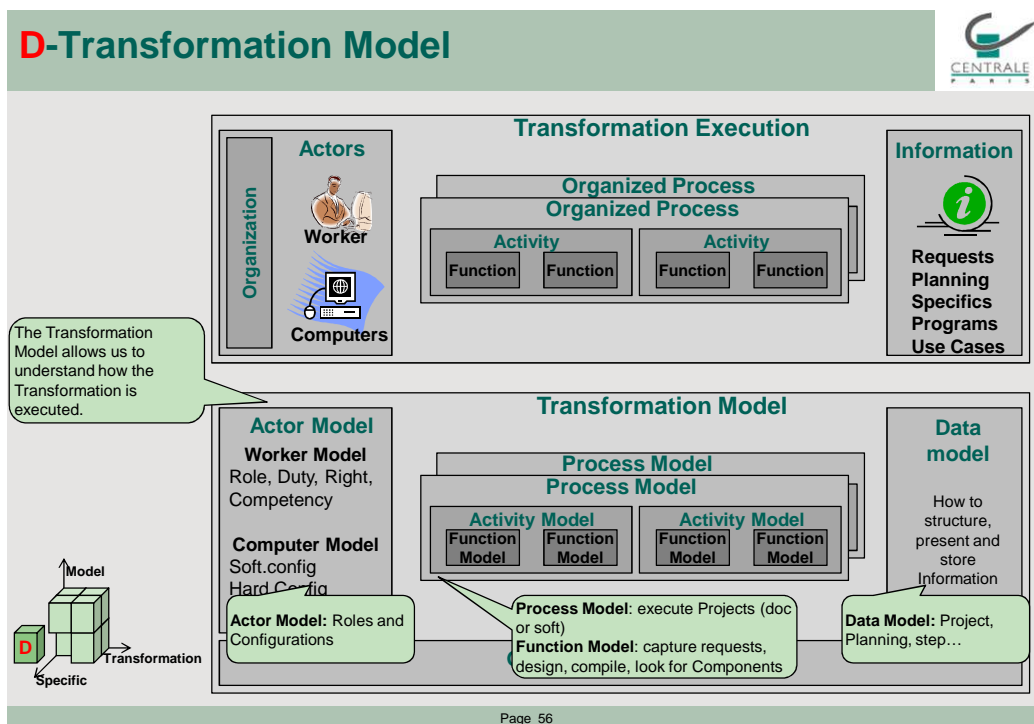
- At Executed Operation level (A): Trained Actors, Migrated Data
- At Operation Model level (B): Documentation and Software which describe the New Target Operation Model

The **Transformation Actors** are not Salesmen or Back Office employees, they are Project Managers, Architects, Business Analysts, Developers or Teachers.

The **Transformation Actions** are not "Sell" or "Produce", they are "Define Global Road map", "Execute Project", "Correct a bug" or "Train an Operation Actor".

The **Transformation Information** is not "Product", "Customer", "Contract", it is "Strategy", "Requests coming from Operations", "Planning", "Delivered Operation Model".

6.6 Transformation Model



To execute Transformation, we need a Transformation Model.

The **Detailed Transformation Model** describes:

6.6.1 Transformation Actor Model

- For Transformation Units, it defines Unit **Roles**: Process Design Unit, Development Unit, Maintenance Unit, Training Unit...
- For Transformation Workers it defines **Roles, Duties, Rights and Competencies** for Business Analysts, Developers, Architects
- For Transformation Computers (Work Stations or Servers) it defines **Roles** (Development Center, Integration Center, Test Center...) and **Configurations** (hard and soft).

6.6.2 Transformation Action Model

It describes Instructions given to Transformation Actors.

- For Transformation Workers: instructions are in Documentation (often called **Methodology**)
- For Computers: instructions are in **Transformation Tools** (Design Tools, case Tools, Test Tools, Compilers...).

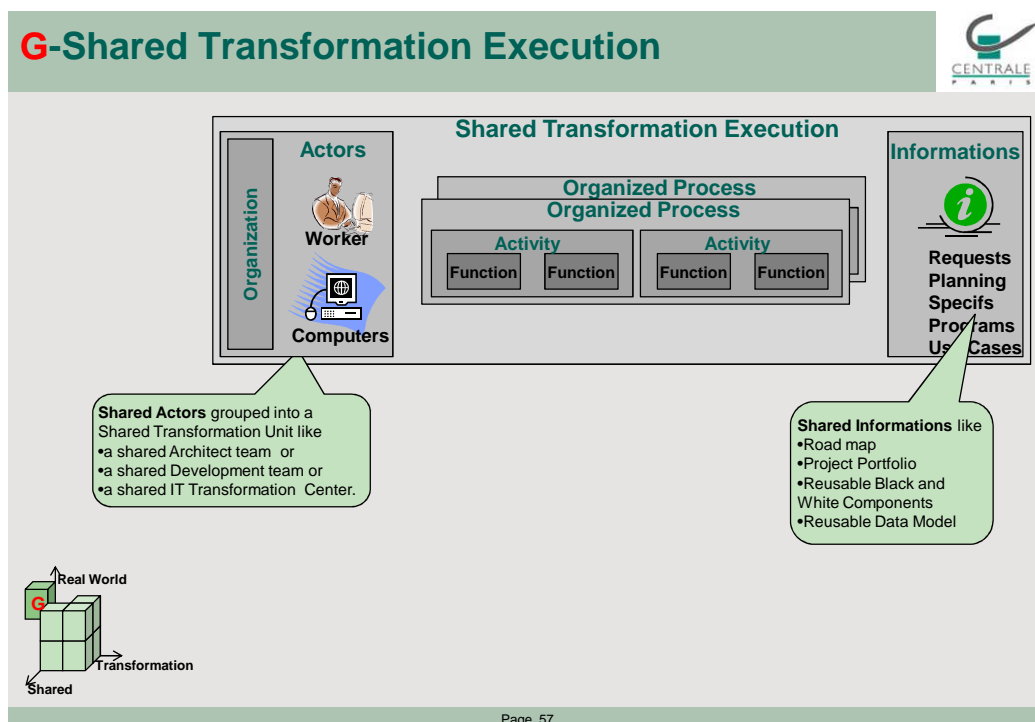
Transformation Processes are more complex than Operation Processes: Execute a Project is more complex than Execute an Order. This is why it is **very difficult to model Transformation Processes**. Some simple Transformation Processes like “Change Product Pricing”, or “Correct a Bug” are well modeled and automated. But **complex Transformation Processes** like “Execute a Solution Project” or “Execute an Architecture Project” are partly modeled on Documentation, but not automated.

Yet some **Transformation Functions** which are part of the Transformation Process are automated: Build Solution map, design Processes, define Data Model, look for Components, manage Software Configurations, execute non regression Tests...

6.6.3 Transformation Data Model

It includes documented definitions and structured Data Model for Transformation Entities like Project, Planning, Architect.

6.7 Shared Transformation Execution



To optimize Resources, the Enterprise shares Transformation Elements.

- An Enterprise may **share a Transformation Unit**: for example:
 - share an **Architecture Unit** which develops Reusable Functions for the different Transformation teams
 - centralize all Transformations in a unique Unit.
- An Enterprise may share **Transformation Master Data** between Solutions: like Information on transformation Actors, Projects, component repository, data model repository...

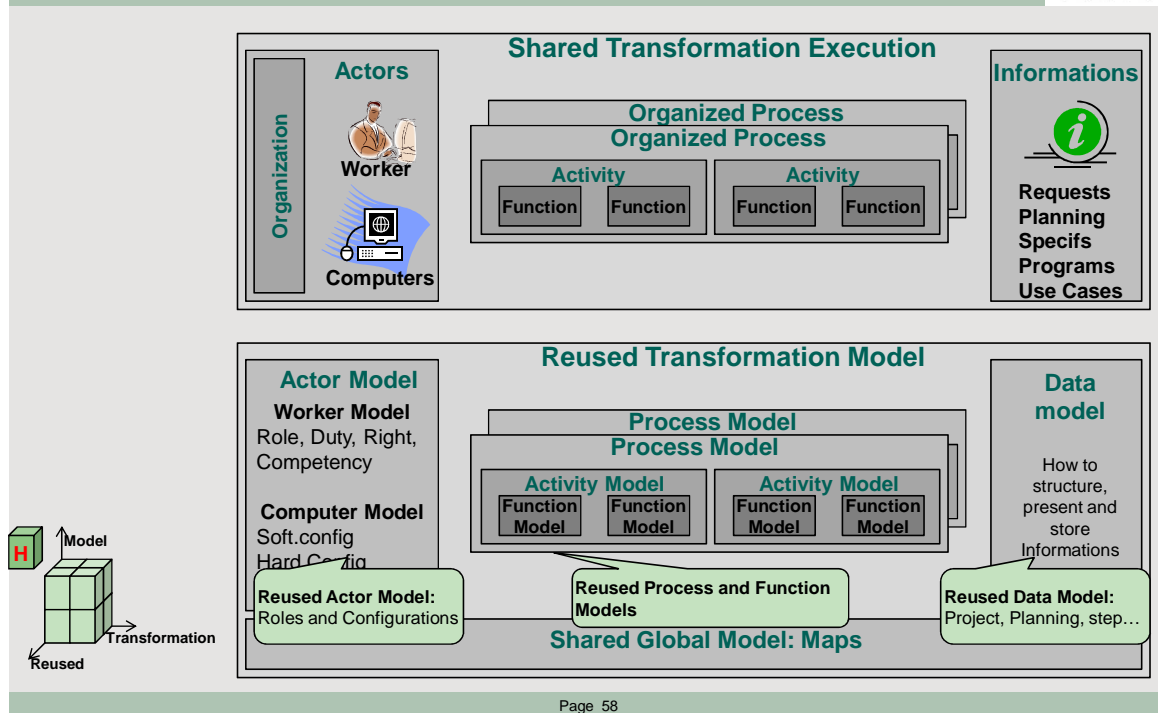
6.8 Reused Transformation Model

Sharing Transformation Resources is much more efficient if Transformation Actors share the same Transformation Model:

- Same global Maps: Transformation Process Map, Software Components Map
- Same Actor Model:
 - same role definitions for Project Manager, Business Analysts, Architects
 - same configurations for transformation computers
- Same Action Model
 - same Transformation Process Models (Methodology)
 - same Transformation Function Models and Tools: design tools, analysis tools, development tools...

- Same Data Model
 - How to represent Specifications, Entity definitions, Reusable components...

H-Reused Transformation Model

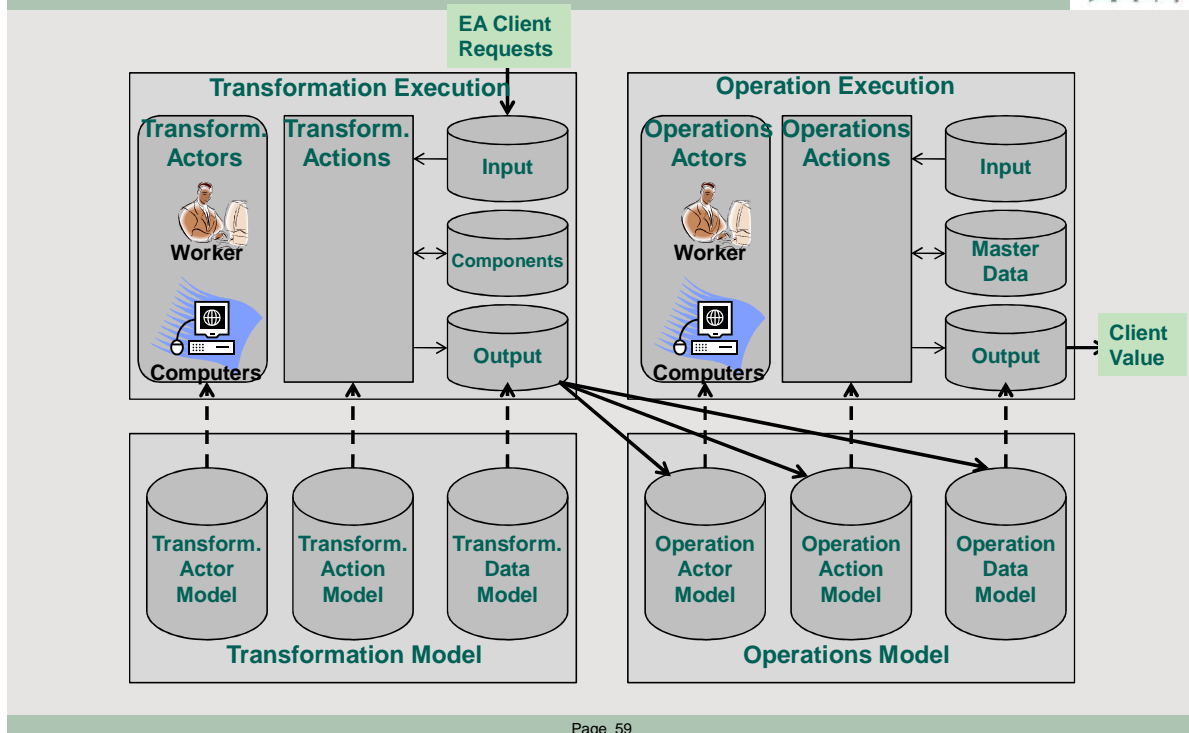


Page 58

6.9 Share and Reuse summary

Transformation (on left) and Operations (on the right) own their Model. Output from Transformation Actions (the Projects) is mainly Operation Models (the Software, data Models and instruction Documents).

Transformation sequence



Page 59

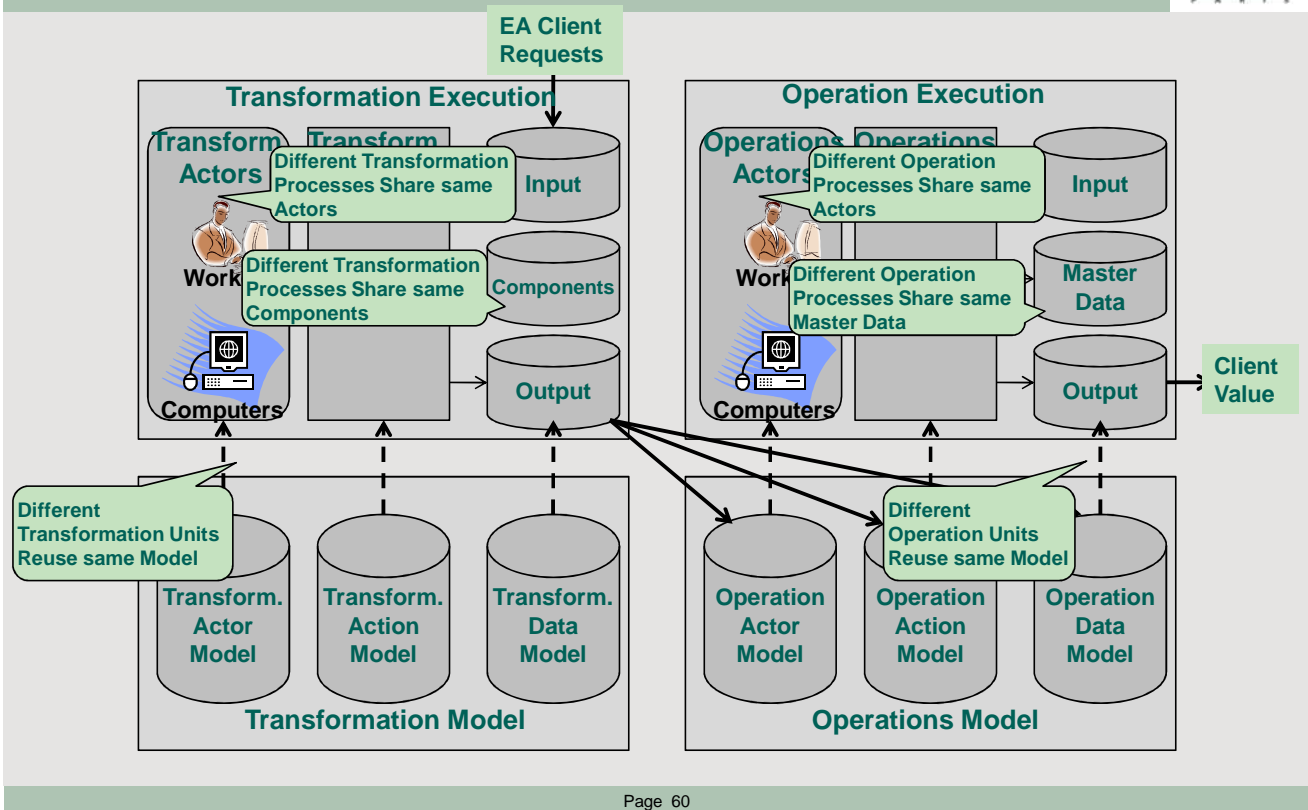
Different Transformation Units may **Reuse** same Transformation Models (ex: same Methodology and Tools), as different Operations Units may **Reuse** same Operations Models (ex: same package software).

Different Transformation Processes **Share** same Components (ex: share Security Function), as different Operations Processes may **Share** same Master Data (ex: Share Customer Information).

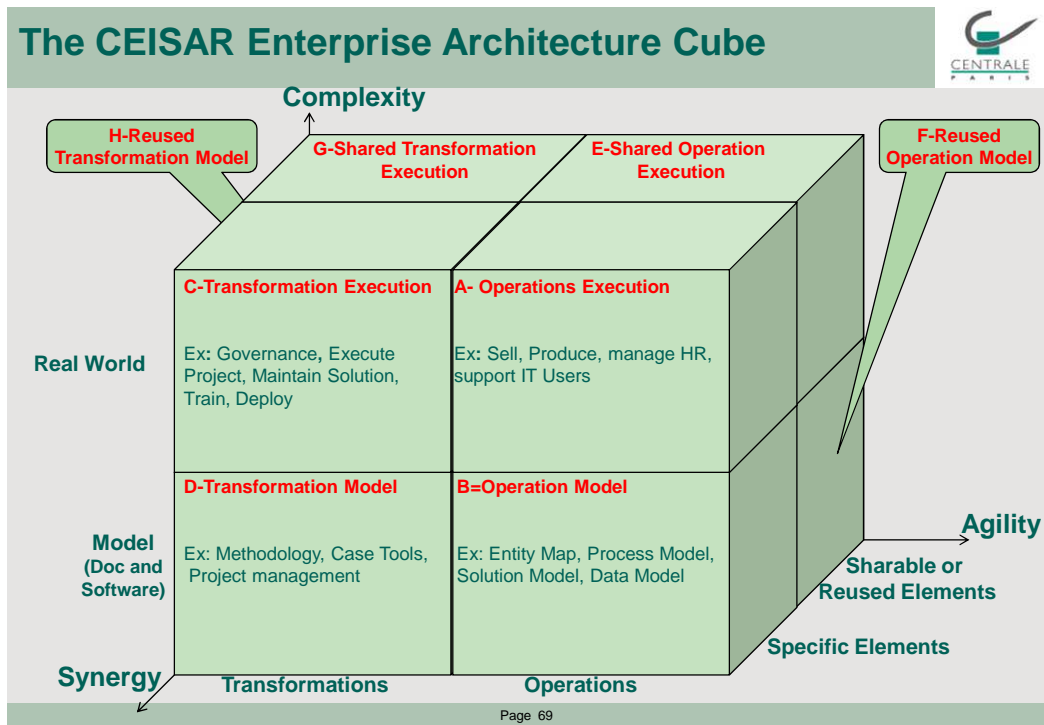
Different Transformation Processes **Share** same Actors (ex: Share an Architecture team), as different Operations Processes may **Share** same Actors (ex: Share a Procurement Unit)



Reuse and Share

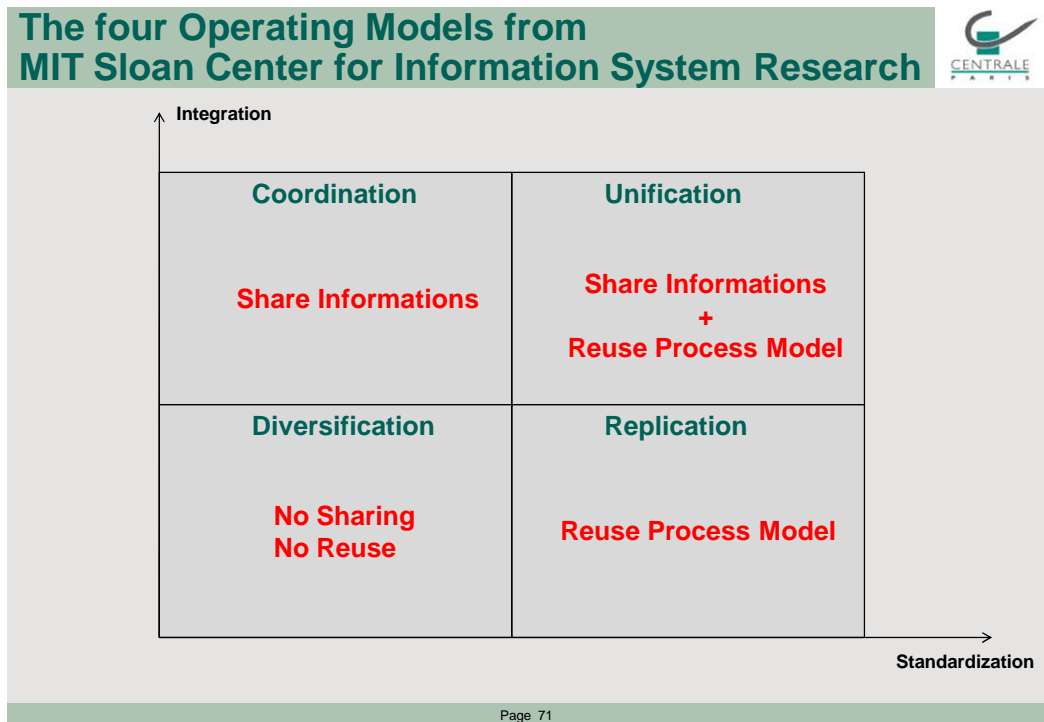


7 How the CEISAR model compares with existing models



7.1 MIT Sloan Center

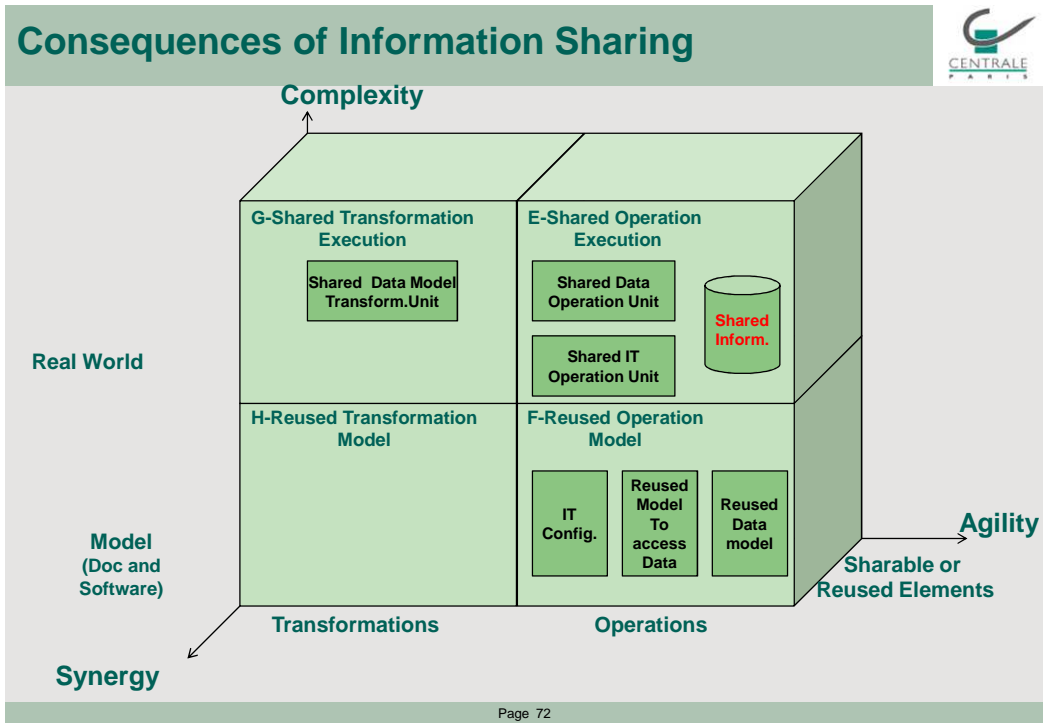
MIT Sloan Center for Information Systems Research has developed a straightforward two-dimensional model with four quadrants, representing different Combinations of the levels of Business Processes Integration and Standardization.



This is a very interesting model and easy to match with the CEISAR approach: in CEISAR language “**Integration**” means “**Share Information**” (on Customers, Products...), and “**Standardization**” means “**Reuse Process Model**”.

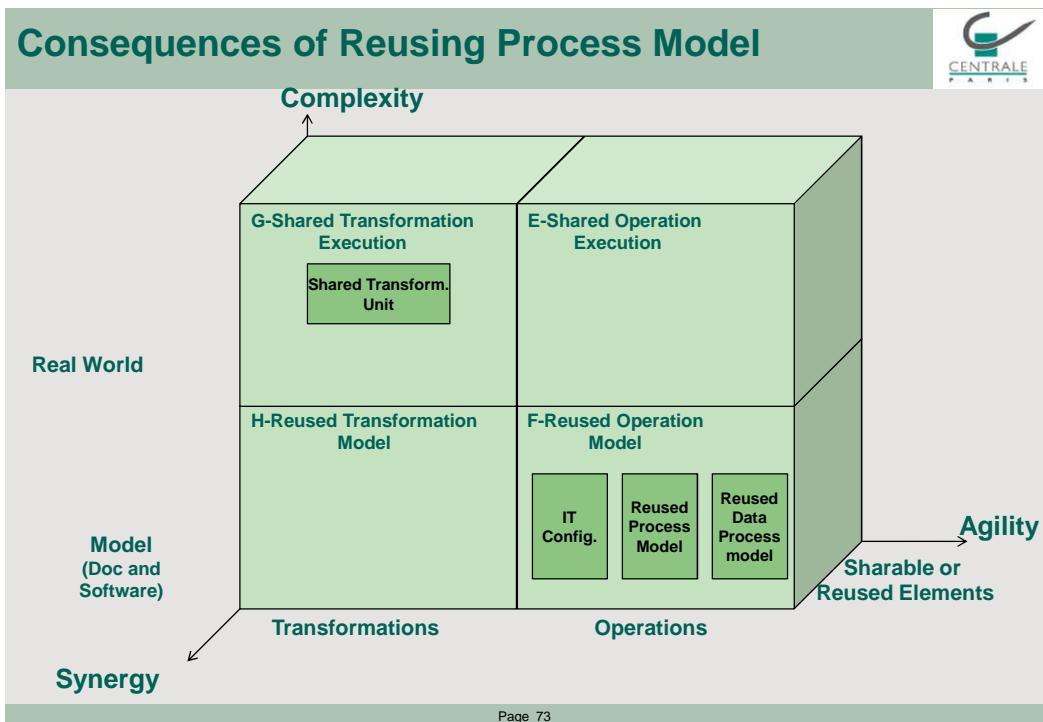
Consequences of Integration:

- A **Shared Data Operation Unit** must guarantee Information consistency
- A **Shared IT Operation Unit** (Servers, Network infrastructure) allows Workers to connect to Shared Information based on the **IT configuration Model**
- The Shared Information reuses the appropriate Model: IT Configuration for data management Server, **Action Model** to access Data, **Data Model**
- A **Shared Transformation unit** Develops and Maintains the Data Model



Consequences of Standardization:

- A Shared Transformation Unit develops and maintains the Reused models.



7.2 TOGAF Architecture (<http://www.opengroup.org/architecture/togaf8-doc/arch/>)

The Open Group Architecture Framework (TOGAF) is a framework - a detailed method and a set of supporting tools - for developing an enterprise architecture.

TOGAF was developed by members of The Open Group, working within the Architecture Forum.

CEISAR converges with TOGAF on many recommendations like:

- Do not separate **business and IT**
- Same **scope**: strategy is not part of EA, Industrial Processes are not inside EA
- To be efficient with EA, we require an Enterprise **Model**
- The Model must be **global** and consistent: it allows different views adapted to each Role
- Good Approaches and Methods do not deliver good results if they are not executed by smart and motivated people

TOGAF is very interesting because successive versions have added new elements every year. But it has become too complex to serve as a basis for training.

This is why we have defined a simpler model which maps with TOGAF.

TOGAF defines 4 Architectures: we **map** these Architectures on the CEISAR Cube (in red):

- A **Business** (or Business Process) **Architecture** - this defines
 - a. The business strategy: **CEISAR does not include Business Strategy inside EA. It is an input to EA. If we include the Business Strategy inside EA, then EA is everything.**
 - b. Governance: **they are Decision Processes. As Governance is mainly about shared Transformation decisions, the Governance Model is in H.**
 - c. Organization: **Organization Unit, Operations Actors and relations between them are defined in A. Their Model is in B (Organization Chart, Roles, Rights and Duties).**
 - d. Business processes: **they are modeled in B.**
- A **Data Architecture** - this describes
 - a. The structure of an organization's logical and physical data assets : **data model is in B.**
 - b. And data management resources: **real Actors who manage Data (persons or computers) are defined in A.**
- An **Applications Architecture** - this kind of architecture provides a blueprint for the individual application systems to be deployed: their interactions, and their relationships to the core business processes of the organization: **these "maps" are defined in B.**
- A **Technology Architecture** - this describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT infrastructure, middleware, networks, communications, processing, standards, etc...
 - a. **Computers, Middleware, Network, are defined in A (in Executing Actors).**
 - b. **IT Operations Process Models are defined in B.**
 - c. **Computer Configurations like Standards are described in B (in Actor Model).**

TOGAF delivers:

1. The **TOGAF Architecture Development Method (ADM)**, which explains how to derive an organization-specific enterprise architecture that addresses business requirements. The ADM provides **(this is the Transformation Model (D) which includes transformation Processes and Tools)**:
 - A reliable, proven way of developing the architecture
 - Architecture views which enable the architect to ensure that a complex set of requirements are adequately addressed
 - Linkages to practical case studies
 - Guidelines on tools for architecture development
2. The **Enterprise Continuum**, which is a "virtual repository" of all the architecture assets - models, patterns, architecture descriptions, etc. - that exist both within the enterprise and in the IT industry at large, which the enterprise considers itself to have available for the development of architectures **(G: Shared Transformation Resources)**. At relevant places throughout the TOGAF ADM, there are reminders to consider which architecture assets from the Enterprise Continuum the architect should use, if any. TOGAF itself provides two reference models for consideration for inclusion in an enterprise's own Enterprise Continuum:

1. The TOGAF Foundation Architecture - an architecture of generic services and functions that provides a foundation on which specific architectures and Architecture Building Blocks (ABBs) can be built (delivered by G to F). This Foundation Architecture in turn includes:
 - The TOGAF Technical Reference Model (TRM), which provides a model and taxonomy of generic platform services (Operation Actor Model in F for Operation standards and Transformation Actor Model in H for Transformation standards)
 - The TOGAF Standards Information Base (SIB), which is a database of open industry standards that can be used to define the particular services and other components of an enterprise-specific architecture (G)
2. The Integrated Information Infrastructure Reference Model (III-RM), which is based on the TOGAF Foundation Architecture, and is specifically aimed at helping the design of architectures that enable and support the vision of Boundary-less Information Flow.
3. The TOGAF Resource Base, which is a set of resources – guidelines (H), templates (G), background information, etc. - to help the architect in the use of the ADM.

7.3 Cobit Architecture

From Wikipedia (<http://en.wikipedia.org/wiki/COBIT>) : the Control Objectives for Information and related Technology (**COBIT**) is a set of **best practices** (framework) for information technology (**IT**) management created by the Information Systems Audit and Control Association (ISACA), and the IT Governance Institute (ITGI) in 1992. COBIT provides managers, auditors, and IT users with a set of generally accepted measures, indicators, processes and best practices to assist them in maximizing the benefits derived through the use of information technology and developing appropriate IT governance and control in a company.

They provide a good list of IT Processes.

Ex: Plan and organize

PO1	Define a Strategic IT Plan and direction (D for Plan Model)
PO2	Define the Information Architecture (D)
PO3	Determine Technological Direction (D)
PO4	Define the IT Processes, Organization and Relationships (D)
PO5	Manage the IT Investment (B for Operation Investments Model and D for Transformation Investments Model)
PO6	Communicate Management Aims and Direction (B or D: different communication to Operations People and Transformation Teams)
PO7	Manage IT Human Resources (B for Operations IT people and D for Transformation IT people)
PO8	Manage Quality (B and D)
PO9	Assess and Manage IT Risks (B and D)
PO10	Manage Projects (B)

For all other Domains (“Acquire and Implement”, “Deliver and Support”, “Monitor and Evaluate”) the same mapping can be done: we just have to split them into Transformation Processes and Operation Processes, and then decide if they are specific to some Solution domains (B and D) or the same for all Solutions (F and H).

8 FAQ

Is Software a Model?

A **Global Model** helps us to understand EA complexity: a Software map can be used.

But Detailed Software resides in **Detailed Model**.

It helps to explain that the same **Reusable Solution Software** can be reused by different Enterprises, each of them executing an **Executable Solution Software**.

What is Architecture?

Depending one's point of view, Architecture may represent:

- Data Architecture: see Data Model and Reusable Data Model
- Computer Configurations, Hardware Architecture, Infrastructure Architecture, Middleware Architecture, network Architecture, Platform Architecture, Portal Architecture, System Architecture, Transactional Architecture, Web Architecture: see Organization Actor Model
- Integration Architecture: see Transformation Model
- Logical Architecture: see Operation Model
- Maps (like Block Cartography): see Operation Models or Transformation Model
- Reusable Component Architecture or SOA Architecture or Service Architecture: see Reused Model
- Organization Architecture: see Organization Actor Model
- Software Architecture: see Operation Model (Maps and Software Services)
- System Architecture (see Operation Architecture)
- Transformation approach and Tools: see Transformation Model

To summarize, Architecture means 2 things:

“Architecture” is “the Art of Construction” or **Transformation Process** in CEISAR Language (**D**)

- Some add that it is the “Art of Good Construction”. When they say that a System is Architected, it means that it is well built
- Some talk about good Architecture or bad Architecture...: we prefer this interpretation

“Architecture” is the **structure** of the built system and its break-down into many pieces

- IT Operations Architecture: the set of Computers, networks, basic operating software like OS, DBMS, Application Server.... CEISAR defines its Computers as Organization Actors (**A**) which own Hardware/software configurations (**B**). The Process of choosing Configurations is a Transformation Process (**D**).
- IT Development Architecture: the set of Computers (**C**), methodologies (**D**), development tools (**D**)
- Software Components, Data Models, Maps which are created in (**C**) and delivered to (**B**). CEISAR uses the word **Architecture** for these shared elements

