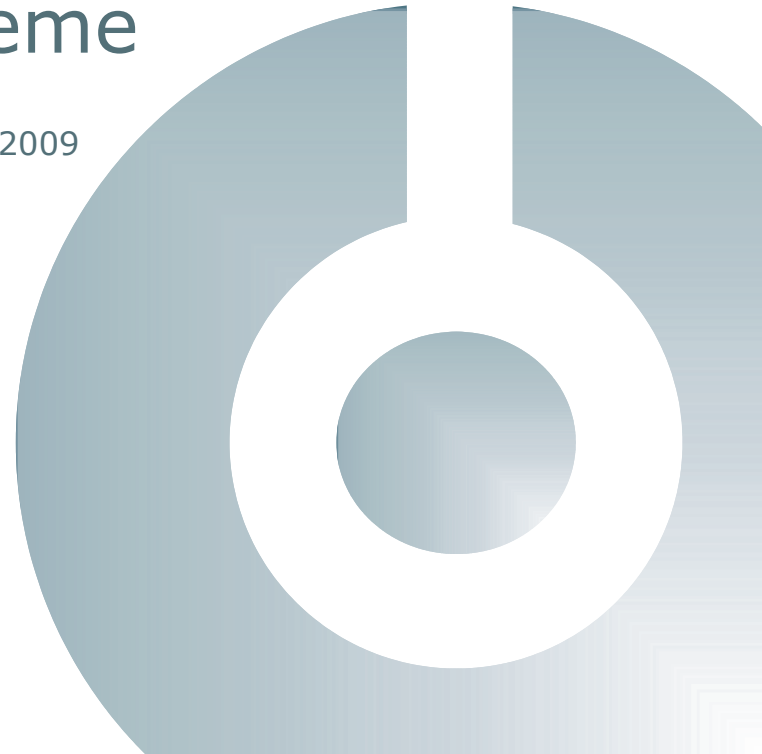


Introducing Praxeme

V1.0 – Mars 2009



Acknowledgement

➤ Partly original work from inno.com (Fabien Villard)

- Including translation of French material
- <http://www.inno.com>



➤ Partly derived

- From S-IT-A material
 - <http://www.sustainableitarchitecture.com/materials>
- From Praxeme Institute material
 - <http://www.praxeme.org>
- From Orchestra Networks material
 - <http://www.orchestranetworks.com/fr/soa>



➤ Praxeme is a free initiative for an open methodology

- <http://www.praxeme.org>
- The Praxeme Institute is the non-for-profit organization supporting the initiative
- Praxeme and Praxeme Institute are registered trademarks

Prelude – A quick dive in the pool



Ecosystem

- Old and obsolete methods
 - Merise, SDM/S, Axial, SA/SD
- Methods that deal with delivery process
 - UP, RUP, SDM/S, XP (eXtreme Programming)
- UML notation (OMG)
 - Reminder : UML is not a method but a notation
- Several best practices in IT / Business governance
 - COBIT, CMMI, ITIL
- Enterprise Architecture Framework (EAF)
 - TOGAF (Open Group) : mainly process oriented
 - Zachman (in most cases too complex and lack of SOA/MDA principles?)
- Other disciplines
 - Business Process Management (BPM), object oriented design, SOA (Service Oriented Architecture)...
 - MDA (Model Driven Architecture) specified by OMG

Today's Facts

- Lack of reference methods
 - Merise is not used anymore
 - UP, RUP, XP, Scrum are delivery processes methods
 - EA insists on processes and forgets models (this is changing)
 - Best practices are not methods
 - ITIL, CMMI, CoBit...
- Recurrent Issues
 - Architecture in silos
 - Redundancy at all level
 - Communication between actors
 - Find a good path to target
 - Knowledge management and documentation
 - Organizational dysfunctions
- Not enough modeling activities
 - Reduce sharing of ideas
 - Increase the silos syndrome
 - Increase the duplication of work
 - Reduce innovations opportunities

Praxeme Foundations

- Syntheses of good principles
 - Decoupling, encapsulation, separation of concerns
 - MDA (Model Driven Architecture)
 - Test orientation
 - UML as a general expression language
 - Open initiative « à la » Open Source
- New approaches added to the package
 - Processes and Use Cases do not come first
 - Function is not the center of the paradigm...
 - ...objects are

Sponsors

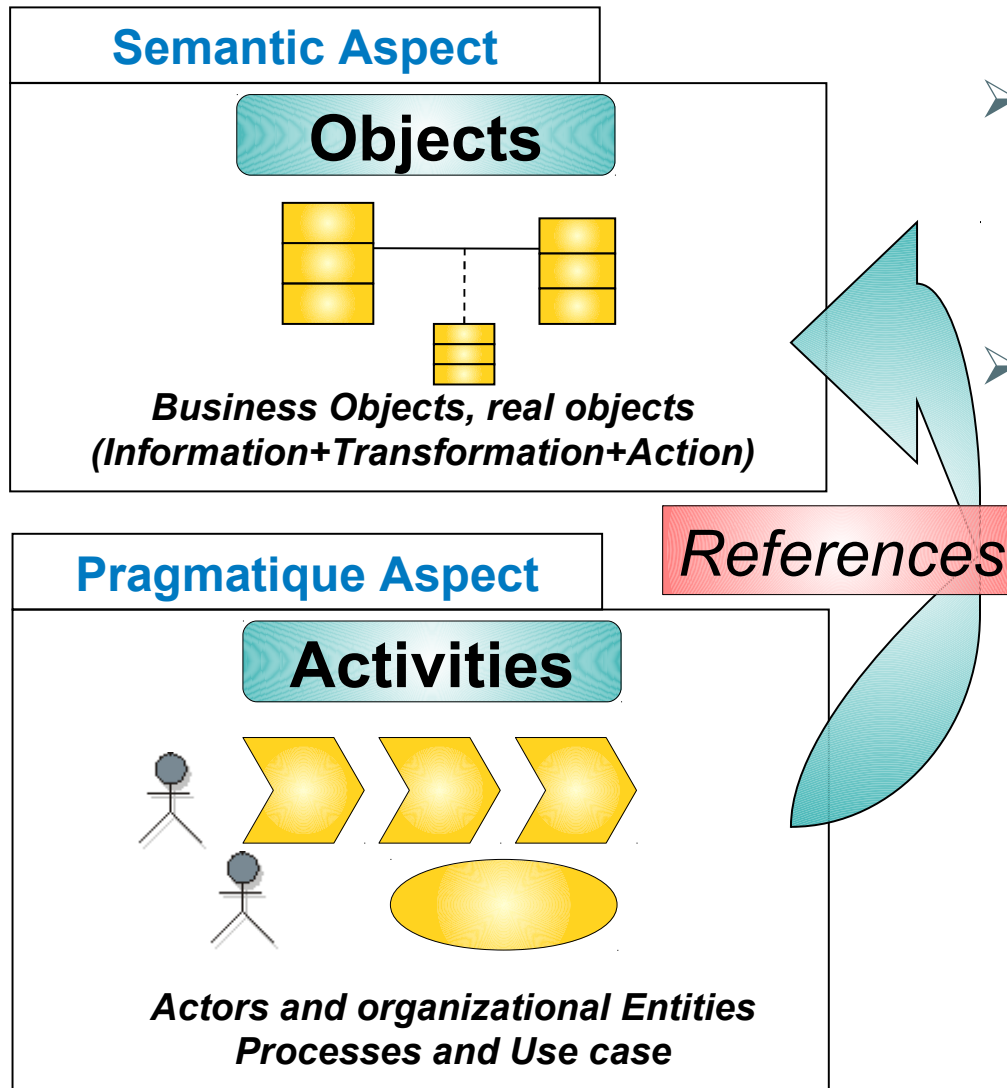
- Companies that contributed to the method
 - SMABTP, SAGEM (Drones conception), CAF (Caisse d'Allocations Familiales), Armée de Terre, Orchestra Networks, Softeam
- Sponsors



A Structural Vision

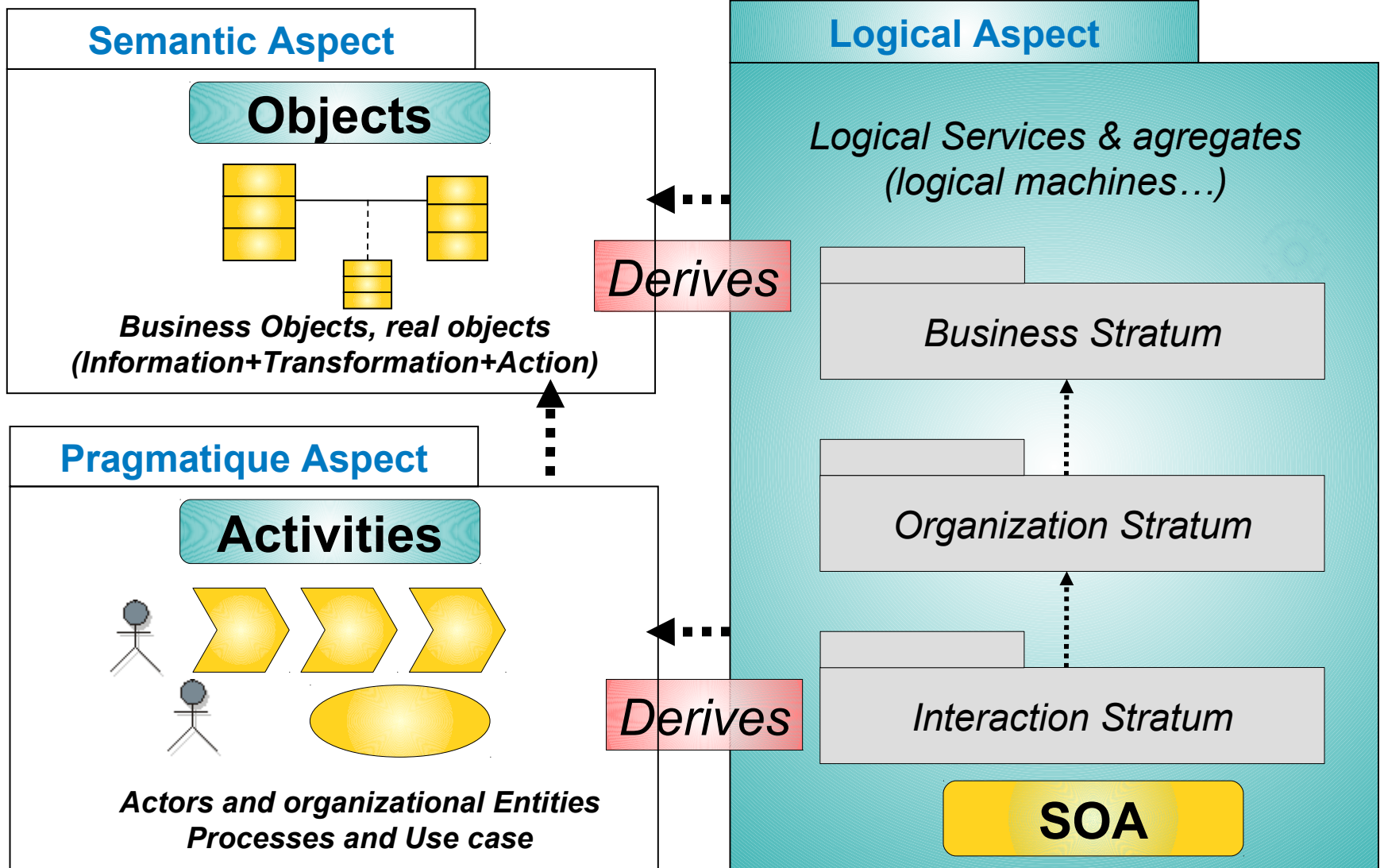


The right description for business



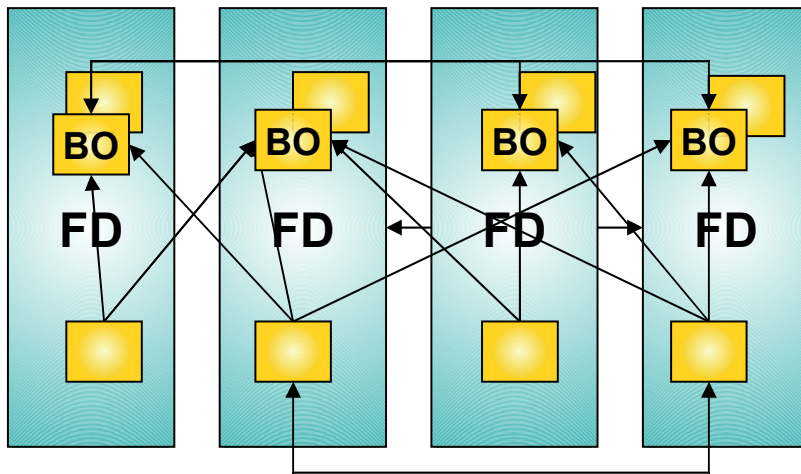
- Activities approach is limited
 - Penalized by local variations
 - Favors redundancy
- Semantic modeling gives
 - Stability
 - Genericity

The right structure for the system



The physiomy change

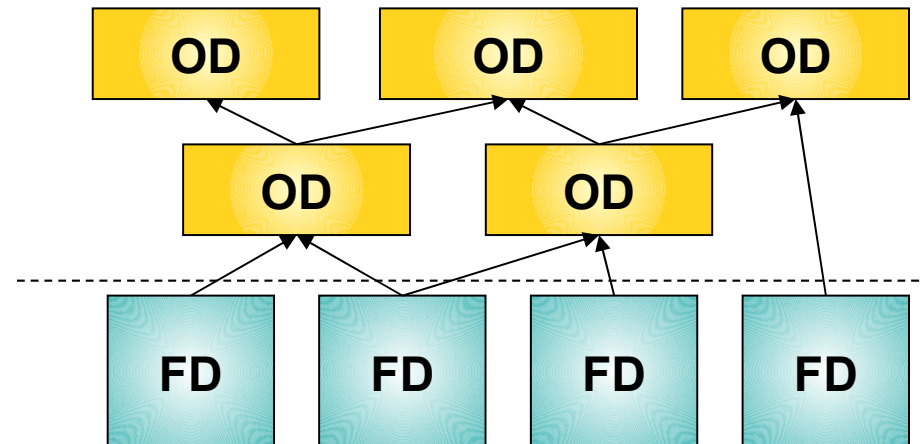
Sample architecture based on functional approach



Logical blocs based on Functional Domains from a pragmatic and historical model
Strong interdependencies and redundancies:

- Business Objects are found multiple times
- All components may be linked to others

Logical architecture diagram following Praxeme



Logical blocs based on Objects Domains structuring the semantic model
Dependencies linked only with topological constraints

- From organization stratum to business stratum
- Mutual relations are forbidden
- No dependencies between OD blocs.

Structuring framework

- A **methodological Framework**
 - Basis for the method
 - Structure for the enterprise perception
 - Of the business domain
 - Of the enterprise itself
- Praxeme answer is the EST
 - **Enterprise System Topology**
 - Structure for the “product”
 - Built with 8 aspects
 - Independent from any point of view
 - The aspects cover all the description needs
 - **PRO³ Scheme**: Product is one of the three dimensions
 - Product: *what*, the target for the building action
 - Processes: *Organization* to achieve the goal
 - Procedures: *How*

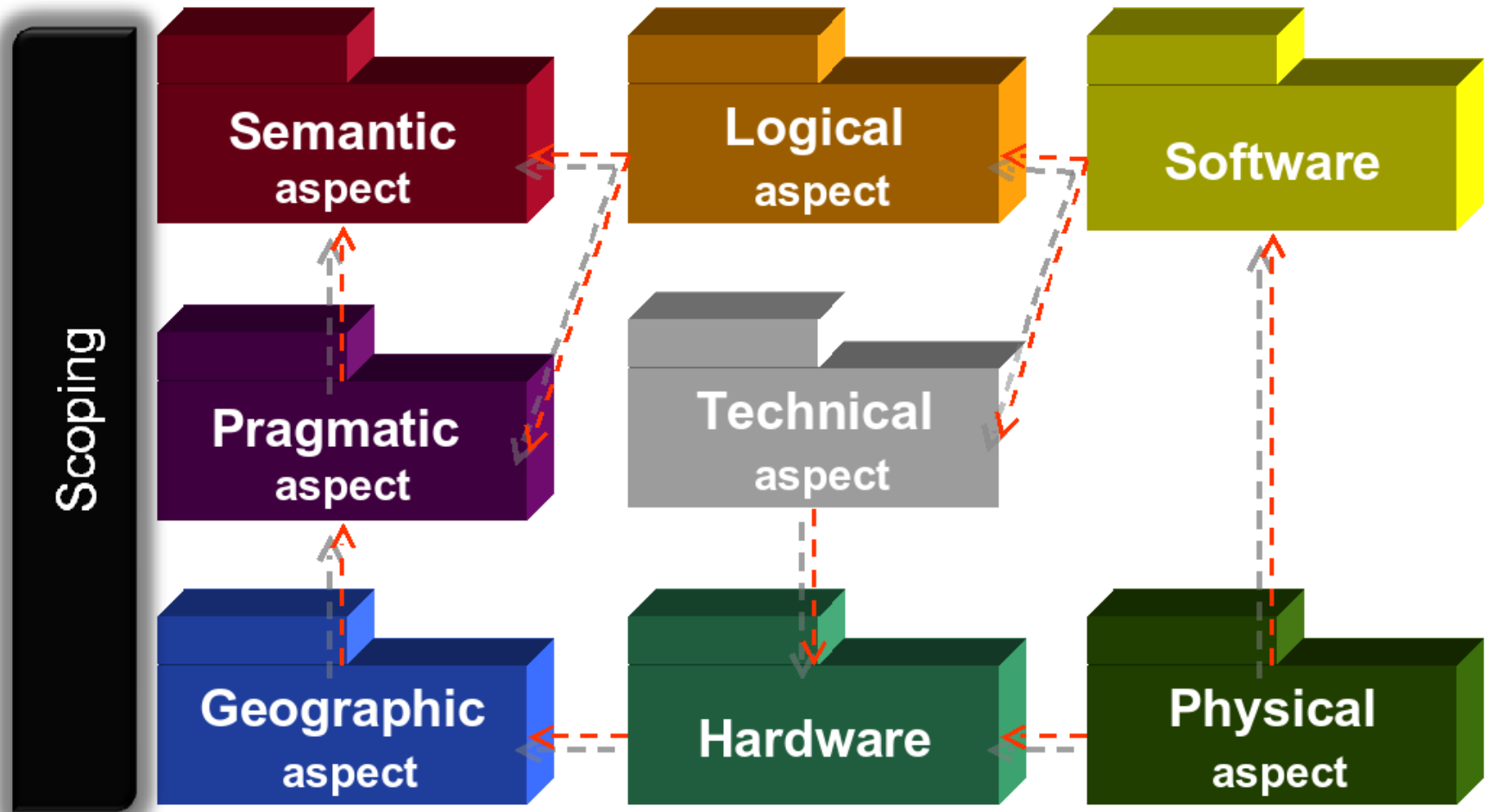
Method Summary

- Business description
 - Not driven by processes only
 - Must isolate the essentials for the business
 - Stable
 - Easy to share across the enterprise
 - Is the real documentation
- Modeling
 - Rigorous **representation of business**
 - Usable in all parts of the exploitation process
 - Red line for all transformation activities
- SOA Added Value
 - A logical **architecture style**
 - Independent from technologies
 - **Alignment is guaranteed** by derivations and traceability

The Enterprise System Topology



Enterprise System Topology

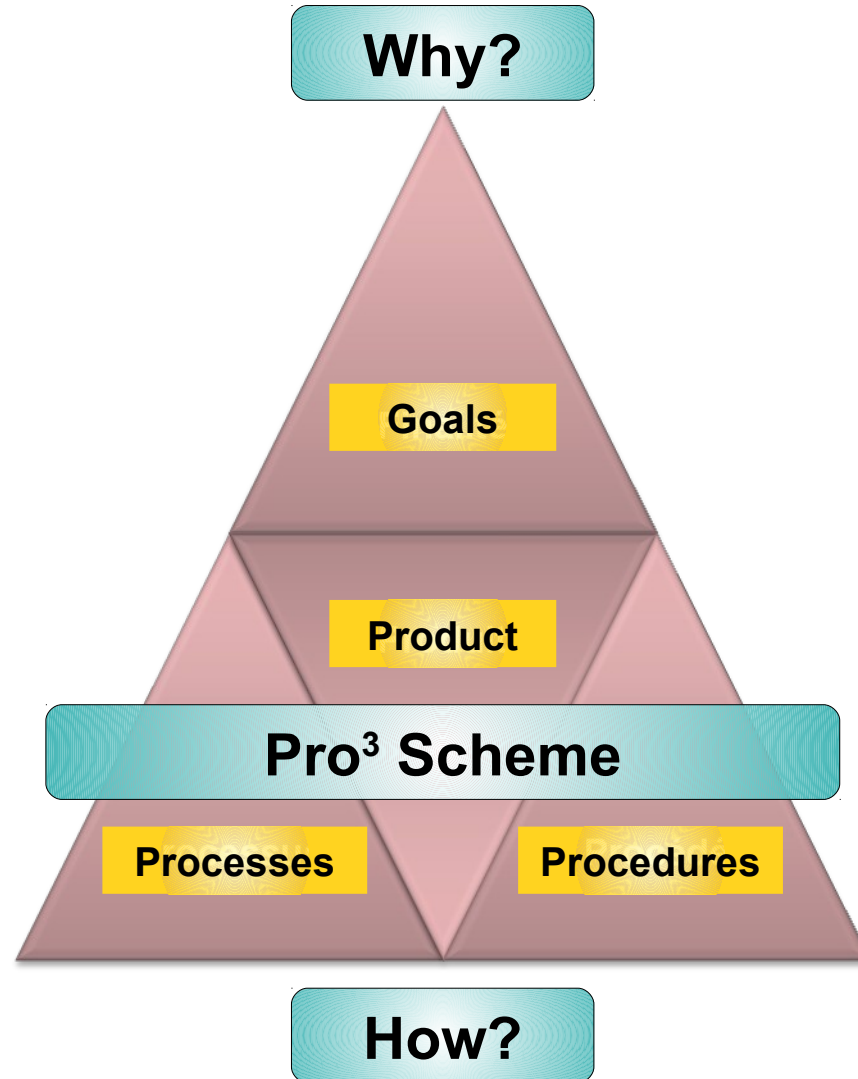


Reading the EST

- Arrows represent dependencies between aspects
 - A package references content of another one
 - OR
 - A package is derived from another
- One notable exception Soft -> Tech
 - Technical Aspect gives
 - Technical choices
 - Recipes to use the technologies
- Arrows are oriented
- Missing links
 - Give more information than existing ones
 - Forbidden dependencies
- The EST provides a traceability map

Warning!

PRO³ Scheme

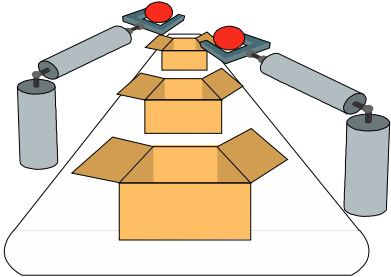
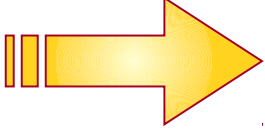
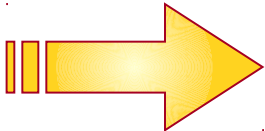
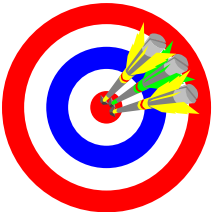


The Dimensions

Why?

What?

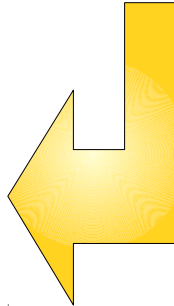
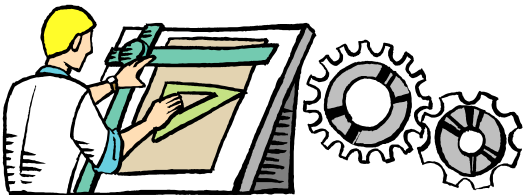
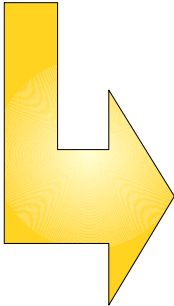
How?



The Purpose

The Product

The Process



The Procedures

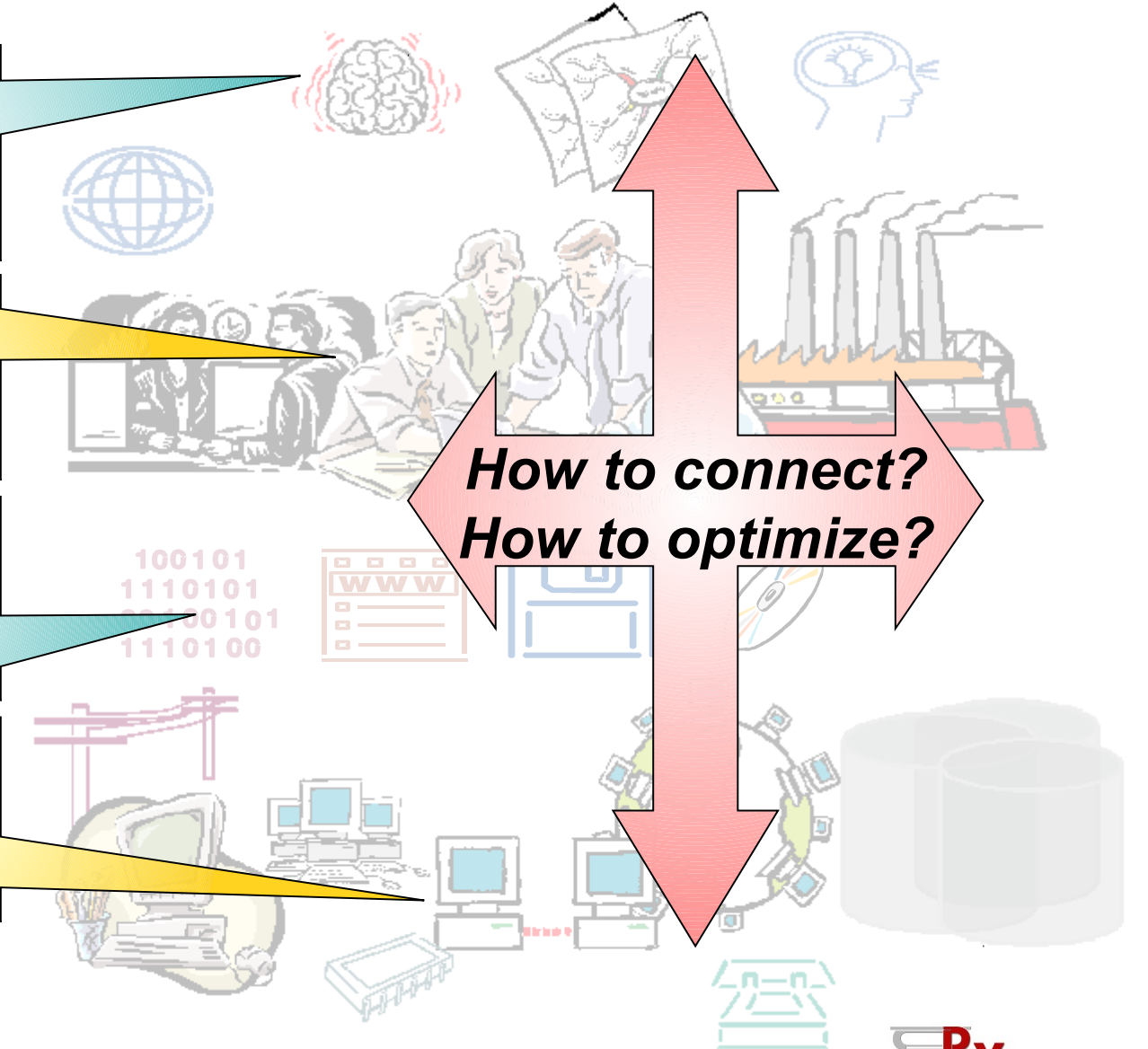
The System Object

The Knowledge:
core business
concepts...

The Activity:
Business
processes...

The Software
Tools

The
Infrastructure



How to connect?
How to optimize?

Before the Aspects, the Scoping



Scoping

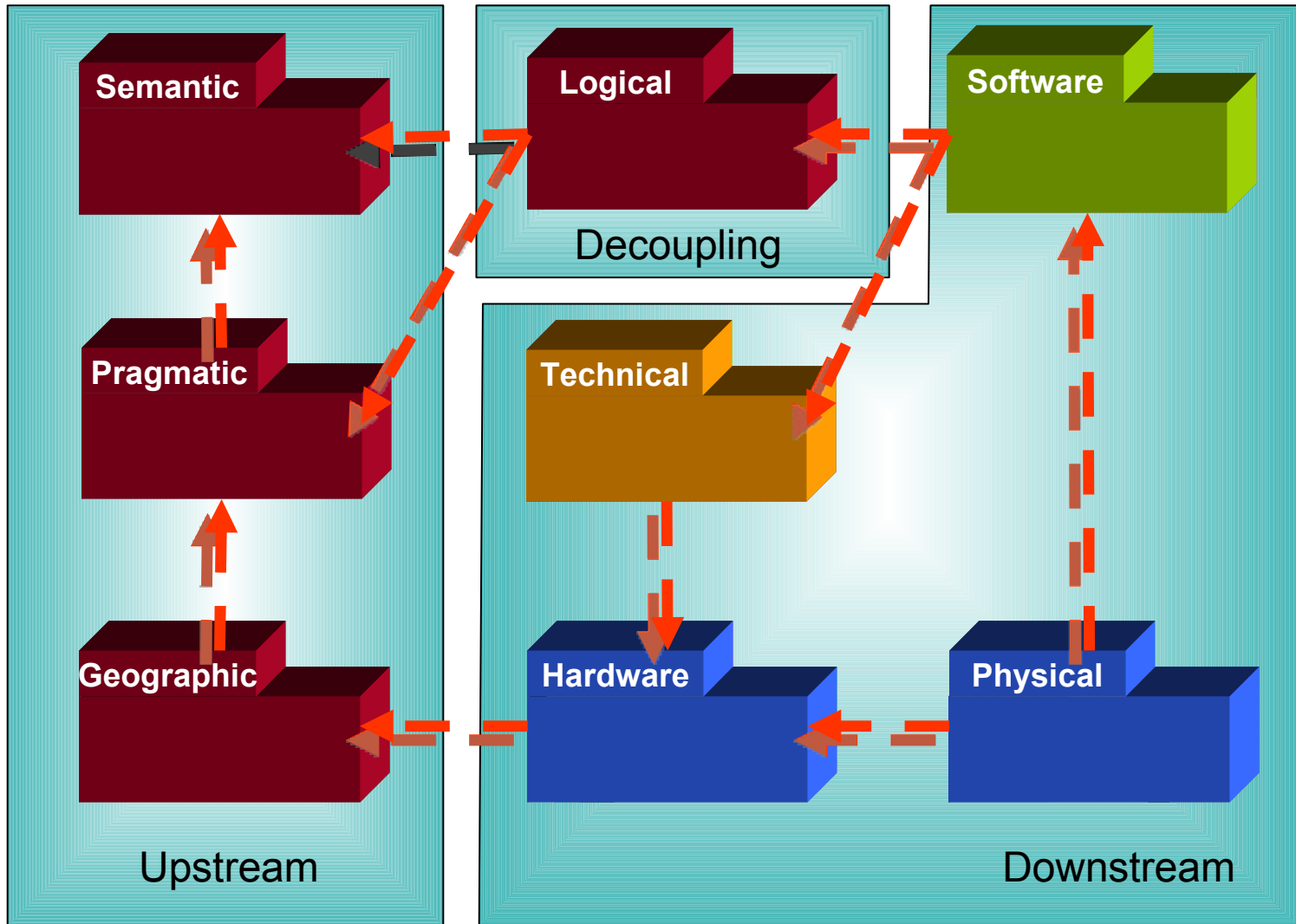
- Bridge between **natural and formal** languages
 - Natural for requirements and goals
 - Formal in models of the modeling chain
- Informal information
 - Requirements, goals
 - Constraints, standards, regulations
 - Strategic considerations
- **All points of view**
 - Same concepts with different names
 - Same names with different meanings
 - Jargons, local conventions
- Informal but **not without tools**
 - Knowledge Management Methods
 - Dictionaries
 - Thesaurus
 - Requirement management tools
- **Traceability** is the main word

The EST Aspects

Enterprise System Topology



Topology Macro Structure



Topology Macro Structure Details

➤ Upstream Aspects

- Represent the **core business** knowledge and the **way to do it**
- Do not include IT considerations
- Keep strict **links with Scoping**
- Is **stable** compared to IT representations

➤ Logical Aspect

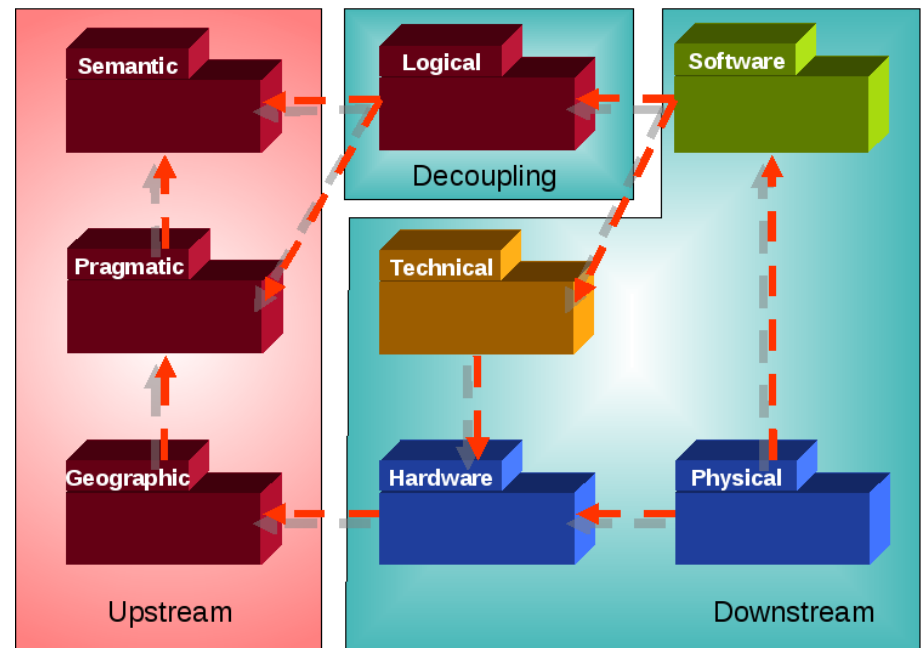
- Represents a **decoupling structure** between Business and IT systems
 - Regardless of technical choices
- Is modeled with an **architecture style**
 - SOA is such a style
 - EDA or SOA+EDA or Functions can be other possible styles
 - New styles will be attached here

➤ Downstream Aspects

- Represent the **IT solutions** to be delivered
- Includes all means to deliver the solutions
- **IT** is located here
- **Technical agility** lives here

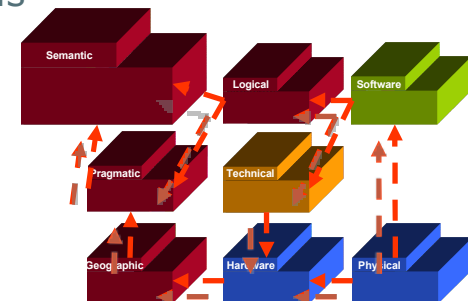
Upstream Aspects

- Semantic Aspect
 - The core business knowledge
- Pragmatic Aspect
 - How an enterprise act in the business domain
- Geographic Aspect
 - Geographic constraints



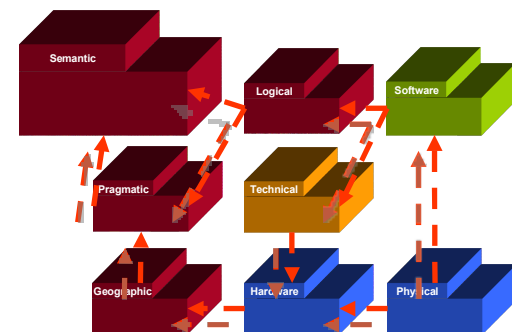
Semantic Aspect

- Captures the being of reality
 - In terms of objects and applying an **Object Oriented Approach**
 - All the being without technical or organizational details
- Does not describe
 - **Actors** and organizational details
 - **Actions** on objects (Processes and use cases)
 - Know-how and recipes on how to deal with business
- Does describe
 - **Real life objects** with their information and relations
 - **Object Statuses** representing the object transformations
 - Pure **Business Rules** (*règles de gestion*)
 - From the business domain reality
 - From external constraints like regulations, standards
 - Enforced rules that can not be avoided



Semantic Aspect Models

- Structural models (Class Diagrams)
 - Rich descriptions
 - No technical limitations
- Functional and contractual models (State Diagrams)
 - Capture of objects life cycle
 - Business rules governing the transformations
 - Forbidden transitions
 - Will be derived into Use Cases and processes
- Semantic Models
 - Are very stable
 - Are closely linked to business fundamentals
 - Are real object for KM (Knowledge Management)
 - Help raise back interest from business people
- Semantic modeling relates to Ontologies
 - Some debates around the notion of context
 - Semantic models could well be ontologies



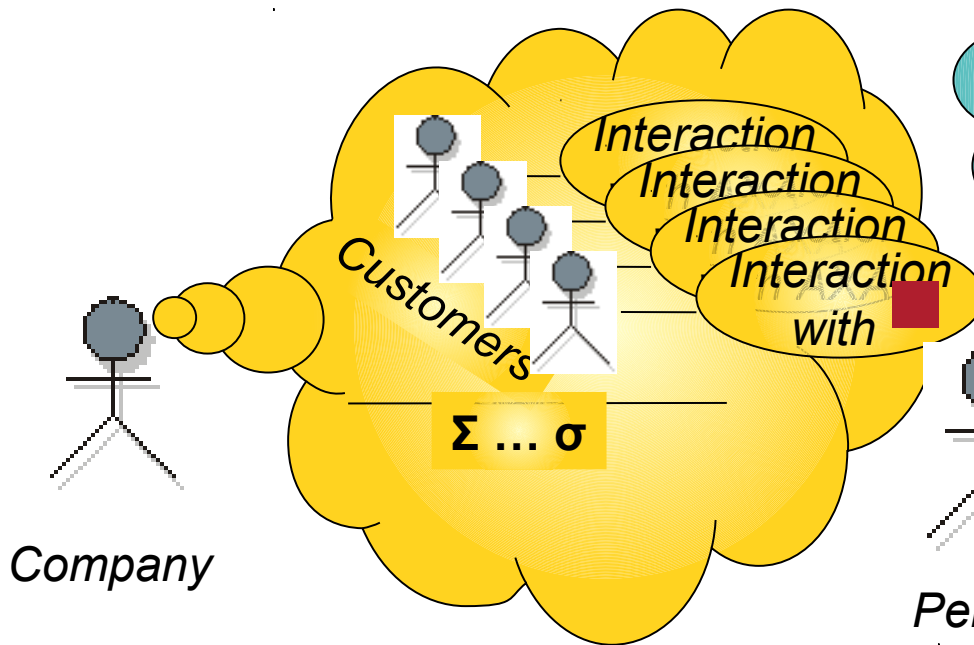
Example: Client Centric Approach

- What **means** « Client Centric » ?
 - Obviousness is questionable
 - **Skepticism** may be a good posture
-
- Semantic modeling helps rethink the business
 - Demonstration

What means Client Centric?

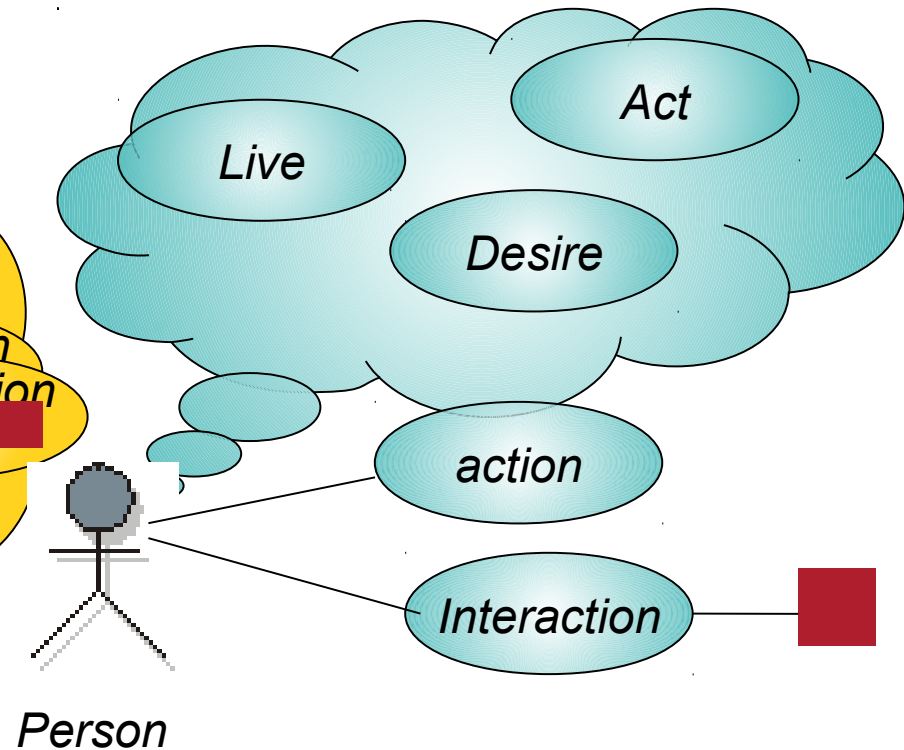
➤ 1st interpretation

- Extract knowledge from the Customer Database



➤ 2nd interprétation

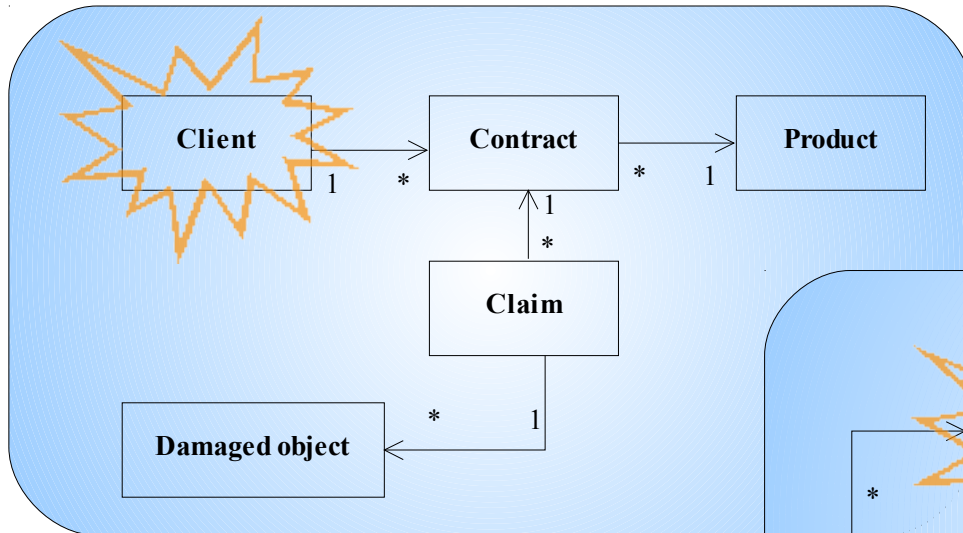
- Get into the client's shoes



Consequences on the model

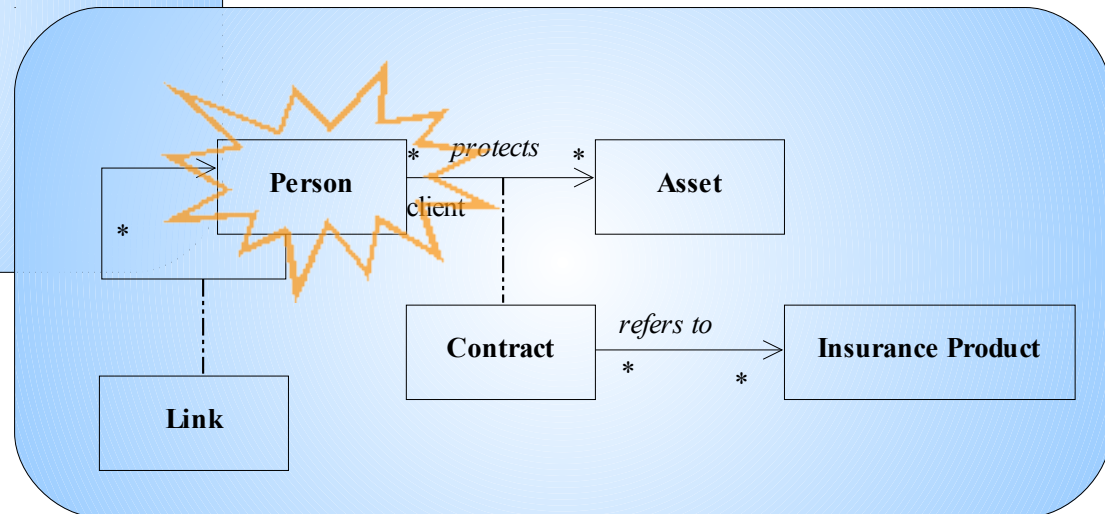
➤ 1st interpretation

- Classical modeling (data)

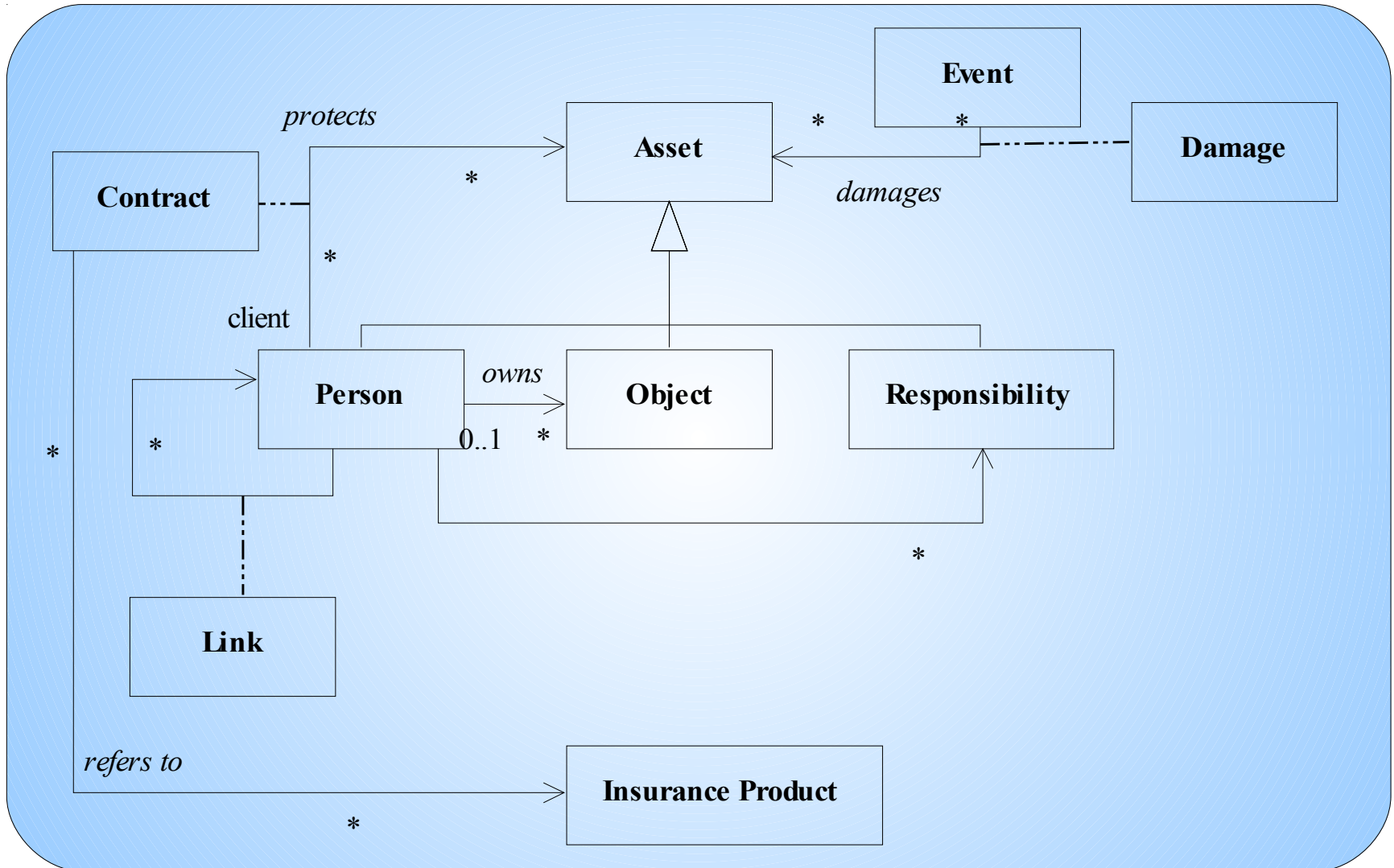


➤ 2nd interpretation

- Semantic Modeling

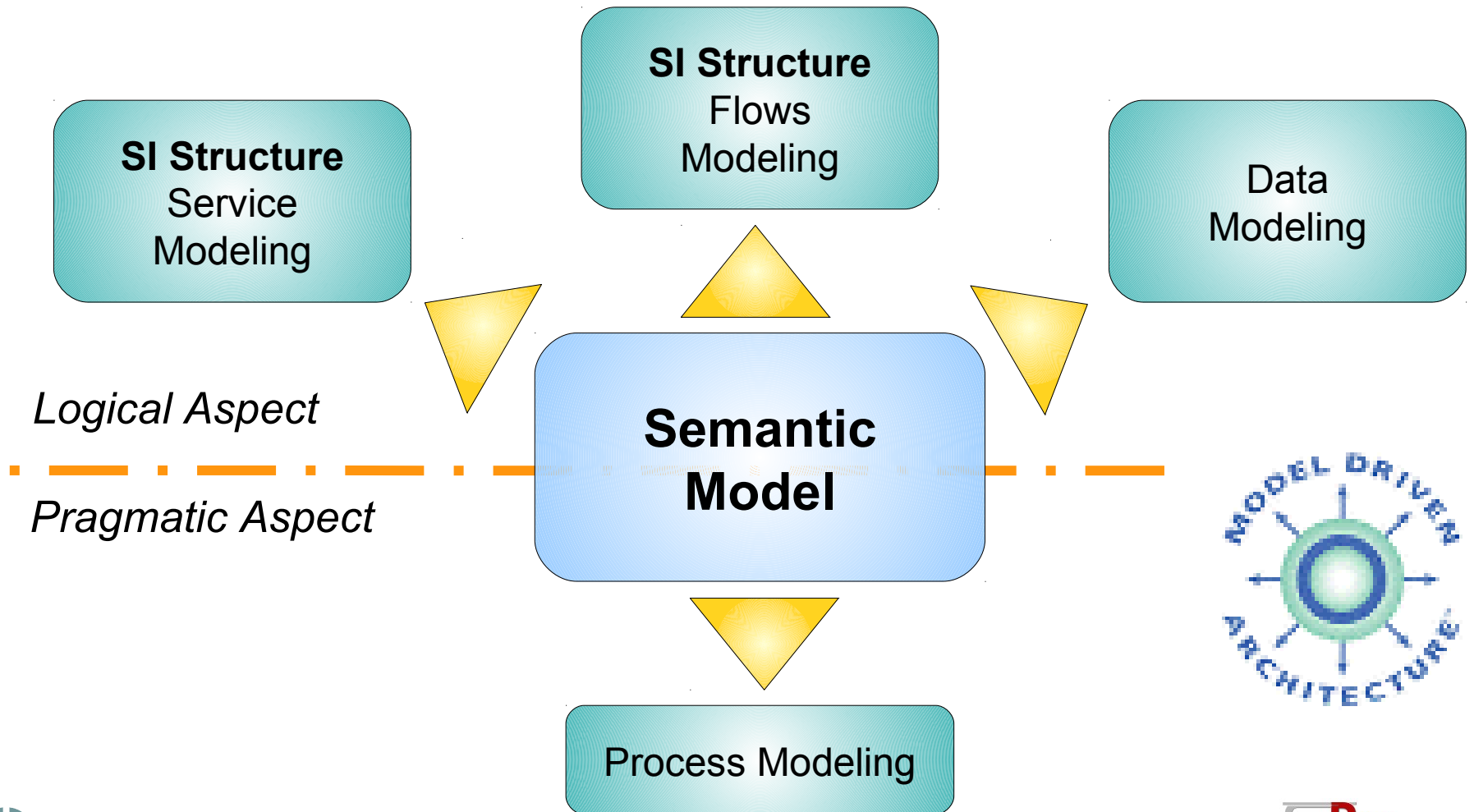


A matter of structure and... Agility



Derivation Paths

- The 4 derivation paths originating from the semantic model

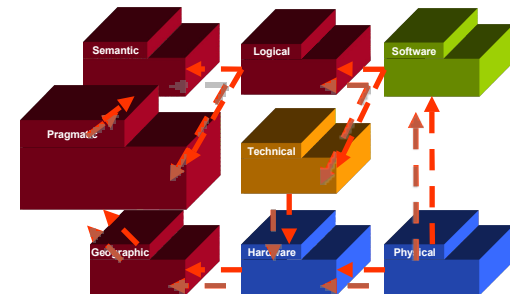


Semantic Key Points

- Semantic Models are **not Data Models**
 - But they will be used to produce them
- They want to express **everything about the reality** of the business
- Big attention is given to differences between concepts
 - Hence an apparent complexity in the structure
- They may be different from the **naive perception** of the real
- They need an **abstraction effort** and a great rigor
- They use the **object logic** and get the mere benefits from that

Pragmatic Aspect

- Captures the **actions**
 - Processes and Use Cases
 - A slightly **different definition of Use Case** compared to RUP
 - Enterprise organization constraints and choices
 - Organization rules (« *règles d'organisation* »)
 - Actors, roles and persons
- Models use UML diagrams
 - **Activity** Diagrams / **State** Diagrams
 - **Use Case** Diagrams with inheritance and such



Process Approach by Functions

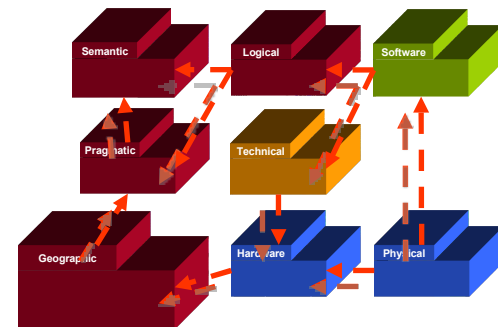
- Difficult to avoid the function approach
 - In nearly all contexts we will **continue to describe** the business **with only activities**
 - Existing descriptions use this paradigm
- Recommendations
 - Start from the results obtained by the function approach
 - Discover **objects inside activities**
 - Take extreme care to built a **unique repository** for these objects

Process Approach by Objects

- Identify **business objects**: the heart of the processes
 - Object approach only
 - It's a major class of the semantic model
 - **Beware of artificial objects**
- Study its life cycle
 - **State** diagram
 - Including non **nominal events**
 - Including **cooperations**
- **Deduce** activities
- **Distribute** activities
 - Actors appear here only

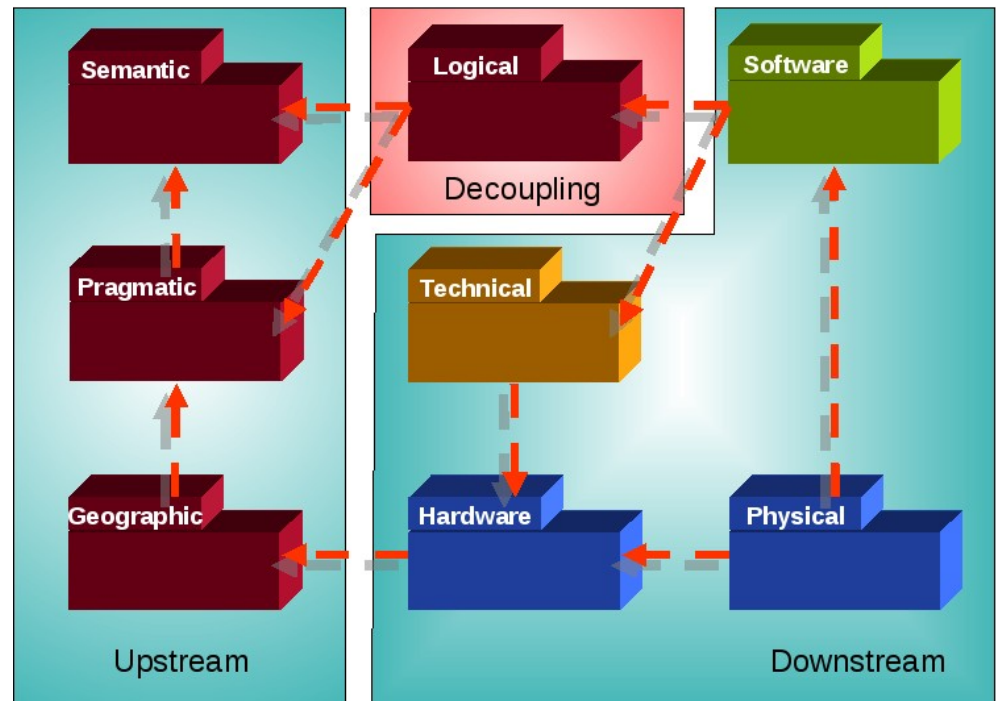
Geographic Aspect

- Includes a lot of non modifiable information
 - More **constraints** than innovation opportunities
 - But must be modeled nevertheless
 - Documentation and references
- Locations
 - Headquarters
 - Specialized locations
 - Machine rooms...
- Role and **duty repartition**
- Links between locations
 - Transportations
 - Digital links
 - Phones, faxes...
- Models
 - **Packages** Diagrams
 - **Non UML representations** like maps



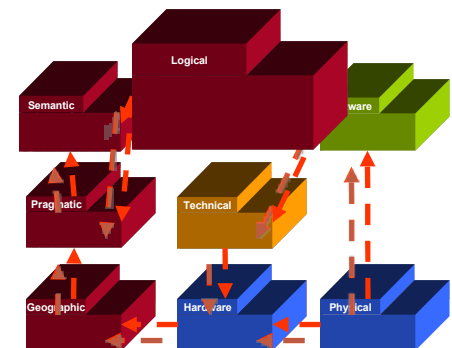
Decoupling Aspect

➤ Logical Aspect



Logical Aspect

- Last **non technical representation**
- **Derived** from Semantic and Pragmatic models
 - Strict derivation rules
- Logical-Technical **Negotiation**
 - To avoid unrealistic architectures
 - Must be **reduced to the minimum**
 - Praxeme proposes a **canonical** list of themes



Logical Architecture

- Objectives for the architecture structuration
 - Cost cuts
 - **Avoid redundancy** and difficulties
 - Evolutivity
 - Flexible and **agile** for changes
 - System control
 - Traceability, readability
 - **Risk reduction**
- Definition

Logical Architecture is a designing discipline. It targets a structured IT system independent from technical choices

Logical Aspect and SOA Style

- With **SOA style** the logical aspect is composed of **2 plans**
 - **Service Plan** covers and hide the **Data Plan**
- Which leads to
 - A path for services
 - A path for data
- **Service Plan**
 - **Stateless** architecture to guarantee the isolation of flows
 - **Pivot** language
 - Other derivation paths
- Design by **Contracts**
 - Pre-conditions
 - Post-conditions
 - Body of execution

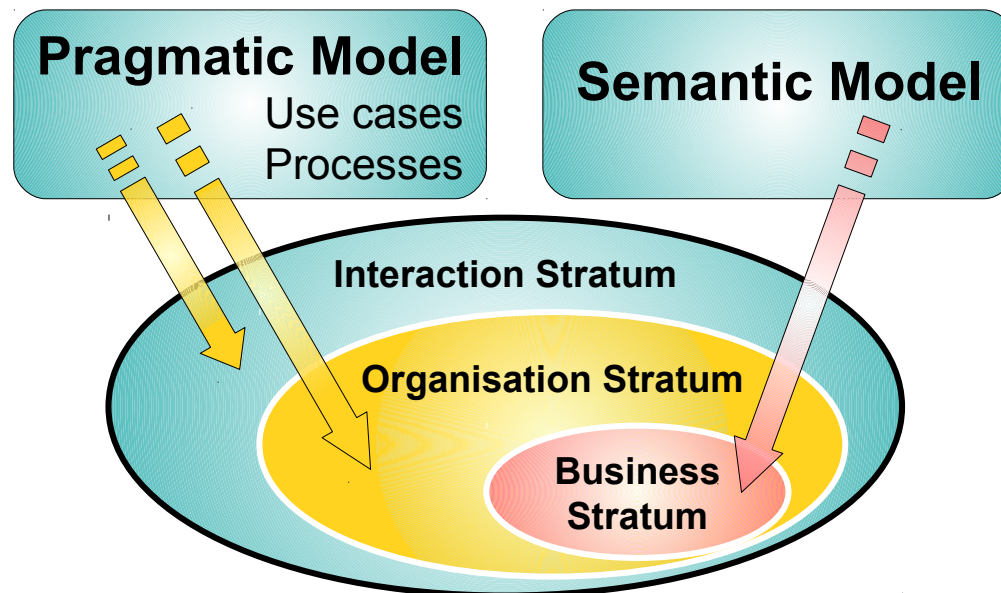
Logical Aspect Stratification

➤ Stratum

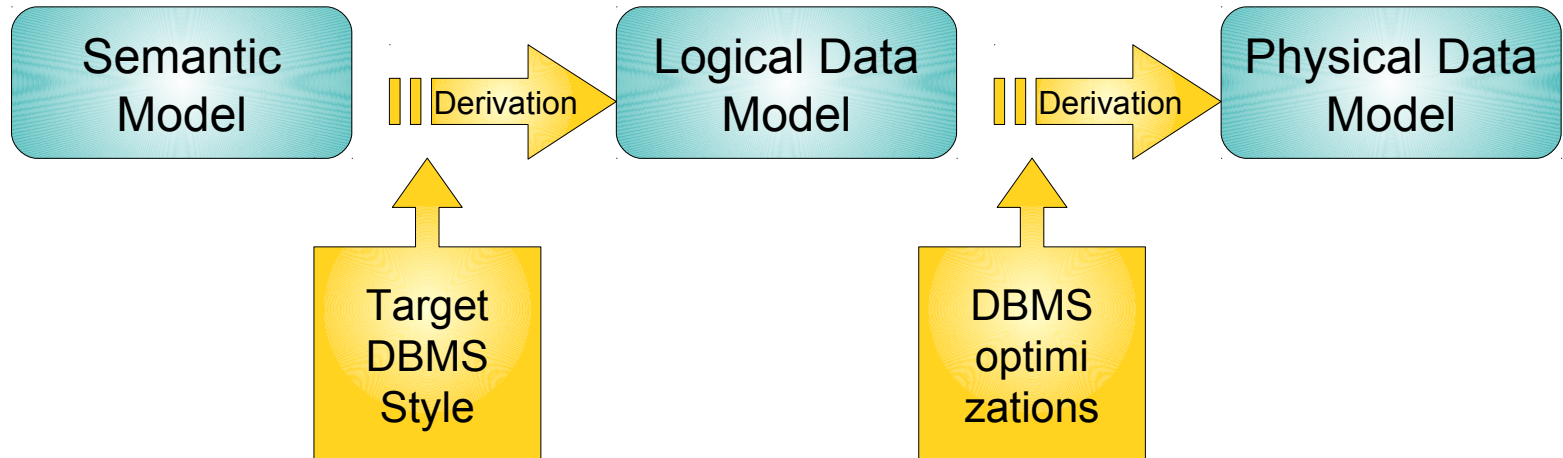
- “Core” stratum, also called “Business” or “Kernel” Stratum
- “Organization” Stratum, also called “Intermediate” or “Activities”
- “Peripheral” Stratum, also called “Presentation” or “Interaction”

➤ Polarization

- Calls go from external to internal
- From one stratum to an adjacent one without jumps



The Data Model



- Derivations apply **precise rules**
- Target DBMS include Relational, Hashtables, flat files...
- Optimizations are brought by **target system experts**
- To go further [fr]
 - Cf. « *La dérivation du modèle sémantique en MLD* »

Downstream Aspects

➤ Software Aspect

- Binaries
- configurations
- versions...

➤ Hardware Aspect

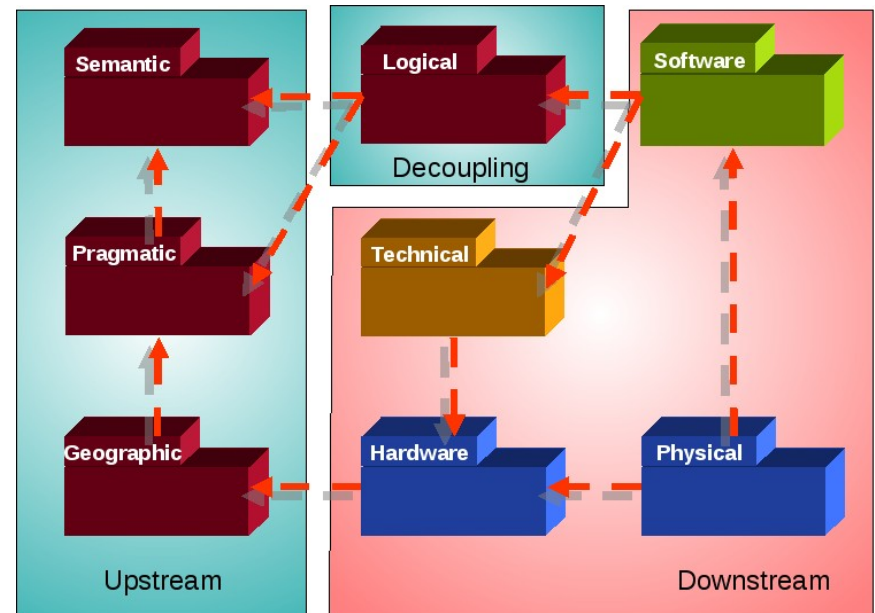
- Machines, networks...

➤ Technical Aspect

- Technological choices and howtos

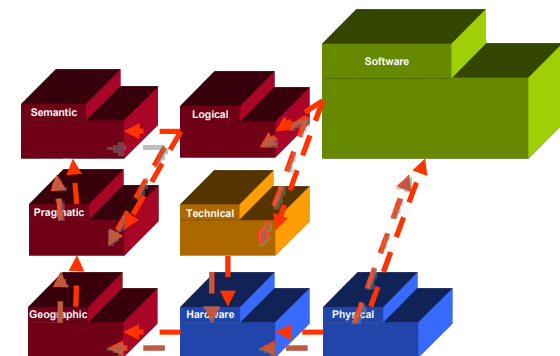
➤ Physical Aspect

- Mapping of software on hardware (deployment)



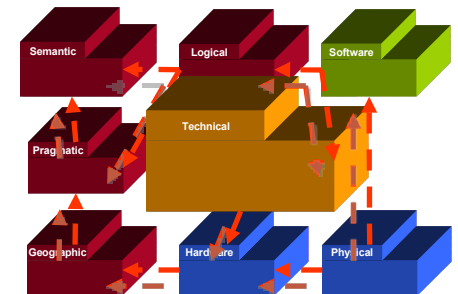
Software Aspect

- Models are extended with **technical UML profiles**
 - These models are **PSM (Platform Specific Models)** in **MDA (Model Driven Architecture)** (Model Driven Architecture)
 - Binding with technical targets: EJB, J2EE, .Net, Web Services, XSD/XML...
- Structural code may be generated
 - From Logical models
 - Technical Aspect describes how to generate



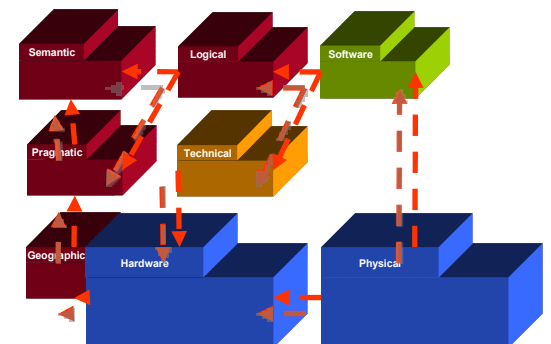
Technical Aspect

- Selecting IT tools and modeling architecture
 - Translate **logical terms** into software
 - Technology **choices**
 - IT guidelines and Howtos
 - Models of technical components, framework
 - The VEP (**Virtual Engine for Praxeme**)
 - A Specification including factories, objects and APIs
 - Needs reference implementations
- Logical – Technical **Negotiation**
 - See Logical Aspect



Hardware and Physical Aspects

- **Hardware** Aspect
 - Models Machines, Networks, Clusters, Grids
- **Physical** Aspect
 - Deployment concerns
 - Mapping of Software elements on Hardware
- Linked with **CMDB** (Configuration Management DataBase)



Praxeme & the Praxeme Institute

- Initial Designer: Dominique Vauquier
- **Praxeme is an open initiative**
 - Documents published under a **Creative Commons**
 - Free: copy-distribute and Derivative Work
 - Conditions: Attribution and Share Alike
 - Some names are trademarks
 - Praxeme, Praxeme Institute
- **Praxeme Institute**
 - **Non profit** French organization
 - “Loi 1901” association
 - Born in 2006 (September)
 - Responsible for the documentation



Etymology

- Praxis (Greek)
 - Action, activities that change the surrounding context
- Séméion (Greek)
 - Sense, meaning, signification
- Hence the subtitles
 - “*Le sens de l'action*” [fr]
 - Meaning in Action [en]

To Go Further



Links and Pointers

- Praxeme Web Site: <http://www.praxeme.org>
- Blogs
 - Dominique Vauquier: <http://dvau.praxeme.org>
 - Community (Starting) (Fabien Villard): <http://friends.praxeme.org>
 - Translators (Starting): <http://translators.praxeme.org>
- Friend Communities
 - Sustainable IT Architecture (S-IT-A):
<http://www.sustainableitarchitecture.com>
 - MDM Alliance Group (MAG): <http://www.mdmalliancegroup.com/>
- Books
 - “Le système d'information durable : la refonte progressive du SI avec SOA” [fr]
 - Sustainable IT Architecture: the progressive way of overhauling information system with SOA

