General guide

Subject	**General methodology**
Purpose of the guide	Introduce and provide a full panorama of the method
Key words	transformation, enterprise, Praxeme, method, methodology, complexity, strategy, modeling, representation, architecture

Reference	PxMDS-01	*Status*	Proofreading in progress, PxI
Version	0.3.0	*Date*	23 February 2014
Authors, contributors	Dominique VAUQUIER	*Translator*	Joanne TOWARD
Proofreaders	Fabien VILLARD, Thierry BIARD		

Summary

Methodological reminders

The Praxeme methodological corpus is composed of:

- Guides, which provide the fundamentals of a subject or a domain;
- Procedures and methods, defined as "ways of doing something, operational modes to execute a task"[1];
- Processes, which describe the sequence of activities and the organizational measures that accompany them.

Outside of the method, the Praxeme open corpus contains models and pedagogic material.

To follow the developments of the open method

- Mailing list
- LinkedIn Group
- Twitter
- the wiki

To participate in the work of the Praxeme Institute

- Become a member of the Praxeme Institute

http://wiki.praxeme.org/index.php?n=Chorus.Join

Document protection

The initiative for an open method rests on voluntary work and the pooling of investments between contributors. It aims to develop and disseminate an open, royalty-free method. Its dynamics only works if this spirit is maintained in the way the documents, which have been made available to the public, are used. This is why the documents are protected with a "creative commons"[2] license, which authorizes the use or reuse of all or part of a document from the Praxeme corpus, the only condition being that the source is quoted. The same conditions should also apply to any documents likely to be derived from Praxeme content. They must refer to the "creative commons" and feature the appropriate symbols:

Updates to this document

To obtain the latest version of this document, please go to the Praxeme Institute website.

Document history

Index	Date	Author	Content
0.0.0	01/01/2014	DVAU	First draft
0.1.0	12/01/2014		Integrated comments from Fabien VILLARD. Chapters 6 and 7 not included.
0.2.0	14/01/2014		Proofread by Thierry BIARD
0.2.2 to 0.3	3/02/2014	JT, DVAU	Translation by Joanne TOWARD and review
1.0.0	March 2014		Addition of chapters 6 and 7
0.3.0	23/02/2014		Current version of the document

[1] Cf. Thesaurus section on the Praxeme Institute website: http://wiki.praxeme.org/index.php?n=Thesaurus.Procedure.

[2] See the philosophy and license detail at: http://creativecommons.org/.

1. The enterprise methodology

This chapter covers:

- the context in which the Praxeme enterprise methodology takes its place and the need that it answers,
- the definition of the enterprise methodology, its objective and its scope,
- its philosophy, its main precepts,
- Praxeme's positioning with regards to other available methods and industry standards,
- the nature of Praxeme: method or methodology?
- the form that Praxeme comes in,
- the initiative for an open method: the development dynamics of Praxeme.

1.1 The need

Enterprises know that they have to constantly adapt themselves. Plunged into the global economy and faced with changes in society, they have to continuously review how they work and behave. They have to exploit innovation, which will give them a competitive advantage, albeit only temporary. More difficult still, they have a duty to reinvent themselves in order to transform themselves, in other words to modify their very being.

Figure PxMDS-01_1. The contextual factors fueling the need for transformation

A combination of factors – geopolitical, economic, technological… – force the enterprise to transform itself.

This need for permanent transformation adds to the complexity of the enterprise and calls for specific approaches. Designing the production system according to technical rationality alone is not enough. We cannot let marketing or financial logic play out while neglecting the other dimensions of the enterprise. On the contrary, it is a case of holding onto the multiple rationalities that run through the enterprise and bringing them together in a coherent and complete vision.

The enterprise must be seen as a complex object in perpetual transformation in order to adapt itself to a changing environment. We can no longer view it only through a succession of stable states that it goes through; it has become movement itself, a permanent transformation, which requires us to be continually on our guard, paying attention to weak signals, distrusting anything that may lull our vigilance into a false sense of security or alter our capabilities.

In this situation, our education, our training and our will are not enough. Indeed, we need not only to mobilize the best competences in each discipline but also to generate synergy between them so that they start to resonate and produce the unexpected, the unseen and the unprecedented. For that, an interdisciplinary approach is required, one that is capable of linking together the many centers of expertise that we must call upon in order to apprehend all the aspects of our organizations, our enterprises and our technical systems.

We cannot rely on intuition alone. We need a framework into which the actions of the different transformation actors can naturally flow. This framework should enable us to answer the questions:

- How can we support the design effort, by making the most of the contributions from various specialties?
- How can we stimulate imagination in order for new ideas to burst forth, upon which future developments will be based?

- How can we square up to complexity, that of enterprises and their systems, and of federations of enterprises and systems of systems?

Conclusion:

The need: master the transformation of complex systems.

1.2 The objective and scope of Praxeme

Praxeme seeks to answer these questions and provide enterprises with the means to master their transformations.

It covers all the activities involved in the thought process surrounding the enterprise and in its transformation, from strategy to deployment, from culture to infrastructure.

The term "enterprise" is taken in both meanings, that of entity and of project. It refers as easily to technical and socio-technical systems as to initiatives or organizations.

Definition:

Enterprise: any type of organized and willful entity or action.

Thus, Praxeme is suitable for use in very diverse fields: the transformation of enterprises and organizations, the improvement and overhaul of information systems, software development, the design of transport or weapon systems, etc.[3]

The practices described by Praxeme lie within the transformation chain, which goes from strategy to deployment and which mobilizes disciplines such as management, strategic design, enterprise architecture, scientific disciplines and design techniques of organizations and systems.

Definition:

Transformation chain: well-ordered set of activities that contribute to the transformation.

A constant concern is present and apparent throughout Praxeme: to create the cultural and practical conditions to connect the centers of expertise and promote the flow of ideas. This results in particular attention paid to the manner in which the different disciplines represent things. Nevertheless, instead of seeing things through the schemas of a particular discipline, Praxeme seeks to reach the reality as it is, before it is shaped by specialized views. It reaches its objective of syncretism, not by combining the modes of perception but by recomposing them from an analysis of the reality.

Conclusion:

To help with the design and transformation of complex systems, Praxeme aims to:

- coordinate the disciplines that contribute to the design of the enterprise, in all its dimensions;
- provide these disciplines with tools and techniques that guarantee a rational, precise and complete approach.

1.3 The definition of Praxeme: method and methodology

The method is, by definition, the answer to the question "how?". Praxeme answers the questions raised in section 1.1 in terms of methods, approaches and procedures. Due to the coverage – the whole enterprise, in all its aspects –, it is necessary to wonder about the factors that determine the quality of the collective action. It is one thing to design a method for a homogeneous group of specialists; it is quite another to bring together the specialties coming from various cognitive universes and generate synergy between them. That is why it is

[3] This point is developped in section 3.2, p. 20.

necessary to remain open and to ask questions about these cognitive universes, what determines them and whether there be any underlying logic. This task falls to the methodology.

Definitions:

- **Method** (etymology: from the Greek *"methodos"*, the path to reach a goal): how to do something.
- **Methodology** (etymology: from the Greek *"methodos"* and *"logos"*, the discourse on the method): discipline whose aim is the controlled creation of methods.

The method meets the needs of the actors of the transformation, according to each person's specialty and role in the transformation. The term "method" conjures up the operating mode, the recipe, the guide to the action. In fact, several methods are required to cover the whole transformation chain. Consequently, one need becomes obvious: each method has to be designed not only in comparison with its own ultimate goal, but also in order to fall within the greater transformation chain as a whole. It is not therefore possible for it to suddenly materialize, complete and ready armed, from within a specialty or corporation. Its design has to start from a wider perspective. The objective, which is to deal with enterprise transformation, requires us to take a step back and to think carefully, which is the methodology part.

The methodology grants a degree of freedom that the method does not: it provides the elements required in the thought process to adjust the method to specific contexts or to articulate it with other contributions. Indeed, a method is a system; as such, its construction follows rules. For a method to work correctly, these rules should be respected. The methodology is the discipline that is concerned with the operating rules and the construction of the methods.

Enterprise methodology develops this thought process on an enterprise scale. It unavoidably handles very diverse material. Its objective is to link together all centers of expertise that contribute to thinking about and designing the enterprise, from every possible angle.

In order to reach its objective, Praxeme has to be both:

1. a method or, rather, a set of methods;
2. a methodology.

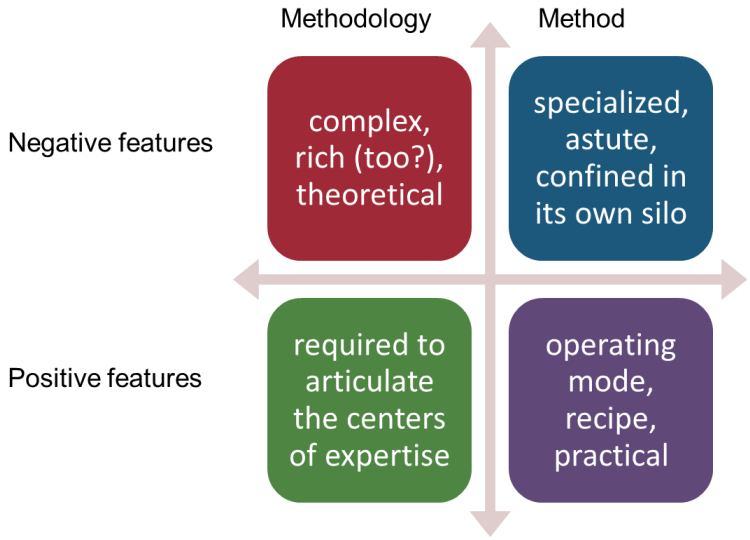
Methods are what practitioners, the different actors of the transformation, seek. But, because these methods are born from methodological thinking, their quality guarantees their integration into the transformation chain. For example, the method of designing processes is conceived in such a way that it takes care of upstream elements (strategic directions, knowledge of the business fundamentals, performance measurement) and obeys the documentation requirements enabling efficient use downstream (automation, design of service architecture…).

In conclusion, Praxeme is both a set of methods for enterprise transformation and an enterprise methodology.

Definition:

Enterprise methodology: methodology covering all aspects of the enterprise.

Figure PxMDS-01_2. Comparison between method and methodology



The method is intended for practitioners; it has to serve as a guide for their daily activity and aims to be highly practical. The risk is being highly specialized and closed in.

The methodology is a wider and theoretical way of thinking. It does not concern all practitioners, but it is necessary to ensure the quality of the methods and their articulation within the transformation chain.

1.4 The philosophy

The enterprise approach, in the meaning given earlier, is what provides the unity of the Praxeme method.

The conscious transformation effort begins with a desire to see the enterprise in its totality. This notion conveys two others:

1. Looking at the enterprise as a whole forces us to consider all the questions from a global perspective and to examine the implication they have on an enterprise scale. This truly represents a real effort. Our everyday practices consist, on the contrary, of isolating questions within organizational silos and reducing them to a project scale.
2. Considering the enterprise in its totality also means recognizing the multiplicity of its aspects and the diversity of its components; organizing a place for elements as varied in type as values, finance, performance, organization, knowledge, culture, logistics, IT, etc.; recognizing too that these elements, although not of the same type, can interact, influence each other and become part of a system whose coherence needs to be verified.

Referring back to the first point, we can speak of a global or totalizing approach. For the second, we can say: holistic and interdisciplinary approach. The interdisciplinary nature is a consequence, in the sphere of human competences, of the multi-aspect character of the enterprise and of the necessity to say everything about the enterprise in order to succeed with its transformation.

Another principle upon which Praxeme relies – as with any method – is that of rationality. The enterprise approach rests on the following credo: to decide well, you have to understand; to understand well, you have to represent. Hence the importance that Praxeme gives to modeling techniques. They extend the rationality principle by adding the effects of formalism to it.

Definition:

Formalism: set of coherent rules that prevail over a type of expression and ensure, for the resulting expressions, certain qualities sought.

On the one hand, turning to formalisms improves our knowledge of the enterprise; on the other hand, it reinforces our capability to intervene and increases our chances of finding innovative ideas. Consequently,

modeling techniques play a key role in Praxeme, as they provide the means to a rational and rigorous approach, highlighting the many factors that make up the reality and allowing us to extract new ideas.

The notion of system links these two principles of totality and rationality, as we will see in chapter 3.

However, in the 21st century, the faith placed in Reason cannot be the same as in the 18th century. Having learnt from history and worried about the impacts our productions have on society, nature and the universe, our enthusiasm is tempered and we review our rational instrumentation from three admissions:

- that of bounded rationality (we cannot know everything; something always escapes from the clutches of reason; we are kidding ourselves when we think we can control everything);
- that of multiple rationalities (several approaches of reality exist, several views based on metaphysics that are equally coherent and legitimate);
- that of the opacity of the ends (the whole of humanity is not devoted to good; nothing more than an "invisible hand" exists to help control our creations; it is by no means certain that we could reach a universal idea of good, nor that humanity could reach a level of maturity that would enable it to behave like a rational being).

Just like the thing-in-itself, the enterprise – for us and compared with our ends – is never fully known nor revealed; its signification never totally exhausted; its moral code never completely guaranteed. This is also what we mean when we speak of the enterprise as a complex system.

The position of epistemological humility follows on from here: we do not know everything, we will never know enough about our domains of study to exhaust them, we will never be able to reach true certainty in our proposals and decisions.

Definition:

Epistemological humility: attitude of humility that follows from the awareness of our bounded rationality.

These philosophical considerations do not distance us from the subject of enterprise transformation; on the contrary, they are at the heart of it, sustaining the appropriate attitude and inspiring practices. They lead to one imperative: openness.

Definition:

Openness: state of mind that keeps attention focused on new or unfamiliar contributions.

Whether it be at the level of developing the methodology or the practices in the field, openness is a mark of respect and a prerequisite for true cooperation. It is expressed in concrete terms by an active listening attitude, the first step in a democratic way of working and a condition for collective creativity.

The principles formulated in this section inspire transformation engineering. Its reason for being is to summon the resources of practical rationality – which the term "engineering" implies – to harness them for the reasoned transformation of enterprises. It joins forces with preexisting disciplines like enterprise architecture, organization or strategic design. Section 7.3 lists all the transformational disciplines.

Summary of the principles of transformation engineering:

- **Global** (or totalizing) and **holistic** (therefore interdisciplinary) **approach**: apprehend the enterprise in its totality and in all its aspects (everything that is connected with the enterprise is included in the methodology).
- **Rationality**: rigorously describe, formalize to guarantee the quality of our knowledge about the enterprise and, on this basis, increase our chances of finding innovative ideas.
- **Epistemological humility**: recognize the limits of our rational instrumentation, deduce from it an attitude of openness and active listening.

1.5 The positioning

Praxeme does not replace specialized methods coming from different practical domains; it seeks more to integrate them and to get the most out of them by including them in the transformation chain.

As an example, Praxeme places the performance indicators in a specific place in its reference framework, a place from where the indicators can be "projected" into the pertinent aspects of the enterprise where they are picked up with a view to automating them. In this way, the method for designing the indicators sees its efforts increased because it is integrated into the enterprise methodology.

The methodology distinguishes the approach and the process or steps to follow.

Definitions:

Approach[4]: way of apprehending the object of study.

Process[5]: way of organizing the activities; the steps to follow.

Praxeme proposes, above all, an approach to the enterprise and to systems. It includes, in addition, certain elements of the transformation process, although it does not stipulate a complete process. Regarding this point, it refers back to industry-standard practices, organized in phases. It is therefore possible and beneficial to combine the Praxeme methodology with industry-standard practices and compliance standards.

The articulation principle between the methods and frameworks rests on the Pro^3 schema, which is introduced in the following chapter. The document PxMDS-03 deals with the question of positioning in more detail (compared with CHAMPS2, UP, TOGAF, COBIT, etc.).

1.6 The form of the method

Praxeme presents itself as a set of documents, available free of charge under a "creative commons" license. Among these documents, we can find:

- the methodological guides, which are the specific methodological foundation of Praxeme,
- the procedure sheets,
- the forms accompanied by their instruction manual.

[4] Definition given in the *Petit Robert* dictionary, meaning 5: "way of tackling a subject of knowledge as to the point of view adopted and method used".

[5] "Way of acting" op. cit.

The structure of the methodological corpus and the document codifications are explained in the document referenced PxMDS-00, "Praxeme method overview".

This corpus is completed by the wiki[6] where we can notably find:

- the thesaurus, network of terms with their definition and discussions on terminology (in English and in French),
- examples of models, including generic architecture;
- the metamodel.

Definition:

Metamodel: model of models, that is to say a formal representation of the means required to build ordinary models.

A metamodel reveals the categories of representation at work in our effort to represent things. It expresses, as it were, the syntax of the language we use to talk about the world. This syntax is intimately linked to the recommended approach, in our case: the approach of the enterprise.

The metamodel is therefore the foundation of the methodology. Establishing a metamodel is an excellent exercise in intellectual hygiene. Moreover, the metamodel can be seen as a model of the tools that we use to represent and document the enterprise.

1.7 The initiative for an open method

We would like to be able to propose a complete, finished method to society and to the market. Unfortunately, the scope and subject matter covered mean that the method will always be under construction. To affirm otherwise would be to contradict the epistemological humility and openness of spirit that characterize our philosophy.

To address the subject of enterprise transformation, a process is therefore needed, the dynamics of construction. This is the role of the initiative for an open method. Launched in 2004 with the publication of a white paper, it is supported by the Praxeme Institute, a nonprofit, state-approved association. The initiative is also supported by a great number of actors, both public and private.

The Praxeme Institute is:

- the depository of the Praxeme open corpus,
- the guarantor of the spirit of openness of the initiative,
- the coordinator of the works on the method.

Extract from the statutes of the association

These actors, the contributors first of all, recognize that the primary quality of a method is to be widely shared, to the point of acting as the reference method. When all the project stakeholders speak the same language, share the same mindset, they can save themselves a lot of trouble and avoid wasting time and resources. Conversely, imposing a proprietary, confidential method, whether it comes from the client or the supplier, leads to an overspend linked to training needs and any necessary adjustments. In addition, methodology is not the business of enterprises in general; they rarely have the skills and necessary resources to develop a method, in accordance with tradition and state of the art. This is why it is essential, from an economic point of view, to provide the market with a reference method. This is the purpose of the initiative for an open method.

Its economic model rests on the pooling of investments: contributors use the existing resources as a starting point, finance the development of further work in answer to an immediate need, then transfer these new

[6] http://wiki.praxeme.org.

contributions to the open corpus. Praxeme has been built in this way, with the flow of contributions, following an overall plan that ensures the coherence and answers the specific requirements of its contributors.

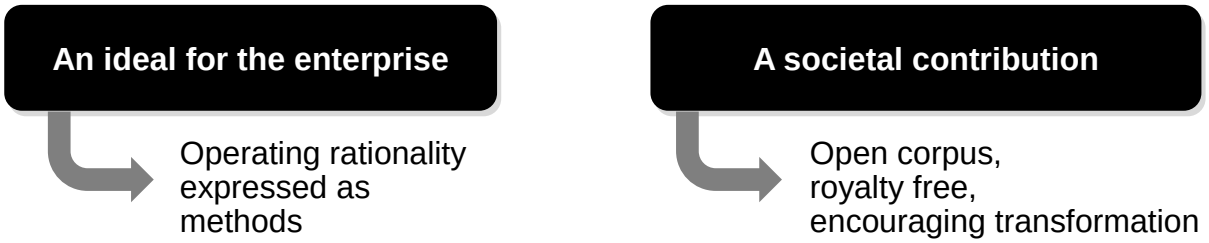
The table below summarizes the main contributions.

Figure PxMDS-01_3. Table summarizing the contributions to the initiative for an open method

Contributor	Year	Contribution
Sagem defense branch	2003	Within the scope of the redesign of the drone control systems, initial version of the nine methodological guides of version 1.
SMABTP	2004-2006	To control the overhaul of the information system, generalization of version 1 and architecture procedures and logical design in SOA. Also, first Praxeme training courses.
Les Caisses d'allocations familiales **(Office of Family Assistance)**	2006	Praxeme metamodel (version 1), required for the implementation of the Information Systems Repository. This metamodel was donated to the open corpus following a request from the French army.
French army	Since 2003	Promoter of the initiative since its very beginnings. Contribution within the scope of the unified platform of software development.
AXA Group	2007-2012	Development around Business Architecture.

A reference method represents a real industrial asset, by the increase in the skill set that it brings and by the cooperation that it facilitates between partners. The actors that support the initiative have a heightened awareness of the societal and citizenship value of this effort.

Figure PxMDS-01_4. The vision upheld by the initiative for an open method



2. The structure of the methodology

Enterprise methodology, by nature, embraces many domains and deals with subjects galore. This chapter maps the content of Praxeme. Around the three structures that fit together and form the backbone of Praxeme, it provides the landmarks to find one's way. These structures are: the Pro^3 schema, the Enterprise System Topology and the Organum.

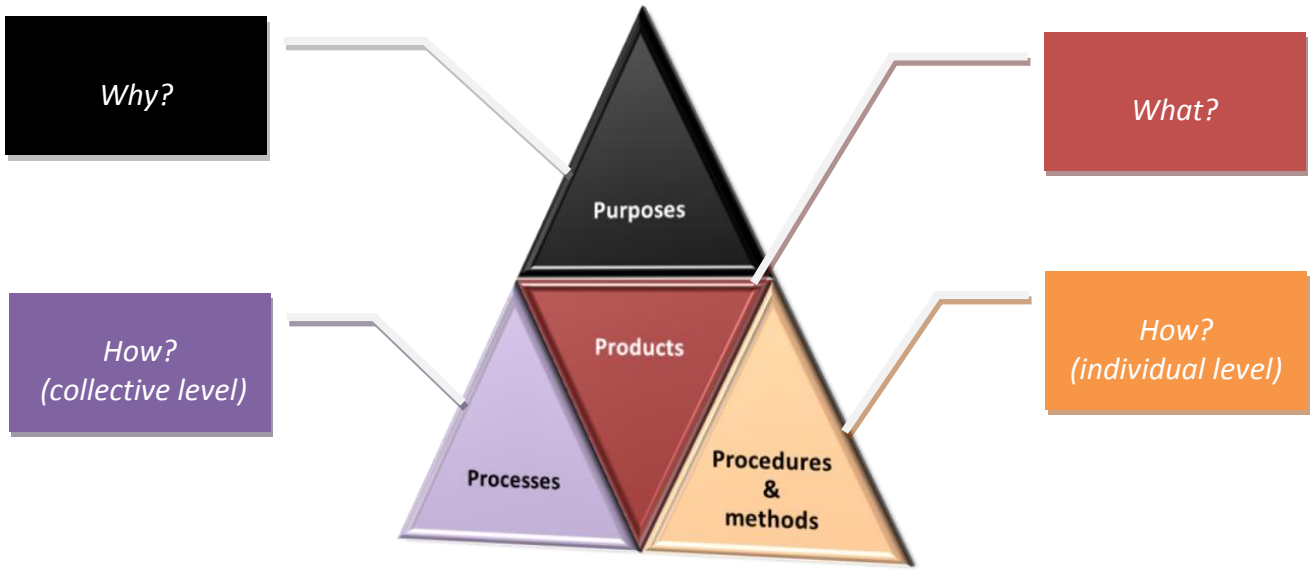
2.1 The field of the methodology

The methodology is concerned with notions that can be categorized in the following manner:

- first, the elements that emerge from the description of the object studied;
- then, the instructions that enable us to organize the collective work;
- finally, the operating procedures that guide the individual work.

These three groups of methodological contributions are relatively independent. We can treat them as dimensions. We name them respectively Product, Process, Procedures and methods. The analysis of the methodological field, split in three chapters, is represented by the Pro^3 schema (pronounced: “pro cube”), shown below.

Figure PxMDS-01_5. The Pro^3 schema



In addition to the three chapters that constitute the methodology in the strictest sense, a fourth has to be added: that of the aims. It deals with the requirements and motivations that preside over the drawing up of the methods. It enables us to broaden our methodological thinking and include the justifications within. In this way, the methodology joins the quality approach, in the sense of giving due weighting to effort.

The table below specifies the outline of each Pro^3 chapter.

Figure PxMDS-01_6. The three chapters of the methodology

Chapter	Key question	Content
Aims	Why? For what?	Requirements and motivation that apply to the method; justification of the measures and level of effort required
Product	What?	Analysis of the objects produced or transformed, study of their composition (notably, theory of enterprise)
Process	How can we organize ourselves collectively?	Organization, roles, processes, procedures, phasing… Most popular component of the methodology. Equally, notions of lot, iteration, increment and regulation advice.
Procedures and methods	How can we work individually?	Operating modes, techniques (of representation, documentation, analysis, design, etc.)

The Pro^3 schema signposts the field of the methodology and enables us to find the answers sought within. The most obvious chapter, when we talk about method, is the one on “Process”. But, if we have to understand the

nature of the enterprise and to find the notions that enable it to be described, the practitioner will turn towards the "Product" section. When the time comes to act – sometimes to represent activities; other times to design equipment, etc. – the practitioner will turn towards the "Procedures and methods" section.

The following four sections take up the Pro^3 chapters.

Definition:

Pro^3 schema: schema of the structure of the methodology.

2.2 The justification of the method

The "Aims" dimension of the methodology is not expressed by one type of isolated component. It appears in the methodological guides as analysis and justifications that accompany the presentation of the method.

As far as methodology is concerned, the main trap is the tendency to dogmatism. It leads to the method being frozen, which leads to its obsolescence. It also fuels intolerance, harming the efficiency as much as the communication around the method.

However, on the pretext of escaping from this tendency to dogmatism, we risk falling into the opposite trap: abandoning all rigor, throwing off the yoke of the method, liberating ourselves from all requirements. It is good form to invoke pragmatism to legitimize this attitude, which is quite the opposite. We have to understand that this attitude is not in the slightest bit pragmatic, unless we accept that the term "pragmatism" only refers to the alibi protecting us from effort and rigor, the hidden face of the anti-method. Over the last decades, one important victim has fallen foul to this overwhelming tendency: modeling, leading to a serious consequence: the loss of control.

It is in the "Aims" dimension that the analysis of the context in which the method operates therefore plays out. It is extended by the demands of principles, precepts and values, which are at work in the action. There are therefore two parts to the analysis:

1. a critical part which, as we have just illustrated, has to reveal the conditions of both the action and the decision;
2. a deontic part which is the positive moment, where the requirements to be respected in the action are affirmed.

The first part is all the more important given that we are tackling enterprise transformation, a complex situation with many determining factors passing through. A poor analysis of this context will lay us open to disappointment, linked to one of the two types of negation of complexity: naivety or brutality.

Based on the results of the critical approach, the method will be able to select and strengthen the precepts of the action and decide on the requirements to put into practice. Nothing could be further from the Praxeme spirit than the procedural side that is often associated with the method. For the Praxemian, the method is above all a guide to the action and a spur to creativity. The guide to the action is useful on an individual level, indispensable on a collective level. Creativity is stimulated thanks to the abstraction principle and the sequencing of competences.

2.3 The "Product" dimension: the reference framework

In accordance with the scope of the method, the product can be a process, a product in the industrial sense, software, etc. For the enterprise methodology, the product is the enterprise, in its dual meaning given above (p. 5). To apprehend this complex reality, we require an analysis frame that reveals the structure and enables us to account for all its substance. This is what we call a reference framework (or conceptual framework, according to the terms of the standard IEEE 1471[7]).

[7] Cf. IEEE Std 1471-2000 *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*.

Definition:

Reference framework: theoretical structure applied to a system so as to organize our perception of it.

The framework clarifies our intellectual departure point. It specifies the categories of representation that we give ourselves to apprehend the reality. Without such a framework, collective action is hampered; it suffers from confusing terminology and approximations in the representation.

The framework has to be seen as a summary or as the structure of the most complex object, the metamodel. The latter is tasked with modeling the intellectual tools, the tools of representation that are at our disposal and is far more detailed; it sets, in detail, the relations maintained by the categories of representation.

The framework obeys certain rules, following the example of architecture (we can, for that matter, see it as our intellectual architecture). These construction requirements are indispensable to remove amateurism and ensure minimal quality. An imperfect framework would ignore dozens of questions that arise thousands of times in different projects, leading to a waste of resources and the heterogeneity of responses.

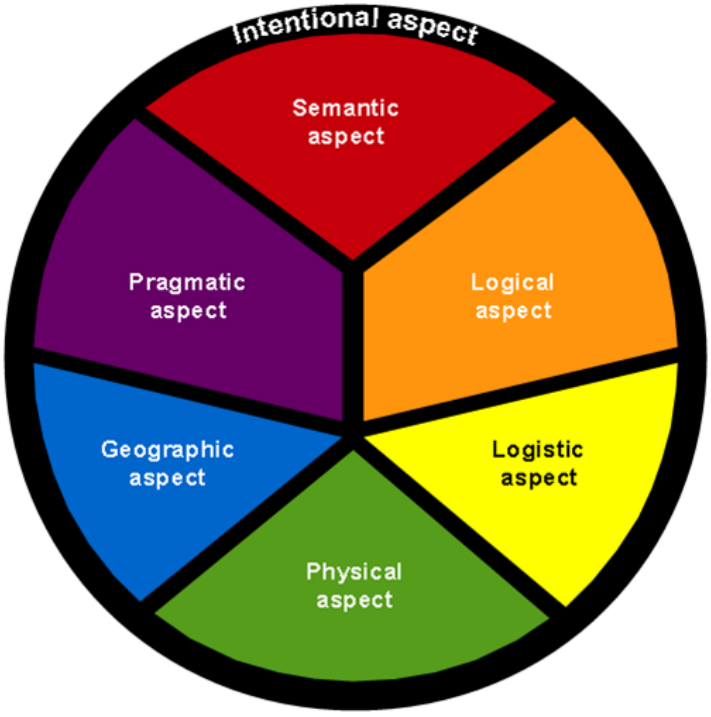
These construction rules, as well as the framework itself, are detailed in the document PxPRD-01. The framework that Praxeme proposes to structure the "Product" dimension is called the "Enterprise System Topology". The notion of Enterprise System is the topic of the next chapter.

To remember:

- **Topology** (etymology: the "discourse of place", from the Greek *"topos"*, the place, and *"logos"*): the term has been chosen to evoke the answer to the question "where should the information and decisions go?"); in addition, the method is concerned with the neighborhood relations between elements, which refer to the meaning that this term has taken on in mathematics.
- **Enterprise System Topology**: framework that applies to the enterprise and approaches it through its aspects.

The figure below shows the esthetic shape of the Enterprise System Topology (EST). Its explanation can be found in the document PxPRD-01. Chapter 4 of the present document introduces the aspects.

Figure PxMDS-01_7. The Enterprise System Topology



The EST identifies seven aspects through which everything there is to be said about the enterprise is organized. On this representation, the sectors are in contact when there are relations between the aspects. The table below introduces the aspects.

Figure PxMDS-01_8. The definition of the aspects

Aspect	Definition	Examples of actions
Intentional	The enterprise ethics and its ultimate goals (values, strategy, culture…)	Elucidate the values Articulate the vision Establish a common language Analyze performance
Semantic	Knowledge and the business fundamentals (the environment, the enterprise offer…)	Capture and formalize the business knowledge (through business objects)
Pragmatic	The enterprise activities and its organization (roles, processes, management and control styles, governance…)	Design the organization Model the business process Simply the processes
Geographic	The location of the enterprise activities (enterprise geography, virtualization, teleworking, mobile equipment…)	Evaluate deployment scenarios (24/7…)
Logical	An intermediary aspect between business and technology, introduced in the transformation chain to facilitate the design of technical systems (equipment)	Design the optimal information system Develop the transformation trajectory for the technical systems
Logistic	The set of technical resources that serves the business activity	Implement the technical system (software, hardware, logistics)
Physical	The Enterprise System fully deployed (with all its resources located)	Prepare the deployment Size the target Accompany the change Analyze the feedback

2.4 The "Process" dimension

In the "Process" dimension, Praxeme does not fix a general structure, but only the distinction between the "project" and "enterprise" scope[8].

Praxeme proposes two types of contribution:

- on the methodological level, certain notions and points requiring vigilance that need to be taken into account to control the actions[9];
- on the method level, some processes, including the steps to follow with innovative projects or those with a high degree of uncertainty, and a process for enterprise transformation[10].

[8] Distinction detailed in section 6.4.

[9] These elements are discussed in chapters 6 and 7 of the present document.

[10] See section 6.4.

2.5 The "Procedures and methods" dimension

The chapter on procedures and methods is the most prolific. It assembles the methods and procedures, and the operating modes that help the practitioner in his or her daily work. The scope of the enterprise methodology means that there are a great number of these procedures. The structure of the Pro[3] chapter is as follows:

1. one part made up of the procedures and methods that can be attributed to one aspect of the enterprise (in the sense of the Enterprise System Topology);
2. one part structured by a classification of activities.

The table below shows this classification of activities.

Figure PxMDS-01_9 The classification of activities.

Type of activity	Definition	Illustration
Management	Management, steering and management of projects and teams	Writing quality plans, organizing projects
Development	Product development	Modeling techniques, interview techniques, programming methods
Judgment	Product and activity evaluation	Quality, test
Equipment	Tooling	Development of UML profiles, customization of modeling tools
Support	Support, training	Pedagogic methods
Environment	Supply, logistics	

Definition:

Procedure: specified manner in which to do something.

2.6 The structure of projects and initiatives

Projects and programs handle huge numbers of documents and presentations. To help leverage these corpora, Praxeme proposes a structure that has been perfected project after project. In order to facilitate its use in international contexts, the sections of this structure have been given Latin names, which have, in the main, been kept as such in European languages, thus avoiding the need for translation. This structure is the Organum[11].

Definition:

Organum: proposed structure to organize the subject matter of a project or an operation.

The table below introduces the sections of the Organum. The document PxPCD-01 provides additional details.

[11] Organum (or Organon) is the name given to the collection of books of Aristotle's logic. Also: the *Novum Organum* by Francis Bacon.

Figure PxMDS-01_10. The sections of the Organum

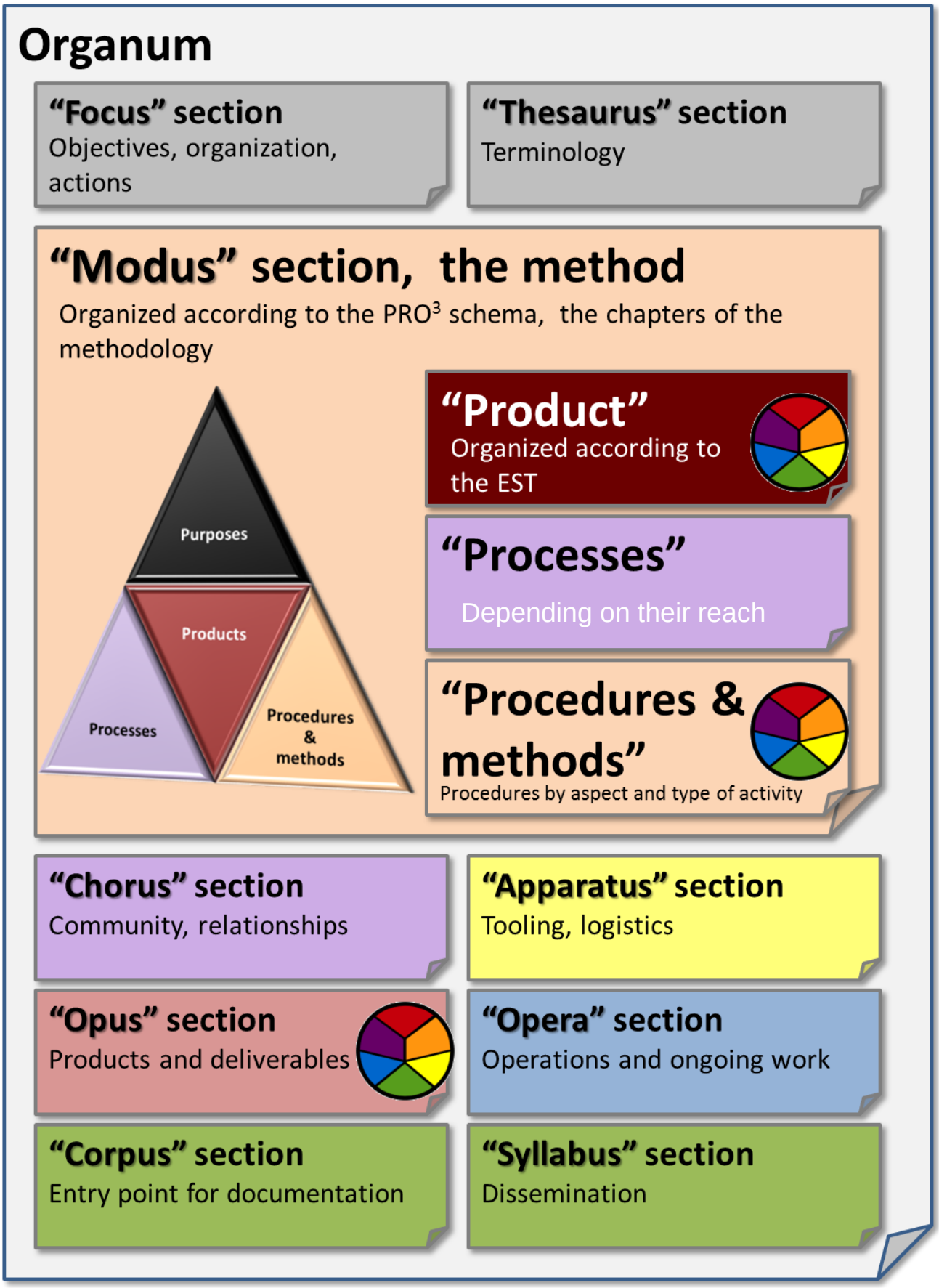
Organum *(broader structure Praxeme methodology)* *Name of rubric*	*Code*	*Meaning* *Item of the Organum (directory, category…)*	*Content*
Focus	**FCS**	Scoping - objectives & roadmap	Plans, Terms of Reference, requests, update
Thesaurus	**THS**	Understanding - vocabulary	Glossaries, dictionaries, discussions on vocabulary (terms, expressions…)
Opus	**OPS**	Building - architecture outcome	Deliverables, models, architecture dossiers…
Chorus	**CHR**	Networking - community	Material for the sessions of the working group (agenda, minutes…)
Corpus	**CRP**	Investigating - Documentation collected	Bibliography, material from external sources, input from the companies
Modus	**MDS**	Guiding - methodology	Method documents, guidelines
Apparatus	**APR**	Tooling - equipment	Works on tooling
Syllabus	**SLB**	Spreading - training, marketing	Material for communication and training
Opera	**OPR**	Supporting	Operations

The Organum applies to the initiative for an open method. In this context, the Praxeme method is situated in the "Modus" section. The material linked to the communication and dissemination is found in the "Syllabus" section; work on tooling the method, in "Apparatus". The projects undertaken within the initiative are located in the "Opera" section.

2.7 The correspondence between the structures

The three structures – Pro^3, Enterprise System Topology and Organum – fit together in accordance with the schema presented below. If we consider things at the level of the initiative for an open method, the broadest structure is the Organum. It is important to note, however, that this structure also applies on a work-in-progress and project scale, placed in the "Opera" section. The "Modus" section, the method, is structured according to the Pro^3 schema. The "Products" chapter is decomposed in accordance with the Enterprise System Topology (EST). The "Procedures and Methods" chapter is split into one part also decomposed by the EST and another structured through the classification of the activities.

Figure PxMDS-01_11. The complete edifice, with the embedded structures



Moreover, it is possible to establish a correspondence between these three structures. In this exercise, the Enterprise System Topology is the reference structure. Indeed, it expresses the most developed theory of the methodology and is the fundamental structure in working with Praxeme. To build the Organum, one just has to consider the project (the initiative, the work in progress, the operation…) as a system and apply the EST to it.

The color code used for the structures shows the correspondences from section to section (see PxPCD-01).

3. The Enterprise System

This chapter explains the central notion of the Enterprise System as well as its applications and corollary notions: system, environment and system quality. It also includes a discussion about the opposition between aspect and view. This distinction greatly determines the practices.

3.1 The notion of Enterprise System

Besides its use in astronomy, the word "system" has been used in European culture since the 17th century to refer to "systems of concepts". It retained this conceptual meaning until the 19th century before progressively taking on the meaning it has today, applied to objects more or less complex or to sets of objects. Established in the middle of the 20th century, systems theory enhanced this concept and turned it into an instrument of analysis and design[12]. The concept of system is applied in many sciences, including human sciences and represents one of the modeling tools, particularly useful when faced with complex realities.

Definition (temporary):

System: a set of interconnected elements, perceived as a whole.

This definition counterbalances a substantial part – the elements – with the subjective, constructivist character of this notion. The system is always the result of an act of knowledge, a decision to organize part of the reality into a coherent whole, endowed with an identity and seat of an activity and change. In one way, the system is, above all, a representation of a reality made by the human mind, seen in its totality and in its interactions with its environment. Henri Poincaré summarized this constructivist value perfectly: "The system exists only in the human mind."

This brings us back to the situation of the modeler. For modelers, the system is not an autonomous "being", given from outside. It is always the result of a commitment, a decision, to carve a more or less coherent part from the substance of reality, upon which we exert our understanding. All modelers must live and breathe this conviction, in order for them to move the line of what is given contingent upon something, towards that which becomes a pertinent construct.

Precept:

Constructivist freedom: Always remember that the system is never given, but results from a decision; be aware of this decision and support it with arguments.

Thus, we will sometimes have to move the boundaries of the systems that we are studying or building. For example, we position the human actor in the technical system. We may judge that there are no technical systems strictly speaking, but that we are always dealing with socio-technical systems. We may ask ourselves whether technical actors (like the IT worker) have to be represented in the business process in order to take into account their role in enterprise performance and reactivity. We will investigate the blurred zones between product system and production system, between operations and transformation (see section 6.6)…

[12] In "*La Théorie du Système Général*" (General System Theory) (1977), Jean-Louis Le Moigne proposes the following definition (in fact, more a digest of the notions attached to the concept of system): "something (a structure) that is functioning and transforming toward a goal in an environment. In more trivial, but perhaps more mnemonic terms: something (anything, presumed to be identifiable) – which in something (environment) – for something (aim or project) – does something (activity = functioning) – through something (structure = stable form) – which transforms itself over time (change)" (PUF, 2nd edition, 1984, p. 61 et 62).

As an enterprise methodology, Praxeme gives itself the enterprise as central object. It calls this object the "Enterprise System", affirming in so doing the rational approach that it implements. This expression reflects the decision to view the enterprise as a complex set of interacting elements, endowed with certain properties. Consequently, we have to identify the important elements, examine their relations and approach the enterprise in all its aspects. To talk about an Enterprise System is to see the enterprise as an organism, aware of itself, controlling its aims and representing itself in order to evolve and have better control over its destiny. The Enterprise System is the enterprise viewed and designed as a complex object presenting the characteristics of a system – in the systemic sense –, gifted with decision-making autonomy and self-organization.

Definition:

Enterprise System: the enterprise which sees itself as a system.

3.2 The applications of the method

To inventory all possible applications of Praxeme is the same as asking ourselves what we consider an Enterprise System to be. The answer is: more or less everything! Praxeme has been applied to organizations, socio-technical systems ("Product system") as well as information systems.

The scope of the domains of application is made possible by the theoretical foundations of the method, built on systems theory. In return, this versatility creates obligations as the terms and notions are organized in different ways when it comes to sharing them between several professional communities. The argument will appear again when we have to position specific notions in the right place in the framework and the metamodel.

Praxeme defends the idea of a single framework applicable to all systems, irrespective of type. The Enterprise System Topology was designed with this in mind[13]. In practice, there is no general discipline of system design, but as many disciplines or corporations as there are types of systems. Notably, Product systems, seen as being essentially technical, mobilize engineering disciplines and narrowly specialized scientific knowledge, often to the detriment of human or societal aspects. Yet, there are different facets to every system. Even a mechanical system must be designed to include its uses and its maintenance, therefore in its relation with people and social aspects. It remains possible to separate, in theory, the different facets of a system and entrust them to ad hoc specialties. However, we should guard ourselves against the danger of reductionism that consists in omitting, if not denying, certain components of the reality. The recommended attitude consists in considering, at the offset, that any system presents all the facets identified in the Enterprise System Topology. For example, we shouldn't design a washbasin without integrating any secondary uses, in this case installation and maintenance, nor dismiss ecological considerations[14].

The domains of application of the enterprise methodology:

- Organizations, organizational entities: enterprises, associations, groups, civil service, state, society…
- Business processes (industrial or tertiary).
- Information systems, IT systems, software, software packages.
- Technical systems, transport systems, weapon systems…

[13] This explains the changes between Praxeme versions 1 and 2. Applying the Enterprise System Topology to a transport system revealed the limits of version 1.

[14] In this example, the uses, even secondary ones, are included in the "pragmatic" aspect; the ecological considerations in the "intentional" aspect. They can inspire design decisions, such as the choice of material, its provenance or the integration of a wastewater recycling process, depending on the quality of the water. This trivial example of the washbasin shows that it is always possible – and generally desirable – to broaden our thinking from the object towards the system regarding its uses and implications. From there to say that the washbasin is an Enterprise System…

3.3 The quality of the Enterprise System

The first characteristic that comes to mind with regard to the enterprise is, without doubt, its complexity. This justifies applying a systemic approach to it. Why do we consider the enterprise as a complex object? Because, whatever its size or sector, it is made up of a large number of elements, particularly of a disparate type. The components are linked between themselves within a same category and between different categories. The relations between the components are part of the categories that must be specified. One essential factor that heightens the complexity lies with the autonomy of willingness and action of certain components – components which themselves can potentially be seen as complex systems.

Definition:

Complexity: (from Latin "*complexus, complecti"*, contain) character of a system composed of several elements linked by multiple relations and unable to be understood only as the sum of its parts.

Complexity arises from the intricate relations of multiple and diverse factors. It makes the emerging properties of the system possible that would otherwise be impossible to deduce analytically from the sum of its parts.

Complexity is often perceived as a risk or, at least, a difficulty. It is also a strength: it shows the range of resources that the system has at its disposal. Its ability to invent and adapt is proportional to it. In the most advanced cases, the system has emerging properties that cannot be deduced from its composition. Does that mean that we should banish the expression "reduce complexity"? The complexity of a system, if it is real, is necessary and constitutes an asset that it should know how to take advantage of. Complexity should therefore be assumed by the system and respected by its designer.

Conversely, complication is an artificial and unnecessary complexity, a result of poor design or accumulated over time. It is the opposite of irreducible complexity and brings no added value. Artificial and cultural systems tend towards complication against which we should fight. This drift particularly affects organizations, proportionally to their size.

Definition:

Complication: artificial and unnecessary complexity.

To apprehend complexity, the traditional approach of top-down hierarchical decomposition (also known as Cartesian) is not the right solution. We opt for the systemic approach, which seeks the links between the separate elements and makes room for system-specific, emerging properties that are not possible to deduce from the sum of the parts. Chapter 4 will show that the approach has to be chosen according to the aspect in question.

One characteristic often associated with complexity is uncertainty. The latter comes, in part, from the complexity itself and, to a greater extent still, from the environment. We seek every possible means of reducing uncertainty, but we have to adapt the system and our action to an irreducible uncertainty, impossible to remove.

The result of complexity is to discourage action and generate resignation. The methodology appears exactly as the remedy to a failing imagination and abdication of will.

Quality characteristics of a system:

- Agility: ability to adapt in a cost effective manner.
- Traceability: ability to piece together the determining elements of the chain (determination chain).
- Auditability: ability to implement and carry out checks.
- Interoperability: ability to link the system to other systems and integrate them into a coordinated way of operating.

3.4 The system and its environment

The systems we study are neither alone nor closed. They are plunged into an environment with which they exchange material and information. Even if we consider an "economy-world[15]" or, today, the globalized economy, we are not able to apprehend it in its entirety if we detach it from the other systems it interacts with, starting with the environment (nature) and society (politics, culture, values, social contract…).

From the system viewpoint, two major features characterize the environment:

- the absence or weakness of control that the system exerts on its environment;
- the uncertainty.

As the Enterprise System is not alone, the method has to provide the means of analyzing the relations with its environment. Within the latter, it would be advisable to isolate other systems and, occasionally treat them as Enterprise Systems themselves. To this aim, Praxeme provides a multi-system approach, which demonstrates its utility in mergers and acquisitions, convergence programs, the design of extended enterprises or enterprise federations. This approach is the topic covered in the document PxPCS-07.

3.5 The approach of the Enterprise System: the aspects

The first measure to approach complex systems has been formulated in a long-held principle:

Principle:

Separation of concerns: it is advisable to deal with the questions in separate, homogenous sets, which enables the decisions to be serialized and the responsibilities clarified.

This common-sense principle is almost constitutive of methodology for more than half a century. Praxeme applies it and associates the term aspect with it. The approach actually consists in approaching the Enterprise System by distinguishing different aspects of its reality. Of course, we have to identify these aspects as coherent sets of strongly linked elements, relatively autonomous and independent from each other. In accordance with the precepts of systems analysis, the relations between these aspects are not neglected. The Enterprise System Topology identifies these aspects and forms the Enterprise System theory.

Definition:

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

The reality must be approached from different angles. The aspects are parts of the reality being studied. Their separation is in response to both formal and practical requirements. The scope of the aspects is such that any piece of information or decision about the system can find one and only one natural place.

Unlike a view or a point of view – that depends on who is looking –, the aspect belongs by rights to the reality observed. The criterion used to delimit the aspects does not include the types of actors who are involved in the transformation chain, nor any other element linked to the way of acting on the object. They are only concerned with the object studied. It is only secondarily that certain disciplines are defined as specialties related to an aspect (semantic modeling, logical architecture…)[16].

Methodology, in particular Merise, has spoken for a long time in terms of levels of abstraction or separation of concerns. These levels were clearly separate from the views, also defined at the same time and in the same methods. Then, the levels were given as more fundamental, more important than the view. These methods first

[15] To quote Fernand Braudel.

[16] NAF (*NATO Architecture Framework*, p. 525) defines the aspect: "A coherent and consistent set of characteristics of a system as seen from a given viewpoint". This definition better suits the idea of view.

defined the level as expressing the internal structure of the system. The views were then defined in their relation to the actors and for communication needs[17].

The term "level", nevertheless, was not a good one in the sense that it conveyed a certain idea of hierarchy, and therefore of value.

3.6 Communication about the Enterprise System: the views

Whereas the reality, in our approach, organizes itself into clearly distinct aspects, the same cannot be said for how the observer or actor sees things. Some aspects are difficult to grasp because they are abstract. Others have a counter-intuitive position, which can only be understood from architectural requirements. Spontaneous perception mixes different types of elements. For example, an expression of needs will mix operating logic with operational requirements; it will contain organizational presuppositions but will remain silent on the motivations behind the operations. This expression is therefore incomplete and mixed up at the same time. The elements have to be separated according to type to be included in a usable order. That said, it is not a question of getting rid of the expression of needs: it remains an essential means of communication. The same applies to the many documents and representations that we cannot directly include in the Enterprise System Topology but that concern elements of one or several aspects, to present them in a communication context. The views satisfy this communication need.

A view requires an actor looking at something. It provides access to part of the reality observed, from this actor's point of view. It is therefore a subjective view: the subject's particular situation faced with the reality. The advantage of a view is for communication as it develops, rightly so, against the needs and language of a particular kind of actor. The price to pay is a certain degree of confusion:

- the views overlap as several types of actors have to know about the same elements;
- the views are incomplete as they are limited to the need for knowledge and possibilities of understanding of a type of actor, sometimes at a given moment;
- the views can encourage confusion because they mix different types of elements, sometimes within a same sentence or representation.

In practice, a view is made up of a selection of elements taken from one or several aspect models. For example: the "Use view" and the "Organization view" are both representations of the pragmatic aspect, one focused on a local stake, the other covering the whole of the organization. A "business view" will combine elements taken from the intentional aspect, the business aspects (semantic, pragmatic and geographic), but also from the logical aspect (to arbitrate the investments) and from the software aspect (mock-ups in particular).

Definitions[18]:

Point of view: perspective of an actor or type of actor.

View: selection of information on the system, from a point of view and for a specific purpose.

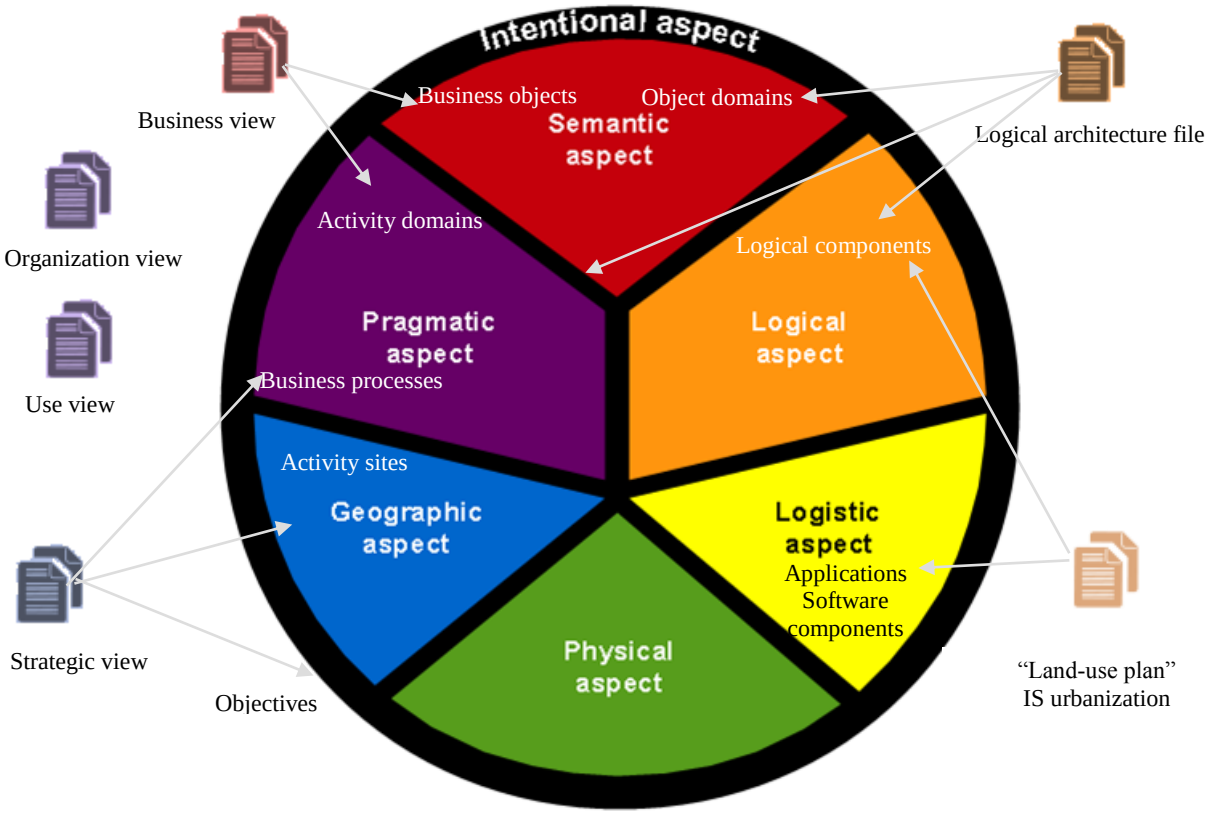
The content, the form and the vocabulary of a view are adjusted to the recipients' profile.

The figure below provides some examples of intra-aspect and composite views.

[17] For example, Merise distinguished between the "external views" and the data model: the latter provided the normalized and complete data structure, whereas the former provided an extract, potentially denormalized, for a particular use.

[18] Cf. IEEE Std 1471-2000.

Figure PxMDS-01_12. Some examples of views in their relation with the aspects



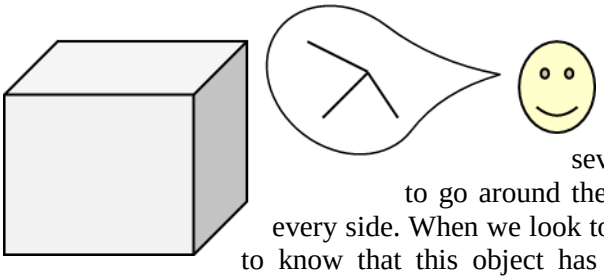
From an actor typology (contracting owner, organizers, strategists, IT workers, subcontractors, etc.), we define the views, which can take elements from one or several aspects.

For example, the Organization view and the Use view are both extracts from the full pragmatic aspect. The former gives the global vision: organization and process. It is of interest to the organizer. The latter provides insights from an enterprise function or project standpoint.

3.7 About the difference between aspect and view

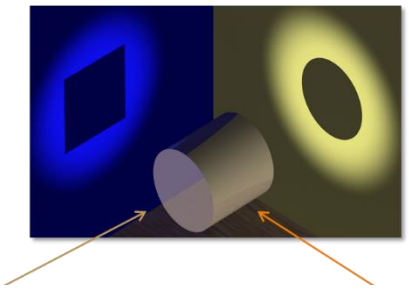
The notion of view has been used for several years and promoted by several methods. It is not to be confused with that of aspect. Praxeme uses both these notions. The notion of view introduces a part of subjectivity, absent from the notion of aspect. The views are subjective in the sense that they are placed on the subject side – the actor or observer considering the system. Conversely, the aspect is objective, on the object side.

Figure PxMDS-01_13. Illustration of the difference between view and aspect



This distinction between aspect and view, essential for the theory of knowledge, can be seen in the use of both terms and in the words that accompany them. When we look at a cube, for example, we never see all the sides. We can make several representations, take several views of it but we will have to go around the object, look at it from several perspectives in order to see every side. When we look to fully represent the cube and articulate the views, it is useful to know that this object has six sides, even if this idea does not come to us through experience but through understanding. The geometry comes with the drawing.

Figure PxMDS-01_14. The object seen from different points of view



The framework at the foundation of Praxeme targets the internal organization of the Enterprise System, independently of who observes it. It is a prerequisite to mastering the mass of knowledge, information and decisions that concern this complex object. Our aim is to isolate the internal logic, before any methodological working out and well before dealing with the questions that involve the actors: responsibility, organization of the transformation, communication, etc. In so doing, Praxeme includes itself in the Merise method filiation as opposed to Anglo-Saxon methods that, with the passing decades, only retained the notion of view and focused on communication to the detriment of the internal logic of the system[19].

Finally, the last argument in this discussion, structuring the Enterprise System into aspects obeys architecture rules: non-redundancy, weak coupling. If the overlap between views is quite acceptable, the same cannot be said when you have to organize the mass of information and decisions concerning the enterprise. The Enterprise System Topology was developed in answer to the requirement for an optimal ranking of the descriptive elements of the enterprise, for their controlled use.

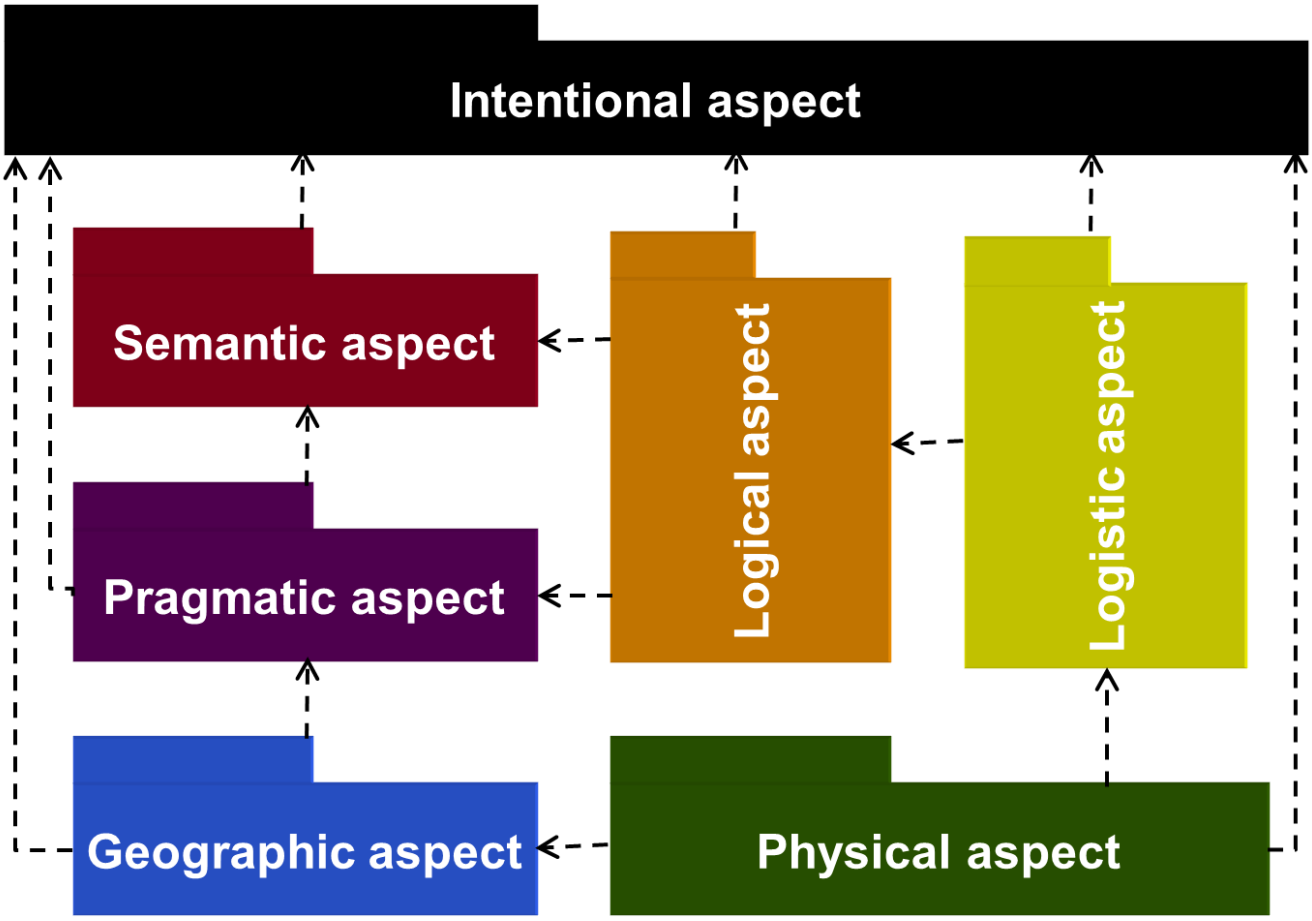
[19] If we refer to the etymology, the choice of the term "aspect" is not particularly good either, as it comes from the Latin "*aspectus*" which means look. The term "facet" would have been preferable. Note, all the same, that if "view" is clearly on the side of the subject who looks, "aspect" is on the side of the object that appears ("appearance presented by something" as defined in the *Grand Robert* dictionary; certainly, far from the thing per se, but we know that we will never attain that point…). Or again: "Aspect is purely objective; that is to say that in the view, what dominates is the idea of the subject who sees and in the aspect, what dominates is the idea of the object that is seen…", Émile Littré, *Dictionnaire*, article *Aspect* (cf. in French: http://littre.reverso.net/dictionnaire-francais/definition/aspect).

4. The content of the Enterprise System

The previous chapter introduced the notion of aspect. The Enterprise System Topology – one of the main structures of Praxeme – identifies them. This chapter introduces the seven aspects. Each one is the subject of a methodological guide.

When we open a repository of the enterprise descriptions built according to the Praxeme method, we find seven containers, each one corresponding to an aspect defined by the Enterprise System Topology. At the root of this repository, we should always find a diagram that strictly replicates the framework: aspects, color code and dependencies, as shown in the figure below.

Figure PxMDS-01_15. The Enterprise System Topology formalized as a package diagram



In this representation, dotted-line arrows show the dependencies between the aspects. They express reference relations: the content of the aspect at the start of the arrow can refer to the content of the aspect shown by the arrow. The framework retains only a small number of dependencies, in the aim of reducing coupling to the bare minimum. The aim is to avoid complicating the management of the information and decisions that supply the enterprise description repository (cf. PxPRD-01).

4.1 The aims of the enterprise: the intentional aspect

Definition:

- **Intention**: "the fact of giving oneself a particular goal"[20].
- **Element of intent**: formulation of a desire recognized in the enterprise and which directs its action (aim, value, rule, objective, requirement, metrics…).
- **Intentional aspect**: aspect of the Enterprise System which gathers together the elements of intent.

The intentional aspect, the first in the order of determination, gathers together all the expressions of the enterprise will: its values, its objectives, its requirements, the indicators (often closely linked to the objectives), and its vocabulary which reveals its culture. The term "intention" has been chosen as generic enough to cover these different notions. It also refers to the aims the enterprise gives to itself. The table below shows the types of elements of the intentional aspect, organized into four facets, 'VWVV' (values, wants, valorization, vocabulary).

Figure PxMDS-01_16. The content of the intentional aspect

Facet	Typical elements	Examples	Discipline	Examples of action
Values	Value, ideology	Respect for persons, enterprise responsibility, justice	Axiology	Elucidate the values Negotiate the values
Wants	Objective, requirement	"Conquer a market" "Design a new, adapted product"	Teleology	Develop the strategy Motivate the personnel
Valorization	Indicator, measurement, improvement potential	Progression of revenue, productivity, success rate of sales appointments	Metrology	Build the Performance Tree Help to define dashboards and objectives
Vocabulary	Term, definition		Terminology	Collect glossaries, give a canonical definition

These four facets do not impose themselves as a structure of the intentional aspect. The method restricts itself to distinguishing the types of elements of intent, leaving us with the possibility of structuring this aspect as we see fit. We therefore have to make a decision about the architecture of the intentional aspect. Rather than reusing the four facets above, we can choose:

- to structure the aspect by the sources (emitters or original documents that often contain several types of elements),
- to opt for a specific criterion, with a notion of domain as with any other aspect.

Thus, the intentional aspect has its own structure and obeys its own rules that do not necessarily reflect current practices. Consequently, it becomes interesting to define *views* to facilitate the communication with actors with specific profiles, for example:

- an ethical view, centered on the enterprise values and also including their impact on the other aspects of the Enterprise System;

[20] Source: *le Grand Robert* dictionary.

- a metrological view, composed of indicators, their links with the transformation objectives, as well as their projection towards the business concepts, activities or any other type of element in the other aspects;
- a terminological view, expressed through the enterprise thesaurus and showing, in addition to the terms, the traceability links that connect them to the other elements.

These sub-products, the views, respond to the needs of communication and manipulation; they are associated to uses. The method has to satisfy these needs, without changing the internal logic of the structure of the Enterprise System. It does this by distinguishing the notions of aspect from view.

The intentional aspect collects the elements of intent, which appear as rather informal expressions. These elements are then linked to elements from the other aspects, either because the latter formalize them or because they provide an answer. We keep the relations between the elements of intent and the modeling elements. This ensures the traceability, a key measure for controlling the description of the system.

Definition:

Projection: fact of associating an element from the intentional aspect to one or several elements from other aspects.

Precept:

Traceability: the link between the element of intent and the modeling elements must be kept (in this way establishing traceability chains to help analyze the impact should the system evolve).

The enterprise terminology has been included in this aspect for the following reasons:

1. The vocabulary reveals the way things are perceived. Implicitly, it conveys the presuppositions closely connected with the enterprise values and the way that the enterprise sees itself (its ideology).
2. The other elements of intent are built from terms whose meaning often needs to be clarified. It is therefore necessary to associate the terminology with the other components of the intentional aspect.
3. Each term is susceptible to be projected (linked or formalized) towards an element from any one of the other aspects. In the Enterprise System Topology, only the intentional aspect benefits from a position that allows this game of projection.

The approach of the intentional aspect is detailed in the guide PxPRD-10.

Summary of the intentional aspect:

Stake: clarify the enterprise intentions, its aims, its vision (how it sees itself, how it would like itself).

Content (expressed by the mnemonic formula: VWVV): values (ethics), wants (objectives, requirements), valorization (metrics), vocabulary (terminology).

4.2 The enterprise knowledge: the semantic aspect

Definitions:

- **Semantics** (etymology: "*semanein"*, signify): related to signification, meaning. Equivalent term: conceptual.
- **Semantic aspect**: aspect of the Enterprise System that isolates the knowledge of the objects and fundamental concepts.

Enterprises and organizations are spontaneously seen through their resources and their activities. In the same way, technical systems are, above all, seen as equipment carrying out functions. Praxeme adds a degree of abstraction, inviting us to isolate the fundamental concepts, ignoring organizational and technical circumstances. This is not a natural attitude; it requires effort, distancing oneself from the solutions in place. Through this abstraction effort, the modeler isolates the essentials: the concepts that describe the business, the required objects and their behavior.

When we say that the semantic aspect isolates the knowledge of the business fundamentals, this expression should be understood negatively: the semantic model discards any element that is not essential because it is linked to individual choices, organizational decisions and specific solutions. In this way, semantic modeling aspires to be universal; it is not the least of its contributions. By pushing for abstraction, the model gets rid of the specific elements, locates the essentials and becomes an easily shareable expression. Of course, this abstraction effort turns out to be difficult but extremely productive. Indeed, it enables a core of notions to be isolated whose implementation is enough to achieve the enterprise mission, whatever the organizational and equipment solutions. It encourages us to reexamine these solutions, practices and techniques. It creates the conditions to reinvent the business, to innovate.

Semantic modeling consists in rigorously representing the real objects, the "business objects", the notions that are manipulated in and by the Enterprise System. Here, the unit of representation is the class, a formal representation of the concept (in intension or comprehension: the meaning of the concept; by development: all the instances, that is to say the objects that fall within the concept). These objects are grouped into coherent sets, the object domains. Here, the architect has a different decomposition criterion to the functional approach.

Definitions:

- **Business object**: concrete or abstract objects essential to the mission of the Enterprise System.
- **Object domain**: knowledge area: coherent set of business objects grouped around a small number of main objects and delimited by extended neighborhood.

Complement:

The object domain is the unit of decomposition of the semantic aspect.

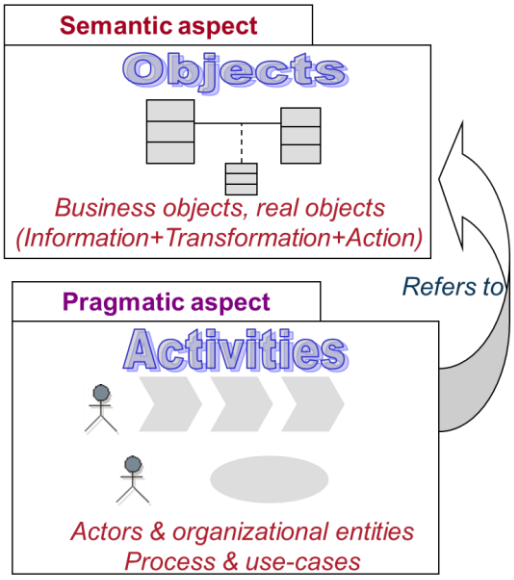
The approach of the semantic aspect is detailed in the guide PxPRD-20.

Summary of the semantic aspect:

Stake: facilitate the understanding of the system; simplify, revisit the business; share and unify knowledge.

Content: knowledge of the enterprise fundamentals, as classes with their relations and their life cycle.

Figure PxMDS-01_17. The business description (first stage of the fundamental sequence)



A business description made only in pragmatic terms would suffer from the effects of a functionalist approach, entailing organizational choices. It would be weighed down by the inevitable redundancy in this approach and would be difficult to share, as it would reflect the current practices of a specific organization.

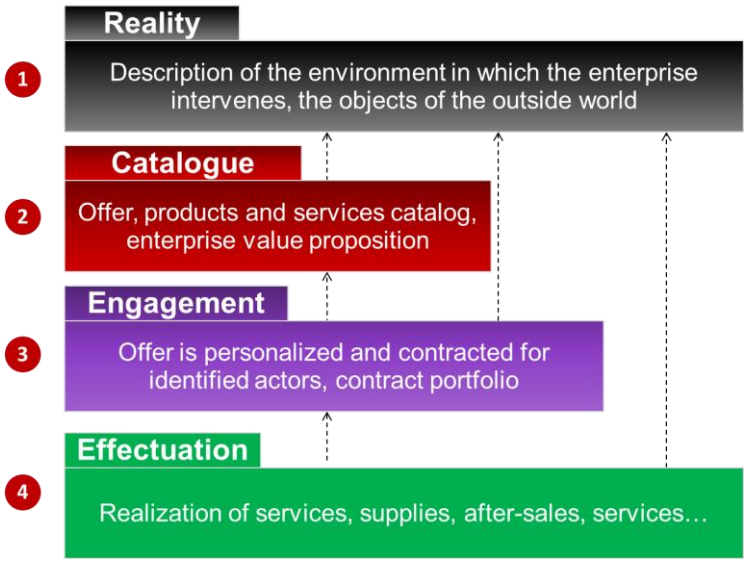
Through an effort of abstraction from existing practices or by deducing the intentions, semantic modeling isolates the fundamental concepts required to carry out the enterprise mission. This model, more compact, states the business essentials, from which it is possible to rethink the practices and solutions.

The pragmatic aspect refers to the semantic aspect, more stable and easier to share. This feature paves the way for the innovative design of processes and organizations.

Praxeme proposes a generic architecture of the semantic aspect, as shown in the next figure. A more detailed presentation is made in the document PxOPS-20.

Figure PxMDS-01_18. The decomposition of the semantic apsect into object domains

The domains are stacked according to the degree of dependency. They are represented here as packages, in the UML notation (Unified Modeling Language; see section 5.7). The diagram also shows the dependencies between the domains. This architecture may seem extremely coupled. It is not a problem in the semantic aspect, which contains the knowledge: the movement between concepts ignores the constraints that are imposed on the other aspects, particularly the technical ones.



4.3 The enterprise activities and organization: the pragmatic aspect

Definitions:

Pragmatic (etymology *"pragma"*: action): applies to the activity. Equivalent term: organizational.

Pragmatic aspect: aspect of the Enterprise System concerned with the activities and way they are conducted.

Under its pragmatic aspect, the enterprise appears as an organization and an activity. This aspect is more easily perceived than the semantic aspect. We are not merely content to describe it: we also have to design it, simplify the processes and exploit the organizational innovations.

The pragmatic aspect is decomposed into activity domains, also known as "functional domains". The functionalist approach remains pertinent in this aspect, even though the methodology challenges it for the other aspects. Indeed, its primacy given to the action or the function perfectly suits the nature of the pragmatic aspect, reserved for the activity of the Enterprise System. The top-down hierarchical decomposition characterizes this approach. Specific additional measures have to be added to decrease the redundancy that it never fails to produce.

Another way of decomposing the pragmatic aspect rests on the inter-functional processes (which cross the divisions of the organization). The design of these processes – in contrast to the representation of the intra-functional processes – is a source of significant changes in the way the enterprise functions.

Definitions:

Activity domain: area of activity; set of activities and the resources required to manage them.

Process: set of scheduled activities.

Complement:

The activity domain is the unit of decomposition of the pragmatic aspect. The criterion of this decomposition is often the function (in the sense of functional direction or important functionality of a system). In this case, the activity domain merges with the functional domain.

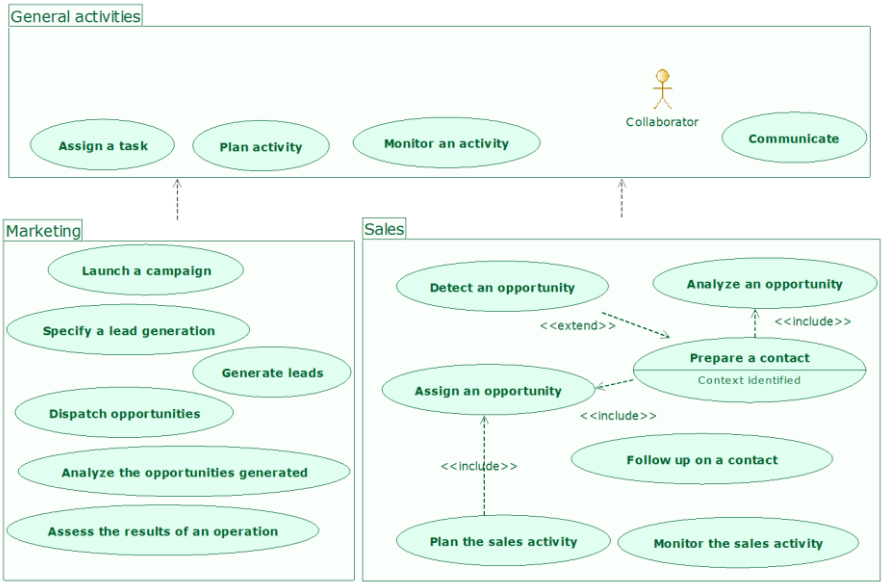
The approach of the pragmatic aspect is detailed in the guide PxPRD-30.

Summary of the pragmatic aspect:

Stake: understand or stipulate how the enterprise functions; optimize the actions; adapt the enterprise.

Content: the organization and related notions (organization rules, procedures, roles, structures, governance, management style…), activities (all levels, from processes to the elementary work situation), objects of an organizational type (files, forms, rights…).

Figure PxMDS-01_19. An example of a diagram in the pragmatic aspect



This diagram shows three activity domains. Two of them (Marketing, Sales) correspond to functions in the enterprise. The other one enables the common or reusable activities to be factorized.

Its presence enables us to greatly reduce the system's redundancy in its pragmatic aspect.

These domains contain activities (represented here by use cases) and roles. They can also contain processes and objects of an organizational type.

4.4 The enterprise locations: the geographic aspect

Definition:

Geographic aspect: location of the activities of the Enterprise System. Enterprise geography.

The pragmatic aspect describes the activities to be carried out. However, they have to be allocated a location. This is not an insignificant act: it leads to consequences not only in financial terms and in terms of human resource management, but also regarding the quality of service. It also involves some elements of intent, for example:

- the operational requirements and quality of service (availability; proximity…);
- the values (work conditions; teleworking; environmental costs…);
- the performances (impact of choice of location on deadlines, quality, stakeholder satisfaction…).

The geographic aspect is described in terms of sites and types of sites, as well as in terms of connections authorizing the (physical and virtual) communications. The design decisions on this aspect can have a strategic reach (examples of relocation, territorial organization or international partnerships). They almost always have a human impact that must be taken into account, in accordance with the enterprise values and those of society in general.

In some circumstances, sites involved in cloud computing solutions must also appear on the geographic model, in particular for legal reasons.

The approach of the geographic aspect is detailed in the guide PxPRD-40.

Summary of the geographic aspect:

Stake: quality and performance of the activities and services; social and environmental impacts.

Content: enterprise geography, full organization chart (with the territorial organization), typology and listing of sites, communication needs, topics such as outsourcing, teleworking and nomadism.

4.5 The design of the technical systems of the enterprise: the logical aspect

Definition:

Logical aspect: description of the logistic system independent of technical choices.

The previous aspects make up what we could call the "business view" of the Enterprise System. From here on, all that remains is to design and set up the equipment that will enable us to carry out the activities which, when taken together and connected, make up the logistic aspect. However, before tackling the detail of the technical choices specific to the logistic aspect, it is important to:

- get an idea of the overall system, its behavior and its quality;
- develop a description of the system, relatively independently of technical choices in order to (1) use it for communication purposes outside of the technical circle and (2) have a description available that can accompany the changes in the system over the long term.

We would not be able to get such a description if we dealt with it on a technical level: on the one hand, the technical detail and vocabulary would prohibit its use in communication with non-technical stakeholders (decision makers, clients, users…); on the other hand, it would be subject to technical obsolescence, even though it has to accompany the transformation over the long term.

"Logical" is therefore the opposite of "technical", "material", "physical". The logical aspect presents itself as an abstraction of the logistic solution (technological, IT). It fits into the transformation chain as an intermediary aspect between the "business" and the technical side. It ensures the decoupling between both terms that live at a different pace.

The logical model has a pivotal role in the design of the Enterprise System, linking the upstream aspects – semantic and pragmatic – to the technical design. The Enterprise System Topology positions it touching the semantic and pragmatic aspects. This position infers a precise approach. It results in changes to the face of the logistic system, around a structure that avoids the shortcomings of the functional approach.

Figure PxMDS-01_20. The position of the logical aspect (second stage of the fundamental sequence)

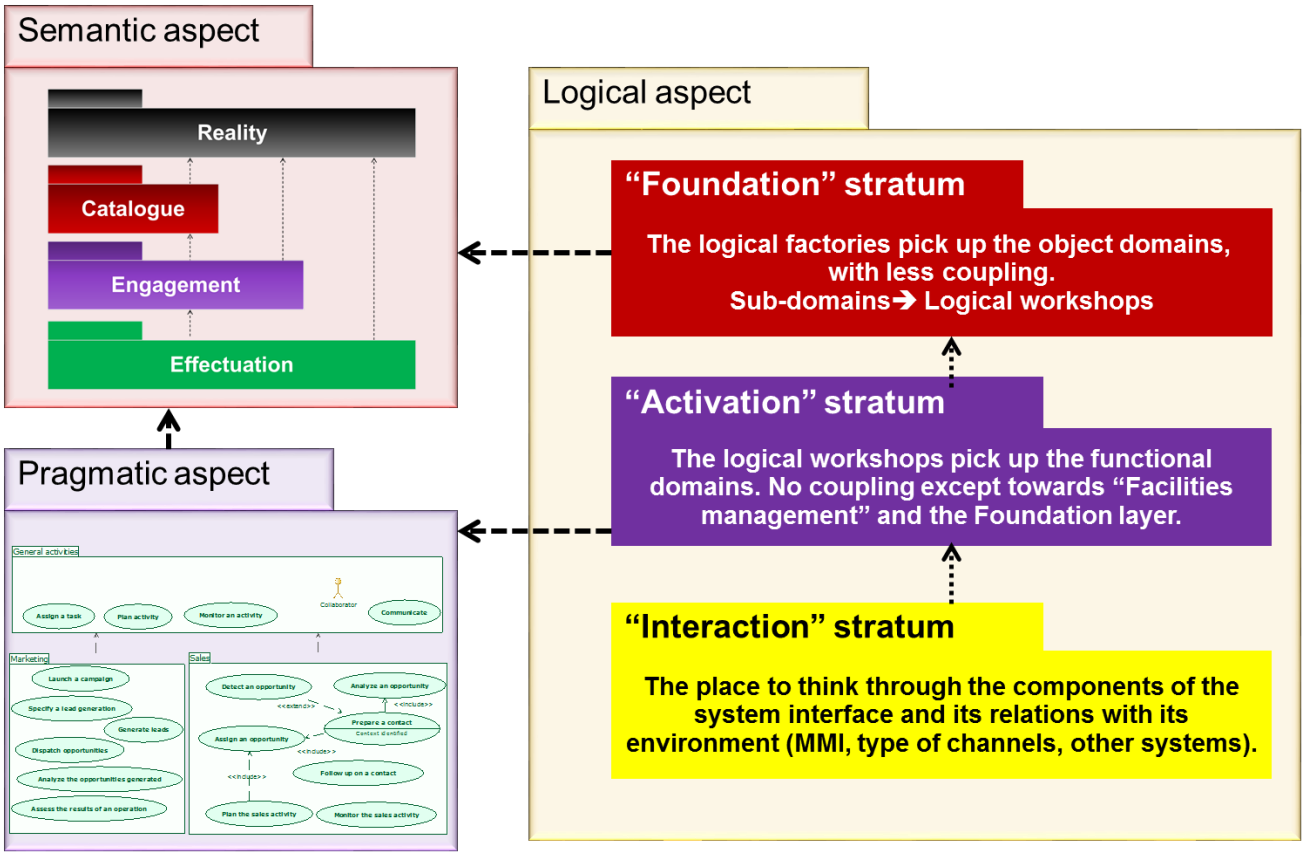
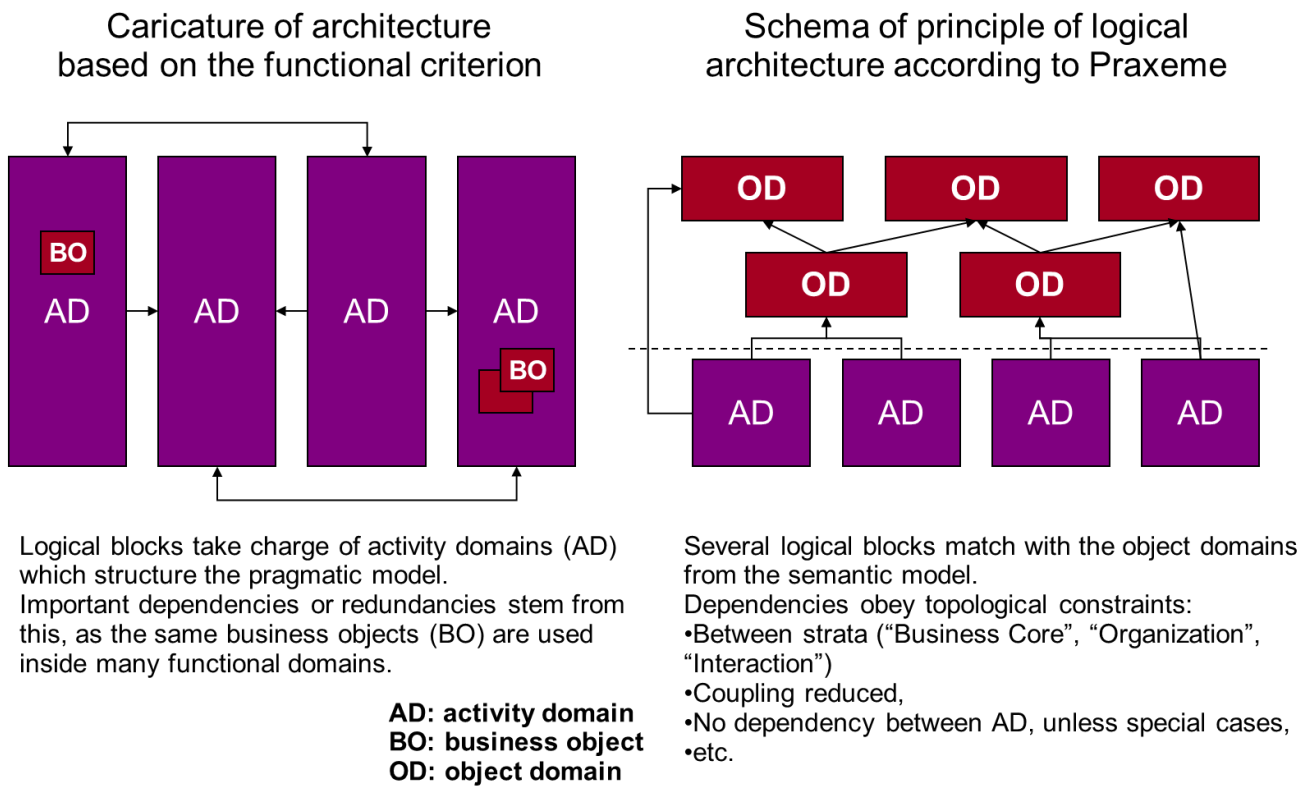


Figure 20 above reminds us of the position that the Enterprise System Topology gives to the logical aspect. It also illustrates the layering principle: the logical aspect is decomposed into strata, defined by the method, in order to keep the separation carried out upstream, between the semantics and pragmatics, within the technical system. The units of decomposition from these last two aspects are reused in the logical architecture, leading to a structurally profound transformation.

Figure PxMDS-01_21. The changing face of the Enterprise System (third stage of the fundamental sequence)



Essentially, the change lies in a very simple decision: isolate the "business" objects in well-identified parts of the system. The core of the system must be structured not in activity domains (or functional domains), but in object domains. The substance isolated this way is largely reusable. In the figure above, the diagram on the left caricaturizes the architecture of the majority of existing systems, marked by the functional approach, which leads to organizational silos, with everything that that implies in terms of redundancy and coupling. The diagram on the right shows the typical structure of logical architecture according to Praxeme: we can see an application core made up of services that have been deduced from the semantic model. The blocks mirroring the activity domains still exist, but they have been emptied of part of their substance as they call on the services shared in the core and organized according to the object domain criterion. Certain topological constraints improve the architecture: for example, we can see on this diagram that the communication is only made up of AD components towards core OD components and that there is no lateral communication between the AD components.

The approach of the logical aspect is detailed in the guide PxPRD-50. This is where the SOA (Service Oriented Architecture) style is discussed.

Summary of the logical aspect:

Stake: quality and control of the logistic system (non-redundancy, reuse, low coupling, control of development…); ability to rethink the technical system from top to bottom.

Content: style of logical design, metaphor, architectural rules, description of the logistic system apart from technical choices.

Principles:

Stratification: the logical architecture keeps the separation between the semantic and pragmatic aspects.

Derivation: the content of the logical aspect is derived from upstream aspects, in the main.

4.6 The technical systems serving the enterprise: the logistic aspect

Definition:

Logistic aspect: set of means (hardware and software; production, communication, processing, transport…) serving the enterprise activity.

The means that enable us to carry out the activities, and potentially automate them, are designed and realized from the logical model. All these means make up the logistic system. We have to:

- on the one hand, select, design, realize and link the hardware solutions (vehicles, computers, sensors, effectors and all sorts of equipment) that fit into the infrastructure of the Enterprise System;
- on the other hand, select, design, realize and link the software components that will be installed on this hardware to ensure its functioning and coordination.

Information technology comes in here and, in a general way, technology. The work consists mainly in expressing the logical specification into the selected technical terms.

The approach of the logistic aspect is detailed in the guide PxPRD-60.

Summary of the logistic aspect:

Stake: control and optimization of the production means.

Content: technical choices, technical architecture (set of technical choices and their justifications, completed by the rules of use and derivation rules from the logical aspect), types of equipment, software components.

4.7 The enterprise resources: the physical aspect

Definition:

Physical aspect: aspect under which the Enterprise System appears as deployed; physical reality of the enterprise.

On the one hand, the logistic aspect brings together the *types* of hardware and software resources. On the other hand, the geographic aspect lists the *sites* and the spatial relations (including the virtual ones) between these sites. What remains to be done is to instantiate the means and locate them on a site-by-site basis. This is the end point of the chain: the physical model describes the enterprise reality deployed in space and equipped.

The passage from the logistic aspect to the physical aspect requires a twofold process:

- first, instantiate the types of hardware and software resources;
- then, locate them in appropriate sites and install them.

The physical aspect is therefore one of deployment.

The approach of the physical aspect is detailed in the guide PxPRD-70.

Summary of the physical aspect:

Stake: know the detail of the resources in order to carry out the follow up.

Content: the individualized and located equipment; the configurations; the states.

The physical aspect marks the end of the transformation chain.

5. The description of the Enterprise System

This chapter introduces some notions and precepts that guide modeling practices:

- the continuity principle,
- the relations between models, the traceability,
- the central mechanism of the "enterprise description repository",
- modeling requirements,
- notations.

5.1 Modeling

The philosophy presented in section 1.4 – especially the rationality principle – instills in our practices a state of mind of rigor and the concern of having a detailed understanding of the Enterprise System. This requirement naturally drives modeling practices, which aim to rigorously describe things, in an effort to fully comprehend everything.

Definition:

Model: formal representation of a part of the reality.

See the definition of "formalism", p. 7.

That does not necessarily mean that we do not confuse the model and the reality. The passion for modeling is tempered by the epistemological humility. We know that a complete representation is not possible and we must be wary of the illusion of having said everything there is to say and knowing everything there is to know. Praxeme already helps us by distinguishing different approaches and by accepting several modeling techniques. It means we avoid the trap of monomania, whereby we apply a single tool to deal with all aspects of the reality. Once the aspects of the Enterprise System have been identified, the methodology considers the most appropriate approaches and proposes the ad hoc tools[21]. These approaches are the subject of the methodological guides PxPRD-10 to PxPRD-70.

A model, no matter which one, will never be able to cover the totality of an aspect. It would be a Herculean job. In general, the model is developed in a specific context, dependent on an aim, often within the scope of a project. The criterion to counterbalance the modeling effort is therefore that of pertinence. It is simply not materially possible to describe the whole enterprise reality fully and with an extreme level of detail. We will only do it as a need, remaining fully aware of what escapes us.

5.2 The continuity principle: models and architectures

A model is developed with a precise intention, often in the context of a project. It thus concerns one part of the Enterprise System, delimited by an immediate need. Nevertheless, this pragmatic attitude must not lead us to abandon the overall quality of the system. To aim for this quality leads to consequences and requirements on each individual model. This ambition results in the primacy of architecture over modeling and, in consequence, over the execution.

[21] This shortcoming can be seen today with the process approach, hailed as the unique and universal panacea, whereas: (1) it leaves whole chunks of the enterprise on one side, starting with its knowledge capital; (2) the process drawing merely scratches the surface of the reality of the activity, steeped in diverging motivations, of games played by the actors who can, at any moment, make the whole workings derail. We reassure ourselves with our representations; they verge on rituals and illusions. Whatever we do, let's not be deceived! These representations are nothing more than the shadows on the cave walls. The formal approach puts aside the depth of human experience and the torments of life in society. Yet these factors condition to a large extent the behavior of the Enterprise System.

Principle:

Primacy of the architecture: any design effort is included in a blueprint, developed beforehand, on the system scale.

This principle applies to the design: it restricts the modeling activity within the scope of one design, that is to say the development of a new solution, whatever the aspect in question.

The architecture as a product (architecture schema, architecture document…) is therefore a prerequisite to modeling. In the field, it results in specific decisions made regarding the location of the modeling elements, to the advantage of the reuse and quality of the system. This topic is linked to the dual scope theme (see section 6.4, p. 44), as well as the coring procedure (data sheet PxPCS-01c). Its consequences affect the dynamics of the Enterprise System transformation.

The dependency between architecture and modeling forces us to keep perfect continuity between both disciplines, as well as between their deliverables. One practical consequence concerns their means of representation: they must be identical or, at the very least, compatible. The architecture diagram must be understood as both the plan anticipating the detail of the models and as the summary of the topic fully described by all the models. The architect has the duty of anticipating the construction detail. In return, the modeler is obliged to include his/her activity in the architect's vision. They have to share the same ambition, the same requirements and the same representations. This point springs up again regarding the choice of notations, discussed further on.

5.3 The relations between the models

The models maintain relations that have to be clarified. The job of clarification takes place first at the relations-between-modeling-element level.

Inside each aspect, the elements are linked to each other by relations depending on their categories and described by the metamodel. Added to which are the relations between packages regarding the reuse of generic or shared elements.

Between the aspects, we can distinguish three types of relations. They are described in the following table.

Figure PxMDS-01_22. Typology of the inter-aspect traceability links

Type of relation	Definition	Illustration
Projection	An upstream element is expressed in a downstream element that formalizes it.	A performance indicator of the intentional aspect (upstream) is projected into an attribute of the semantic or pragmatic aspect.
Justification	A downstream element participates in the response given to an upstream element.	For example, an objective (upstream) justifies the organizational or equipment measures. An operational requirement motivates an infrastructure choice.
Derivation	A downstream element results from the mechanical product from an upstream element.	A logical component derived from a semantic class or a use case, through the application of derivation rules defined for the SOA style.

In all cases, the relation between the elements is kept as a traceability link.

Derivation plays a key role in the Praxeme method. It provides a powerful accelerator in the transformation chain that covers all the aspects, and a guarantee that the technical system will be aligned with business needs. This approach complies with the MDA (Model Driven Architecture) standard published by the OMG (Object Management Group).

5.4 The enterprise description repository

Definitions:

Repository: set of elements shared by a community.

Enterprise description repository (or system description repository): repository containing all the elements collected during the work to describe the Enterprise System and structured in accordance with the chosen architecture.

NB: the term "repository" applies to all the dimensions of the Pro^3 schema (repositories of practices, for example). In the case of the enterprise description repository, it belongs to the Product dimension.

The models serve the investments that have a short- or medium-term objective. The continuity principle aims to increase their effects by including them in a more extensive effort to build the system in the long term. This process relies on the enterprise description repository, the central mechanism whose structure mirrors the architecture decisions and into which the models flow, project after project.

This repository plays a pivotal role in the transformation. It materializes the will to administrate the investments at the enterprise level and to achieve the architecture target. From this mechanism, we will have to review the interactions between transformational activities, as mentioned in the previous two chapters.

5.5 The architecture documentation

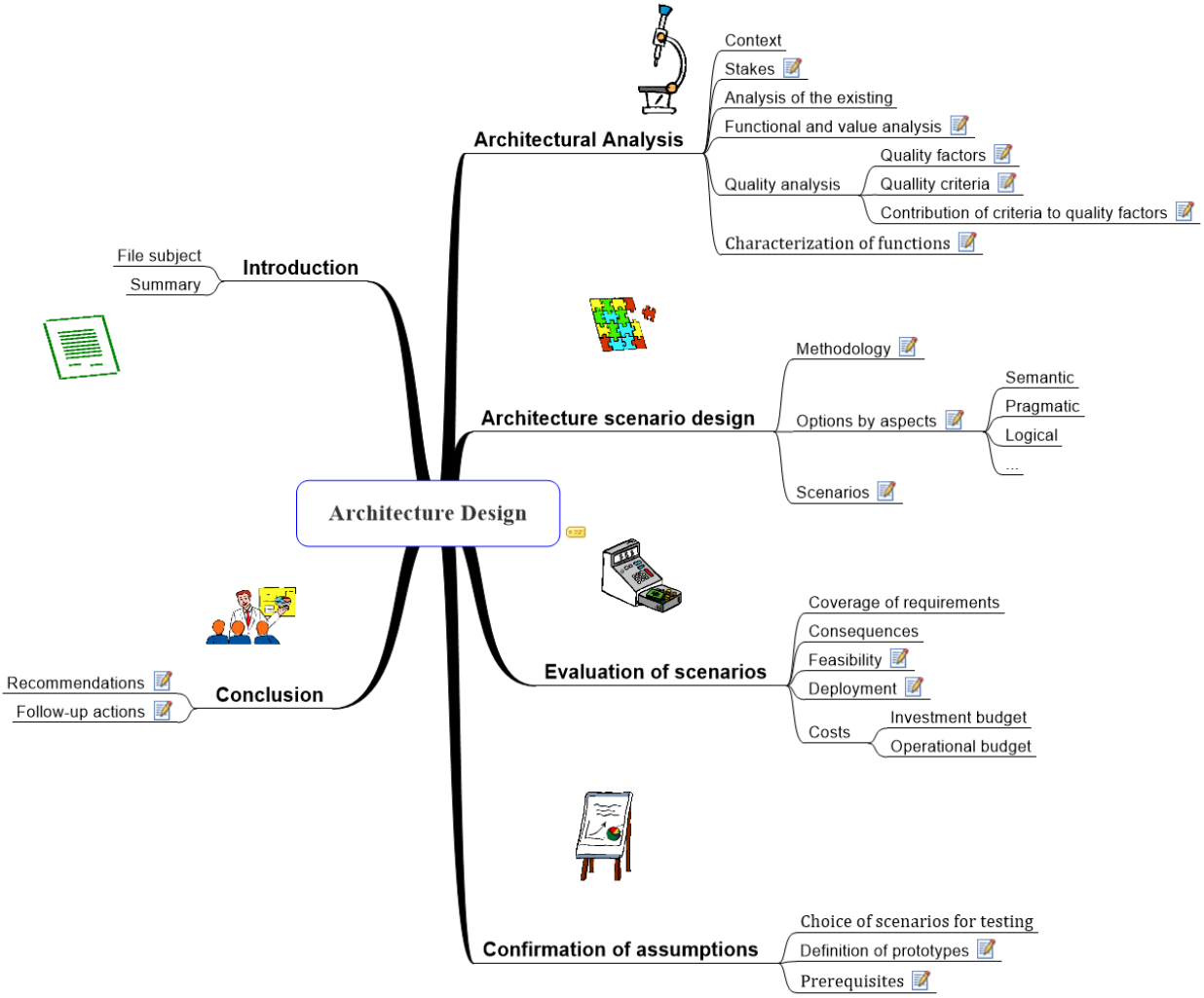
Architecture is a structural affair. It begins with the choice of framework, for example the Enterprise System Topology. This framework becomes the first-level structure of the enterprise description repository. Then, the enterprise architecture decides on the first levels of decomposition of each of the seven aspects, by supporting them with the relevant arguments.

This structuring prefigures the future Enterprise System. At this level, the architecture decisions, if they have to be carried out, have a considerable reach. One error, one weakness in the architecture could lead to consequences whose financial impact, although often diffuse and hidden, may add up to millions of euros.

It is therefore of paramount importance to weigh up each decision, to consider several scenarios and to document the choices that have been settled on. The figure below proposes a typical summary for architecture files. Its logic unrolls as follows:

1. The first step is to analyze the transformation context.
2. Then, aspect by aspect, the architect considers different options.
3. These options combine to form scenarios that must be evaluated in terms of costs and benefits.
4. The final choice of the transformation scenario falls to the decision maker, based on the evaluation elements established in all the dimensions of the Enterprise System.
5. When the scenarios have been established and evaluated, sometimes even after the decision has been made, there may be some areas of uncertainty regarding the validity or interest of the transformation options. In this case, the architecture file includes measures for verifying the assumptions made. The transformation trajectory can thus be rectified earlier on, if the assumptions cannot be confirmed.

Figure PxMDS-01_23. Typical summary for an architecture file



5.6 The requirements concerning the models

The requirements concerning the modeling products (including architectures) can be classified into two categories:

- requirements which target the quality of the modeling;
- those that serve the integration of the models into the enterprise description repository.

Among the first category, we can find:

- the required level of detail (it must be sufficient to allow the transition to the next stage in the transformation chain, according to the process adopted);
- the documentation (a model is not restricted to a set of diagrams, nor to a set of modeling elements; it has to contain enough comments so that these diagrams and elements are perfectly understandable, now and in the future; it is also important that any future intervening parties understand the reasons that guided the modeling decisions);
- the means of verification.

Every model must contain its own proof. This amounts to including the test cases in the models. This precept has an obvious economic justification. By asking the modeler to design specific test cases for the aspect in

question, it forces him/her to reread the model and to run it in theory. This measure significantly improves the quality of the models. In addition, who is better placed than the modeler of a given aspect to have the knowledge required to design the proof? It is an organizational and economic aberration to push this exercise down the chain, for example to the software developer, who is then expected to know the business side, on top of the technical side.

This precept applies to all aspects. Even an abstract model, like the semantic model, justifies such rigor.

The requirement concerning the level of formalization can be pushed even further with executable models. This is one ideal that is not justified in all contexts.

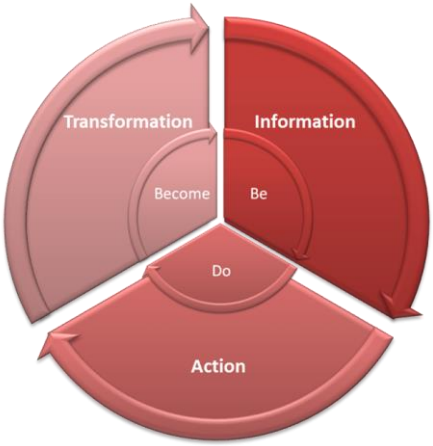


Figure PxMDS-01_24. The threefold approach to modeling

The requirement of correctly apprehending the reality leads to the triangulation principle that structures all modeling approaches. Many methods compare a static model with a dynamic model, thus putting to one side the main source of our difficulties: change. Praxeme substitutes this binary paradigm with a threefold one: information, action and transformation. It refers to the three ways of seeing the world, through the *to be*, the *to do* and the *to become*. The triangulation principle is embodied in structural modeling, functional modeling and contractual modeling. The methodological guides define their content, aspect by aspect.

5.7 The notation and tooling

Thinking about the modeling requirements raises the subject of notation. Notation brings a level of formalism and facilities of expression that support the modeling effort. What is important in the notation is not the graphical key, but the set of categories of representation and use constraints that are associated with it (the syntax), or in a nutshell: the metamodel.

The thinking that determines the choice of notation behind the Praxeme practices is the following:

- The first requirement is to assimilate all the representations in the enterprise description repository.
- It is obvious that, wherever possible, we should favor the notations that are recognized as international standards. On the one hand, this represents a guarantee of quality as the standards have been designed and tested by a wide community; on the other hand, turning to international standards quickens the consensus and adoption within groups and between partners. A third reason is an economic one: tooling standard notations is, in general, less expensive as it is amortized on a wider market.
- Incorporating the representations in the repository alone is not enough. We also have to be able to link the modeling elements with each other, including between the different aspects (traceability requirement discussed p. 37).
- In some cases, we must be able to apply derivation rules to produce new elements, for example from the software code (MDA approach, mentioned p. 37).
- The MDA approach is quite naturally tooled with the UML (Unified Modeling Language) notation and UML profiles, both easily accessible international standards.

In conclusion, the most commonly used notation in the implementation of Praxeme is UML. It enables us to cover all the aspects, especially if it is completed with UML profiles. The latter provide the means of adapting the notation to the method. They allow the notation to be enhanced by new notions, associated with syntactic or productive rules.

This choice is absolutely a pragmatic one. First, UML is a standard, established in 1997 and which has continued to be consolidated ever since. Its metamodel is the richest that is known today and it has benefitted from contributions from the best methodologists, coming from diverse domains. Numerous modeling tools exist, including open source ones. Moreover, this toolbox is suitable for uses that concern all aspects of the Enterprise System. Of course, being general, the toolbox is a little on the heavy side! Yet, it does not come with

an instruction manual. The method provides precisely this instruction manual; it defines the uses of the diagrams established in the standard. A same diagram can be used to represent different things in several aspects.

There are some biases that still need to be defused. As much as it seems impossible to ignore UML in the IT domain, this choice may cause surprise in "business" aspects, even abstract ones. For example, how could this notation represent knowledge throughout the semantic model? A simple reminder from history is enough to clear up the misunderstanding. It is true that UML emerged in the IT community in response to its needs for representation. The notation tools the object-oriented approach, associated with IT programming languages. The central notion is the class. Yet, if this term today evokes a software component (e.g., the Java class), the designers of these languages borrowed it, quite deliberately, from knowledge philosophy and logic. Their aim was to bring the machine closer to the natural representation made by the human mind. It is therefore no surprise that the notation is perfectly suited to knowledge representation, the notion of class merging with that of concept.

Although a large place is given to UML in the Praxeme practice, it is not an exclusive choice. Apart from UML, we cannot today find a standard notation that, with some extensions, covers the entire enterprise description repository. However, more specialized notations can be of great use. This is the case of BPMN (Business Process Model and Notation) for example, a notation that only represents processes, but which does so in an extremely efficient way. The requirements set out above must be verified however: the process models should be included in the enterprise description repository and be connected to other elements. The guide "Approach of the pragmatic aspect" (ref. PxPRD-30) specifies the articulation with the use cases and the semantic model.

Other notations are used, for example: objective diagram, performance tree of the metrological model, terminological diagram (thesaurus)… Every time, the initial selection criterion of the notation and the tool is the integration into the enterprise description repository.

Figure PxMDS-01_25. Notation requirements according to the aspects

Aspect	Possible notations	Requirements
Intentional	No standard	The representations of the 4 models (see p. 27) are not complicated. They are limited to a connection between the elements, with qualified links. One major criterion is the ability to establish the traceability towards the other aspects.
Semantic	UML	The semantic model is where the standard must be respected the most scrupulously, with possibilities of expression like n-ary associations, qualified associations, properties of class scope and, of course, state machines.
Pragmatic	UML, BPMN	BPMN has greatly improved process design by providing very elegant means to deal with disruptions. The requirements: connections with UML classes and use cases needed for derivation toward the logical model.
Geographic	UML, representations	It is possible to define the sites and types of sites as "nodes" in the UML sense. For the representation, nothing is better than a geographic map.
Logical	UML, SysML	UML provides component, interface and package notions that we can potentially reinterpret according to the selected style of architecture. The logical elements refer to the elements from previous aspects.
Logistic	UML	UML is perfectly suited to the IT part (types of hardware and software components). We can add stereotypes to it to deal with other equipment.
Physical	UML	The deployment diagram comes into its own here.

Table of figures

Index

R

S

T

U

V