

Preface

Today, dealing with information systems appears to be very complex because the term, by now, is used in order to indicate any application capable of solving any problem in a specific business area.

Over the years, we have used this term in a very general way, often with divergent meanings between them.

Moreover, several and different design methodologies have been used, from the functional decomposition to those relative to the big framework, as those realized by SAP® (that put together different specific applications with the goal—above all marketing goals—to supply integrated information systems simply choosing some configuration parameters) and, besides these, several and different philosophies of design like ERP, MRP, CRM, and so forth.

Moreover, the advent and the extraordinary spread of the Internet and, therefore, the expansion of information systems on the Web, have generated a remarkable increment of methodologies and technologies for the design and the development of applications for the Web, forcing to review well-known methodologies, such as the object-oriented, with the result to make the effective use of UML® (Rumbaugh, Jacobson, & Booch, 2004) for the design of information systems extremely complex.

In general terms, we can assert that any methodology, to be truly effective, must allow the designer to concentrate him/herself on the problem of solving rather than trying to understand the use of the same methodology in order to express the specific problem.

It appears, therefore, necessary to try to make order in this chaotic universe of acronyms, philosophies, and methodologies that often approach similar problems and offer similar solutions, even if they address innovation and oneness, supplying a clear point of reference to which we will relate within this book.

INFORMATION SYSTEMS

The term information systems means “the set of the procedures and the infrastructures that support and describe the flowing of the information inside an organizational structure” (Pighin, 2005).

In other words, an information system allows describes, in detail, as any company (public or private) carries out the job to which it is deputy and to which the material and immaterial resources are available.

Such definition does not take account, and it must not take account, of the use of an eventual automation related to the use of computer-based systems.

For an information system to become “automated,” partially or completely, means to design and to develop a computer-based system that supports and implements how much described previously; in other words, the application must adapt itself completely to the way the company works, and not vice versa.

Very often, today, many software houses try to impose on the companies the way of working already implemented within their applications, prescindendo from the peculiarities of the companies, their history and the resources available.

Of course, to develop an ad hoc application requires remarkable economic investments and it is not always affordable for the average and small productive realities and, therefore, often it is indispensable to balance these two divergent requirements.

The goal of the book and of the methodology that is described inside is to realize a compromise between the economic effort and the necessity, for the companies, to differ from other concurrent economic realities.

Information Systems: From Business Process Design to the Business Process Reengineering

In the 1990, Hammer (1990) published his theory on the necessity of reengineering the business processes starting from the observation of the reality and from the experience that many computer experts had experimented in their professional life.

He asserted that the introduction of innovative technology on old business processes had the highest probabilities of failure.

Such affirmation, however obvious, had the merit to raise the problem of the understanding of how the company carried out its own job and with which resources before proposing innovation.

This is, in our opinion, the guideline that would have to always drive the proposal of innovation to a company, but as pointed out previously, it is not how many things happens in the majority of the cases.

The theory of the BPR—business process reengineering—(Hammer & Stanton, 1995), in brief, is made up of three main phases:

1. *AS-IS*: the definition of present BPs (business processes). This is supposed to be the fundamental phase of the analysis because it allows identification of the primary processes (or macro-processes) that they describe as the company works currently.
2. *Comparison and diagnosis*: it is provided for a quantitative and qualitative evaluation of what is found, using well-known techniques of benchmarking, and eventually a comparison with companies of the same typology.
3. *TO-BE*: re-definition of BPs according to the requirements of a company.

There are least four variables to be considered in each phase, in particular:

- **The flow of BPs**: the decomposition of the processes in activity flows
- **The organizational structure**: the contribution and the responsibility of the productive units to BP
- **The human resources**: the characteristics, skills and the availabilities to the change
- **Systems of values (motivation and boosting)**: a system that aims both to reduce the opportunistic individual factor and to measure the productivity

Globally considering these variables, there are three strategies of reengineering that have substantially been asserted:

1. **Buy-side strategy**: related to the part of the information system regarding suppliers of raw materials and/or services (B2B, e-procurement, etc.)
2. **In-side strategy**: related to the part of the information system inside the company (ERP)
3. **Sell-side strategy**: related to the part of the information system that involves the final users (B2C, etc.)

Any design of reengineering always begins with the necessity of a change and the definition of a new business vision.

Therefore, in the first phase (AS-IS), it is indispensable to understand how the company currently operates, by which organizational structure and the specific level of responsibility within the activities, with which kind of human resources (skills, availability to the change, etc.), and which system of values defining both the individual and structure goals and the system of incentive in order to reduce most possible what in literature it is called “opportunistic individual behavior” (Williamson, 1975).

This phase is, in our opinion, the most difficult and most expensive from the economic and employed resource point of view, but if well executed, it concurs to carry out BPR with success.

In fact, starting from the detailed analysis of the present situation and using opportune tools of simulation, it is possible to carry out a first rationalization of the organizational structure and the flows of business processes.

Such affirmation, justified from the experience of a quarter of a century in the field of the high-level information systems of one of the authors, is based on the fact that often the operating procedures and the information exchange inside the companies is the result of a series of adjustments that they have contributed to create, in the course of time, a superstructure not justified and often unknown to the company's people.

AS-IS analysis would have to identify a few primary processes for being able to effectively manage a reengineering of the same ones and, in particular, we think that for a big company, approximately 15 processes would have to be characterized.

Both the primary processes and the support process (or secondary processes) are represented using the value chain of Porter (Millar & Porter, 1985) identifying the processes that create value for the company according to their mission.

The first activity, therefore, concerns the decomposition of these processes in a harmonic and reasoned flow of tasks.

For many years, from the introduction of the concept of business process, it has been a complete anarchy in the use of an effective notation for the graphical representation of the flow of the activities regarding the first variable into play. Some big software houses have developed their own method (for instance, FILENET™, now an IBM® subsidiary) while others have used standard methods like IDEF0 (IEEE, 1998).

The fundamental characteristic of these models of representation of the processes is that they are not user-oriented, that is, they are particularly complex to use and too cryptic so that the final user could understand them.

Finally, in 2004, the more important companies in this sector have gathered in a consortium to aim to develop a common and effective notation to be used to model the BPs in an understandable way for the final customer.

Such notation, BPMN™—business process management notation—which is the base of this book and will be shortly illustrated later on, (OMG, 2006) is now maintained from the OMG™.

The representation of the processes through BPMN uses, for its formal representation, specific machine-readable languages, such as the initial BPEL and the actual XPD, that the companies of the sector use in order to develop all of the support tools that are not free of charge.

For many years, our research group has used OWL (Web ontology language) (W3C, 2004) as the main language for the modeling and the definition of meta-model because it is considered more suitable to represent the concepts and the base rules understanding a methodology. The choice to use the ontologies in the field of software engineering has been confirmed by a recent paper published by IEEE (Cardoso, 2007).

The use of ontologies, that will be discussed, allow us homogeneity of modeling being a platform/company-independent.

The second phase, defined as comparison and diagnosis, provides a first activity of appraisal using the techniques of benchmarking, certainly complex and articulated, but basically determinist.

Instead, the comparison with other similar companies is different. In literature, often it refers to such a comparison with the term of “best practice” that means some success cases of companies that operate in the same segment of market with equal dimensions: in other words, concurrent companies.

It is obvious that the companies cannot have approached, legally, sensitive data of concurrent companies in order not to incur in the crime of industrial espionage.

It is equally obvious that the big consultant companies, thanks to the elevated number of customers, could have such information, but they would not have to use them for the same reason expressed previously.

The term “best practice” comes instead from wide use in the business field but with obvious scope to propose to the customer something pre-assembled.

According to our opinion, the true comparison would have to instead be carried out between how much is found using benchmarking and the goals of the company according to the vision that has generated the change necessity.

Finally, the third phase, called TO-BE, consist of redesigning the way by which the company must work in order to aim its goals.

The change, of course, will include all of the involved variables described previously and the result will be absolutely integrated.

It is interesting to notice that, until now, we have spoken exclusively about information systems and not about applications by agreement with the separation between the two systems, information systems and computer-based systems, described previously.

During the last few years, the business process management (BPM), that “is a field of knowledge at the intersection between management and information technology has been asserted, encompassing methods, techniques and tools to design, enact, control, and analyze operational BPs involving humans, organizations, applications, documents and other sources of information” (Van der Aalst, 2003).

This mixture between information systems and computer-based systems was already present *de facto* on the market for many years.

In fact, when the big consultant companies with their associated software companies propose the notorious “best practices,” they certainly sell effective software applications, but these applications are pre-manufactured and customizable through an opportune set of parameters, so it is possible that these applications are not the best possible solution for that specific company.

In short, the consultant proposes to the company a solution to the problems that have driven the change, but this solution does not take into account some of their own peculiarities.

Some of the more important companies of the sector pursue this type of approach, which could be useful for some companies of small and average dimensions, but the economic effort required is often times unsuitable for these kinds of companies.

The market is currently global and too competitive, so there are various reasons that induce change. Moreover, the change necessities take part very frequently (in some cases, also many times in the same year), so the information system must be flexible for being able to adapt itself easily to the new reality without the necessity of dramatic reengineering.

In the same way, the support computer-based system must be equally flexible for effectively being able to adapt itself to the new requirements.

The philosophy of the BPM proposes exceeding the logic of the BPR, just for the ability to implement flexible systems in order to follow the continuous evolution of BP; it considers as the main software instrument the workflow engines that have the job of automating the repetitive activities above all (the operating job) usually oriented inside the company, putting into effect the transformation of the processes according to the inside strategy that generated the ERP.

In this book, we will not deal with ERP, because however many software companies currently continue to advertise their products like ERP, practically having unique DBMS and allowing the sharing of the information between all the business software systems is a norm for good design.

Therefore, the BPM is more modern, more flexible and a softer approach of the BPR. However, is it true?

No, it is not in our opinion. In fact, very often before would it be opportune to carry out, where necessary, a radical transformation using the BPR and successively to follow the continuous evolutions through the techniques of the BPM.

In fact, using directly BPM in order to model BPs in the point of view of their automation, it is the high risk to fall back in the error described by Hammer: to graft computer-based innovation on an inadequate organizational structure represents a high probability of failure.

The failure risk increases considering which professional figure should model the BPs.

From the point of view of automating the execution of this flow of activities, it seems natural to design such a flow using the necessary sagacity to the implementation and, therefore, the key figure would have to be an expert of design of applications.

Such a choice does not seem adequate to us. In the methodology proposed in the book, the first hypothesis regards just the knowledge that a cultural gap exists between the specialists of business and organization and the specialists of software applications.

Web Application: The Concept of User Experience

Today, Web applications (WA) are widely dealt with, but everyone interprets this definition in their own way.

We think, therefore, it is opportune to clarify for the reader what we mean by Web application.

First, it is necessary to clarify what we certainly do not mean by such a term, and that unfortunately, many specialists of the field adopt instead.

The WA is not a traditional application that uses the browser as interface.

The WA is, instead, a tightened marriage between the necessity to carry out operations, being an active part of a business process, with the typical usability and the navigation of a static Web site.

The greater causes of failure in the area of Web applications are determined from the inability of the visitor to find the information that he/she needs and to surf between these.

For this reason, some methodologies of Web application design have been developed in the last years that will be further illustrated in this book.

At the dawn of the Web, also with the knowledge of the concept of hypertext, that is, a semantic relation between information, many have undertaken the development of such applications without using a design methodology; they discovered absolutely unusable WA in which the visitor “got lost” and that the user desperately abandoned the application never to return.

When it became apparent that it is necessary to use this type of application for design methodology also, many turned to the only noted and widely used UML®.

Many practitioners continue to think about this methodology without taking into some account the fact that it has been created in order to model system-oriented applications, and they continue to ignore the requirements of usability of the user. UML® has evolved in order to try to model anything, without losing its own original sin represented by the base rules of the Object-Orientation, thus becoming potentially a methodology able to model anything (also BPs) but, de facto, so complicated to be unusable even for the same specialists of applications.

The user of a WA needs, instead, “to feel at ease” in a world to which it belongs, and that it agrees before still to find the information which it needs.

Modeling the user experience means modeling the interaction of the user with the application by agreement with its requirements and not modeling the application according to the requirements of the system.

The approach used in this book aims to model the interaction of the user with the WA using the paradigm of the “dialogue,” considering both the peculiarities of the user (multi-user) and those of the device that it is using (multi-device).

Moreover, the user of a WA uses information not closely tied to the data, but that instead is related to the marketing and to the creation of a world able to attract a new user.

The approach used for modeling the WAs is based on three levels: conceptual level, logical level and page level. The details of every level will be illustrated more ahead; in this moment, we only notice that the more advanced point of the modeling, mainly close the implementation (page level), loses a few semantic to advantage details to the necessary developer.

The attention of the specialists, therefore, is moving more and more towards the application domain rather than towards the classic design of applications (OO).

This area of research, confirmed from the result of the workshop on the DSM (domain-specific modeling) in the conferences OOPSLA 2006 and 2007, does not consider the effective use of standard methodologies in order to model anything in any application domain.

The main reason is the limitations that these methodologies of course introduce and the key concepts of an application domain “make up” that they force to express that make it lose clarity and acquaintance (Paiano, 2006; 2007).

During a conference in 2006, IBM® tried to illustrate the use of UML® from the point of view of the DSM, and the result was a sort of old-style functional decomposition, brought up-to-date with some of the typical terms of the OO.

In any case, one of the key concepts of this area of research is represented by the automatic generation of the final application starting from the model.

We almost perfectly agree with this tendency.

The domain we consider is the Web, and we model the WAs using a methodology that is absolutely suitable for the Web in order to model applications in different application domains with the awareness that, if will be further clarity and representativeness requirements, it is possible to modify the below meta-model thanks to the use of ontology that allow us to express very complex concepts with the needed semantic and clarity.

Finally, starting from the model of the application, we automatically generate the final WA using two different open source frameworks.

Complex Web Information Systems: The Challenge

At this point, a question cannot be deferred anymore: What is a complex Web information system?

A complex Web information system is an information system, usable via the Web, which includes the way a company performs its activities inside, the BPs. This system must be usable and must be arranged to easily follow the changes of the activities of the company according to the vision of a company as an “open system” (Galbraith, 1973), adopting, therefore, a contingent perspective about its position in the business world. This vision, after 35 years, is not only actual but unavoidable in the global market of the third millennium.

As the reader can see, the answer appears to be quite simple, however, to aim the goal of designing a “contingent” information system capable to effectively adapt itself to the continuous changes forced by the solicitations coming from the outside of the company, a complex work of analysis and integration of several methodologies is needed.

Therefore, the **challenges** of this book are:

- To design information systems approaching two main problems: on one hand those relative to the development of Web applications and on the other hand to the design and integration, inside of the information system, of the business processes that, although their importance and unquestioned usefulness, they found it hard to enter in a pervasive way in the design and the development of the Web information systems.
- To implement the information systems through the automatic code generation tools that, starting from the design model in a machine readable format, help the IT expert to obtain the Web information system very close to the design and without the little personal choices that are very often dangerous.

The methodology that is the foundations of this book has the goal of proposing a solution to several problems related to the development of information systems usable via the Web.

Regarding the BPs, as argued previously, it is thought to be fundamental to start from a deepened analysis of the actual situation without considering eventual automations.

Moreover, being that this activity is much more complex and embraces every aspect of the business life, the experience and the skill of the analysts who, apart from some cases ascribable to small business reality, are experts of organizations and models of business and are not expert of development of applications, are decisive.

This clear separation of the jobs represents our first hypothesis.

In fact, currently we can identify two situations.

The first represents how much happens using the techniques of the BPM. The consultant companies aim to design BPs from the point of view of the developer of applications, allowing, therefore, the ability to use the workflow engines or similar techniques.

In this case, they completely lose sight of the company in its wholeness, so this technique represents just a way to design applications with look and feel more attractive but are substantially similar to the functional decomposition of approximately 20 years ago.

Of course, in some cases, this technique can be equally effective, but it is impossible to generalize its use.

The second situation represents, instead, exactly the opposite. A big consultant company designs the new BPs correctly considering the company in its wholeness and acting on all of the variable ones into play.

Successively, when they will develop the computer-based system, which must support the new way to work of the company, the developers will use well-known techniques of applications design considering BPs just as requirements.

Therefore, it needs to bridge the gap between the design of BPs and the design of the Web applications.

Our research work, that is the base of this book, has the goal of bridging this gap through a methodology that takes into account the requirements and the peculiarities that are apparent.

The research activity, therefore, starts from the foundation that the activity of the business experts (those who have the task of redesigning a company in its wholeness) stop at a detail level unsuitable to those who, instead, must implement the applications.

Therefore, the first methodological step carried out from the designers of applications is that of refining the flow of the processes in order to render them apt to being a true input (and not just as a requirement) for the developers.

Moreover, two possible scenarios are proposed. The first regards the part of Web information systems turned to the internal users by agreement with the strategy of reengineering of the BPs defined “in-side” to which often it is associated, erroneously to our opinion, the acronym ERP.

This type of user often has the necessity of applications data or process-driven, and therefore, the philosophy of designing the user experience could be not be suitable, in the sense that in the majority of the cases, it coincides with the semantic structure of an ER model with the timing based on the flow of the operations.

For the internal users, the use of a workflow engine could be suitable also, but we have thought it opportune to give a greater freedom to such users in order to build a “virtual desktop” according to their needs.

For this reason, and by agreement with the philosophy of the DSM, we have preferred to generate automatically the Java™ Portlet or the Webparts in Microsoft® environment.

In this way, the user could personalize his/her virtual desktop, adding to the business activities also others Portlet about individual productivity or social communication.

The second scenario concerns instead the external users of the company and therefore essentially, but not only, by agreement with the strategy of reengineering of BPs defined “sell-side.”

The external users essentially use a WA in order to navigate between the information and to activate BPs having integrated part of the same ones.

This scenario allows making a design of the user experience independent from the processes and by agreement with how much is described in the previous paragraph to which successively will be integrated the BPs.

According to these considerations, this book aims to introduce two new methodologies: the first one is a result of the extension and a reasoned integration of existing methodologies at conceptual and logical levels which introduces a new publishing model; the second one, is oriented to the internal users, as an enhancement of the generation of applications through workflow engines.

This book deals with new methodologies and is clearly oriented to the scholars demanding their contribution to improve our approach; however, since the book also deals with tools and automatic code generation starting from the models of the Web information systems, it could be an essential guide for practitioners that have to design, manage and maintain information systems. The development process model is not completely classified into the well-known models such as, waterfall, agile, and so forth.

According to the philosophy of domain-specific modeling, the programmers have to develop the tools for the automatic generation of the code to build the final applications. They are completely free in using whatever they like in terms of development process.

Furthermore, to build the final complex Web information system, our methodology does not constrain a specific process; in fact, it is possible to generate a prototype starting from the only conceptual model to verify with the customer the good quality of the modeling analysis. This iteration could be always done. In the last analysis, it is possible to affirm that the development process is quite similar to the agile unified process (AUP) (Ambler, 2002), but not completely equal.

ORGANIZATION OF THE BOOK

The research work, that is the base of this book, realizes an integrated and flexible methodology in order to model complex Web information systems, which safeguard both the peculiarities of the Web and adopting a right methodology for the design of the information system, performed by experts of the field, that becomes integrating part of the final model being reused and adapted to the requirements of the development.

All the models are in OWL format and through a code generator an application is obtained very closer to the final.

The book is divided into three sections with an overview for each section that summarizes the content and the goals of the section providing the reader with a helpful orientation.

The first section provides a background about the existing methodologies with a critical discussion about their strengths and their weakness; in Chapter I, the main notations for modeling the BPs will be described in detail; in Chapter II, the main methodologies for the modeling of WAs will be described; in Chapter III, the IDM (interactive dialogue model) methodology for designing WAs will be presented; in Chapter IV, after a short review of the fundamental concepts of the ontology, the use of OWL language for the realization of meta-model and models will be described.

The second section is of this book concerns the presentation of the two new methodologies using case studies to better understand them. Chapter V describes, in detail, the overall design vision of the architectures that are the base of the methodologies; in Chapter VI, the complete methodology of design for Web information systems for external users will be described using a simple case study to improve the readability; in Chapter VII, such methodology is applied to a more complex real case study; in Chapter VIII, the methodological approach for the internal users will be presented through a simple case study; furthermore, in Chapter IX, the methodology for internal users is applied to a real and complete case study.

The third section concerns the description of the support tools, used essentially as editors, and the tools that allow for generating the final application in an automatic way. In detail, in Chapter X, we will deal with the problems related to the automatic generation of code; in Chapter XI, the technological choices will be described; in Chapter XII, a configurable editor as an indispensable tool in order to model BPs and WAs will be described; in Chapter XIII, the code generators will be described and in the Chapter XIV, the application of these tools to real case studies will be presented.

Finally, in Chapter XV, we will draw a conclusion and a panoramic view of the future work.

REFERENCES

- Ambler, S. (2002). *Agile modeling: Effective practices for extreme programming and the unified process*. John Wiley & Son, Ltd.
- Cardoso, J. (2007). The semantic Web Vision: Where are We?. *IEEE Intelligent Systems*, 22-26.
- Galbraith, J.R. (1973). *Designing complex organizations*. Reading, MA: Addison-Wesley Publishing Company, Inc.
- Hammer, M. (1990). *Reengineering work: Don't automate, obliterate*. Boston, MA: Harvard Business Review.

Hammer, M., & Stanton, S. (1995). *The reengineering revolution*. New York, NY: Harper Business.

IEEE Std 1320.1-1998. (1998). IEEE Standard for Functional Modeling Language—Syntax and Semantics for IDEF0. New York: IEEE.

OMG. (2006). *Business process modeling notation specification*.

Paiano, R., Pandurino, A., & Guido A. (2006). Conceptual design of Web application families: The BWW approach. *Proceedings from the 6th ACM OOPSLA Workshop on Domain-Specific Modeling*, (pp. 23-32). Portland, OR.

Paiano, R., Sánchez-Ruíz, A., Motoshi S., & Langlois, N. (2007). Domain-specific software development terminology: Do we all speak the same language. *Proceedings from the 7th OOPSLA Workshop on Domain-Specific Modeling (DSM'07)*. Montreal, Canada.

Pighin, M., & Marzona, A. (2005). *Sistemi informativi aziendali*. Pearson Education.

Porter, M.E., & Millar V.E. (1985). *How information gives you competitive advantage*. *Harvard Business Review*, 63(4), 149-161.

Rumbaugh, J., Jacobson I., & Booch, G. (2004). *Unified modeling language reference manual*. Addison-Wesley.

Van der Aalst W.M.P., der Hofstede, A.H.M., & Weske, M. (2003). Business process management: A survey. *Proceedings from the First International Conference in Business Process Management*, (pp. 1-12) Eindhoven, PAYS-BAS.

W3C. (2004). *OWL Web ontology language reference*.

Williamson, O.E. (1975). *Markets and hierarchies: Analysis and antitrust implications*. New York, NY: Free Press.