Overview of a Framework to Hypermedia Process Modeling

Luis OLSINA
Faculty of Engineering, at UNLPam,
also at Ciencias Exactas, UNLP, and UNLu - Argentina
E-mail olsinal@ing.unlpam.edu.ar
Web http://gidis.ing.unlpam.edu.ar/personas/olsinal/olsinal.html
TelFax +54 (0) 302 30497 Int. 6501

ABSTRACT

In this work, we discuss aspects of a framework to support the understanding and the improvement in hypermedia development processes. The framework is built taking into account two modeling approaches and four domain levels. The modeling approaches are called prescriptive and descriptive respectively, and the domain layers are called, namely, enactment, customization, representation, and reuse/knowledge domains. We feed a potential project regarding this framework by applying a mixture of expression and analysis-oriented descriptive process modeling strategy, and expression, analysis and guided-oriented prescriptive process modeling strategy. That is, the actual or desired entities (process, artifact, and resource) are just described, represented, and analyzed in a somewhat formal way. This potentially allows us to understand, communicate, guide and improve different aspects in hypermedia development projects. Ultimately, we summarize some techniques and mechanisms for prescriptive and descriptive software modeling customized to the hypermedia domain. In addition, we present in-progress researchs and developments.

KEYWORDS: Software Engineering, Framework, Descriptive, Prescriptive, Hypermedia Process Modeling, and Web-based Developments.

1. INTRODUCTION

Despite the growing and renewed interest in the wide community of researchers and developers about Web-based Developments (for instance, see the last-held well-known international Hypertext and WWW conferences and workshops [7, 17]), the building of hypermedia application still lack very documented lifecycles and process modeling mechanisms. This happens not only in the academia but also in the industry. Within projects of this arena, development processes are poorly managed and the basically used strategy to produce artifacts is rather ad hoc, often producing applications of uncertain quality and hard to maintain. Likewise, hard to evaluate and predict are development costs, process duration, and productivity estimations. Many of these troubles (and others do not cited here), are mainly due to the lack of a common understanding how to deal with hypermedia processes.

Therefore, we have a strong need of a broader engineering-based software approaches and mechanisms that support the understanding, the control and the improvement in hypermedia projects. In this way, we propose a framework to hypermedia process modeling that should take into account descriptive and prescriptive approaches and mechanisms. By process modeling, we mean the actual or desired entities that could be modeled in a hypermedia project such as activities, artifacts and resources, among others. Even if we know research efforts in traditional Software Engineering about process modeling [1, 3, 6, 14, 16, among others] we could consider it a new concern in the Hypermedia community [9, 10, 11].

Five basic research concerns for process modeling ranging from understanding aids to automated-execution support, were enumerated [3]. They are 1) the facilitating human understanding and communication; 2) the supporting process improvement; 3) the supporting process management*;* 4) the automating process guidance; 5) the automating execution support. Our current research concern in hypermedia process modeling is centered mainly in the first, the second, and the fourth issues.

On the one hand, we try to answer the question, *how should hypermedia software be developed*? To deal with the inherent complexity of processes we have proposed views on a conceptual model that represents a schema of classes and relationships of process modeling domain [12]. We abstract essential concept such as process, task, activity, artifact, resource, agent, role, performer, process constructor, condition, and constrain, among others. A view is a particular approach to specify and to communicate information about some entities of the conceptual model. Our taxonomy, enlarging on Curtis´ classification, represents functional, methodological, informational, behavioral, and organizational views. Therefore, the schema of conceptual model and views are at the reuse/knowledge level of the proposed framework (that we see later). In this way we are developing the Hypermedia Flexible Process Model (HFPM), this is, a process model customized to the hypermedia field.

On the other hand, we are evaluating hypermedia projects trying to answer the question, *how is hypermedia software actually developed*? This potentially allows us to study, analyze, and improve entities such as artifacts, processes and resources. One of the main goals in developing hypermedia applications is to produce artifacts of planned quality, which should be governed by a set of desired and observable attributes, using for such end the most effective processes and the most appropriate resources. We should assure the mechanisms to build hypermedia artifacts that accomplish such attributes. So, we present a reduced list of measurable attributes for these entities [13] that could be used by means of Goal-Question-Metric approach [2] in the context of our proposed framework.

Therefore, in the following section, we discuss a reference model to process-modeling domain and a panorama of a framework whereby descriptive and prescriptive process modeling approaches can work. Next, we present some mechanisms for prescriptive and descriptive modeling customized to the hypermedia domain. Finally, we consider some concluding remarks and future directions.

2. OVERVIEW OF A FRAMEWORK TO DESCRIPTIVE AND PRESCRIPTIVE PROCESS MODELING

2.1 Motivation

From the early 90's, the advent of the multimedia CD-ROM and the coming of Web-based Developments have marked a quick growth in the construction of hypermedia components and applications. However, as previously indicated, a much defined development process that leverage the understanding and the improving, mainly in medium and large-scale projects, have not been accompanied by that applications growth. Rather the utilized development strategy has been an undocumented, and ad hoc one (also known as quick-and-dirty, or just-do-it strategy centralized mainly in code, validation, and error-fixing activities). Unfortunately, this lack of a much defined hypermedia process that cover vital aspects of a project could make the ghost of the software crisis grows. Despite the fact that this discipline has turned an extremely active research area there has really been very little specific literature about hypermedia process models, evaluation mechanisms, and integrated frameworks to process modeling. Thus, it is being observed the need of having an engineering approach, i.e. a disciplined, quantified, and systematic employment of Software Engineering principles for the evaluation, control, guidance, and prediction of hypermedia entities (artifacts, process, resources).

Therefore, if this situation is to be improved then, we need to develop an understanding of hypermedia processes and, ultimately, and understanding of a framework whereby hypermedia process modeling can fits.

2.2 Process Modeling Domain

In this way, we should first see what canonical concepts in process modeling are embraced. Figure 1 represents a schema, a conceptual model for process-modeling domain useful to support the reuse/knowledge level of the framework (see figure 3). This model is integrated by primitive packages, classes and relationships that abstract the essential concepts for process modeling. This gives us a basic repository (an O-O static model) of the fundamental set of responsibilities and collaborations among entities. In software engineering, key concepts for process modeling are handled, such as process itself, task, activity, artifact, resource, performer, agent, role, process constructor, process description, goal, condition, constrain, among others. Important efforts have been made with the purpose of establishing a solid conceptual and reusable base [4, 8, 14].

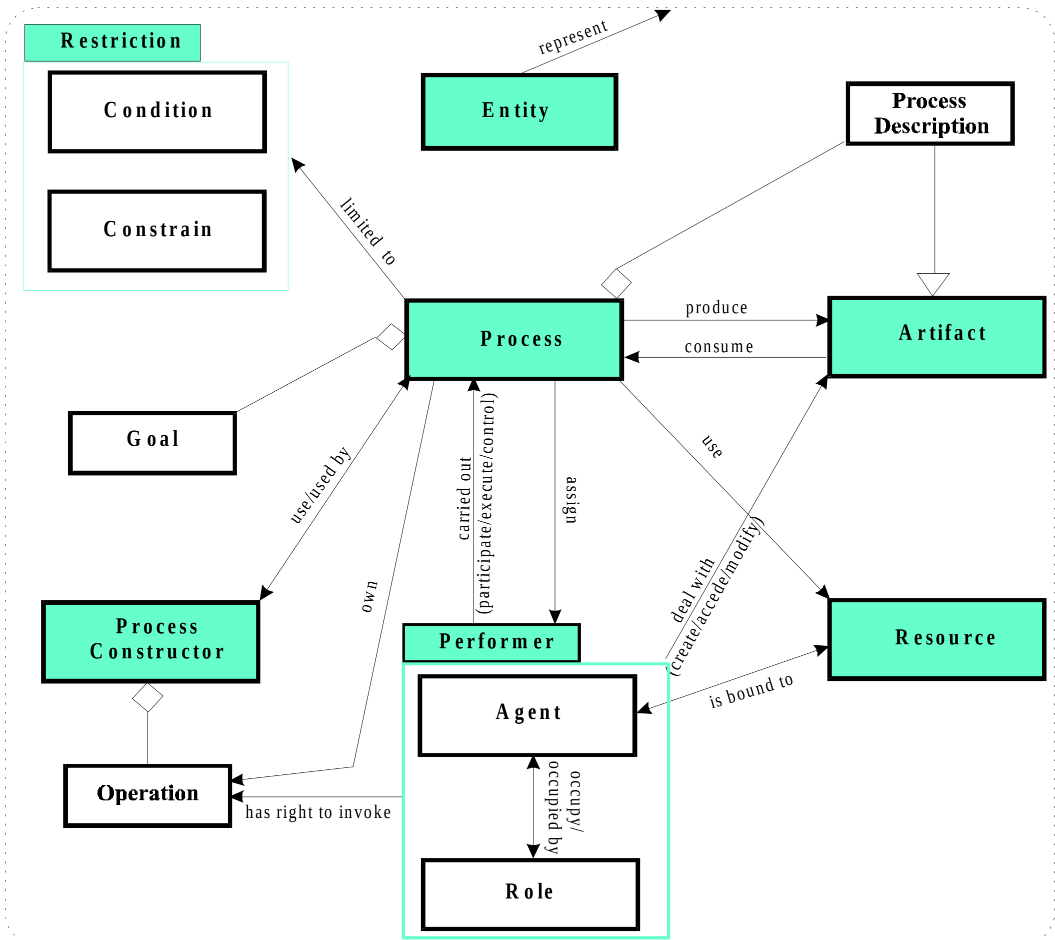

Figure 1. *Schema of the conceptual model of Process-Modeling Domain (specified in UML notation)*

We think that in our conceptual model, the representation into packages, classes and relationships, contributes to specify the problem domain in a clear and powerful way. The classes (or essential model concepts) represent behavior and state. They support operations and encapsulate attributes. For instance, a class "Artifact" can have attributes such as its identity, its creation, last modification and approval date, the given version, etc. It can respond to a performer request by means of creation, destruction, modification operations, both of the content, structure, and of its attributes.

On the other hand, classes support different relationship mechanisms. The relationships can be established among entities of a class itself, or among different classes. We can name inheritance, aggregation, and association mechanisms. For instance, we could depict an inheritance hierarchy for the “Role” class (e.g., an "Interface Designer" is-a "Designer"); a compound “Artifact” is related by an aggregation mechanism (see figure 2b). Classes are also related to each other by means of association (for example, see the produce and consume relationships between "Artifact" and "Process" classes).

We will make a brief description of key components of the conceptual model, and we will value how this schema facilitates the understanding of the process-modeling domain. So, an “Activity” is-a "Process" (see fig. 2a) and a process could be composed by other processes. The differences between “Task” and “Activity” are sometimes very subtle. A former is a managed process; e.g., resources can be allocated to it as well as it can be scheduled. Instead we see an "Activity" as an unmanaged process (this could be useful in prescriptive simulation and analysis).

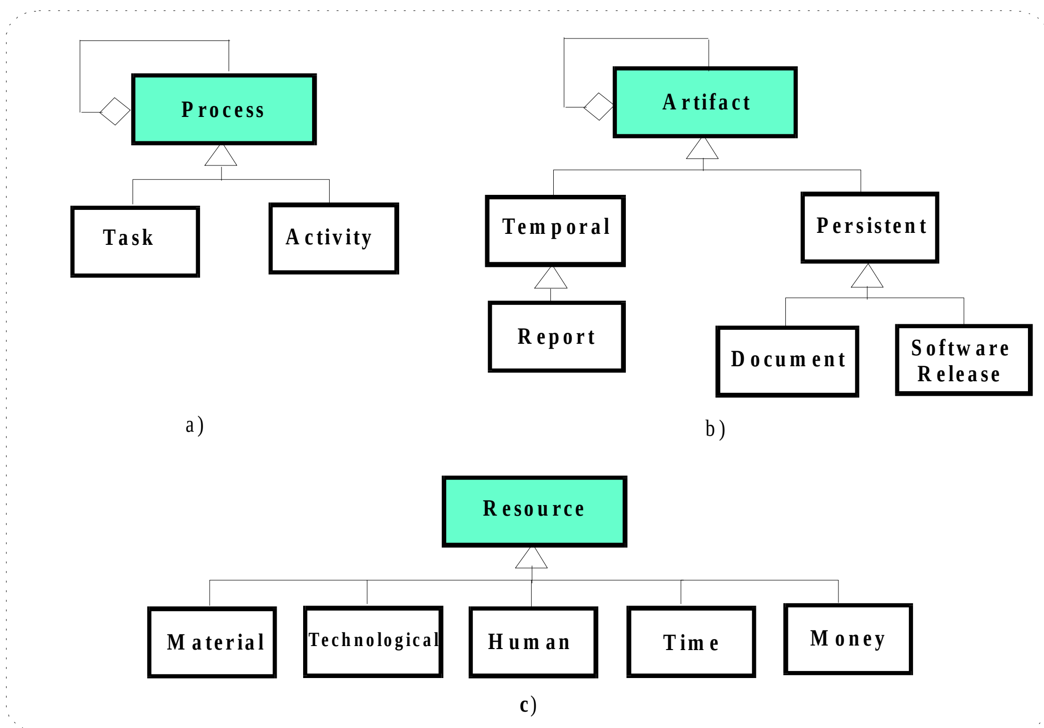

Figure 2. *Reduced diagrams of three entities: a) Process, b) Artifact, c) Resource.*

Thus, a "Task" represents a unit of work that is assigned to a “Performer” for its performance. A “Task” contains a “Process Description”, which can include (in a formal, semi-formal or informal way) a collection of alternatives to specify the same unit of work. In addition, the process description can be specified in different formalisms that will be understandable for the different involved performers. Agent is the entity that associated to a role carries out processes. (The agent can be as much a human entity as a computerized tool or device. Task taxonomy in correspondence with the agents is, namely: manual, automatic and interactive). In this way, a class of “Agent” type occupying or playing a type of “Role” (a "Performer" package) carries out a type of “Process”.

On the other hand, artifacts are persistent or temporal objects that represent the product of performing a task. An artifact can be a simple or compound object and it could be subject to a configuration management and version control system.

If we continue observing some diagram classes and relationships we can say that a “Performer” carries out a “Task” contained in the “Process Description” using some “Process Constructor”. For the fulfillment of the task the “Performer” is entitled to invoke the “Operations” that are part of the “Process Constructor”, this is, the specific techniques, mechanisms, or method's primitives to do a task. The “Process” is limited to certain “Restriction”. On the other hand, the class “Agent” has a many-to-many relationship with the class “Resource” (see fig. 2c). A resource can associate several agents and an agent can use several resources. For example, a person that is a resource can associate several agents, and an agent can use several resources like a person, computer hardware and software, physical space, etc. An agent can use a composition of resources of the same type (e. g. people’s team, etc.).

Therefore, the definition of this conceptual or reference model and the distinction among different types of abstractions and concerns is of paramount importance for process modeling for several reasons. Firstly, because it represents the essential abstractions of the problem domain, containing common behaviors and making explicit a set of relationships. This favors the distribution of responsibilities and identification of collaborations (for instance, we are at the initial steps in the building of a web-based, guidance-oriented hypermedia process environment, regarding that reference model). Secondly, it favors the outline of views, that is to say, the division of concerns in sub-models from the conceptual model. In the figure 1, we consider explicitly the essential relationships among classes. However, with seven primitive classes we could have a seventh factorial of potential relationships. (For this reason, when designing a process model, a division of concerns into views, as elsewhere discussed [10], can diminish the complexity in process modeling.)

Thus, to diminish the complexity, it is convenient to separate different types of information of the processes to specify, to communicate, and to control portions of the model. In our research, we define the functional, informational, behavioral, organizational and methodological views. In the following paragraphs, we summarize the main concerns of each view.

The *functional view* specifies, at the knowledge level, what elements intervene into the model of a generic process. For instance, we can represent a process in function of the input and output artifacts, the constrains and conditions, the hierarchical process structure, among other issues. At the customization domain of the framework (see figure 3), this view could be tailored to represent specific phases and tasks of a specific kind of hypermedia project (as we will present in the subsection 3.1).

The *informational view* is centered on those artifacts produced or required by processes, on the structure of the artifacts and their interrelationships, on the intervening performers, and on the strategies of configuration management and traceability models.

The *methodological view* represents the process constructors or primitives to be applied to accomplish the different actions specified in the activity description. For a specific activity description, we can have one or more methods’ primitives that give support to that task (for instance, for some activities of “Navigational Modeling” process we can employ primitives of OOHDM [15] or RMM). On the other hand, for a method in particular we can have one or more tools that support it.

The *behavioral view* represents the dynamics of the process model. For instance, the sequencing and synchronization of tasks, the information about how the activities are carried out, that is, parallelisms, iterations, feedback loops, beginning and termination conditions, milestones, among other issues. At the customization level, we can specify workflows. Besides, we can also specify the different states of an entity like an artifact with formalisms such as statecharts or Petri nets.

In the *organizational view* we are considering what performers and their associated resources perform what tasks; what roles (and skill attributes) are assigned to agents, what communication strategies and groups' dynamic are applied, among other aspects.

Ultimately, and taken into account our reference model, the functional perspective focalizes mainly on the "Task" class and its relationships. An informational view focalizes it on the "Artifact" class, its components and relationships. A methodological view concentrates on the "Process Constructor" class, and so on. These views and the reference model fit in the reuse/knowledge domain, at the highest level of abstraction. Next, we consider spaces and domains of the proposed framework.

2.3 Framework to Process Modeling Spaces and Domains

The main steps in conceiving a process model for different hypermedia projects is to develop first a framework within which such modeling can happen. We need to have both general and more specific understanding of those elements and issues that should be included into a modeled project. For that reason, we think as a mandatory issue the working in the context of a process-modeling framework.

Figure three represent an integrated framework to descriptive and prescriptive modeling spaces. The framework is built taking into account two modeling strategies and four domain levels (see also [9]). The modeling strategies are called prescriptive and descriptive respectively, and the domain levels are called enactment, customization, representation, and reuse/knowledge domains. We conceive this framework by applying a mixture of expression and analysis-oriented descriptive process modeling strategy, and expression, analysis and guided-oriented prescriptive process modeling strategy; i.e. the actual or desired entities (process, artifact, resource) are just described, represented, and analyzed in a somewhat formal way.

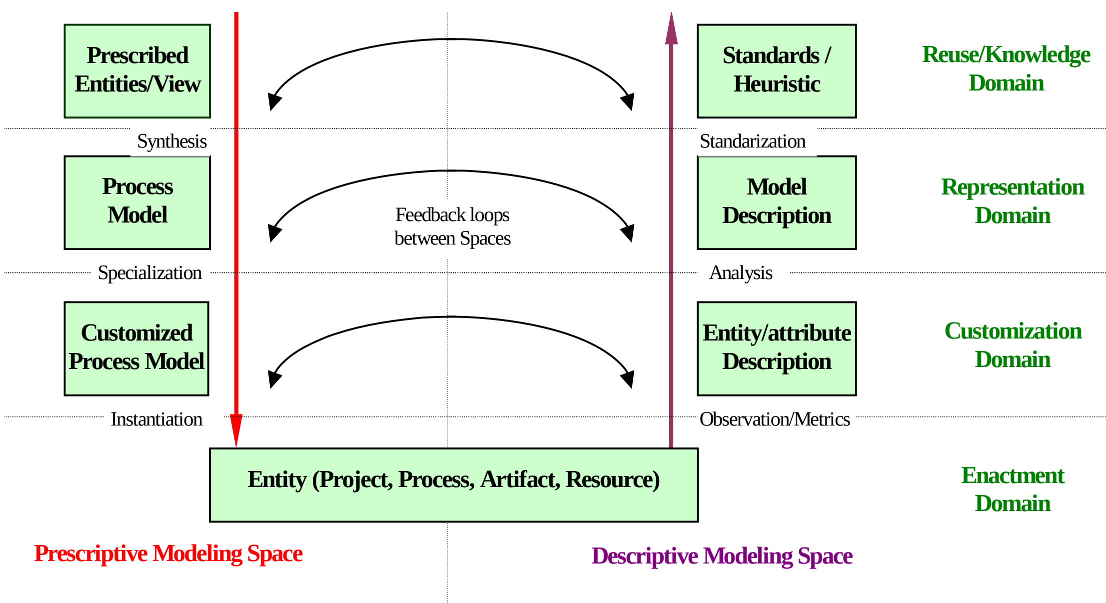

Figure 3 *Framework to prescriptive and descriptive modeling spaces and domains.*

In order to explain the basic rationale between spaces and domains it will be opportune write down the concepts and terminology used by Lonchamps [8]. "*People dealing with software processes may adopt two different attitudes of mind:*

Descriptive: they study existing processes to answer the question "how software is (or has been) actually developed?"
Prescriptive: they define desired processes to answer the question "how software should be developed?"...

Within these attitudes people dealing with software process descriptions may aim at:

Expressing: the actual or desired process is just described more or less formally for understanding, communication, education, reuse, or standardization.
Analyzing: the description of the actual or desired process is studied through more or less formal techniques (such as validation, e.g. simulation, or property verification) for a deeper understanding, comparisons, improvement, impact analysis, or forecasting.

And, within the prescriptive attitude only, people may aim at:

Guiding: software process performers (developers, managers, ...) receive indirect support through information which help them to perform their work..."

On the one hand, and taken into account the prescriptive modeling space, we have prescribed a conceptual model and views at the highest level of the framework. The conceptual model is abstract enough to serve different proposes to process modeling. At the knowledge level, we describe views as a way to diminish the inherent complexity in process modeling. Thus, we can derive more specific process models at the representation level of the framework (fig. 3). For instance, we can derive a functional process model (regarding the conceptual model and the functional view), that takes into account general phases, tasks, and relationships. This prescriptive model could be specified in function of inputs, outputs, constrains and conditions, supertasks, subtasks, and performers. In turn, to model a more specific domain like Hypermedia, we should consider more specific tasks, roles, and hypermedia mechanisms; that is, we should customize the process model (at the representation level) to hypermedia domain (at the customization level). Finally, in the real and raw world only actual entities fit into specific project type and attributes. In this way, by means of an instantiation, we have actual entities in the enactment domain.

On the other hand, and regarding the descriptive modeling space, we can collect data, measure, describe, analyze, and model actual entities such as processes, artifacts, and resources. This potentially allows us to evaluate, predict, and improve such entities. In the next section, we will discuss some mechanisms that fit into the framework discussed here.

So, to achieve these objectives we see the task of designing a process model (as part of a tailorable project) as a two-dimensional workspace, i.e. observing and studying existing hypermedia development processes, and abstracting and prescribing desired processes and models. This task evolves into feedback loops among those spaces feeding and enhancing the target model. This potentially should permit us to understand, communicate, guide, and improve different aspects in hypermedia development projects.

3. MECHANISMS TO PRESCRIPTIVE AND DESCRIPTIVE APPROACHES

3.1 Some Prescriptive Aspects for Hypermedia Projects

Next, and for the sake of conciseness, we exemplify some prescriptive aspects regarding the *functional view* at the customization domain of the framework. At this point, we take into account entities and mechanisms to the hypermedia domain, in a medium and fine level of granularity. For instance, we can show phases and processes, and given a process, we can "navigate" for its sub-processes. Then, we can arrive to an atomic activity, to browse its specific actions, to see the input artifacts, the produced artifacts, the performer associated to each activity, and so on.

In the right frame of the screen, in the figure 4, the reader can see phases and processes of the Hypermedia Flexible Process Model (the HFPM is a process model customized to the hypermedia domain). The whole picture represents a dumped screen of the Web-based guidance-oriented hypermedia process environment that we are currently developing (the GoHyPEr project).

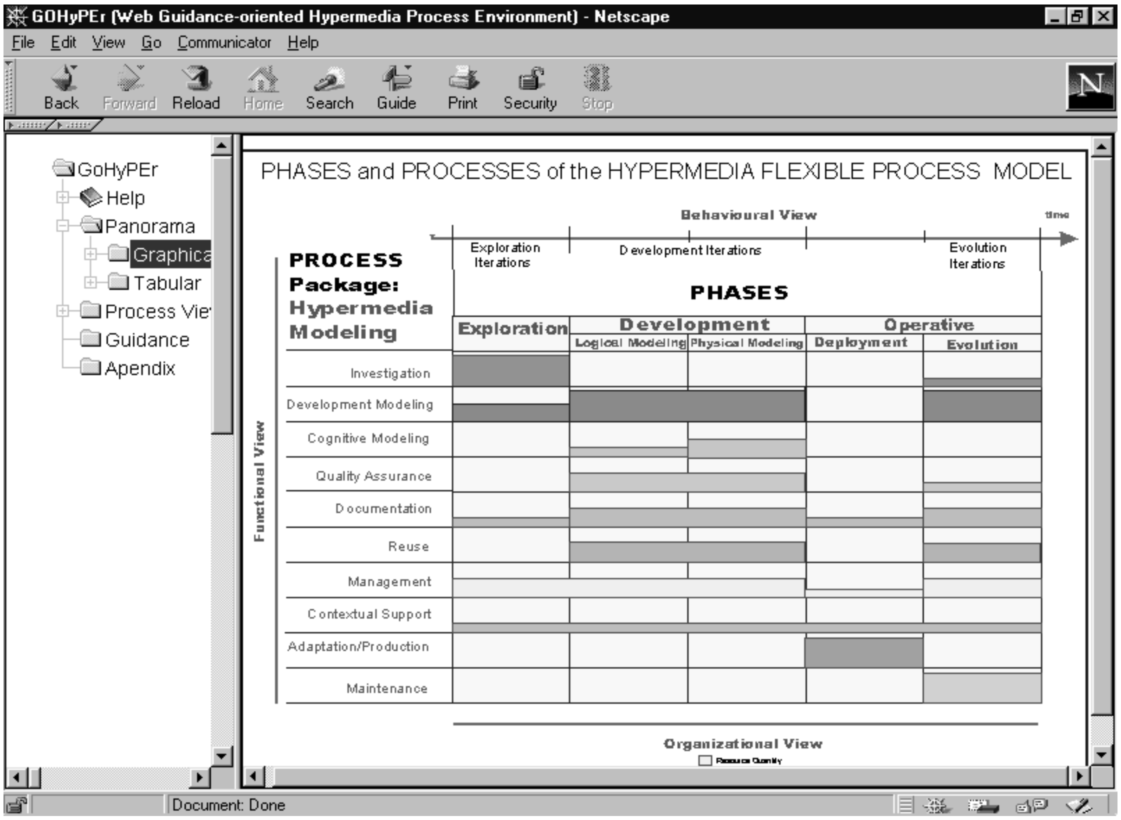

Figure 4. *A dumped screen of the Web-based guidance-oriented hypermedia process environment that show prescribed phases and processes of the HFPM .*

There are three cohesive phases in the HFPM, namely: the *exploration*, the *development*, and the *operational phase*. Also, there is a set of processes like "Investigation", "Development Modeling", "Cognitive Modeling", etc. If the practitioner activate the anchor "Development Modeling", the tool shows a similar picture but containing its sub-processes (recall the aggregation relationships shown in fig. 2a). For instance, "Software Requirement Modeling", "Conceptual Modeling", "Navigational Modeling", "Interface Modeling", "Physical Modeling", are some prescribed sub-processes of "Development Modeling" process. In turn, if we select the "Navigational Modeling" anchor into the tool, we will obtain a table with atomic activities and their associated (and hyperlinked) information like actions, input and output artifacts, performers, as previously indicated.

All the entities and views prescribed at the reuse/knowledge level, could be customized in some degree to the hypermedia domain. Finally, the left frame of the tool shown in the figure 4, is a menu (an applet), containing starting points to hyperlinked pictures, tables, templates, and hypertext, that serve to the propose of guidance in the development process. This tool will allows us (in a near-future step of the GoHyPEr project) the collaborative development environment, by editing a shared hypermedia process model.

3.2 Some Descriptive Mechanisms for Hypermedia Projects

In the introduction section, we argue that all development process, in the context of an organization should continually take care of three essential objectives: to produce artifacts of quality, to use effective processes, and to employ appropriate resources. To accomplish this, we should select a balanced mixture of observable characteristics or attributes that could contribute to the quality of these entities.

From the point of view of the evaluation and improvement of actual processes, artifacts and resources (regarding entities at the enactment domain of the framework depicted in fig. 3), it is necessary to perform measurements on the attributes of the entities. The analysis and interpretation of the collected data can be used to evaluate, to feedback, to predict, and to improve actual or desired entities.

Table 1 shows a reduced set of these observable characteristics [5] that derive in measurable attributes -or potentially measurable in some cases, given the novelty of the hypermedia field. Unfortunately, there is a very few metrics reported in the hypermedia domain, so far. However, the attribute/entity descriptions should be validated an enriched by performing increasingly mature projects in the context of actual organizations. (Recall that there exist ways to assess the maturity of processes and organizations, e.g., CMM and SPICE frameworks).

Table 1 *Reduced set of characteristics to Artifact, Process, and Resource entities*

Attr./Entity	**Internal (Objective)**	**External (Subjective)**
Artifact	• Size (node quantity, link quantity per node, etc.) • Complexity (centrality, compactness, etc.) • Cohesiveness in concepts organization (at Local Level– navigational context, and at Global Level) • Completeness • Defect Quantity (for instance, dangling links, invalid link, etc.)	• Navigability (interconnection level, distance, orientation, path) • Reliability (destination node reachability, link validity, etc.) • Usability (search mechanisms, index and annotation mechanisms, self-evidence of the shown objects, interface adjustment, etc.) • Readability • Quality
Process	• Flexibility • Time • Completeness	• Cost • Quality • Stability
Resource	• Skill Level, Age • Size • Communication Level • Speed	• Cost (e.g., price of a human agent according to the skill level and age) • Productivity

In the following paragraphs, we show one mechanism to select metrics in function of goals, regarding the point of view of the performer, and considering the context and maturity of the organization (as addressed by Fenton and Pfleeger [5]). That descriptive mechanism is call Goal-Question-Metric approach [2]. This is a goal-oriented rather than a metric-oriented approach: we can define and plan desired situations so that, by means of the gathering of the data allow us further decisions on target entities.

Therefore, given a selected set of project goals, a set of questions can be placed and refined for each goal and depending on each question the appropriate metrics can be chosen (keeping in mind some attributes from the table). The interpretation of the outcomes can be fair to analyze, predict, understand, and improve processes, products, and resources (feeding primly the descriptive modeling space -see fig. 3). For example, the following are some goals that could be formulated in a hypermedia project, taken into account only de conceptual level of GQM model:

✓ *To improve the navigability of a hyperdocument from the final user's point of view* **Objective:** *Improve*; **Characteristic:** *Navigability;* **Object (type):** *Hyperdocument (artifact);* **Agent assigned to a role:** *Final user;*

- ✓ *To evaluate the content relevancy and completeness of a Web site* **Objective:** *Evaluate*; **Characteristics:** *Content relevancy and completeness;* **Object (type):** *Web site (artifact);*
- ✓ *To improve the personnel expertise in the graphic design of Web pages* **Objective:** *Improve*; **Characteristic:** *Skill level;* **Object (type):** *Personnel (resource);*

Thus, starting from these goals, questions can be formulated and from these, metrics can be refined.

Finally, and as previously stated, a core issue in GQM approach is the interpretation of the captured data depending on the questions from which these measures were derived. In [13], we have discussed the interpretation of the compactness metric for a specific performed project.

4. CONCLUDING REMARKS

Hypermedia applications are continuously and rapidly growing, mainly those developments based on the Web. Unfortunately, the most common used strategy to develop applications has been rather undocumented and ad hoc. There are up to now very few documented lifecycles and process modeling mechanisms. So, we have argued in this paper about the necessity of using explicitly and systematically a well-defined engineering-based approach to hypermedia process modeling in order to pursue the understanding, communication, and improvement. Onto this direction, we see as one of the first concerns to develop a hypermedia process model, the designing of a framework within process modeling strategies, mechanisms, and techniques can take place.

So, as previously reported, we are working based on process modeling framework that potentially promotes communication, guidance, and improvement. We have shown aspects of an integrated framework whereby descriptive and prescriptive process modeling strategies can work. We have aimed to answer both questions how software is (or has been) actually developed (the descriptive process modeling strategy side), and how hypermedia software should be developed (the prescriptive process modeling strategy side). A division of concerns into views from the conceptual model helped us to focus on some views in almost an independent way. We might continue formalizing and refining some views and arranging them into process architecture. Ultimately, these in-progress investigations are allowing us to feedback and evolve the target process model (the HFPM), and to gain experience in mechanisms of hypermedia process modeling.

Moreover, within the prescriptive approach and regarding automation there are two important objectives that people may aim, i.e. guiding and enacting. An initial area of research is to design and to build a Process-centered Software Engineering Environments that support the guidance and/or enactment of hypermedia developments taking into account some or all of the discussed views. In addition, our current research concern (and our future direction) is focused on the construction of a Web-based guidance-oriented hypermedia process environment (the GoHyPEr project) that will give support to stakeholders of hypermedia developments.

ACKNOWLEDGMENT

The author would like to acknowledge Dr. G. Rossi for his valuable guide. MS Luis Olsina is partially supported by the *"Programa de Incentivos de la Secretaría de Políticas Universitarias del Ministerio de Cultura y Educación de la Nación, Argentina"*. The author thanks the collaboration of Godoy, D.; Lafuente, Guillermo; Nicolau, S. and other participants of the currently-in-progress GoHyPEr project.

REFERENCES

1. Bandinelli, S; Alfonso, F.; Lavazza, L.; Loi, M.; Picco, Gian Pietro, 1995, *"Modeling and Improving an Industrial Software Process"*. TSE 21(5): 440-454
2. Basili, V.R., Caldiera, C., Rombach, H.D., 1994, *"Goal Question Metric Paradigm"*, Encyclopedia of Software Engineering, Vol. 1, John Wiley & Sons, pp. 528-532.
3. Curtis, B.; Kellner, M.; Over, J., 1992, *"Process Modelling"*, CACM 35, 9; pp. 75-90.
4. Dowson, M.; Nejmeh, B.; Riddle, W., 1991, *"Fundamental Software Process Concepts"*, Proceed. of the First European Workshop on Software Process Modeling, AICA Press.
5. Fenton, N.E.; Pfleeger, S.L., 1997, *"Software Metrics: a Rigorous and Practical Approach"*, 2nd Ed, PWS Publishing Company.
6. Humphrey, W.S., Kellner, M.I., 1989, *"Software Process Modelling: Principles of Entity Process Models"*, 11th Int. Conference on Software Engineering, IEEE Computer Society.
7. Hypertext '98 Conference, 1998, http://www.ks.com/HT98/
8. Lonchamps, J., 1993, *"A Structured Conceptual and Terminological Framework for Software Process Engineering "*, ICSP 2, Berlin, IEEE Press.
9. Lowe, D.; Webby, R., 1998, *"The Impact Process Modelling Project - Work in Progress"*, 1st International Workshop on Hypermedia Development, at Hypertext 98, Pittsburgh, US.
10. Olsina, L., 1997, *"Applying the Flexible Process Model to build Hypermedia Products"*, Hypertext and Hypermedia: Products, Tools, Methods (H2PTM´97), V. 1, pp. 211,221, Hermes Editorial, Paris, Fr.
11. Olsina, L, 1998, *"Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy ";* 1st International Workshop on Hypermedia Development, at ACM Hypertext 98, Pittsburgh, US (The paper is available at http://ise.ee.uts.edu.au/hypdev/).
12. Olsina, L., 1998, "*Functional View of the Hypermedia Process Model*, Fifth International Workshop on Engineering Hypertext Functionality, at IEEE ICSE´98, (The paper is available at http://www.ics.uci.edu/~kanderso/htf5/papers/) Kyoto, Japan.
13. Olsina, L., 1998, *"Toward Improvements in Hypermedia Process Modeling"* Accepted paper at the Symposium of Software Technology, 27 JAIIO, Bs As., Argentina
14. Scacchi, W; Mi, P., 1996, *"A Meta-Model for Formulating Knowledge-Based Models of Software Development"*, Decision Support Systems, Vol 17, Nº 3 pp. 313-330.
15. Schwabe, D.; Rossi, G., 1995, *"The Object-Oriented Hypermedia Design Model"*, CACM 38,8.
16. Verlage, M; Webby, R; et. al., 1998, *"The Spearmint Approach to Process Definition and Process Guidance"*, First International Workshop on Software Engineering over the Internet at IEEE ICSE´98, Kyoto, Japan. (The paper is available at http://sern.cpsc.ucalgary.ca/~maurer/ICSE98WS/ICSE98WS.html).
17. WWW '98 Conference, 1998, http://www7.conf.au