Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1996 Proceedings

Americas Conference on Information Systems (AMCIS)

8-16-1996

Using Machine Learning Techniques to Improve Information Systems Development Methods

Nicolas PRAT
University of Paris- Dauphine, prat@lamsade.duphine.fr

Using Machine Learning Techniques to Improve Information Systems Development Methods

Nicolas PRAT
LAMSADE
University of Paris-Dauphine
Place du Maréchal de Lattre de Tassigny
75775 Paris cedex 16,
CRI
University of Panthéon-Sorbonne
17, rue de Tolbiac
75013 Paris
FRANCE
email: prat@lamsade.dauphine.fr

1. Introduction

In this paper, we concentrate on the first phase of Information Systems (IS) development. This phase, called Requirements Engineering (RE) [NATURE96], aims at producing a high level specification of the system, the so-called conceptual schema. The specification is built using a specific method, e.g. OMT. A method is based on one or several models, e.g. the ER model, and prescribes the use of these models for the construction of schemas.

Traditionally, the IS community has focused on models. However, it is clear today that RE requires not only model knowledge, but also detailed method knowledge, i.e. guidance on the use of models. Therefore, the focus of research has shifted from models to methods, i.e. from a product view to a process view of IS development. To support this view, PCEEs (Process Centered Engineering Environments) are needed. They permit to define various methods, use these methods for IS development, and improve them gradually based on experience. PCEEs currently are at the research stage and examples of such environments are scarce. Apart from the MENTOR prototype of the CRI (Centre de Recherche en Informatique) [NATURE96], the only example we know of is PRO-ART [POHL95].

This article introduces a framework for the definition of PCEEs and focuses on the experience-based learning module of the framework. The originality of the learning method lies in the use of machine learning techniques to facilitate learning.

The paper is organized as follows. Section 2 addresses the framework for the definition of PCEEs. Section 3 addresses the learning module. Section 4 concludes.

2. A Repository-Based Framework for Methods

This section briefly presents the architecture defined in the NATURE project [NATURE96] with the contribution of the CRI. The architecture is illustrated in figure 1.

The process repository, composed of three abstraction levels, constitutes the center of the architecture.

-> At the lowest abstraction level, process traces are recorded. A process trace records what actually happens during a development run. There is one trace for each application developed.

-> At the intermediate level, process models and trace models are defined. A process model is a method, e.g. the ER method. It is prescriptive in that it guides the analyst in schema construction. A process results from the execution of a process model; however, it is not a pure instanciation of the process model since

the analyst may always deviate from the prescriptions of this model. A trace model is descriptive, as opposed to a process model. It is a structure to record traces i.e. what actually happened during a development run.

-> The highest level consists in the process meta-model defined by the CRI. This meta-model provides a set of generic concepts to define any process model or trace model.

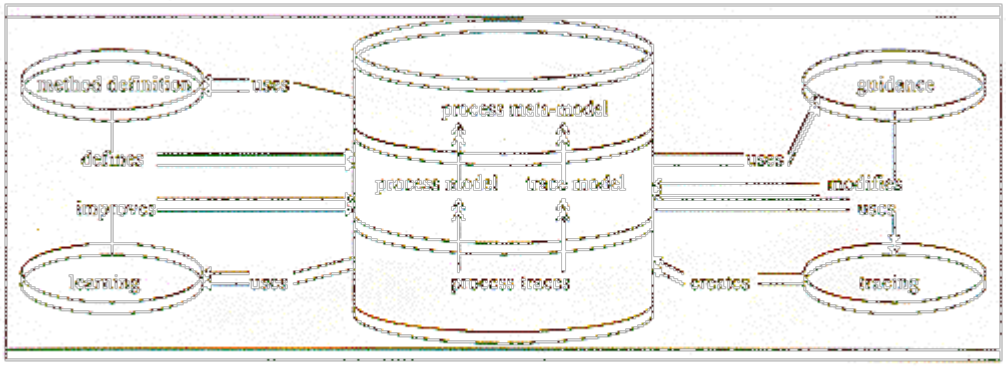

Figure 1. A repository-based framework for methods

The process meta-model [ROLLAND96] is centered around the notion of context, defined as a <situation,decision> couple. The situation part is a part of the RE product it makes sense to make a decision on; the decision part may be viewed as an intention, a goal. It represents a decision made on the situation. Three types of contexts are distinguished:

-> An executable context corresponds to a decision which is directly applicable through actions, leading to a transformation of the RE product. For example, in the ER method (figure 2), the context <RT with at least one participating ET; historize RT> is an executable context: it is not necessary to detail the decision to historize a relationship-type, which is directly applicable.

-> A choice context is a context whose intention may be reached through several alternative contexts. For example, in ER, the intention to refine a relationship-type may be reached by historizing or by retyping the relationship-type. Choice criteria (not shown in figure 2) are associated with the alternatives to help the analyst to choose between them.

-> A plan context is a context which is decomposed into other contexts, which must be executed to achieve the decision of the plan context. For example, in figure 2, the context <Problem statement; define ER schema> is a plan context (only two of the component contexts are shown).

The contexts of a method are organized into trees. The nodes of the trees are choice or plan contexts, while the leaves are executable contexts.

The four modules of the framework (see figure 1) are articulated around the process repository. We shall focus on the learning module.

3. The Learning Module

The learning module aims at improving a process model (intermediate abstraction level) based on experience i.e. on traces (lowest abstraction level) produced when executing the process model. This is a typical empirical induction problem [PRAT95a].

We view trace-based learning as a decision process. Therefore, the learning method is modeled as any other method, i.e. by instanciating the decision-oriented process meta-model. The combination of machine learning techniques with this decision-oriented approach is useful to guide the learner and to partly automate learning.

Our learning method is based on process deviations, which are recorded in traces (for example, if the analyst makes a decision which is not predefined in the process model used, this is considered as a deviation). The learning method considers each of the deviations in turn. Every identified deviation is explained (step 1) and then taken into account to improve the considered process model (step 2), if appropriate. Induction takes place in step 2.

As mentioned above, a trace is not merely an instance of a process model. It may contain <situation,decision> couples (called trace objects) which do not result from the instanciation of a context. In step 2, the induction process consists in inducing the trace objects into contexts of the considered process model. We distinguish three alternatives for the induction of a context from a trace object:

-> free induction:

The learner freely induces a context from the trace object.

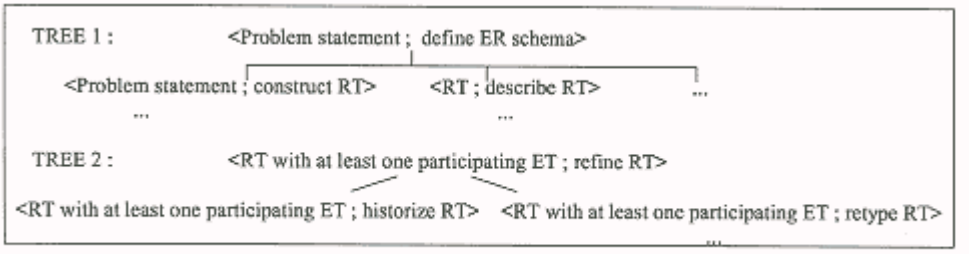

Figure 2. Excerpt from the ER method (ET=Entity-Type ; RT=Relationship-Type)

-> induction by means of the generalization rules:

The induction process is supported by the generalization rules [MICHALSKI83], which the learner uses freely. The advantage of this alternative is to provide the learner with induction expertise without restricting his creativity, but consistency with the existing contexts of the process model is not guaranteed.

-> induction by means of the induction method:

In this case, the learning process is supported not only by the generalization rules, but also by a predefined way of using them. The purpose is to maximize consistency with the already defined contexts (conservatism principle): first, using the generalization rules, the induction method checks if the trace object may be considered as an instance of an existing context of the process model; if this is not the case, a new context is defined using the generalization rules to maximize the similarity of this context with the existing contexts. Assume for example that the used process model is ER and that a trace object is:

<RT "orders", "orders".Collection.[1]="customer" and "orders".Collection.[2]="product"; decompose(RT "orders")>

(The situation part consists in the relationship-type "orders", whose collection consists in the entities-types "customer" and "product".)

Assuming that this trace object cannot be considered as an instance of an existing context, a new context is defined by repeatedly applying the generalization rules : first, using the turning constants into variables rule, the trace object is abstracted into the context :

<RT R1, R1.Collection.[1]=E1 and R1.Collection.[2]=E2; decompose(RT R1)>. Using the constructive generalization rule, this context is then generalized into the context :

<RT R1, nb_values(R1.Collection)=2; decompose(RT R1)>. (Note that nb_values is a predefined predicate which represents the number of values of an attribute.). Finally, the application of the extending reference rule results in the context :

<RT R1, nb_values(R1.Collection)>0; decompose(RT R1)>. This context almost structurally matches [KODRATOFF86] the context <RT with at least one participating ET; refine RT> of figure 2 (formalized as <RT R1, nb_values(R1.Collection)>0; refine(RT R1)>), or, equivalently, one of the two other contexts of tree 2.

4. Conclusion

We have presented a learning method for process model improvement. The originality of our work lies in the combination of process meta-modeling with machine learning techniques : the learning method is modeled as an instance of the decision-oriented process meta-model and uses induction techniques to guide the learner and partly automate learning.

Our work is described more thoroughly in [PRAT95b]. We have started implementation in the MENTOR prototype. In the immediate future, we will concentrate on further formalization and experimentation of the learning method.

Although our primary focus is process model improvement, the use of induction techniques is also relevant for trace-based process model definition.

In addition to induction, we believe that case-based/analogical learning techniques could be used in our repository-based framework : case-based reasoning techniques could be applied to develop new applications (solve new problems) by reusing process traces (solutions), and analogical reasoning techniques could be adapted to define new process models by analogy with existing ones.

5. References Available Upon Request