A Data Mining Method to Support Decision Making in Software Development Projects

J.L. Álvarez and J. Mata
Dpt. Ing. Electrónica, Sist. Informáticos y Automática
Universidad de Huelva
{alvarez, mata}@uhu.es

J.C. Riquelme and I. Ramos
Dpt. Lenguajes y Sistemas Informáticos
Universidad de Sevilla
{riquelme, ramos}@lsi.us.es

Key words: Knowledge Discovery in Data, Data Mining, Dynamic Models, Software Engineering.

Abstract: In this paper, we present a strategy to induce knowledge as support decision making in Software Development Projects (SDP). The motive of this work is to reduce the great quantity of SDP do not meet the initial cost requirements, delivery date and the quality of the final product. The main objective of this strategy is to support the manager in the decision taking to establish the policies from management when beginning a software project. Thus, we apply a data mining tool, called ELLIPSES, on databases of SDP. The databases are generated by means of the simulation of a dynamic model for the management of SDP. ELLIPSES tool is a new method oriented to discover knowledge according to the expert's needs, by the detection of the most significant regions. The method essence is found in an evolutionary algorithm that finds these regions one after another. The expert decides which regions are significant and determines the stop criterion. The extracted knowledge is offered through two types of rules: quantitative and qualitative models. The tool also offers a visualization of each rule by parallel coordinate systems. In order to present this strategy, ELLIPSES is applied to a database which has already been obtained by means of the simulation of a dynamic model on a project concluded.

1 INTRODUCTION

Currently, many Software Development Projects (SDP) do not meet the initial cost requirements, delivery date and the quality of the final product. The reason of this situation is the great quantity of attributes that influence on the development process, whose values should be established by the manager of the project. These values depend on the different management policies as well as the maturity level of the organization of development.

In the traditional method, the manager takes a decision of the values of these attributes according to his experience, the initial available resources and the requirements of the project. This decision is a very difficult task because it is necessary to decide each attribute individually and furthermore to bear in mind the attributes altogether.

The simulation of dynamic models for the management of SDP (Abdel-Hamid and Madnick, 1991) produces an improvement as opposite to traditional static models. The dynamic models allow to analyze, before beginning the development, the result of the manager's decision. However, if the manager has uncertainty about many attributes then many simulations will be necessary and he cannot make an exhaustive analysis of the process.

Data mining method can be used to solve this problem (Aguilar et al., 2001). Data mining is a machine learning process that induce patterns from databases (Chen et al., 1996; Fayyad et al., 1996).

Thus, a database can be generate through the simulation of a dynamic model of the development process. Each instance of the database is composed of the attributes (parameters) used on the simulation and the attributes (variables) obtained from the simulation. A data mining method can be applied to this database.

In this paper, we present an overview of this process. Thus, we offer the results after applying our data mining tool, called ELLIPSES (Álvarez et al., 2002), on a database. This database has been generated using the dynamic model for the management of SDP described in (Ramos and Riquelme, 1999).

The remaining part of the paper is organized as follows. In section 2, a brief description of the use of the dynamic model on SDP is introduced. In section 3, the strategy to apply data mining methods on SDP is given. In section 4, the ELLIPSES tool is described.

The case study and experimental results are described in section 5 and the paper is conclude in section 6.

2 DYNAMIC MODELS AND SOFTWARE ENGINEERING

At the beginning of the 90's a significant event took place in software engineering field: the use of the first dynamic model (Abdel-Hamid and Madnick, 1991) applied to SDP. In the last years, new dynamic models for SDP and powerful simulation environments (Stella, Vensim, iThink, PowerSim, etc.) have strongly supported the complex process of decision making. The potential of the simulation of the dynamic models for the formation and training of the manager of projects has been proved in (Ruiz et al., 2001; Dhiel, 1991; Graham et al., 1992). These systems offer to the manager, without risks, the impact that can have on a project the application of management policies.

Thus, the simulation of software projects by dynamic models can be used to accomplish:

A priori analysis: the simulation of the project is made before beginning the development. This analysis guarantees the live cycle of the project applying the analyzed policies.

Monitoring analysis: the simulation of the project is made during the development process. This analysis permits to compare the real state project with the results of the simulation.

Post-mortem analysis: the simulation is made after the development process. This analysis helps to improve the results on future projects.

Obviously, the great quantity of situation that the manager needs to simulate in a project prevents to make an analytical exhaustive. Thus, the manager only will be able to accomplish some simulations. In the next section, a solution for this problem is given.

3 DATA MINING AND SOFTWARE ENGINEERING

Data Mining methods have been on many fields offering excellent results. However, there are no many researchers using these methods on the software engineering field. A possible contribution of the data mining on the software engineering is the improvement of the development process. Thus, data mining algorithms analyze development process databases to induce a set of management rules.

This strategy has an inconvenience: there are not many real databases of development process. This problem can be solved using a simulation environment with a dynamic model for the management SDP.

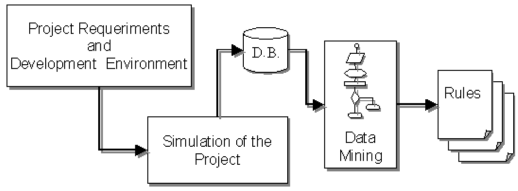

Figure 1: Process to induce management rules from software development projects.

Thus, the attributes with certain uncertainty level are selected by the manager according to his experience, the requirements of the project and the management policies. For each one of them, the manager chooses a range, since he do not know that the value of a attribute will be 37, but know that will be between 30 and 40. The simulation environment will choose a randomly value for these attributes and will simulate the project generating the results of cost, delivery date and quality of the final product. This process is repeated until the database has the necessary instances. Figure 1 shows the sequence of this procedure. Finally, a data mining algorithm is applied on this database obtaining a set of management rules.

The manager can be use the interesting knowledge that these rules offer. There are some criteria in order to choose the rules:

- To choose the rules that cover more instances.
- To choose the rules with less attributes.
- If it is a post-mortem analysis, to choose the rules for those the range of each attribute contains the initial value.
- If it is a priori analysis, to choose the rules whose attributes can be controlled easily.
- To choose the rules that offer the best results.

3.1 Adjustment of the strategy

The described strategy previously has a problem if we want to apply our tool. A database generated according to the previous procedure is composed by attributes: parameters (input attributes) and variables (output attributes), whose values are all continuous. However, ELLISPES tool needs a training set whose attributes are continuous values, but the class is a categorical value.

In order to solve this problem, the continuous variables should be transformed in only one categorical attribute using the experience of the project manager. This way, the manager needs to establish the maximum values to each output attribute (cost, delivery date and quality), in such a way that if these values

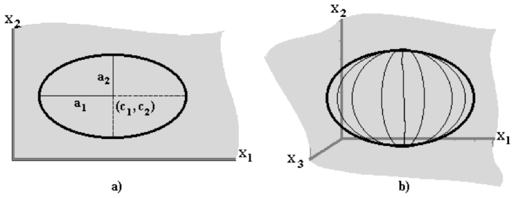

Figure 2: Graphical representation of an hiperellipse a) two b) three dimentional.

are surpassed the project is considered as not acceptable. In forward, each set of maximum values for the variables will be denoted ”cutting section”.

Thus, each ”cutting section” produces a different training set on the generated database, where the instances whose output attributes surpass those values are considered as ”good”, and the instances whose output attributes maintain those values are considered as ”not good”. The ”good” instances are labelled with ”G” and the ”not good” instances are labelled with ”N”. The new training sets are composed of the input attributes and the new class.

4 ELLIPSES TOOL

ELLIPSES tool is a data mining method to induce interesting regions in databases. The core of the algorithm is a evolutionary process (Goldberg, 1989; Holland, 1975; Michalewicz, 1999) whose individuals are elliptical surface in the search space.

The results are shown by quantitative and qualitative rules, and a visualization by parallel coordinates systems is also shown.

In the next sections, the preliminary details, a brief description of the algorithm and the visualization are described.

4.1 Preliminaries

Our method uses conical regions to find the most significant rules. These regions contain the features of each class. This section offers the basic definitions of the models of rules used in our tool.

An hyperellipse (the wrapper) is equal to an ellipse or circumference in a two-dimensional space R^2. An hyperellipsoid (the wrapped volume) is equal to an ellipsoid or circle in a two-dimensional space R^2. Figure 2 offers a graphical representation of these concepts. Figure 2a) represents an ellipse of center (c_1, c_2), greater axis a_1 and smaller axis a_2 to two attributes x_1 and x_2 (two-dimensional space R^2) and figure 2b) shows an hyperellipse to three attributes x_1, x_2 and x_3 (three-dimensional space R^3).

$$\frac{(x_1-c_1)^2}{a_1^2}+\frac{(x_2-c_2)^2}{a_2^2}=1 \quad (1)$$

$$\frac{(x_1-c_1)^2}{a_1^2}+\frac{(x_2-c_2)^2}{a_2^2}\leq 1 \quad (2)$$

$$\frac{(x_1-c_1)^2}{a_1^2}+\frac{(x_2-c_2)^2}{a_2^2}+...+\frac{(x_d-c_d)^2}{a_d^2}\leq 1 \quad (3)$$

The equation of the ellipse in R^2 is shown in 1. The equation of an ellipsoid is shown in 2. This equation is obtained changing $=$ by $\leq$ in the equation of the associated ellipses. Generalizing, in R^d, the equation of an hyperellipsoid is shown in 3.

$$If\ x_1(c_1,a_1)\ and\ ...\ and\ x_d(c_d,a_d) \Rightarrow C_i \quad (4)$$

$$h(x_i,a_i)=\begin{cases} Large\ if\ a_i > 40\%A_x \\ MLarge\ if\ 25\%A_x < a_i \leq 40\%A_x \\ Medium\ if\ 15\%A_x < a_i \leq 25\%A_x \\ MShort\ if\ 5\%A_x < a_i \leq 15\%A_x \\ Short\ if\ a_i \leq 5\%A_x \end{cases} \quad (5)$$

$$If\ x_1(c_1,E_1)\ and\ ...\ and\ x_d(c_d,E_n) \Rightarrow C_i \quad (6)$$

The models of the rules (quantitative and qualitative) used in our tool are based on 1, 2 and 3. Thus, the quantitative model is obtained directly by the equation of the ellipse. This model is shown in 4 and it offers the central c_i value and the extent (width) a_i for each attribute, and the associated class C_i. The qualitative model uses five labels to specify the extent. For each attribute x_i, a E_j label is generated by $h(x_i, a_i)$ function, according to 5, where A_x is $x_{iM} - x_{im}$, x_{iM} is the maximum and x_{im} the minimum for x_i attribute. The qualitative model is shown in 6. The interpretation of these models of rule is very intuitive because the rule does not differ from the typical classification rules. Thus, let be $t : (y_1, y_2, ..., y_n)$, if $y_i \in [x_i - a_i, x_i + a_i] \forall i$ then the item t is associated with the class C_i, according to 4. In the qualitative model, the label establishes the difference between y_i and x_i.

The C_i of an hyperellipsoid is assigned as follows. Let be $t : (x_1, x_2, ..., x_d, C_i)$ item, if i satisfies the equation 3 then the item is within the volume of the hyperellipsoid. Thus, the majority class within the hyperellipse is the associated class to it.

4.2 Evolutionary Process

The main objective of our tool is to induce the regions of the search space with a greater number of

```
ELIPSES Algorithm
T ← Read Training set
Repeat
    iter ← iter + 1
    P_i ← Initialize population on T
    Repeat
        Evaluate P_i on T
        Select the best in P_i to P_{i+1}
        Select 10% in P_i to P_{i+1}
        Crossover P_i individuals to P_{i+1}
        Mutate P_{i+1}
        P_{i+1} is P_i
    Until number generations
    r ← Select the best of P_i
    if alpha(r)>ALPHA THEN add(R, r)
Until (iter=ITER or beta(r)>BETA)
Show R rules
Visualization R by Parallel Coordinates
END.
```

Figure 3: ELLIPSES Algorithm.

the instances belonging to the same class and to permit the human-expert interaction in order to establish some criteria for the search process. The final result shows a reduced and easier interpretable set of rules. ELLIPSES is based on Evolutionary Algorithm (EA). EA are a heuristic search technique that has demonstrated to be robust for a variety of complex search space.

Figure 3 shows the ELLIPSES algorithm. Iteratively, the EA finds the best hyperellipse r based on the number of positive and negative items in the hyperellipsoid. Let be $alpha(r)$ the percentage of the same class items in r, if $alpha(r)$ is greater than the predefined human-expert percentage $ALPHA$, then region r is considered. This process is repeated until reaching a predefined human-expert number of rules or predefined human-expert percentage $BETA$, where $beta(r)$ is the number of covered cases. Finally, the rules are shown according to 4 and 6 (quantitative and qualitative models), and they are shown by parallel coordinate systems.

4.2.1 Representation

An individual (a feasible solution) is a set $I = \{c_1,...,c_d,\ a_1,...,a_d\}$ where d is the number of attributes and $c_i, a_i \in \Re$ are, respectively, the center and extent of the x_i attribute and they represent the equation of an hyperellipsoid according to 3. Figure 4 shows a graphical representation of the individuals.

In practice, an individual represents a search space region. Each region will be associated to a class that will be deduced by the majority class of the data items

c_1	c_2	$\cdots$	c_d	a_1	a_2	$\cdots$	a_d

Figure 4: Representation of an individual in the ELLIPSES evolutionary algorithm.

in the hyperellipsoid.

4.2.2 Fitness function

The fitness (or merit as a solution) of an individual is obtained by training set item analysis. An item can be in or out of the hyperellipse. The out items are ignored. The different classes of the items in the hyperellipse are counted and the associated class to the individual is the majority class. Thus, the items with the same class are positive cases and the items with different classes are negative cases.

Furthermore, next iteration must direct the evolutionary process to other regions. Thus, the positive cases covered by discovered rules are considered covered cases. Finally, our method needs to obtain the greatest region. Thus, the amplitude of the hyperellipse is the hyperellipse volume divide by search spaces volume.

$$f(i) = Pos(i) - Neg(i) - Cover(i) * FC + Ampl(i) \quad (7)$$

Our algorithm maximizes the fitness function f for each individual i. The fitness function is given in 7, where $Pos(i)$ and $Neg(i)$ are the positive and negative cases in the hyperelipsoid that represent the individual i, $Cover(i)$ are the covered cases by previous hyperellipses, FC is the coverture factor and $Ampl(i)$ is the hyperellipse amplitude. Coverture factor (FC) is a value in the interval $[0..1]$, and it offers the possibility of relaxing the covered cases, so, if FC is closed to 1, then the covered cases are considered negative cases, and if FC is closed to 0, then the covered cases are ignored.

4.2.3 Genetic Operators

There are tree genetic operators: selection, crossover and mutation. To form a new population (the next generation), the individuals are selected according to their fitness by the selection operator. Many selection procedures are currently in use, our algorithm uses roulette wheel procedure, where individuals are selected with a proportional probability to their relative fitness. This ensures that an individual is chosen in a expected number of times approximately proportional to its relative performance in the population. Thus, high-fitness (good) individuals stand a better chance

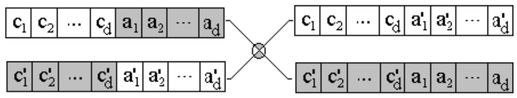

Figure 5: The middle point crossover operator in the ELLIPSES evolutionary algorithm.

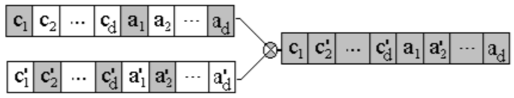

Figure 6: The uniform point crossover operator in the ELLIPSES evolutionary algorithm.

of selecting, while low-fitness individuals are more likely to disappear.

Selection cannot introduce any new individuals into the population. These individuals are generated through cross-over and mutation operators. Crossover operator is performed by selecting two individuals called parents, and generating new individuals called offspring. In our algorithm, the crossover operator has two components: the middle point crossover and the uniform crossover. They are performed with a probability p_{cross} that chooses between the middle point crossover and the uniform crossover. The middle point crossover randomly splits the individuals in two parts. Then the fragments are exchanged generating two new individuals. Figure 5 graphically shows this process. The uniform crossover decides , independently for each coefficient of an individual, whether it contribute or not to the new individual. An example of this procedure is shown in figure 6.

$$v_{ij} = v_{ij} \pm Quant * PerMut * v_{ij} \qquad (8)$$

Finally, the mutation operator is introduced to prevent premature convergence to local optimum by randomly sampling new points in the search space. Three variants are implemented: center mutation, amplitude mutation and extreme mutation. Mutation is performed with probability p_{mut} on an individual. When an individual must be mutated, a probability chooses between the different operators. The center and amplitude mutation operators alter the center $(c_1, ..., c_d)$ and the extent $(a_1, ..., a_d)$ of the hyperellipse, respectively, according to 8, where v_{ij} is the factor to alter, $Quant$ and $PerMut$ take their values from [0..1], $Quant$ is the random quantity that v_{ij} is altered and $PerMut$ is the percentage of mutation that determines how the mutation influence on v_{ij}. The extreme mutation operator alters both center (c_i) and extent (a_i) of an attribute (x_i). Thus, the mutation let the middle value of $x_{iM} - x_{im}$ to c_i and let $\frac{x_{iM} - x_{im}}{2}$ to a_i. The objective of this operator is to cover the attribute.

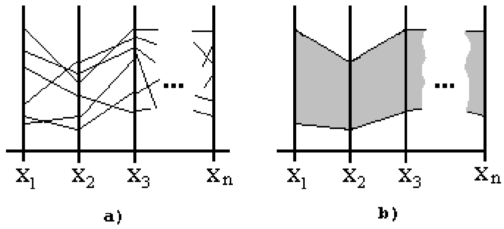

Figure 7: ELLIPSES Parallel Coordinate Systems.

4.3 Parallel Coordinate Systems

Although our tool offers two models of rules and the qualitative model is easily interpreted, sometimes it is necessary to provide the information using another philosophy. Thus, a visualization of the relationships among the attributes offers a good support to the expert. The visualization technique used in our algorithm is shown in this section. This technique offers the relationships among attributes by parallel coordinates (Inselberg, 1985).

A parallel coordinate system is composed by a set of parallel axes separated by a fixed distance. Each axis corresponds with an attribute and they are escalated on the range of the attribute. Thus, d axes are necessary to represent d attributes. In this system, a line represents each data item. This line intersects with each axis on the value of the item for that attribute. Figure 7a) shows the traditional parallel coordinate system.

In our method, each region is represented on a parallel coordinate system. But, all data items in a region are not represented on parallel coordinate system. Thus, only the minimal value and the maximal value, for each attribute, are represented on each axis and these values are joined by filled polygonal. Figure 7b) offers an example of this method. The internal lines are eliminated. The objective of this variant is to offer a clearer and compact vision of the relationships between the attributes.

5 CASE STUDY

In this case study, we will analyze the influence of the policies of personnel management on the variables of a SDP with restrictions in the delivery time.

Thus, the used attributes on this study are: the average delay of the new technicians' adaptation that has incorporated to the project (technicians' integration), the average delay of the technicians' exit of the project (technicians' discharge), the average delay of the new technicians' incorporation in the project (technicians' recruiting) and the maximum delay on the delivery time. Table 1 shows more information

Table 1: SDP attributes

Attributes	Description	unit	initial value	range
inputs				
ASIMDY (A)	The average delay of the new technicians' adaptation that has incorporated to the project	days	20	[5,15]
HIREDY (H)	The average delay of the new technicians' incorporation in the project	days	30	[5,10]
MXSCDX (M)	The maximum delay on the delivery time	%	1.16	[1, 1.2]
TRNSDY (T)	The average delay of the technicians' exit of the project	days	10	[5,10]
outputs				
JBSZMD	Necessary effort to carry out the project	technicians-days	1111	-
SCHCDT	Development time	days	320	-
ANERPT	Final product quality	errors/tasks	0	-

Table 2: "Cutting section" on CRCCRT database

CRCCRT	JBSZMD	SCHCDT	ANERPT
Cutting section 1	-	≤ 352 (10%)	≤ 0.45 (0.45%)
Cutting section 2	-	≤ 352 (10%)	≤ 0.35 (0.35%)

about these attributes. Furthermore, this table shows the variables that will be analyzed: delivery time, cost and quality of the final product.

The database, that we will call CRCCRT, has been generated using a quick recruiting with restriction on the delivery time strategy[1]. Thus, the attributes related with personnel management (ASIMDY, HIREDY and TRNSDY) take value inside the interval considered as quick and the attribute related with the delivery time management (MXSCDX) takes values inside the intervals of fixed term and moderate term.

In this study has been analyzed two "cutting sections" of the CRCCRT database. The values of the variables for each "cutting section" are shown in table 2. The objectives of each "cutting section" are different according to the used values. Thus, the objective of the cutting section 1 is to induce management rules where the delivery time cannot overcome at the estimated (320 days) in more than 10% (352 days) and quality of the project cannot greater than 0.45 errors/tasks. The objective of the cutting section 2 is equal concerning delivery time, but there are restrictions stronger concerning quality. That is to say, the final product quality cannot greater than 0.35 errors/task. This new restriction will reduce the number of instances and, consequently, the reliability of the induced knowledge, but, however, the quality of the final product will be greater than in the previous cutting section.

[1]A quick recruiting strategy has been analyzed since some researches (Ramos and Ruiz, 1998) have shown that the fulfillment of the delivery time is favored by these policies though the cost could be increased.

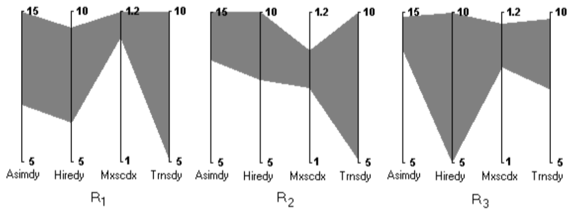

Figure 8: Visualization by parallel coordinates of the management rules induced from cutting section 1.

5.1 Cutting section 1: restrictions of time and quality

If a data mining method, e.g. ELLIPSES, is applied to the training set generated by the cutting section 1, then a set of rules for the management of SDP can be obtained. These rules will offer information on the policy of personnel management (ASIMDY, HIREDY, TRNSDY) and the maximum postponement of the delivery time of the project (MXSCDX) when the objective is to obtain a final product with a delivery time (SCHCDT) less than 352 days and a quality (ANERPT) less than 0.45 error/tasks; regardless of the cost or the necessary effort to carry out the project (table 2, cutting section 1).

The rules induced by ELLIPSES, labelled as "good", on this training set are:

```
R1:  A(11.9,3.14) & H(7.9,1.57) & M(1.185,0.015)
R2:  A(13.4,1.65) & H(9.3,1.53) & M(1.160,0.018)
R3:  A(13.6,1.01) & M(1.178,0.021) & T(8.6,1.17)
```

The qualitative models of these rules are[2]:

```
R1:  A(11.9,ML) & H(7.9,ML) & M(1.185,MS)
R2:  A(13.4,M) & H(9.3,ML) & M(1.160,MS)
R3:  A(13.6,MS) & M(1.178,MS) & T(8.6,M)
```

The visualization o graphical representation by parallel coordinates of these rules are shown in figure 8.

The interpretation of this knowledge would be:

[2](L) Large, (ML) Medium Large, (M) Medium, (MS) Medium Short and (S) Short.

- Figure 8R1): Asimdy, defined in the interval [5, 15], takes middle high values (with center 11.9 and a margin of ± 3.14), Hiredy, defined in [5,10], takes middle high values, but without reaching the extreme values (center 7.9 with margin of ± 1.57) and Mxscdx, defined in [1,1.2], takes very high values (center 1.185 and margin of ± 0.015), practically, to the extreme.
- Figure 8R2): Asimdy takes high values (center 13.4 with a margin of ± 1.65), Hiredy takes high values (center 9.3 and margin of ± 1.53) and Mxscdx takes middle high values (center 1.160 and with a margin of ± 0.018).
- Figure 8R3): Asimdy takes high values, but without reaching the extreme values (center 13.6 with margin of ± 1.01), Mxscdx takes high values, but without reaching the extreme values (center 1.178 and margin of ± 0.021) and Trnsdy, defined in the interval [5,10], takes middle and high values, but without reaching the extreme values (center 8.6 and with a margin of ± 1.17).

5.2 Cutting section 2: restrictions of time and greater level quality

The set of rules for the management of SDP that can be obtained from the training set generated by the cutting section 2 offer information on the policy of personnel management (ASIMDY, HIREDY, TRNSDY) and the maximum postponement of the delivery time of the project (MXSCDX) when the objective is to obtain a final product with a delivery time (SCHCDT) less than 352 days and the quality of the final product (ANERPT) is less than 0.35 error/tasks; regardless of the cost or the necessary effort to carry out the project (table 2, cutting section 2). Thus, the cutting section 2 establishes a greater level of quality that the cutting section 1.

The rules induced by ELLIPSES, labelled as "good", on the training set obtained by cutting section 2 from CRCCRT database are:

```
R1:  A(13.5,1.57) & H(8.7,0.28)
R2:  A(12.4,2.04) & M(1.175,0.006) & T(6.9,1.19)
R3:  A(14.8,0.74) & H(9.9,1.20) & M(1.095,0.055)
```

The qualitative models of these rules are[2]:

```
R1:  A(13.5,M) & H(8.7,S)
R2:  A(12.4,M) & M(1.175,S) & T(6.9,M)
R3:  A(14.8,MS) & H(9.9,M) & M(1.095,ML)
```

The visualization of the obtained rules by ELLIPSES on this training set is shown in Figure 9.

The interpretation of this knowledge would be:

- Figure 9R1): Asimdy takes high values (center 13.5 with a margin of ± 1.57) and Hiredy takes high values, but without reaching the extreme values (center 8.7 with a margin of ± 0.28).

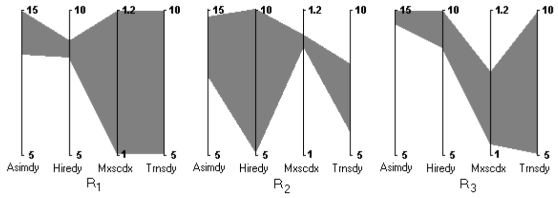

Figure 9: Visualization by parallel coordinates of the management rules induced from cutting section 2.

- Figure 9R2): Asimdy takes high values, but without reaching the extreme values (center 12.4 with a margin of ± 2.04), Mxscdx takes high values, but without reaching the extreme values (center 1.175 and margin of ± 0.006) and Trnsdy takes low and middle values (center 6.9 with a margin of ± 1.19).
- Figure 9R3): Asimdy takes very high values (center 14.8 with a margin of ± 0.74), Hiredy takes high values (center 9.9 and margin of ± 1.20) and Mxscdx low and middle values, but without reaching the extreme values (center 1.095 with a margin of ± 0.055).

5.3 Analysis of the induced knowledge

The manager can choose the management rule that he wants to apply, when the rules have been induced. This decision must take into consideration others features of the project: the initial available resources, the requirements of the project, the management policies as well as the maturity level of the organization of development. That is to say, all induced rules cannot apply directly.

Analyzing the obtained results of both cutting section, the best rule to apply is R1 of the cutting section 2 according to the criteria in order to choose the rules established in section 3. The rule R1 is very easy to apply, because the manager only must supervise two attributes (ASIMDY and HIREDY) in order to obtained an acceptable level of quality. But this rule has an inconvenience with regard to other induced rules: the intervals for ASIMDY and HIREDY are less wide than in other rules. Thus, the manager does not have a lot of margin when this rule is applied.

Figure 10 shows the evolution of the project if the rules R1 and R3 of the cutting section 2 are applied. Analyzing this figure, we can see that the delivery time is similar in both rules (350 and 351 days, respectively), but the necessary effort to carry out the project in R1 (2615 technicians/days) is lower than the effort in R2 (2769 technicians/days). Summarizing, the two rules obtain similar results of delivery time and quality, but R1 reduces the cost of the project

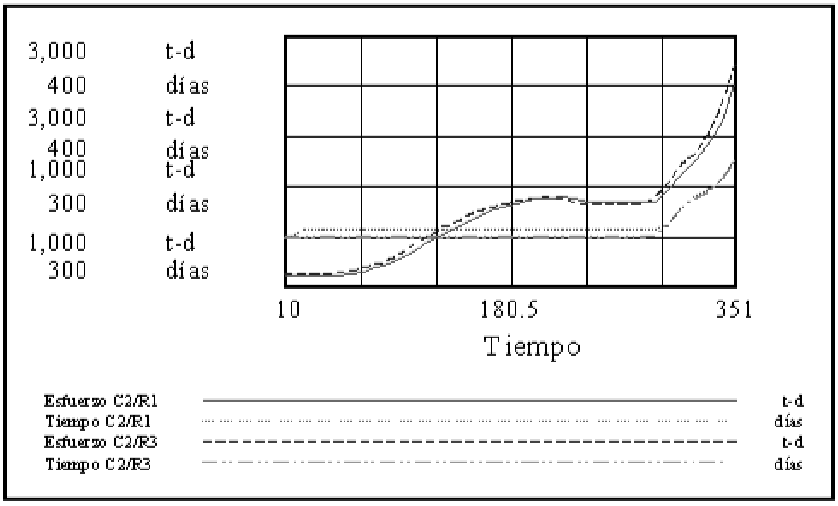

Figure 10: Results of the project simulation with the rules R1 and R3 of the cutting section 2.

diminishing the necessary effort of development.

6 CONCLUSION

In this paper, we present a strategy to induce knowledge in databases of software development projects. The databases are generated by simulations of dynamic models for the management projects. A data mining tool analyzes these data inducing the new knowledge.

This strategy can be used to make three analysis: a priori analysis, monitoring analysis and post-mortem analysis

We uphold the use of this new strategy as opposed to traditional static model or simple dynamic models.

Acknowledgments

This work has been supported by Spanish Research Agency CICYT under grant TIC2001-1143-C03-02.

REFERENCES

Abdel-Hamid, T. and Madnick, S. (1991). *Software Project Dynamics: an Integrated Approach*. Prentice-Hall.

Aguilar, J., Ramos, I., Riquelme, J., and Toro, M. (2001). An evolutionary approach to estimating software development projects. *Information and Software Technology*, 43(14):875–882.

Álvarez, J., Mata, J., and Riquelme, J. (2002). Mining interesting regions using an evolutionary algorithm. In *ACM SIGAPP Symposium on Applied Computing (SAC)*, pages 498–502.

Chen, M., Han, J., and Yu, P. (1996). Data mining: An overview from database perspective. *IEEE Trans. on Knowledge and Data Engineering*, 8(6):866–883.

Dhiel, E. (1991). Participatoty simulation software for managers: The design philosophy behind microworld creator. *European Journal of Operational Research*, 59(1):210–215.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Graham, A., Morecroft, J., Senge, P., and Sterman, J. (1992). Model-supported case studies for management education. *European Journal of Operational Research*, 59(1):151–166.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Inselberg, A. (1985). The plane with parallel coordinates, special issue on computational geometry. *The Visual Computer*, 1:69–97.

Michalewicz, Z. (1999). *Genetics Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.

Ramos, I. and Riquelme, J. (1999). The dynamic models for software development projects and the machine learning techniques. In *International Conference on Product Focused Software Process Improvement*.

Ramos, I. and Ruiz, M. (1998). Aplicación de diferentes políticas de contratación de personal en un proyecto de desarrollo de software. In *IV International Congress on Project Engineering*.

Ruiz, M., Ramos, I., and Toro, M. (2001). A simplified model of software project dynamics. *Journal of Systems and Software*, 59(3):299–309.