

DATA MINING FOR THE MANAGEMENT OF SOFTWARE DEVELOPMENT PROCESS

J.L. ÁLVAREZ and J. MATA

*University of Huelva, Escuela Politécnica Superior, La Rábida
Huelva, 21071, Spain*

and

J.C. RIQUELME

*University of Sevilla, Escuela Técnica Superior de Ing. Informática, Avda. Reina Mercedes
Sevilla, 41012, Spain*

Received (received date)

Revised (revised date)

Accepted (accepted date)

In this paper we present a new method for the application of data mining tools on the management phase of software development process. Specifically, we describe two tools, the first one based on supervised learning, and the second one on unsupervised learning. The goal of this method is to induce a set of management rules that make easy the development process to the managers. Depending on how and to what is this method applied, it will permit an a priori analysis, a monitoring of the project or a post-mortem analysis.

Keywords: Project Management; Software Metrics; Software Project Simulation; Data Mining.

1. Introduction

In spite of its great diffusion in the last years, the application of data mining techniques to the field of software engineering has not been very much used, nevertheless, it is presented as a task that can offer great advantages and possibilities to this area. One of the possibilities in this respect, although not the only one, consists in the application of data mining techniques to databases on Software Development Projects (SDP). Thus, if we could have a database with the possible values of the initial parameters of a software project and the results at the end of the project, with regard to cost, time and quality for each one of these values, the data mining techniques could induce a set of rules that should permit to relate the initial parameters with the results of the project. It is obvious that this set of rules would offer to the project manager an extraordinary knowledge that would make easy both, the decisions to take in order to undertake new projects and the actions to realise in a development project or the analysis of an already finished project.

This idea has not arisen spontaneously, but as a necessity requested by the software engineering. All of us know the "crisis" or chronic affliction, as it is described by Diehl [1], that has suffered the software development: a great amount of the undertaken software development projects has not fulfilled the initial expectations of delivery time, cost and quality.

This situation is due, mainly, to the estimations or management policies that the manager of the project must carry out at the beginning of the project. Since the development of software projects began on a great scale, one of the main concerns for the managers has been to optimise the attributes that measure the viability of the final process. At the beginning, such task was not easy to carry out, since the manager himself was who, basing on his own experience, estimated the values where the attributes (management policies) that affect the development process had to move. The main problem in this case, lies in the amount of attributes that must be estimated, and besides, it is not enough to establish their values in an independent way, but it is also necessary to know the influence of ones on the others.

With the incorporation of the simulation of the development process by dynamic models [2], a quite significant advance in the management of SDP was achieved, since these are simulated, without any risk, before undertaking the real project. Thus, the manager of the project decides the values for the attributes and simulates the process to verify its behaviour. Obviously, this means an improvement, but again, the amount of existing possibilities does not permit an exhaustive verification of the process.

Data mining techniques can be used to solve this problem. In this case, the task of data mining techniques is to give the manager of the SDP a knowledge where to be based on. This knowledge helps the manager to estimate which are the most appropriate values, where the attributes that have an influence on the project must move, in order to obtain the desired values at the end of the project with regard to delivery time, cost and quality.

It is necessary to emphasise that the application of data mining techniques to the software development process has an added problem that does not exist in other fields: the lack of real databases with information on the parameters that have conditioned the development of other software projects. Nowadays, such deficiency is solved thanks to the existence of powerful simulation systems and to the possibility of constructing dynamic models for the simulation of software projects. For that, the manager establishes the ranges for the doubtful attributes of the project, and by means of the simulator a database with the possible values for these parameters and with the result of the project (time, cost and quality) is generated.

In this paper we offer a vision of the application of data mining techniques to SDP with the aim of giving to the manager a new method capable of providing a knowledge that will help him in the management of the project. This method will permit the manager, depending on its application, to have a priori information in the case of starting a new project, to have information for the following of a

project in progress or information for the evaluation and analysis of an already finished project. This project uses as SDP simulation model the one described in [3] and the tools GAR [4] for unsupervised learning and ELLIPSES [5] for supervised learning. Nowadays, similar works are using datamining tools for the induction of management rules for software development project [6].

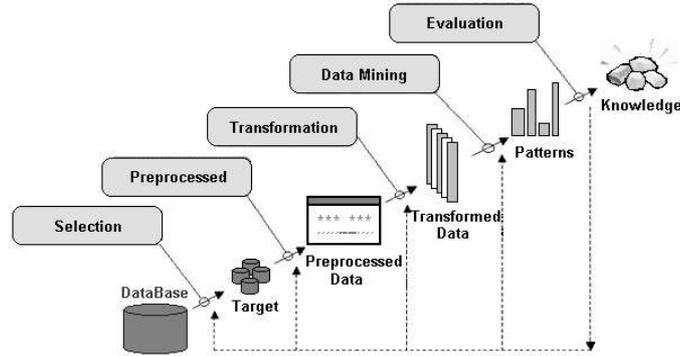


Fig. 1. Knowledge Discovery in Database.

1.1. Process of Knowledge Discovery: Data Mining

The knowledge discovery in database can be defined, in an informal way [7], as the non trivial process for the induction, departing from a set of data, of previously unknown, recent, potentially useful and understandable patterns (knowledge). As figure 1 shows, it is a process made up of several steps where data are selected, pre processed and transformed before we can discover the knowledge. Next, we offer a brief description of the goal to be covered in each step.

Selection. It is the step where data that are object of interest are chosen departing from the data initially available.

Pre-processed. Selected data are processed with the aim of removing noise (outliers) and of treating the absent values.

Transformation. The pre-processed data are treated for the reduction of attributes or instances, fact that will make easy the application of the mining algorithm.

Data Mining. This step takes the previously treated data and induces the set of patterns.

Evaluation and Interpretation. The induced patterns are analysed by part of the experts with the aim of deducing useful knowledge.

Despite that all the steps have an important influence on the process, the data mining phase is, in our opinion, the most important within the whole process. In this phase, it is necessary to decide the task and the algorithm to employ on the data. There are two tasks of special interest in the mining step: classification and association.

The aim of classification is to induce a set of rules that permits to predict, departing from a data set, the class of a future instance [8]. In a more accurate way, given a data set D (training set), composed by attributes $X = \{x_1, x_2, \dots, x_d\}$ and a class E that takes the discrete values e_1, e_2, \dots, e_l , it has as goal to induce a classification model depending on the attributes x_1, x_2, \dots, x_d , that permits to assign a determined label e_1, e_2, \dots, e_l , for the future instances, previously unknown with a certain degree of certainty.

Even though the association concept is quite wide and includes different forms of representing and interpreting the knowledge (correlations, sequential patterns, negative associations, etc.), association rules has been the most outstanding ones within data mining. The aim of the mining association rules process consists, basically, in discovering if the presence of a set of pairs attribute-value that it is found in a determined number of records can be used to induce the presence of another set of pairs attribute-value in the same transactions [9]. In a more formal way, if $X = \{x_1, x_2, \dots, x_d\}$ is the set of all the attributes of a database D , an association rule is an implication $C_1 \rightarrow C_2$, where $C_1, C_2 \subseteq X$ and $C_1 \cap C_2 = \emptyset$. C_1 and C_2 are respectively called antecedent and consequent of the rule and it is read "if C_1 then C_2 ".

The probability of a given association rule to be fulfilled is known with the name of confidence. Besides this measure, that shows, up to a certain point, the importance of a rule, there is another one to indicate the covering of the rule, that is to say, the percentage of records in which the rule is fulfilled, and has the name of support.

2. Data Mining Techniques

The selection of the algorithm or tool to be used in the data mining step is an essential element in such step. In this paper we use two tools that offer different visions of the knowledge we want to obtain. Thus, we use an association rule algorithm, named GAR [4] and a classification algorithm named ELLIPSES [5], both based on Evolutionary Algorithms [10].

In this paper we offer the details of each one of these algorithms, without studying in depth the implementation details.

2.1. GAR: Association Rules

GAR is a algorithm that discovers association rules in databases that have numeric attributes. The core of the algorithm is an evolutionary process that is in charge of finding, in an automatic way, the most suitable intervals for each attribute

that forms part of the rule. This is the main advantage of this technique with regard to others, since it is not necessary to carry out a previous discretization of the continuous domains.

$$\begin{aligned} & \text{if } X_a \in [l_{ai}, l_{as}] \text{ and } X_b \in [l_{bi}, l_{bs}] \text{ and } \dots \text{ and } X_z \in [l_{zi}, l_{zs}] \\ & \text{then } X_k \in [l_{ki}, l_{ks}] \text{ and } \dots \text{ and } X_j \in [l_{ji}, l_{js}] \end{aligned} \quad (1)$$

The model of rule we present has the format described in equation 1, where X_i are the attributes that form the rule and $[l_i, l_s]$ are the minimum and maximum values of the interval discovered by the algorithm.

The fitness of a rule is determined by several factors:

- **Support.** This parameter rewards rules with a high value of support.
- **Confidence.** It is the other most used parameter to evaluate the quality of association rules together with support. Therefore, it is also a positive parameter that makes that the search process tend to find rules with high values of confidence.
- **Before covered cases (recov).** It is used to indicate that an attribute of a record has previously been covered by a rule, that is to say, it belong to one of the rules discovered by the algorithm. With this, we get that the algorithm tend to discover different rules in later searches.
- **Number of attributes (natrib).** Association rules with a high number of attributes give an information with more quality to the final user. But we are also aware that, in many cases, it is difficult to find rules with a high number of attributes or that the user is only interested in finding associations among a limited number of attributes. The parameter natrib rewards the number of attributes that has a rule.
- **Amplitude (ampl).** This parameter controls the amplitude of the intervals of the attributes that form the rules. It is a negative parameter that penalises individuals with very wide intervals.

Equation 2 shows how the fitness of a rule is obtained.

$$\begin{aligned} \text{Fitness} = & (\text{support} * w_s) + (\text{confidence} * w_c) - (\text{recov} * w_r) + \\ & + (\text{natrib} * w_n) - (\text{ampl} * w_a) \end{aligned} \quad (2)$$

Since, it is very difficult to establish some fixed values valid for the solving of all the problems, each parameter of the function is weighed up to increase or to decrease its effect.

Besides of presenting the rule in the format "if-then", we offer a graphic representation by means of a parallel coordinate system [11] of each discovered rule. In this system only appear the attributes that form part of the rule. By means of this representation we show, in a graphic way, the intervals of the attributes, in such a way that a qualitative reading of the rule can be done.

2.2. ELLIPSES: Classification Rules

ELLIPSES is a classification system based on an evolutionary algorithm. The evolutionary process, iterated for each rule to be induced, has as goal function to find the best region in each iteration. The induced regions are elliptical spaces, within the search area, described by equation 3.

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

$$Fitness = clasif - nclasif - recov * frc + vol \quad (4)$$

The fitness of a region is determined by the number of instances that exist inside it and by its volume (*vol*). As for the instances, we have to distinguish between the instances (success) of the majority class, established as class of the region (*clasif*), the instances (errors) of other class (*nclasif*) and the instances (before covered cases) of the class previously covered by other rules (*recov*).

This last term can be weighted up ($frc \in [0, 1]$) by the expert, depending on its necessities with regard to the rules overlapping. If *frc* takes values near to one, the rule overlapping will be much lesser than if this takes values near to zero. In this way, the fitness of a region is determined by equation 4.

$$if \ x_1(c_1, a_1) \ and \ x_2(c_2, a_2) \ and \ \dots \ and \ x_d(c_d, a_d) \ then \ e_i \quad (5)$$

$$h(x_i, a_i) = \begin{cases} Large \ if \ a_i > 40\%A_x \\ MLarge \ if \ 25\%A_x < a_i \leq 40\%A_x \\ Medium \ if \ 15\%A_x < a_i \leq 25\%A_x \\ MShort \ if \ 5\%A_x < a_i \leq 15\%A_x \\ Short \ if \ a_i \leq 5\%A_x \end{cases} \quad (6)$$

$$if \ x_1(c_1, E_1) \ and \ x_2(c_2, E_2) \ and \ \dots \ and \ x_d(c_d, E_n) \ then \ e_i \quad (7)$$

The information is presented by two models of rules: quantitative and semi-qualitative, and also by a graphic representation. The quantitative model is directly deduced from equation 3. Thus, a rule in the quantitative model will have the format described in equation 5, where c_i and a_i represent, respectively, the centre and the amplitude for attribute i ; and e_i the class or majority label in the region.

In the semi qualitative model, the information of the amplitude for each attribute is replaced by a label obtained depending on it, according to equation 6. In equation 7 we show the format of this model, where E_i are the labels which represent the amplitude.

The graphic representation we offer, the same that in GAR, consists in a system of parallel coordinates for each rule. The difference is that the representation for ELLIPSES shows all the parameters of the model, and we can distinguish in it those that do not influence on the decision because they can take any value of their domain.

3. Software Engineering, Dynamic Systems and Data Mining

A priori, the design of a method for the union of these three concepts can seem complex. But, the application of dynamic systems to SDP is something quite known and used since the early 90s. Simply, what we need is to incorporate the term "data mining" to this formula.

On the other hand, data mining is a task that is giving huge benefits in a countless amount of knowledge fields, so the fitness seems to be easy. And, so it is. With the pair dynamic systems and SDP we can get to obtain a data set that describes, depending on the initial parameters (input) of the project, how this evolves, according to its expected results of cost, time and quality. This database is the link and departing point for data mining.

In the following sections we offer, in a more exhaustive way, the relation between this three concepts and we detail the influence of each one of them on the method proposed in this paper.

3.1. *Dynamic Systems for Software Development Projects*

The software development process can be defined as a set of tools, methods and practices that are used to produce software. As any other organisation, those devoted to software development keep among their main aims the production of software according to the initial planning, besides a constant improvement, with the purpose of achieving the last three goals of any production process: high quality and low cost in the least time.

The reduction of the software development cycle without risking the quality of the final product has become a basic goal. Not only for being an advantage from the point of view of the customer but for offering, according to [12], a series of important benefits among which we can emphasize : a larger commercial life of the product and the possibility of beginning to produce as later as possible ending within the limit, what permits to employ the last technological advances available at the moment.

Most of the advances produced in the Software Engineering field, directed to the reduction of the development cycle, try to achieve their goals by a substantial improvement of the development technology employed. Such as it is assumed in the model CMM (Capability Maturity Model) of the SEI (Software Engineering Institute) and in the proposal of the norm ISO 9000. The CMM model classifies software development organizations on a scale from one to five resting on the maturity of the process that take place in them. According to [13] we can assure that as a software development organization progress from a level to the one immediately superior, the length of the development cycle diminish, always guaranteeing the quality of the final product.

Traditionally, we have tried to face the known problem of the software crisis from the point of view of the development technology used. Thus, significant advances relating to the use of new methodologies and development approaches, CASE tools,

code reuse, etc, have been produced. At the beginning of the 90s, a significant leap in the management of software development projects is produced with the appearance of the first dynamic model applied to this field [2], which models the process carried out in the management of such projects.

In the last years, a significant advance in the field of tools for project management and in the advice for the complex process of decision making has taking place, with the appearance of dynamic models for SDP and powerful simulation environment (Stella, Vensim, iThink, PowerSim, etc.). These tools have been called "Software Project Simulators" (SPS), and allows the manager of the project to test with different management policies without cost, making easy the most adequate decision taking [14].

The making of dynamic models can constitute a formal methodology according to which we can express the knowledge on the system. Besides, the own model building process forces the researchers to have a high knowledge about which are the key parameters and how they are related among them.

The potential of simulation models for the education and training of the project managers is stated in [1], [15],[3]. Thanks to the simulation environment, the managers can be placed in front of real situation they can find in practice and that permit them to acquire experience without risk. Besides, the formal character of the dynamic model allows the managers to share their viewpoints in a not ambiguous way and to harmonise the mental models of the process. Definitively, a software project simulator allows to carry out the following analyses:

- An a priori analysis of the project: consisting in simulating the project before starting the development. These simulations can be carried out under different management policies and/or different estimations of initial resources and/or different development technologies. The knowledge of these management rules can be obtained before starting the project performance and it is going to guarantee the obtaining of good results for the variables that the project manager considers of interest (delivery time, cost, quality,...). Besides, the obtaining of management rules permits the manager to analyse which are the most significant management policies in order to achieve the initials goals defined in the SDP and to know if these policies can be applicable or not.
- Monitoring of the project: simulation of the project during its development to adapt the realised estimation to its real evolution. For this application it is necessary to have a metric system that permits to pick up the real situation of the project and its comparison with the results of the simulation.
- A post-mortem analysis of the project: simulation of an already finished project in order to know how the obtained results could have been improved.

In other words, by means of the SPS we can answer questions such as "What would happens if...?", before beginning the project, "What is happening..?", during its development, and "What would have happens if...?" once the project is finished.

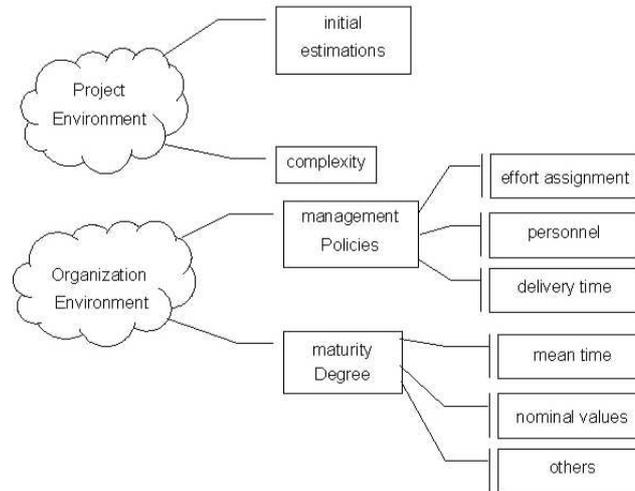


Fig. 2. Classification of the attributes of a dynamic model for SDP.

3.2. Data Mining in Software Development Projects

Once we have arrived at this point we can ask: Could it be possible for the project manager to carry out all the possible situations and to analyse their results? The answer is no, unless the manager could spend all his life doing this. Thus, it is necessary an automatic mechanism able to induce some patterns that describe the behaviour of the project in a global way. In order to do that, we will use data mining together with dynamic models.

Dynamic models for SDP include a set of attributes that allows us to define the management policies that can be applied to the projects, both those related to the project environment and those related to the development organization and the maturity level of the organization (see figure 2).

Inside the group "Project environment", we insert the attributes related to the initial estimations of the project and to the complexity of the same. In the group named "Organization environment", appear the attributes related to the different management policies that can be applied to SDP and with the maturity degree of the own development organization. Within the subgroup "Management policies" we collect the different parameters and tables (attributes) that define the management policies related to effort assignment, management policies related to personnel (hiring, dismissal, adaptation, etc.) and management policies related to restrictions on the project delivery time. On the other side, the subgroup "Maturity degree", fundamentally includes the attributes bound to the average time in which certain activities of the project are realised and the nominal values used. The subgroup "Others..." contains the attributes related to the maturity degree of the organization

but that are difficult to insert in the previous sections.

Once the attributes of the model are defined, the manager of the project must decide which are the variables that are going to be analysed. The usual options are the variables that define the project development: delivery time, cost, number of generated errors, number of corrected errors, average development productivity, etc. Therefore, the management policies that the mining process can find will be those that relate attribute values with the variables over whose influence we want to establish rules. Every time we assign a specific value to each attribute of the model, this stay completely defined and by its simulation we will obtain some values for this variables.

To generate a set of training cases for the mining process, the project manager must choose a range of values for the attributes of the model over which there is some level of uncertainty. For example, the manager of the project does not initially know, if the average dedication of the technicians in the project will be exactly of 72% or 50%, but he knows for sure that the value will be between 50% and 80%.

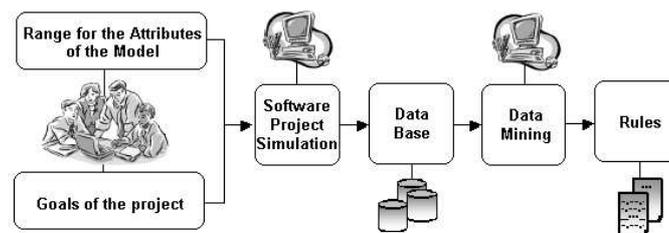


Fig. 3. Process of obtaining management rules for a SDP.

Next, the simulation tool randomly generates, for every attribute, a value in each one of those intervals. Then, a tuple of values for the variables resulting from the simulation, will correspond to each tuple of attributes so defined,. In this way, we generate a record for the database with the values of the attributes and the values obtained for the variables of the project that we wish to analyze, that is to say , a specific scenario of the project. Doing again this process a certain number of times, we can obtain a training file that serves as base for the mining process. This process is shown, in a graphic way, in figure 3 [6].

In this case, the mining process must find behavior patterns that relate the values of the attributes with the ones of the variables. These patterns can be easily interpreted by an expert and transformed into a knowledge on the project that is being analyzed.

3.3. Processing of Data and Techniques

The database generated by the previous procedure consists of a set of parameters or, input attributes, that take values in a continuous domain and several variables

or, output attributes, that take values in a continuous domain depending on the evolution of the project for those parameters. Thus, the learning process would be a supervised learning process with all the continuous attributes, where there is more than one output attribute.

If we analyze again the tools described in section 2, "Data mining techniques", we can check that the first one is an unsupervised learning technique and that the second one is a supervised learning technique, where the input attributes take values in a continuous domain, but the output attribute or class must be unique and takes values in a discrete domain.

In this way, in order to analyse the database generated by the former simulation process with the proposed techniques, it is necessary to carry out an adjustment both of the data and the GAR tool.

3.3.1. *Adaptation in GAR*

Next, we are going to detail which are the necessary adjustments to adequate the GAR algorithm to the particular case of applying it to the database generated by the procedure described in the previous section and to achieve that it obtains suitable results for the necessities of the managers. We have to remember that this database of simulations of the project is composed by a set of continuous attributes where part of them are input attributes and the rest are output ones.

As we have described previously, the goal of the GAR algorithm is to induce association rules hidden in the database using an unsupervised learning process, without taking into account which attributes will take part in the antecedent and which ones in the consequent.

In this particular case, the rules of interest for the manager of the project are those that relate the input attributes with the output ones. That is, GAR must induce only those rules that have as antecedent input attributes and as consequent output attributes. For this reason, it has been necessary to add to GAR the possibility of establishing the attributes that can take part of the antecedent and the attributes that can take part of the consequent.

Besides, most of the times, the project managers are only interested in knowing the rules that determine a "suitable" result (as for cost, delivery time and quality) for the project. However, GAR, in its original task, can obtain both, those association rules of interest for the managers and those that do not present a suitable result of the project, which, a priori, have no interest for the managers.

In this case, and in order to realize the fitness of the tool, it is necessary to rely on the experience of the project managers. Thus, the project manager must decide the intervals of values of the output variables that make a project to be "suitable". To decide these values, the manager defines for each output variable, a maximum percentage over the value initially estimated. These percentages establish the limits to consider the project as suitable.

We are going to clarify this idea with an example. We suppose that the initial value estimated for the attribute *TpoDllo* (delivery time) is 320 days. If the manager

decides that a project can be considered acceptable if does not overcome a 10% of the estimated initial value, the value of the variable *TpoDllo* must always been less than 352 days. In the same way, if the initial value estimated for the attribute *EsfNec* (necessary effort) is 1111 technicians-day and the maximum percentage allowed is 20%, the projects considered as acceptable must not overcome 1333 technicians-day.

From now on, we will use the word "cut" to determine each one of these decisions, since this establishes some limits or divisions, settled by the manager in order to consider that a certain scenario of the database is suitable.

In order to make the search process easy and to satisfy the necessities of the project managers, we have also adjust the algorithm to permit, in case it is desired, to show only the association rules with the values of the consequent shifting in a pre-established ranges.

Departing from these considerations, the GAR algorithm gives the project manager a set of association rules, with a specific level of support and confidence that relate intervals of values of the input parameters with intervals of values of the output variables when these shift in a specific range.

3.3.2. Adequacy of ELLIPSES

As we have said before, the generated database has two parts, input and output attributes, that determine how the project would behave if the values for the input attributes at the beginning of the project, were the established ones in that tuple or scenario. It is obvious to think that this knowledge could be used as information within a supervised learning process.

For this reason, a technique based in supervised learning, as ELLIPSES, can be quite useful to obtain a set of rules that describe the content of the database. But, the generated database has several output variables that take their values in a continuous domain, however, ELLIPSES as a traditional classification technique, uses a unique attribute as class, that besides, must take its values in a discrete domain.

In this case, although it is not necessary an adjustment of the algorithm, we need a pre-processing of the data in order to adequate them to the requirements of ELLIPSES. Thus, the output attributes must be expressed as a unique attribute that takes its values in a discrete domain.

A technique similar to the one described for GAR algorithm is used to solve this problem. In this case, the project manager must assigns labels to the tuples of the database depending on the values of the output variables. We will call cuts to these assignments, as in the previous case, since they establish a percentage over each output variable, departing from which the process would not be considered suitable.

Thus, for example, the manager can establish that the project can be considered as good if the delivery time is less than 352 days, a 10% more than the one initially estimated (320), and the effort less than 1333 technicians-days, a 20% more than the effort initially estimated (1111).

RmAnvos	RmCtrata	TecCtrata	RmDesp	EsfNec	TpoDllo	Quality	Class
14,99	9,26	1,16	8,49	2452,2	350,6	0,32	G
14,92	6,52	1,02	6,44	3395,5	349,5	0,36	B
12,14	9,47	1,16	6,93	2168,3	350,3	0,32	G
13,72	8,38	1,11	5,03	2546,4	350,1	0,33	G
14,73	9,72	1,12	8,11	2538,9	356,4	0,36	B
14,14	7,14	1,14	7,94	2518,0	350,3	0,34	G
...
12,75	7,68	1,15	8,01	2294,4	350,1	0,34	G
13,22	8,85	1,11	6,18	2483,6	350,0	0,35	G
12,77	9,78	1,19	5,15	2144,7	361,8	0,35	B
14,34	7,83	1,17	5,24	2385,7	356,5	0,36	B
13,85	9,55	1,20	5,55	2238,6	354,3	0,36	B

Fig. 4. Conversion of the original database to ELLIPSES.

In this way, we can generate, by analyzing the scenarios of the database, a new database in which the output variables are transformed into a unique attribute *Class*, with values of the kind 'good' (G) and 'bad' (B) or similar, depending on the established cut. This new attribute can be used as label in a process of classic supervised learning. Departing from this, the possibilities of obtaining rules that relate the attributes with the labels go through the use of classic techniques of supervised learning, such as ELLIPSES. In figure 4, we show an example of this conversion, in which, we have considered as suitable scenarios those whose delivery time (*Tdllo*) does not overcome the initial one (320) in 10% and the rest of the output variables can take any value.

4. Results

To demonstrate the importance of the method proposed in the previous sections, we have simulated an already finished project, generating two databases that have been analyzed with the tools GAR and ELLIPSES.

4.1. Study Cases

To realize this research, we have focused on determining the influence that the management policies of hiring, adequacy and dismissal of personnel have over the variables (cost, time and quality) of an already finished project, when there are also initial restrictions on delivery time. That is to say, we want to analyze the influence that the personnel management policies have on the output variables (the necessary effort to carry out a project (*EsfNec*), delivery time (*TpoDllo*), and quality or average number of errors per task (*Quality*)), in a project over which there are strong initial restrictions on delivery time.

To sum up, the attributes whose values we want to know are: average delay in hiring (*RmCtrata*), average delay in the adaptation of new technicians (*RmAnvos*),

Table 1. Interested attributes in our Software Development Project.

Attributes	Description (unit), initial value
<i>DmTecn</i>	Average daily manpower per staff (%), 0.5
<i>RmAnvos</i>	Average delay in adaptation of new technicians (days), 20
<i>PesfDes</i>	Percentage of effort dedicated to the development (%), 0.85
<i>RmCtrata</i>	Average delay in hiring (days), 30
<i>TecInicio</i>	Initial understaffing factor (%), 0.4
<i>TecCtrata</i>	Most new hires per experienced staff (technician), 3
<i>Pm̄Ret</i>	Maximum delay percentage permitted in delivery time (%), 1.16
<i>RmDesp</i>	Time delay to transfer people out (days), 10
<i>Dformar</i>	Number of trainers per new employee (%), 0.25
<i>IEsfEst</i>	Man-days underestimation fraction (%), 0.05
<i>ItmoLcf</i>	Tasks underestimation fraction (%), 0.35
<i>EsfNec</i>	Necessary effort to carry out the project (technicians-days), 1111
<i>TpoDllo</i>	Delivery time (days), 320
<i>Quality</i>	Average number of errors per task (errors/task), 0

Table 2. Estimated intervals of the attributes of SDP.

Attributes	CRCCRT	GENERAL
<i>DmTecn</i>	-	[0.5, 1.0]
<i>RmAnvos</i>	[5, 15]	[10, 120]
<i>PesfDes</i>	-	[0.5, 0.95]
<i>RmCtrata</i>	[5, 10]	[5, 40]
<i>TecInicio</i>	-	[0.2, 1.0]
<i>Pm̄Ret</i>	[1, 1.2]	-
<i>TecCtrata</i>	-	[1, 5]
<i>RmDesp</i>	[5, 10]	-
<i>DFormar</i>	-	[0.05, 0.4]
<i>IEsfEst</i>	-	[0.05, 0.6]
<i>ItmoLcf</i>	-	[0.05, 0.6]

Table 3. Cut values considered as good by the project manager.

Cuts	<i>EsfNec</i>	<i>TpoDllo</i>	<i>Quality</i>	Num.Cases
CRCCRT				
Cut 1	-	≤ 352 (10%)	≤ 0.45 (0.45%)	45
Cut 2	-	≤ 352 (10%)	≤ 0.35 (0.35%)	12
GENERAL				
Cut 1	≤ 2092 (90%)	≤ 387 (20%)	≤ 0.40 (0.40%)	48
Cut 2	≤ 2092 (90%)	≤ 387 (20%)		221

Time delay to transfer people out (*RmDesp*) and the maximum delay percentage permitted in delivery time (*PmxRet*). In table 1 we describe the attributes of the process and we show their initial values.

In this research we have analyzed two strategies, and, therefore, we have generated two databases. The first one, which we call CRCCRT, has been generated by establishing a fast hiring policy^awith initial restrictions on delivery time^b. The second one, which we call GENERAL, has been generated with a less restrictive policy, in such a way that, in the simulation, both the number of attributes to estimate and the range of values for the output variables that establish when the development of a project is acceptable, have been expanded (see table 2).

We collect in table 3 the values of the cuts established for the output variables used in both tools in this research, distinguishing both databases. It have defined two cuts (percentages) for each one of them, which establish certain values for the output variables. Each one of these cuts establishes different correct scenarios. These cut values indicate the goals we want to cover. For example, cut 1 of CRCCRT has as goal to obtain management rules that will permit to maintain the delivery time and the quality of the project below the indicated values, independently of the value obtained by the effort necessary to carry out the project.

Together with the cut value, we show the maximum percentage, departing from the initial value estimated by the manager of the project, that must not overcome each output variable in order to realise a project that adjusts to the initial estimations. For example, in cut 1 of CRCCRT we consider good values for delivery time those included between the initial estimation (320 days) and a permitted margin of 10% over such estimation (352 days), independently of the value obtained for the cost of the project.

In this table, we also offer information about the number of cases that have been categorized as acceptable. As can be seen, and as could be supposed from the very beginning, the number of cases decreases as the restrictions on the project increase. We can deduce, that too many restrictions could cause a low probability of carrying out an acceptable development project or even, that the probabilities were non-existent. For example, the fact of imposing a restriction on the effort (*EsfNec*) for the database CRCCRT, would provoke that the number of cases were practically nothing. In this case the cut that establishes a larger number of correct scenarios (221) is cut 2 for database GENERAL. That is obvious, since the restrictions are only established on two variables and, besides, they are quite permissive (90% for *Quality*, and 20% for *TpoDllo*).

4.2. Analysis of the database CRCCRT

^aFast hiring implies that in the organization, hiring (*RmCtrata*), dismissal (*RmDesp*) and adaptation of new technicians (*RmAnvos*) are carried out quickly (*Pmxret*), that is to say, in a short period of time (see table 2).

^bFast hiring policy has been analyzed since previous studies has shown that these kind of policies favor the fulfilment of the project delivery time although it cost increases.

The database CRCRT has been generated imposing, in the simulation, restrictions on the attributes related to personnel hiring, making it to be fast, and, besides, imposing strong initial restrictions on the delivery time [16].

With the strategy followed in the generation of this database, we pretend to know the value of the attributes related to personnel management that permit to obtain good results for delivery time and to maintain acceptable levels in the project quality, independently of the value obtained for the necessary effort to carry out the project.

4.2.1. *Cut 1: Restrictions on Time and Quality*

For this database, cut 1 induces a set of rules on the input attributes, only fulfilling restrictions on delivery time (*TpoDllo*) and quality (*Quality*), according to the cuts established in table 3.

Results of GAR

The association rules discovered by GAR, where the consequent is formed by the intervals of the variables $TpoDllo \in [320, 352]$ and $Quality \in [0, 0.45]$, that is to say, the management rules that would have permitted to obtain good results simultaneously for delivery time and project quality, are the following ones:

- R1:** If $RmAnvos \in [12.4, 14.6]$ and $RmCtrata \in [6.7, 9.3]$ and $PmxRet \in [1.10, 1.18]$. support = 7%, confidence = 91.3%, covered cases = 21, rc = 0.
- R2:** If $RmAnvos \in [12.5, 14.4]$ and $RmCtrata \in [7.6, 9.9]$ and $PmxRet \in [1.11, 1.19]$ and $RmDesp \in [5, 7]$. support = 3%, confidence = 100%, covered cases = 10, rc = 7.
- R3:** If $RmAnvos \in [12.1, 14.8]$ and $ReCtrata \in [6.9, 9.7]$ and $PmcRet \in [1.11, 1.17]$ and $RmDesp \in [7.9, 9.6]$. support = 3%, confidence = 90.91%, covered cases = 10, rc = 8.
- R4:** If $RmAnvos \in [11.8, 14.7]$ and $ReCtrata \in [9.1, 9.9]$ and $RmDesp \in [6, 9.3]$ support = 3%, confidence = 76.92%, covered cases = 10, rc = 8.

The quantitative interpretation we can withdraw from the rules discovered by GAR is the following: the support indicates the number of cases (percentage) where the conjunction of values of the rules is produced. The confidence shows the percentage of cases that fulfil such conjunction and besides, fulfil the values of the consequent. Together with this information, we show the number of cases that fulfil the rule in a correct way (covered cases) and the number of cases that also belong to other rules (*rc*). This interpretation will be applicable to the rest of the examples that are shown next.

In this particular case, rule R1 shows that 7% of the cases (21) fulfil the conjunction of values $RmAnvos \in [12.4, 14.6]$, $RmCtrata \in [6.7, 9.3]$ and $PmxRet \in [1.10, 1.18]$ with the correct consequent, that is to say, with the values of the variables within the range established by the manager. The confidence value of 91.3% shows that in the database there are 23 cases that fulfil the antecedent, but that 2 of them do not fulfil the consequent. Rule R2 shows that 3% of the cases(10) fulfil the antecedent and the consequent. Besides, a 100% of confidence for this rule, means

that all the cases (10) that fulfil the antecedent, also fulfil the consequent. We also want to show that 7 of the cases covered by this rule are also covered by other rules ($rc = 7$). In figure 5 we offer the graphic representation of these rules. We can check that the average delay in the incorporation of new technicians to the project and their adequacy, are attributes that appear in all the obtained rules. Such attributes take, in all of them, values from medium to high within the interval defined as fast personnel management. With regard to the obtained results, R1 would have been the easiest rule to be applied, since, although it has the same number of attributes than rule R4, we would have only to modify the values of $RmAvnos$ y $RmCtrata$.

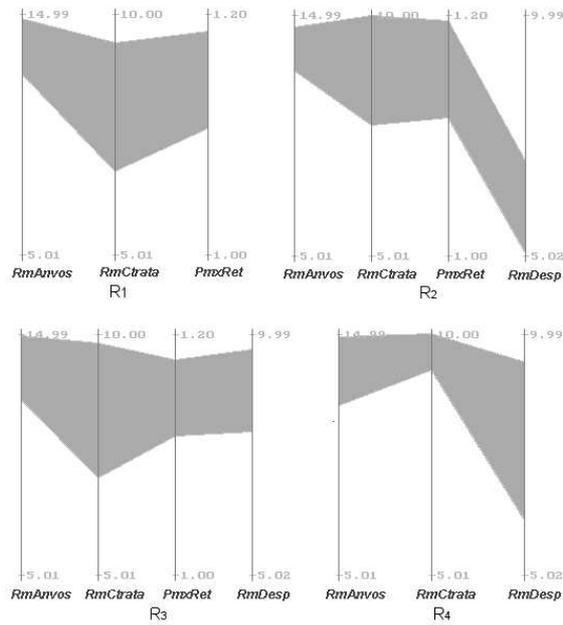


Fig. 5. Results of GAR on CRCCRT database for cut 1.

Results of ELLIPSES

The management rules induced by ELLIPSES that show information about the results labeled as 'Good' are the following (see figure 6):

R1: $(RmAvnos-11.9)/3.14+(RmCtrata-7.9)/1.57+(PmxRet-1.18)/0.02=1$.
Success = 10, Errors = 1, before covered cases = 0.

R2: $(RmAvnos-13.4)/1.65+(RmCtrata-9.3)/1.53+(PmxRet-1.12)/0.02=1$.
Success = 10, Errors = 1, Before covered cases= 0.

R3: $(RmAvnos-13.6)/1.01+(PmxRet-1.16)/0.03+(RmDesp-8.6)/1.17=1$.
Success = 5, Errors = 0, Before covered cases = 0.

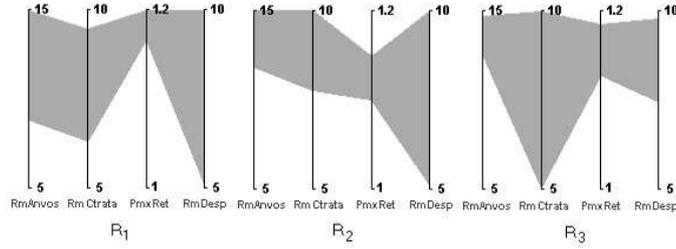


Fig. 6. Results of ELLIPSES on CRC CRT database for cut 1.

In figure 6 we offer the graphic representation of these rules. The interpretation that can be followed from them would be:

- Figure 6.R1): *RmAnvos*, defined in the interval $[5,15]$, must take medium high values (center 11.9 with a margin of ± 3.14), *RmCtrata*, defined in $[5,10]$, must take medium high values but without reaching the limit (center 7.9 with a margin of ± 1.57) and *PmxRet*, defined in $[1,1.2]$, must take very high values (center 1.18 with a margin of ± 0.02), practically at the limit.
- Figure 6.R2): *RmAnvos* takes high values (center 13.4 with a margin of ± 1.65), *RmCtrata* takes high values (center 9.3 and a margin of ± 1.53) and *PmxRet* takes medium high values (center 1.12 and margin ± 0.02).
- Figure 6.R3): *RmAnvos* must take high values without reaching the limit (center 13.6 with a margin of ± 1.01), *PmxRet* must take high values, without reaching the edge (center 1.16 and margin of ± 0.03) and *RmDesp*, defined in the interval $[5,10]$, must take medium and high values, but without reaching the limit (center 8.6 and margin ± 1.17).

4.2.2. *Cut 2: Restrictions on Time and a High Quality Level*

Cut 2 for database CRC CRT induces a set of rules on the input attributes fulfilling the restrictions on the same variables, but being less permissive as for the average number of errors for task (*Quality*) (see table 3). We have to observe that, in this situation and for being a so restrictive policy, there is only 12 cases categorized as good.

Results of GAR

The association rules discovered by GAR, where the consequent is formed by the intervals of the variables $TpoDllo \in [320, 352]$ and $Quality \in [0, 0.35]$, that is to say, the management rules that would have permitted to obtain good results for delivery time, demanding, besides, a high level of quality from the project, are the following ones:

- R1:** If $RmAnvos \in [13.1, 13.8]$ and $RmCtrata \in [8.3, 8.9]$ and $PmxRet \in [1.11, 1.18]$ and $RmDesp \in [5, 6.2]$. Support = 1%, confidence = 100%, covered cases = 3, rc = 0.
- R2:** If $RmAnvos \in [12.1, 13.6]$ and $RmCtrata \in [8.5, 9.2]$ and $PmxRet \in [1.17, 1.18]$ and $RmDesp \in [5, 6.1]$. Support = 1%, confidence = 100%, covered cases = 3, rc = 1.
- R3:** If $RmAnvos \in [12.7, 14.1]$ and $RmCtrata \in [7.1, 8.6]$ and $PmxRet \in [1.13, 1.15]$ and $RmDesp \in [7.9, 8.2]$. Support = 1%, confidence = 100%, covered cases = 3, rc = 0.
- R4:** If $RmAnvos \in [14.4, 14.7]$ and $RmCtrata \in [9, 9.8]$ and $PmxRet \in [1.02, 1.10]$ and $RmDesp \in [6.1, 8]$. Support = 1%, confidence = 60%, covered cases = 3, rc = 1.

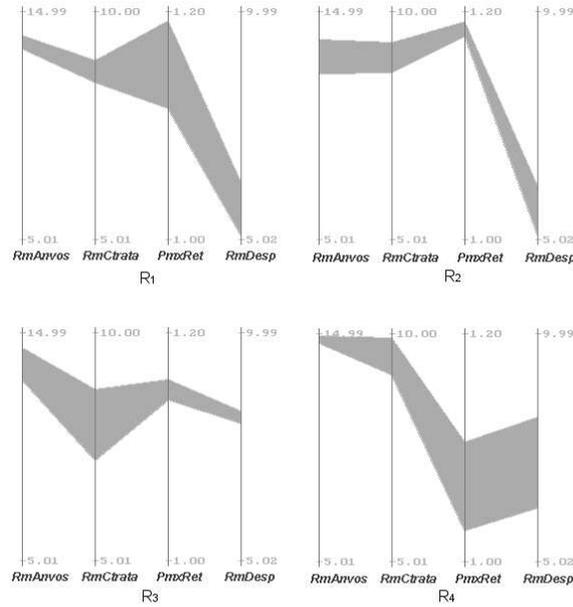


Fig. 7. Results of GAR on CRCRT database for cut 2.

Figure 7 shows the graphic representation of these rules. With regard to the previous cut, we can check that in these rules appear all the attributes implied in the database, and besides, that the ranges are narrower, which implies that, in order to reach the desired goals in this second study, the control on these attributes by the project manager must be larger. We can check that in all the obtained rules, specially in R1, R2 and R4, the tendency is to realize the incorporation and adequacy of the new technicians slowly, while the leaving of the project technicians must tend to be realized quickly (always within the intervals we have defined for these attributes).

Although, the 4 attributes that have been used to generate the database appear in all the rules, in this case, rules R3 and R4 indicate that not only the attributes related to the personnel management had to be modified but also the one related to

the initial restrictions on delivery time. Thus, it would have been more suitable to apply the rules R1 or R2, since this last attribute was estimated within the interval obtained in the rules.

Results of ELLIPSES

The rules induced by ELLIPSES whose results show restrictions on development time and a high level of quality are the following:

R1: $(RmAnvos-13.5)/1.57 + (RmCtrata-8.7)/0.28=1$.
Success =5, Errors =2, Before covered cases=0.

R2: $(RmAnvos-12.4)/2.04+(PmxRet-1.16)/0.01+(RmDesp-6.9)/1.19=1$.
Success =3, Errors =0, Before covered cases =0.

R3: $(RmAnvos-14.8)/0.74+(RmCtrata-9.9)/1.20+(PmxRet-1.06)/0.05=1$.
Success =2, Errors =1, Before covered cases =0.

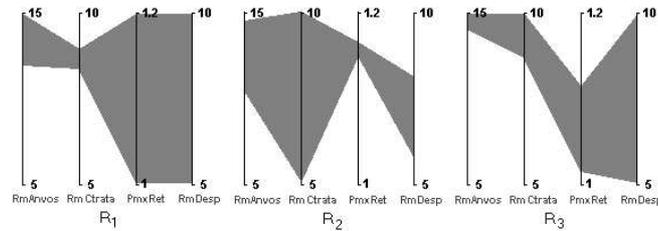


Fig. 8. Results of ELLIPSES on CRCRT database for cut 2.

In figure 8 we offer the graphic representation of these rules. Next, we offer the interpretation of these results.

- Figure 8.R1): *RmAnvos* must take high values (center 13.5 with a margin of ± 1.57) and *RmCtrata* must take high values but without reaching the limit (center 8.7 with a margin of ± 0.28).
- Figure 8. R2): *RmAnvos* takes high values without reaching the limit (center 12.4 with a margin of ± 2.04), *PmxRet* takes high values without reaching the limit (center 1.16 and margin ± 0.01) and *RmDesp* takes medium low values (center 6.9 and margin ± 1.19).
- Figure 8.R3): *RmAnvos* must take very high values (center 14.8 and margin ± 0.74), *RmCtrata* must take high values (center 9.9 and margin ± 1.20) and *PmxRet* must take medium and low values, but without reaching the limit (center 1.06 and margin ± 0.05).

4.3. Analysis of the database GENERAL

The database GENERAL has been generated with a less restrictive policy than the previous one, in such a way that we have expanded both the number of attributes to estimate and the cut percentage that establishes when the development of a project is good.

Given that the real values of this project for time and effort were of 387 days and 2092 technicians-day respectively, what we are searching in this analysis are the management rules that would have improved the final values of this project, keeping, at the same time, the quality within acceptable levels, as it is shown in table 3. We can check again, the same than in the previous section, that if we limit the goals, that is to say, if we only want to obtain, in a global way, good results for time and effort, independently of the values obtained for the project quality, the number of suitable scenarios considerably increases (goes from 48 to 221).

4.3.1. Cut 1: Restrictions for time, effort and quality

The first cut for this database is the only one that establishes restrictions on the three output variables. Therefore, the rules that are induced on the attributes must fulfil the three restrictions simultaneously, according to the cuts shown in table 3.

Results of GAR

The association rules discovered by GAR, where the consequent is formed by the intervals of the variables $EsfNec \in [1111, 2092]$ and $TpoDllo \in [320, 387]$ and $Quality \in [0, 0.40]$, that is, the management rules that would have permitted to obtain, simultaneously, good results for delivery time, effort and project quality, are the following ones:

- R1:** If $RmAnvos \in [33.2, 78.3]$ and $PesfDes \in [0.58, 0.84]$ and $TecInicio \in [0.57, 0.93]$ and $DFormar \in [0.05, 0.26]$ and $ItmoLcf \in [0.05, 0.22]$.
Support = 2.2%, confidence = 84.62%, covered cases = 11, rc = 0
- R2:** If $DmTecn \in [0.54, 0.88]$ and $RmAnvos \in [41.4, 78.4]$ and $RmCtrata \in [15.1, 36.4]$ and $TecInicio \in [0.62, 0.98]$ and $ItmoLcf \in [0.05, 0.24]$.
Support = 2.6%, confidence = 86.67%, covered cases = 13, rc = 6
- R3:** If $DmTecn \in [0.79, 0.91]$ and $RmAnvos \in [41.6, 119.2]$ and $RmCtrata \in [33.1, 39.6]$ and $TecInicio \in [0.75, 0.99]$. Support = 2.2%, confidence = 84.62%, covered cases = 11, rc = 2

Figure 9 shows the graphic representation of the discovered rules. In this case, R1 is the chosen rule for being realizing a post mortem analysis of the project (rule R2 is similar). This rule, although involves 5 attributes (one more than rule R3), is the one that is closer to the estimations realized for this project. That is to say, we would have improved the obtained results if we had improved the initial estimations on the size of the project ($ItmoLcf$), if we had increase the average delay of adequacy of the new technicians ($RmAnvos$) and if we had increase the number of technicians at the beginning of the project ($TecInicio$). Rule R3 would imply the modification of the four attributes since, the estimated values for each one of them are not in the range of the values indicated in the rule.

Results of ELLIPSES

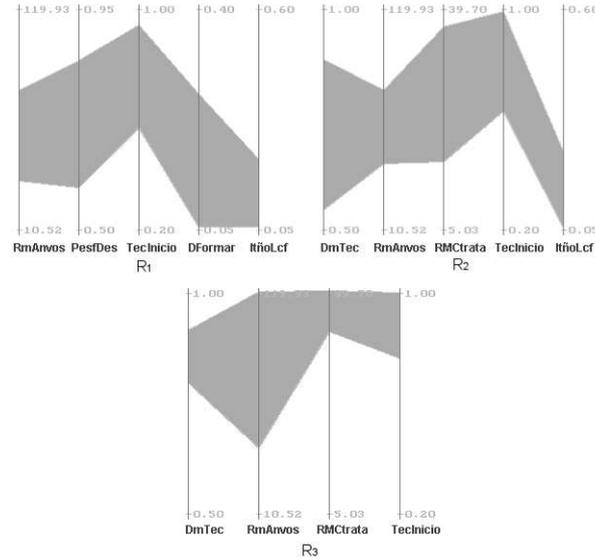


Fig. 9. Results of GAR on GENERAL database for cut 1.

The management rules induced by ELLIPSES to obtain results labeled as 'Good' are the following^c:

- R1:** $(a-0.8)/0.16+(e-0.8)/0.15+(h-0.2)/0.05=1$.
Success =9, Errors =1, Before covered cases=0.
- R2:** $(a-0.9)/0.12+(b-94.3)/25.6+(d-33.0)/7.0+(e-0.9)/0.07=1$.
Success =7, Errors =0, Before covered cases=2.
- R3:** $(b-54.3)/44.32+(e-0.5)/0.22+(f-2.7)/0.24=1$.
Success =6, Errors=1, Before covered cases=0.
- R4:** $(d-36.0)/4.00+(e-0.8)/0.20+(h-0.2)/0.21+(i-0.2)/0.20=1$.
Success=10, Errors =1, Before covered cases =5.

In figure 10 we show the graphic representation of the rules obtained by ELLIPSES for time, effort and quality. Next, we offer the interpretation of these rules.

- Figure 10.a). *DmTecn* must take medium-high or high values (center 0.84, margin 0.16), *TecInicio* must take high values (center 0.85, margin 0.15) and *IesfEst* must take medium-low values (about 0.2 with a margin of 0.05). Thus, in order to secure a development process that adjust to the initial necessities, the dedication must be high (>70%), the percentage of technicians at the beginning of the project must be high (>70%), but the initial estimation of effort must not overcome medium values (<25%).

^cKey: a) *DmTecn*, b) *RmAnvos*, c) *PesfDes*, d) *RmCtrata*, e) *TecInicio*, f) *TecCtrata*, g) *DFormar*, h) *IesfEst*, i) *ItmoLcf*.

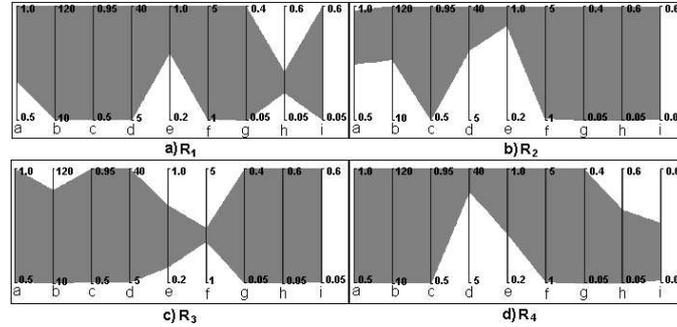


Fig. 10. Results of ELLIPSES on GENERAL database for cut 1.

- Figure 10.b). This rule indicates that an acceptable development process will be carried out with the three conditions (effort, time and quality) if: *DmTecn*, *RmAnvos* and *RmCtrata* take medium-high or high values (centers 0.9, 94.3 and 33 and margins 0.12, 25.6 and 7.0 respectively) and *TecInicio* takes very high values (center 0.9 and margin 0.1). We can deduce that if the delay in the adaptation (*RmAnvos*) and in the hiring of new technicians (*RmCtrata*) is high, more than 70 days and 25 days respectively, then, both the percentage of initial technicians (*TecInicio*) and the average dedication (*DmTecn*) must be high, more than 80% in both cases.
- Figure 10.c). Practically, *RmAnvos* has no influence, although it must not take very high values (center 54.3 and margin 44.32). *TecInicio* must take medium low values (center 0.5 and margin of 0.22), although not very low, and *TecCtrata* must take medium values (center 2.7 and margin 0.24). That is, if the number of new technicians to be hired (*TecCtrata*) has a medium value (≈ 2) then the percentage of initial technicians (*TecInicio*) can be a medium low value (between 25% and 75%), but the delay in the adaptation of technicians (*RmAnvos*) must not be very high (< 100 days).
- Figure 10.d). *RmCtrata* must take very high values (center 36 and margin 4), *TecInicio* must take medium-high values (0.8 and 0.2) and both *IestEsf* and *ItmoLcf* must take low or medium-low values (center 0.2 and margin 0.2). That is to say, if the initial estimation of effort (*IestEsf*) and code lines (*ItmoLcf*) are medium-low values ($< 40\%$ in both cases) and the hiring delay (*RmCtrata*) is very high (> 30 days) then the percentage of initial technicians (*TecInicio*) must be a medium high value ($> 60\%$).

4.3.2. Cut 2: Restrictions on time and effort

As the goals of the project are less ambitious, we can check in this case that a

larger number of possible scenarios have been found and that the attributes that we have to control in order to fulfil the goals of the project are smaller.

Results of GAR

The association rules discovered by GAR, where the consequent is formed by the intervals of the variables $JBSZMD \in [1111, 2092]$ y $SCHCDT \in [320, 387]$, that is, the management rules that would have permitted to obtain, simultaneously, good results for delivery time, and effort, are the following ones:

- R1:** If $TecInicio \in [0.64, 0.99]$ and $ItmoLcf \in [0.05, 0.48]$.
Support = 32%, confidence = 93.02%, covered cases = 160, rc = 0
- R2:** If $TecInicio \in [0.59, 0.86]$ and $ItmoLcf \in [0.25, 0.44]$.
Support = 11%, confidence = 100%, covered cases = 54, rc = 48
- R3:** If $RmAnvos \in [17.8, 81.6]$ and $TecInicio \in [0.55, 0.92]$ and $ItmoLcf \in [0.05, 0.29]$.
Support = 10%, confidence = 98.04%, covered cases = 50, rc = 44

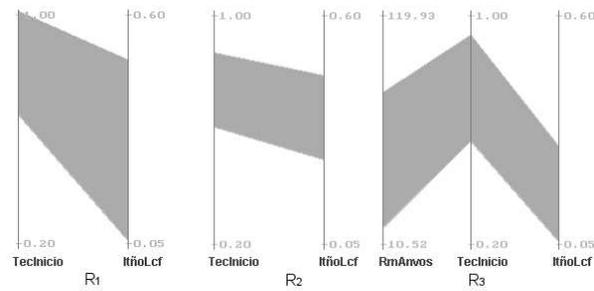


Fig. 11. Results of GAR on GENERAL database for cut 2.

In figure 11 we show the graphic representation of the rules discovered by GAR. Following the criteria proposed by the project manager, the easiest rules to be applied, among the rules obtained, would have been rules R1 and R2. We only would have to modify an attribute in them, since, the underestimation of the size of the project is maintained within the levels initially established and we would only have to modify the value of the percentage of technicians that begin the project. The selection between R1 and R2, will depend exclusively on the project manager, as we have said before, according to his knowledge on the possibility of maintaining the implied attributes within the intervals provided by the rules.

Results of ELLIPSES

The management rules induced by ELLIPSES to obtained results labeled as 'Good' are the following ones:^d

^dKey: a) *DmTecn*, b) *RmAnvos*, c) *PesfDes*, d) *RmCtrata*, e) *TecInicio*, f) *TecCtrata*, g) *DFormar*, h) *IesfEst*, i) *ItmoLcf*

- R1:** $(e-0.9)/0.09 = 1$.
Success =121, Errors =14, Before covered cases =0.
- R2:** $(e-0.7)/0.10 + (i-0.4)/0.12 = 1$.
Success =33, Errors =0, Before covered cases =2.
- R3:** $(b-21.9)/11.42 + (e-0.6)/0.11 = 1$.
Success =19, Errors =2, Before covered cases =2.

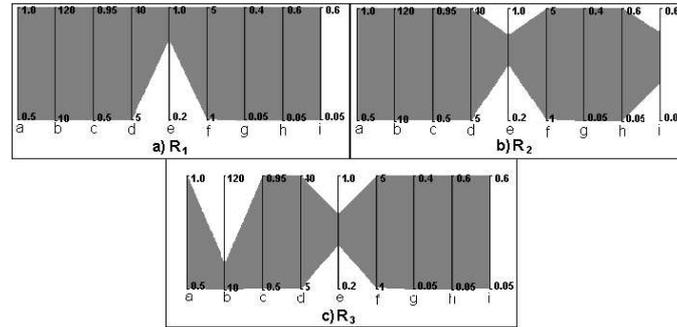


Fig. 12. Results of ELLIPSES on GENERAL database for cut 2.

In figure 12 we show the rules obtained by ELLIPSES for time and effort. Next., we offer their interpretation.

- Figure 12.a). We can notice how very high values (center 0.9 and margin 0.09) of attribute *TecInicio* (>80%) are determinant for the realization of a good software development project (121 cases out of 221), although there is a 11.5% (14 cases) of projects that had not an acceptable development.
- Figure 12.b). In this rule, we can notice again, the significance of attribute *TecInicio*, since its medium high values (center 0.7 and margin 0.10) and medium high values (center 0.4 and margin 0.12) of the attribute *ItmoLcf*, are enough to reach to a good development. That is to say, if the percentage of technicians estimated at the beginning of the project (*TecInicio*) has a medium high value ($\approx 60\%$) then the initial estimation of code lines (*ItmoLcf*) must not be very high, between 25% and 50%.
- Figure 12.c). In this rule we can notice how medium values (center 0.6 and margin 0.11) of *TecInicio* and low values (center 21.9 and margin 11.42) of *RmAnvos* would produce a suitable development, that is, if the percentage of technicians estimated at the beginning of the project (*TecInicio*) is not very high, an average value between 50% and 70%, then the delay in the adaptation of the technicians that are incorporated (*RmAnvos*) must be a very low value (<30 days).

5. Analysis of the results and criteria of selection of rules

The use of a tool to obtain management rules, allows the managers of a SDP to face different situations when they have to define the most suitable management policies in order to optimise the final values of certain variables either individually or globally:

1. There are certain management policies that are already imposed, so they can not be modified. For example, that the average dedication of the technicians must be 40%, or that the delivery time must not overcome the estimated one in more than 10%, or that it should not be possible the incorporation of new personnel once the project has began, etc. This situation becomes more evident in those cases where the maturity level of the organization is difficult to modify.
2. There are no initial restrictions as for the management policies to be applied. In this case, the project manager has a wide margin of decision.
3. Another situation, perhaps the most common, is produced when we know the characteristics of the project environment and the organisation environment , but we do not know or there is some uncertainty about the specific characteristics of the SPD to be realised, fundamentally what affects the initial estimation of delivery time, cost of the project and size of the product to be developed.

On the other hand, once we have obtained the management rules, the project manager will be the one who decide which rules are the easiest to be applied depending on the specific project and of the software organisation with which he is working. He will also be who, before starting the development and once the attributes that appear in the obtained management rules are analysed, decide which attributes can be handled to keep them within the values that appear in the rules and which not, in order to optimise the results of the variables or group of variables he is interested in. In any case, the manager will also know that if he does not keep within the obtained values, he will not have guaranteed the optimisation of the results.

The criteria that the project manager use to consider, in order to decide which are the best rules to be applied of all the ones obtained in the data mining process, are :

1. To choose the rules that collect the largest number of scenarios and success.
2. To choose the rules with the least number of attributes.
3. If a post mortem analysis is carried out (state of the project that we are going to use), we consider that the best rules are those that implicate the modification of a smaller number of attributes. That is to say, from among

the attributes that appear in the rule, which ones are within the initial values and which ones would have to be changed.

4. If an a priori analysis is realised, we must choose those rules that have attributes easy to be controlled throughout the development process.
5. Finally, to choose the rule or rules with which we can estimate the best results.

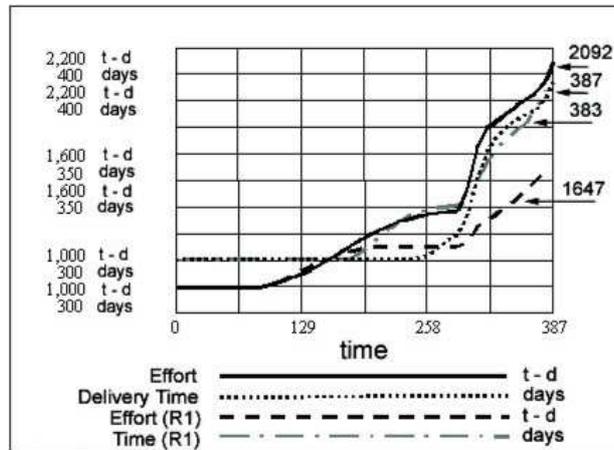


Fig. 13. Comparison between the nominal simulation with the one of the project for R1.

Finally, by means of the project simulator it is possible to check and compare graphically the results that we could obtain in the project by applying the rules induced by the previous process. Next, we show the evolution of the project by applying some of the rules induced by the former tools.

Thus, in figures 13 and 14 we compare the nominal evolution of the project with the evolution that we would have obtained if we had applied rules R1 and R3 for cut 1 of the database GENERAL. The values obtained for delivery time in R1 and R3 were 383 and 380 days respectively, while the values for effort were 1647 and 1869 technicians-day respectively. In the light of these results, the project manager will be who decide the rule to be chosen, given that we obtain better results in time in R3 than in R1 at the expense of a greater effort.

As for ELLIPSES, with regard to the results obtained for both cuts of CRCRT, the project manager will always be, as we have said before, who decide the manager rule that had to be applied in order to obtain the desired results. From the point of view of the criteria indicated for the selection of the rules, rule 1 from cut 2 had been the easiest to be applied, since we get to improve the project quality levels with it and, we should only have to control two attributes, *RmAnvos* and *RmCtrata*, although the intervals where they can move are less wide than in other rules.

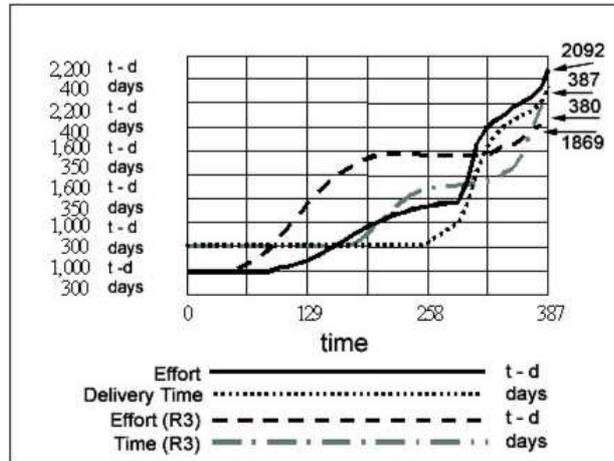


Fig. 14. Comparison between the nominal simulation with the one of the project for R3.

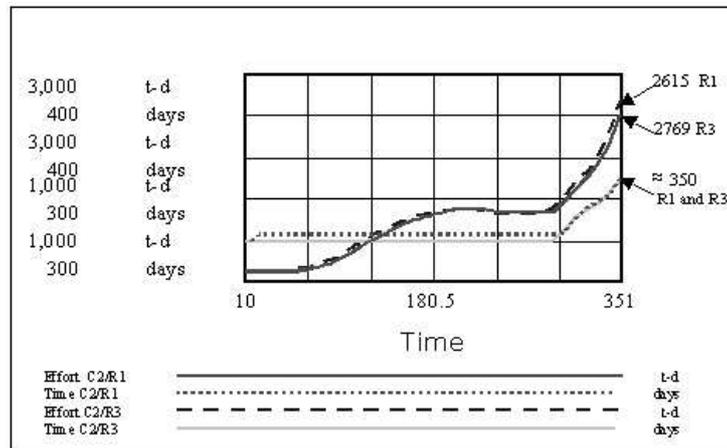


Fig. 15. Checking of the results obtained with rules R1 and R3 for CRCRT.

In figure 15, we can see which would have been the evolution of the necessary effort and of the delivery time if we had applied R1 or R3. We can check that both in R1 and R3 the time would have been practically the same, 350 and 351 days respectively, but the effort would have been of 2615 technicians-day (t-d) in R1 and 2769 t-d in R3. Therefore, as it was expected, both rules allow to obtain the desired goals for time and quality and, besides R1 would have permitted to improve the results of the necessary effort to carry out the project.

6. Conclusions

With the current simulation systems it is possible to model and simulate the complex behavior of the software development processes and to generate different scenarios for the same project. One of the most important advantages of this process is that we can "test", without any cost, the effect that will have on the project the application or not of different management policies. On the other hand, one of the greatest disadvantages in the development process, lies in the necessity of estimating a great number of attributes (more than 60 in some of them) that determine its evolution. That is, the manager of the project has the difficult task of estimating, not only the most suitable value for each one of the attributes that influence on the development process, but, also, the optimal combination of the values of such attributes.

This disadvantage has been solved in recent researches, by means of the use of automatic learning techniques and databases generated by a project simulator [17][18][6].

The conjunction of both techniques, dynamic systems simulation and data mining, allows to obtain automatic management rules in order to make easy the decision-taking in the management process of software project. The obtained management rules give very useful information about the attributes or characteristics of the project and about the organization to control the development process and to fulfil with the goals of the project.

Before beginning the development, some management policies are known with certain security for being supported by the experience in similar projects, but many others are imposed by the own development organization and/or the client. The obtained management rules are going to permit us to know within which intervals the attributes about which we really have any uncertainty can move. Thus, we can decrease the uncertainty we have about certain management policies when we are estimating the project and check, in already finished projects, the accuracy of the realized assessments.

In this paper we have presented the application of the tools GAR and ELLIPSES as a method to analyze databases of software development projects. These data mining techniques are able to generate management rules in order to make easy the decision-taking to the project managers.

Finally, we want to indicate that, the application of the automatic learning techniques to real historical bases, would permit to obtain management rules based on

the experience of the work carried out to date by the development organization. So, if this happens, we can say that the experience acquire by a development organization will be useful to the managers to estimate and to manage future projects and will not be exclusive of only one person.

References

1. Diehl E. Participatory simulation software for managers: The design philosophy behind microworld creator. *European Journal of Operational Research* **59(1)** (1991) 210–215.
2. Abdel-Hamid, T. and Madnick, S. *Software Project Dynamics: an Integrated Approach*. (Prentice-Hall, 1991).
3. Ramos I. and Riquelme J. The dynamic models for software development projects and the machine learning techniques, *International Conference on Product Focused Software Process Improvement*. (1999).
4. Mata, J., Álvarez, J.L. and Riquelme, J. Discovering Numeric Association Rules via Evolutionary Algorithm. *6th Conf. on Knowledge Discovery and Data Mining*. (2002).
5. Álvarez J.L., Mata J., and Riquelme J.C. Mining Interesting Regions using an Evolutionary Algorithm. In *ACM Symposium on Applied Computing*, (2002). pp 498–502.
6. Aguilar J., Ramos I., Riquelme J.C. and Toro M. An evolutionary approach to estimating software development projects. *Information and Software Technology*, **43(14)**, (2001). 875–882.
7. Fayyad U., Piatetsky-Shapiro G., and Smyth P. From data mining to knowledge discovery in databases. *AI Magazine*, **17(3)** (1996). 37–54.
8. Chen M., Han J., and Yu P. . Data mining: An overview from database perspective. *IEEE Trans. on Knowledge and Data Engineering*, **8(6)**, (1996), 866–883.
9. Agrawal R., Imielinski T. and Swami, A. Mining Association Rules between Sets of Items in Large Databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. (1993). 207–216.
10. Goldberg D. *Genetic algorithms in search, optimization, and machine learning*. (Addison-Wesley, Pub. Company, inc. 1989).
11. Inselberg A. The Plane with Parallel Coordinates, special issue on computational geometry. *The Visual Computer*, **1**, (1985), 69–97.
12. Collier K.W. and Collofello J.S. Issues in Software Cycle Time Reduction International Phoenix. *Conference on Computers and Communications*. (1995).
13. Paulk M.C. Capability Maturity Model, version 1.1. *IEEE Software*. (1993), 18–27.
14. Chichakly K.J. The Bifocal Vantage Point: Managing Software Projects from a Systems Thinking Perspective American Programmer. (1993) 18-25.
15. Graham A., Morecroft J., Senge P., and Sterman J. Model-supported case studies for management education. *European J. of Operational Research*, **59(1)**, (1992). 151–166.
16. Ruiz M., Ramos I., Toro M. Using Dynamic Modeling and Simulation to Improve the COTS Software Process.. In *5th International Conference on Product Focused Software Process Improvement*. (2004). 568–581.
17. Tuya J. and all. Integration of Information in a Training Environment for Software Project Management. *9th Annual International Conference of Software Quality Management*. (2001).
18. Ramos I. and Aroba J. and Riquelme J.C.Improvements in the Decision Making in Software Projects. *3th Int. Conf. on Enterprise Information Systems*. (2001). 431-436.