# Visualization Techniques of Management Rules for Software Development Projects

Jacinto Mata, José L. Álvarez
Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática
Universidad de Huelva, Spain
{mata,alvarez}@uhu.es

José C. Riquelme, Isabel Ramos, Jesús S. Aguilar, Francisco Ferrer
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla, Spain
{riquelme,isabel.ramos,aguilar,ferrer}@lsi.us.es

## Abstract

*The application of data mining techniques to the managing of Software Development Projects (SDP) is not an uncommon phenomenon, as in any other productive process that generates information in the way of input data and output variables. In this paper, a set of tools developed by the authors, that generate, in a visual way, managing rules suitable to cover minimum goals in a SDP are presented. Although the techniques used are able to generate quantitative rules, giving numeric values suitable for these goals, the visual representation of these rules helps their easy and quick understanding by a manager of a SDP. The application to a database generated from the simulation of a project allows to establish a profitable comparison and to demonstrate the validity of the techniques.*

## 1 Introduction

The use of different techniques belonging to the field known as *Data Mining*, *Machine Learning* or as a whole, *Knowledge Discovery in Databases (KDD)*, to the management of Software Development Projects (SDP) is relatively new [4][9][7][1]. Its justification for this lies in the difficulty that the software project manager finds in taking decisions. This is motivated by the diversity of management policies that influence on the development project and the necessity of controlling them simultaneously to obtain the desired objectives. The manager fundamentally has his own experience and the historic information about the development organization to solve this problem. The main handicap to carry out this application is the lack of data departing from which we can withdraw information. While numerous industrial productive processes are able to generate millions of records, the management of a SDP will hardly take some tens. Nevertheless, the use of dynamic models [8] to carry out the simulation of hundreds of different scenarios of a same SDP is a tool capable of providing data that should be of use as source for a KDD process.

The parameters or attributes of a dynamic model represent the resources and management policies of the SDP. Not all parameters or attributes are susceptible to variation (mean rotation of the experienced staff, restrictions on delivery time), but for those that display certain level of uncertainty the model gives an interval. In table 1 we show for each one of the attributes: its abbreviation, a brief description, its measurement unit, its initial estimated value and the interval of possible values that will depend on the organization.

In figure 4, the evolution of the project according to the estimated data (Table 1) is shown. With the data collected in the nominal simulation of the project, the development time was of 387 days instead of the estimated 320, the necessary effort was of 2092 man-days instead of the estimated 1111 and the quality, measured in average number of errors for task, was of 0,26.

In general, our aim is that the management project should dispose of easy-to-apply management rules that permit him to estimate good results. Our goal for this project in particular, as we know both, the initial and the final data for being a concluded project, was to obtain management rules that had permitted us to estimate better results (for time, effort and quality, simultaneously) from

Proceedings of the Third International Conference On Quality Software (QSIC'03)
0-7695-2015-4/03 $ 17.00 © 2003 IEEE

**Table 1. Attributes used to generate the database**

| Abbreviation | Description | Unit | Initial value | Interval |
|---|---|---|---|---|
| ADMPPS | Average daily manpower per staff | % | 50 | 10-100 |
| ASIMDY | Average assimilation (of new personnel) delay | days | 20 | 10-120 |
| DEVPRT | % of effort assumed needed for development | % | 85 | 50-95 |
| HIREDY | Hiring (of new personnel) delay | days | 30 | 5-40 |
| INUDST | Initial understaffing factor | % | 40 | 20-100 |
| MNHPXS | Most new hires per experienced staff | technician | 3 | 1-5 |
| TRPNHR | Number of trainers per new employee | % | 25 | 5-40 |
| UNDESM | Man-days underestimation fraction | % | 0 | 0-60 |
| UNDEST | Tasks underestimation fraction | % | 35 | 0-60 |

the ones that finally were obtained. That is to say, which attributes we would have to modify to obtain better results.

For this reason, we indicate the criteria that must be followed to realize a comparative analysis of different techniques of obtaining the rules:

1. To select rules that collect the greater number of scenarios and hits.
2. To select rules with the less number of attributes.
3. We consider that, if a post-mortem analysis is realized (our case), the best rules are those that involve the modification of a less number of attributes. That is to say, which ones of the attributes that appear in the rule, keep within the initial values and which ones should have been modified.
4. If an a priori analysis is carried out, rules with attributes that are easy to be controlled throughout the development process must be selected.
5. Finally, select the rule or rules with which the best results are estimated.

In this paper in particular, we present three techniques to show in a visual way the rules that a SPD manager must apply to cover goals of time, effort and/or quality.

## 2 Description of the tools

Next the three tools used and the interpretation of their results will be described.

### 2.1 HIDER

HIDER (HIerarchical DEcision Rules) [2] is a tool that generates decision rules for a labelled database using as a searching technique an evolutionary algorithm. A decision rule is as follows:

*If* $p_1 \epsilon [a_1, b_1]$ *and* $p_2 \epsilon [a_2, b_2]$ *and ... and* $p_n \epsilon [a_n, b_n]$ *then Label* (1)

where *a* and *b* are real values in the membership interval of each parameter. The performance of HIDER also allows to obtain two approximations for each interval $[a_i, b_i]$ of each rule. One possibility is a confident interval, to the effect that during the training process examples that comply with their belonging to this interval have been found. The second possibility is an expanded interval that includes the confident interval and besides, it expands to regions of the space of parameters where there is no examples. That is to say, it is an interval that includes what could be named "no-mans land" where, if there are no accurate examples, there are neither inaccurate ones. The graphic representation that it generates, is a kind of histogram with a bar for each parameter and where the safe interval is represented in black within the expanded interval which is represented in grey.

### 2.2 GAR

The aim of GAR (Genetic Association Rules) [6] is to discover the association rules that exist among the attributes or parameters of a database. Association rules are a data mining technique used to discover the relationship existing among the values of some attributes in relation to others. In an association rule there are attributes or parameters that are on the left part of the rule (antecedent) and others that are on the right one (consequent). Association rules, unlike other tools such as classifiers, do not take into account which are the attributes that must be in one or in the other part of the rule.

However, in this research, there is a clear difference between parameters considered as input (average hiring time, effort in error tests, etc.) and the output parameters (ending time, necessary effort, etc.) In this case we can direct the tool in order that it only searches for those rules with the antecedents and the consequents chosen by the user.

The final user, besides of being interested in knowing the rules that exist in a database, will also need to know to which extent these rules must be valued. For that, association rules convey two values implied that are used for establishing the quality of the rule:

**Support:** It is an statistical measure that gives a vision of

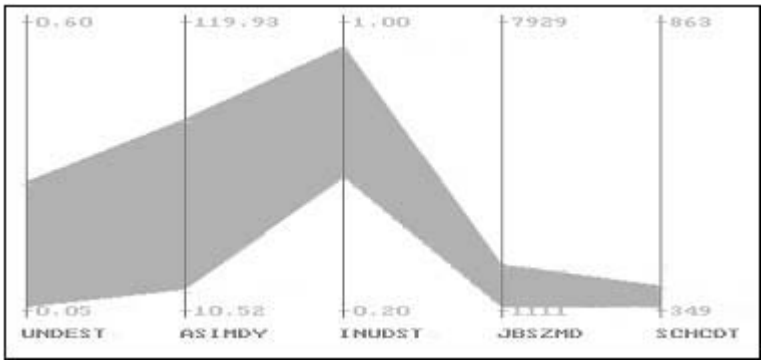


**Figure 1. Representation through parallel coordinates of an association rule**

the frequency of the rule in the database. In this way, rules with a low support are not interesting from the informative point of view since they are produced in a few cases.

**Confidence:** It is an indicator of the strength of the implication underlying in the rule. Its percentage indicates the rate of accuracy with which the rule is fulfilled.

Besides the quantitative interpretation, GAR offers a graphic representation through a parallel coordinate system [5] of the rules obtained. In this way, a rule expressed by the graphic in figure 1 can be read in a more qualitative way: *if UNDEST has low values and ASIMDY has low-medium values and INDUST has relatively high values then JBSZMD and SCHCDT have very low values.*

### 2.3 Ellipses

The aim of this tool [3] is to induce, departing from a set of labelled data, a set of rules that determine the relationship of the attributes of the data set with each one of the existent rules in itself. These rules are described with a central value for each parameter and an acceptable amplitude range departing from this value.

The name *Ellipses* comes from the fact that, in two dimensions, its graphic representation would be elliptic. With the induced set of data we do not pretend to obtain a pattern of classification with a reduced error ratio but to offer to the human expert a reduced set of rules, easy to interpret and with information of interest to take decisions.

*Ellipses* uses three formats for representing the rules in order to show the induced results: quantitative, qualitative and graphic. In the quantitative format a rule presents the format shown in equation 2, which interpretation would be "*If $x_1$ takes a value about $c_1$ with a maximum amplitude of $a_1$ and $x_2$ takes a value about $c_2$ with a maximum value of $a_2$ and ... and ..., then class $E_i$*".

$$\textit{If } x_1(c_1, a_1) \textit{ and } x_2(c_2, a_2) \textit{ and ... and } x_n(c_n, a_n) \Rightarrow E_i \quad (2)$$

$$\textit{If } x_1(c_1, etq_1) \textit{ and } x_2(c_2, etq_2) \textit{ and ... and } x_n(c_n, etq_n) \Rightarrow E_i \quad (3)$$

$$h(x_i, a_i) = \begin{cases} Large\ if\ a_i > 40\%A_{x_i} \\ MLarge\ if\ 25\%A_{x_i} < a_i \leq 40\%A_{x_i} \\ Medium\ if\ 15\%A_{x_i} < a_i \leq 25\%A_{x_i} \\ MShort\ if\ 5\%A_{x_i} < a_i \leq 15\%A_{x_i} \\ Short\ if\ a_i \leq 5\%A_{x_i} \end{cases} \quad (4)$$

$$\textit{If } x_1(c_1, \text{"}Short\text{"}) \textit{ and } x_2(c_2, \text{"}Medium\text{"}) \Rightarrow E_i \quad (5)$$

For the semi-qualitative model the amplitude is replaced by a label generated from it, with respect to the total range of the interval of the attribute. In equation 3 we show the format of semi-qualitative rule, where the labels for the amplitudes have been generated from equation 4. In this case, $A_{x_i}$ is $x_{iM} - x_{im}$, where $x_{iM}$ is the maximum value of the range for the attribute $x_i$ and $x_{im}$ is the minimum value. Thus, the interpretation for the rule shown in figure 5 would be: "*If $x_1$ takes a value with a difference "Short" on $c_1$ and $x_2$ takes a value with a difference "Medium" on $c_2$ then the class is $E_i$*"

The graphic interpretation is through a parallel coordinate system similar to the one of GAR, where each parameter is represented by a vertical line where it is shadowed the interval of suitable values for this parameter.

## 3 Comparison of results

Next, we are going to show the results obtained from the application of HIDER, Ellipses and GAR techniques to the database generated with different scenarios of the project used in this research. We want to point out that in this paper, only the rules that incorporate a greater number of scenarios and hits, as was said in the former paragraph, are presented for each technique.

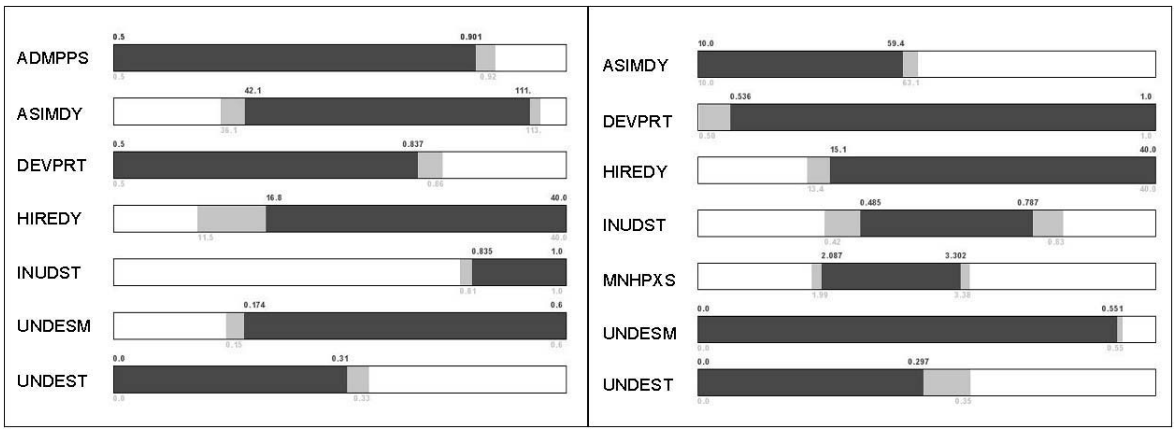


**Figure 2. HIDER rules**

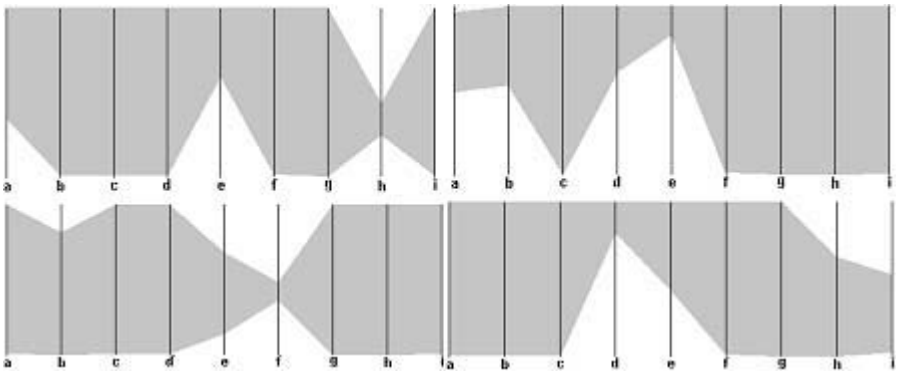


**Figure 3. Rules obtained for Ellipses. (Key: ADMPPS (a),ASIMDY(b),DEVPRT (c), HIREDY (d), INUDST (e), MNHPXS (f), TRPNHR (g), UNDESM (h), UNDEST (i))**

### 3.1 Experiment 1

If we impose restrictions on the three variables (effort, time and quality, simultaneously) in such a way that we label as good those records that fulfil a time that is less than or equal to 2092, an effort that is less than or equal to 387 and a quality that is less than 0.4, these conditions are fulfilled by 48 of the 500 records, that is to say a bit less than 10%.

HIDER finds two rules with 14 success each one, where 7 of the 9 parameters take part, although some of the restrictions are rather lax. Figure 2 shows in a graphic way these rules. In figure 2(right), we can be see that the most important restrictions can be summarized by saying that:

*ASIMDY must take low or low-medium values, HIREDY must take medium or high values, INDUST and MNHPXS must take medium values, UNDESM must not take high values and UNDEST must take low or low-medium values*

Figure 2(left) shows that another rule to find projects fulfilling the imposed restrictions is:

*ADMPPS must not take high values, ASIMDY must take medium or medium-high values, DEVPRT must take low or medium values, HIREDY and UNDESM must take medium or high value, INUDST must take high values and UNDEST must take low or low-medium values*

Following the former criteria, the most suitable rule, out of those obtained with HIDER, would be the second one, since is the closest to the initial estimations and besides, it is the one with which the best results would have been obtained. That is to say, ”final results would have been improved if we had increased the number of technicians in relation to the average estimated value at the beginning of the project and if we had slightly improved estimations on the size of the project”. Good values were estimated for the rest of the attributes of this rule, that is to say, the estimated value is within the range indicated in the rule. On the other hand, we want to indicate that, from the point of view of the graphic representation, it is easy to read ,since the maximum range of values, the optimum range to fulfil

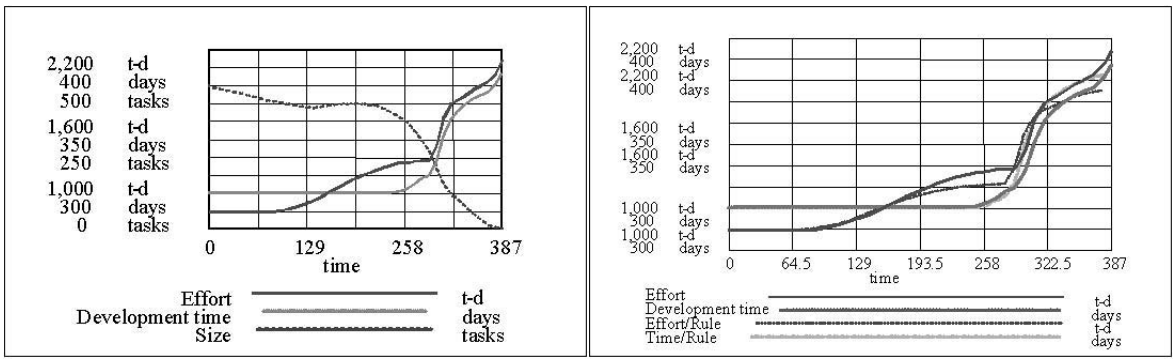


**Figure 4. Evolution of the project (nominal simulation) (left). Simulation of the project if we had applied rule 2 of HIDER or rule (c) of ELLIPSES (right)**

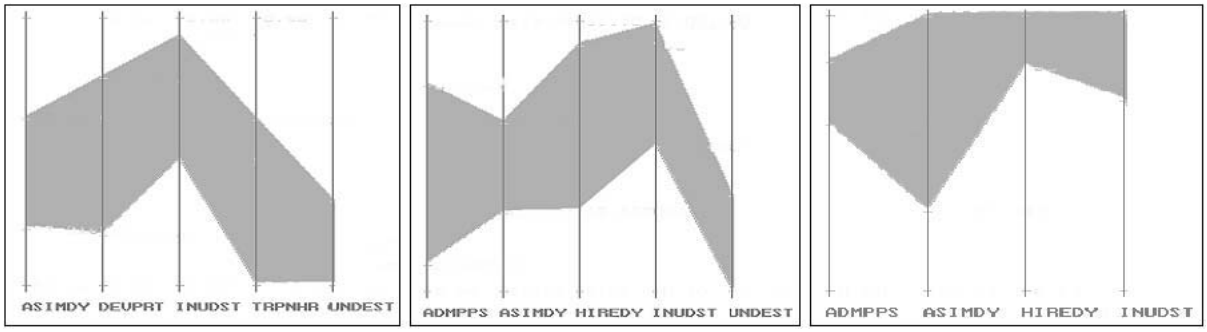


**Figure 5. Rules found by GAR**

with the goals and the uncertainty margins are indicated for each of the attributes.

Ellipses finds rules that are shown in figure 3. Figure 3-a is explained as follows:

*ADMPPS must take medium-high or high values (centre 0.84, margin 0.16), INUDST must take high values (centre 0.85, margin 0.15), and UNDESM must take medium-low values (around 0.2 with a margin of 0.05)*

Rule in figure 3-b shows that projects with the three conditions will be given if:

*ADMPPS, ASIMDY and HIREDY take medium-high or high values (centre 0.9, 75 and 33 and margins 0.1, 25 and 7 respectively) and INUDST takes very high values (centre 0.9 and margin 0.1)*

Figure 3-c shows a region quite similar to the one defined by the second rule of HIDER specially as regarding the restrictions on INUDST and MNHPXS parameters. Finally, rule 3-d is related to rule 3-b, since restrictions on HIREDY and INUDST are similar (although they are more restrictive to HIREDY and less to INUDST), but it changes restrictions on ADMPPS and ASIMDY by restrictions on UNDESM and UNDEST. The rule remains as follows:

*HIREDY must take very high values (centre 36 and margin 49), INUDST must take high values (0.8 and 0.2) and UNDESM and UNDEST must take low or medium-low values (centre 0.2 and margin 0.2)*

This is the best of the three rules analyzed as regarding the attributes that must be modified, since we obtain rules where only 3 of the 9 attributes are implied. Rule (c) has been selected (similar to rule 2 of HIDER). Besides, the attributes of this rule that must move in a narrow range of values (INUDST and MNHPXS) use to be attributes over which the project manager has easiness of decision and control.

In figure 4, the nominal evolution of the project is represented together with the evolution that it would have had if rule 2 of HIDER and rule (c) of ELLIPSES had been applied. The results for these rules are equal since we can take the same values for the attributes in both of them.

GAR technique finds three rules with a support larger than 85%. They can be seen in figure 5. The description of these rules is the following:

- *ASIMDY and DEVPRT must take medium values, INUDST must take medium-high or high values, TRPNHR must take low or medium values and UNDEST must take low values.*

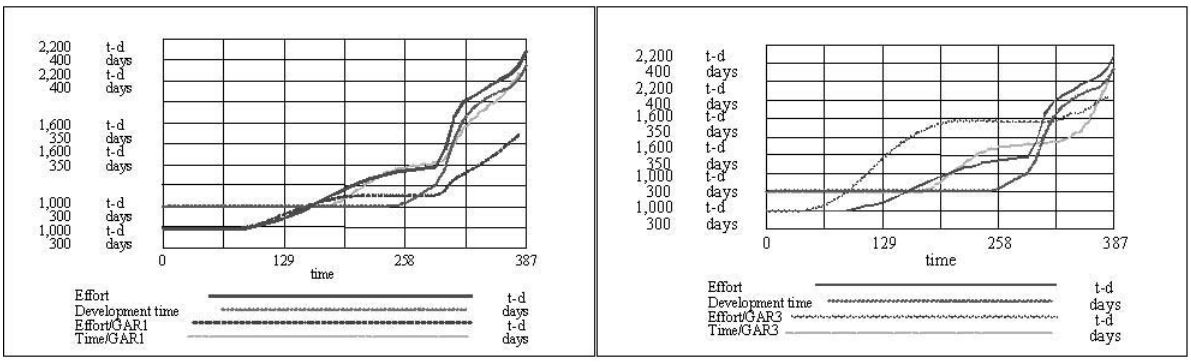


**Figure 6. Comparison of the nominal simulation with the project with the rules 1 and 3 of GAR**


**Figure 7. HIDER rules**

- *ADMPPS and ASIMDY must take medium values, HIREDY and INUDST must take high values and UNDEST must take low values.*

- *ADMPPS must take high values but not in excess, ASIMDY must take medium or high values and HIREDY and INUDST must take high values.*

With regard to the number of attributes involved in the rules obtained (4 or 5 from 9) it improves HIDER but not Ellipses. The chosen rule is, in this case, 1, for being carrying out a post-mortem analyzes of the project (rule 2 is similar), and, though this rule involves 5 attributes (one more than in rule 3) is the one closest to the estimations carried out for this project. That is to say, "the results obtained would have improved if we had improved the initial estimations on the size of the project (UNDEST), if we had increased the average delay of adaptation for new technicians (ASIMDY) and if we had increased the number of technicians at the beginning of the project (INUDST)". Rule 3 would imply the modification of the four attributes since the estimated values for each one of them are not in the range of values shown in the rule.

In figure 6 the nominal evolution of the project is compared to the evolution that we would have obtained if we had applied rules 1 and 3 of GAR respectively.

In table 2, we show, in contrast with the nominal values, the final values that would have been obtained if we had applied the rules described in each technique. In all the cases, time, cost and quality are improved simultaneously, although the rule GAR-3 maximizes the improvement in time, GAR-1[1] in cost and HIDER or Ellipses[2] in quality.

**Table 2. Values and improvement percentages obtained with each technique**

| Rule | Time (387) | Cost (2092) | Quality (0.26) |
|---|---|---|---|
| HIDER-2 | 382 (1.3%) | 1899 (9.2%) | 0.19 (26.9%) |
| Ellipses-2 | 382 (1.3%) | 1899 (9.2%) | 0.19 (26.9%) |
| GAR-1 | 383 (1%) | 1647 (21.3%) | 0.26 (0%) |
| GAR-2 | 380 (1.8%) | 1869 (10.9%) | 0.26 (0%) |

## 3.2 Experiment 2

If we obviate the restriction imposed on the quality variable, the number of records with good values is

[1]The explanation could be that, in general, when the attributes related to the incorporation of new technicians take high values (that is, the incorporation is realized slowly), results obtained in the effort needed to carry out the project are improved.

[2]We'll have a similar result if the chosen value for INUDST (within the range obtained in the rule) is the same that the one in rule 2 of HIDER.

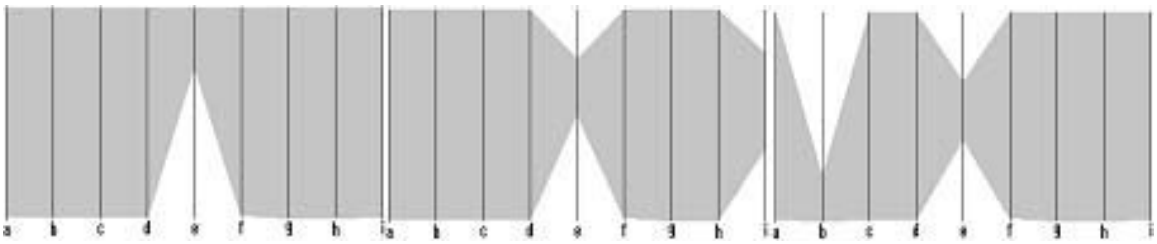

**Figure 8. Rules obtained by Ellipses. (Key: ADMPPS (a),ASIMDY (b), DEVPRT (c), HIREDY (d), INUDST (e), MNHPXS (f), TRPNHR (g), UNDESM (h), UNDEST(i))**

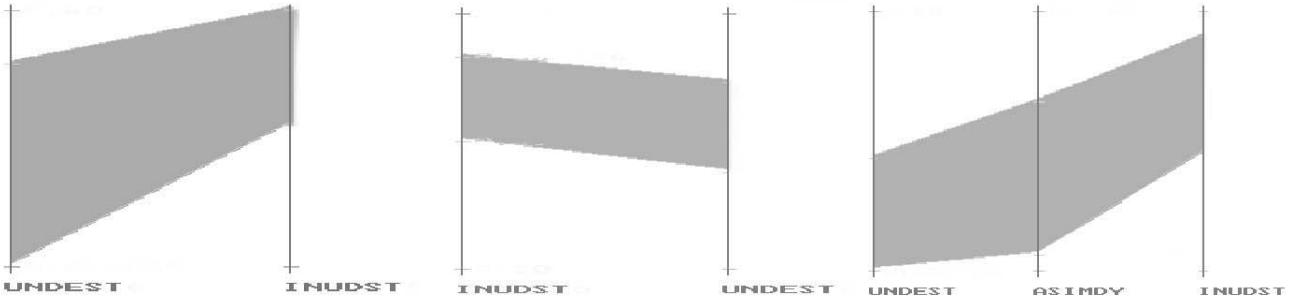


**Figure 9. GAR rules**

221. The main characteristic of the rules found with the different techniques is that the number of restrictions on the parameters is less than in the former case.

HIDER finds rules that collect a great number of cases although with some erroneous records. Thus, rule in figure 7-a collects 153 records within the imposed limits and 7 errors with only one strong restriction: INUDST must take values larger than 0.6 and, three more slight restrictions on ASIMDY, TRPNHR and UNDEST which are kept from taking only extremely high values. Rule in figure 7-b collects 141 good records and 7 errors. Restrictions are only on INUDST for values higher than the previous one and on UNDEST for not very high values. This rule is a refinement of the former, in the sense that, restrictions on ADMPPS, ASIMDY and TRPNHR (slights in any case) disappear and are replaced by a larger restriction on INUDST that goes from having a limit of 0.59 in the first one to have a limit of 0.67 in the second.

Ellipses also finds rules that collect numerous records. Surprisingly, it is the first of the rules (figure 8-a) with only one restriction on INUDST, imposing very high values (about 0.9 with a margin of 0.1) that collect 135 records, 121 of which fulfil restrictions on time and effort.

Other interesting rules, although with a less number of covered records, are shown in figures 8-b and 8-c. In the first one, the restriction on INUDST is relaxed to values about 0.7 with a margin of 0.1, but appears a restriction on UNDEST with values about 0.4 and a margin of 0.12. Figure 8-c, relaxes a bit more the conditions on INUDST to values about 0.6 but imposes conditions on ASIMDY to low values (about 21.5 and a margin of 11.5). To sum up, it can be seen that the parameter INUDST is the one that marks the trend, in such a way that by itself, when taking high values, determines a rule. This restriction of very high values, can be relaxed if some restriction to the parameters UNDEST or ASIMDY is added.

GAR finds rules similar to the previous ones. Thus, figure 9-a shows a rule similar to the one in figure 7-right, forcing INUDST to take values greater than 0.65. Nevertheless, it is also capable of giving other rules that relax conditions on INUDST, adjusting more UNDEST (figure 9-b) or establishing restrictions on ASIMDY (figure 9-c).

## 4 Conclusions

The three techniques we have presented are able to generate management rules suitable for the searched goals. In this paper we have emphasized the visualization and the qualitative interpretation of such rules. If we establish a comparison among them, we can see that HIDER and Ellipses obtain good rules to optimize quality while GAR obtains better results for effort. Ellipses and GAR obtain rules with less number of implicated attributes although

with less support (they collect less examples).

The visualization of the rules in HIDER is more complete because of the presentation of the uncertainty intervals and in GAR is the simplest because it only shows the parameters implicated.

With regard to the section where it is only considered the obtaining of good results for time and effort simultaneously without considering quality and from the point of view of the analyzed techniques, we reach the same conclusions that in the former section. As we expected, we can check that when we limit the goals of the project, the rules obtained:

- Collect a great number of records and hits so they guarantee in a higher degree the obtaining of good results.
- The number of attributes in the rules is lesser so they are easy of applying and of controlling their application.

In conclusion, in this paper we propose the application of new techniques to the management of projects in order to facilitate the decision-making and at the same time the fulfilment of the goals of the project. The election of one or another rule will depend on the priorities that the project manager have.

## References


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

[2] J. Aguilar-Ruiz, J. Riquelme, and M. Toro. Evolutionary learning of hierarchical decision rules. *IEEE Trans. on Systems, Man and Cybernetics*, 2002.

[3] J. Alvarez, J. Mata, and J. Riquelme. Mining interesting regions using evolutionary algorithms. *17th ACM Symposium on Applied Computing (SAC 2002)*, pages 498–502, 2002.

[4] N. Fenton and M. Neil. A critique of software defect prediction models. *IEEE Trans. on Soft. Eng.*, 25(5):675–689, 1999.

[5] A. Inselberg. The plane with parallel coordinates. *Special Issue on Computational geometry, The Visual Computer*, (1), 1985.

[6] J. Mata, J. Alvarez, and J. Riquelme. Discovering numeric association rules via evolutionary algorithm. *LNCS 2336*, pages 40–51, 2002.

[7] M. Mendonca and N. Sunderhaft. Mining software engineering data: A survey. *Report DACS-99-3*, 1999.

[8] M. Ruiz, I. Ramos, and M. Toro. A simplified model of software project dynamics. *The Journal of Systems and Software*, 3(59):77–87, 2002.

[9] J. Tian and J. Palma. Analizyzing and improving reliability: A tree-based approach. *IEEE Software*, 2(15):97–104, 1998.