

TC2006. Programming languages.

Assign employees to a Project depending your skills and requirements of the project.

Jose Eduardo Cadena Bernal | A01704641

Benjamín Valdés Aguirre PhD

May 26th 2021

Contents

Assign employees to a Project depending your skills and requirements of the project.....	1
Abstract.....	2
Context of the problem.....	3
Solution	6
Implementation	7
Diagram.....	8
Business rules.....	9
Defining the Data.	9
Results.....	11
Conclusions	12
Possible improvements.....	12
Set up	13
References	13

Abstract

The present project makes usage of the logic programming paradigm to be able to simulate a staff assignment program based on project requirements and staff availability, role, and skills. In order to make the solution more user friendly I decided to connect the logic implemented in prolog with Java for the graphical user interface, using the JPL library.

Context of the problem

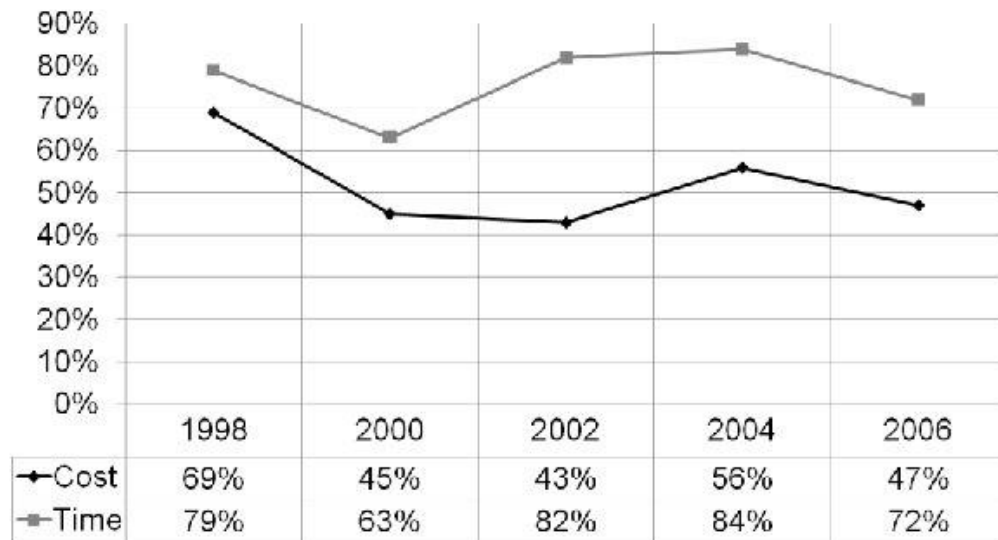
There are a lot of factors why the project end wrong:

- Unclear objectives
- Wrong selection of projects
- Wrong staff assignment
- Non-detailed work plans
- Unrealistic commitments
- Non-existent risk management
- Constant change of scope
- Inconsistent processes
- Poor communication
- Lack of clear and timely indicators

According to Standish group in North America only the 30% of the project are finished successful.

What does that mean? A successful project are the one who finished on time, on cost and scope.

Time and Cost over run



Source Standish Group 2007 Chaos study

25

CHAOS RESOLUTION BY AREA OF THE WORLD

	SUCCESSFUL	CHALLENGED	FAILED
North America	30%	53%	17%
Europe	29%	54%	17%
Asia	23%	57%	20%
Rest of World	26%	51%	23%

The resolution of all software projects from FY2012-2016 by the four major areas of the world.

When we are talking about manage and start a project, its fundamental to choose the right member who is going to perform specific tasks. A big part of an efficient administration is based on be sure that the right members are performing the right task and participating in the corresponding project based on their skills.

According to the site CNN expansion, many times the error in the companies is that projects are assigned to people according to their availability, luck or favoritism, by organizing in that way there are a lot of probabilities that they are wasting resources. That's the worst way to assign people to a project. Not all of us have the same training, capacities, aptitudes, or skills. Assign human resources that fit the needs of the project ensures a very high probability of success

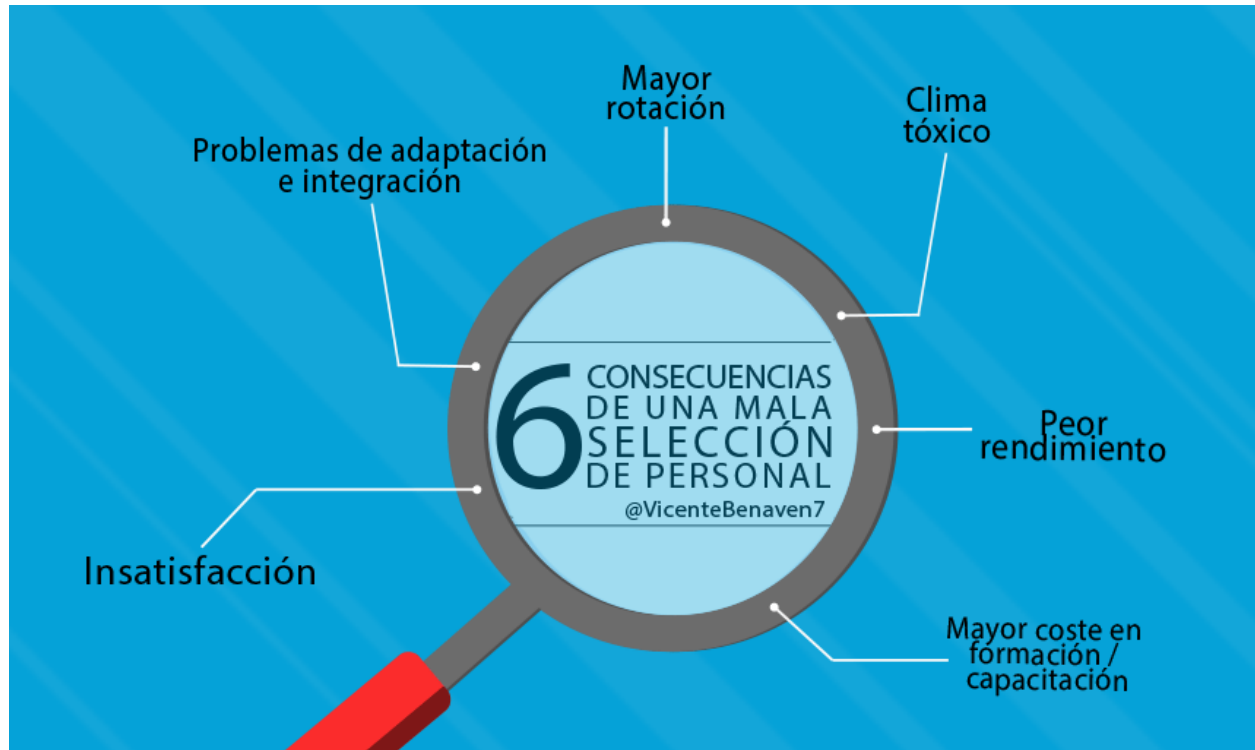
Not having a good selection and recruitment method can trigger a series of errors that lead to loss of time, money, image and work environment, among others.

What consequences can a bad decision have when it comes to recruiting?

Make a mistake in an election of assign a employee will lead to loss in different levels, first economic cost in terms of return it will not be profitable. It will provoke a lack of trust. This can be other negative effects:

- Loss of productivity during the integration process. This loss is normal in new hires, but if the learning curve is too slow because integration is being problematic, the profitability of the operation can be seriously damaged and cause additional stress on all those involved.
- Administrative costs involved in the selection and hiring. In addition to those derived from an eventual dismissal.
- Decrease in the confidence of the rest of the actors that make up the company's ecosystem. And trust in a company affects both employees and customers, suppliers and partners. The discredit is directly linked to the decline in profits.
- Increased turnover. If the error is not corrected in time, it is not refined to adjust and improve the process of selecting new recruits, this instability in the staff may be too recurrent. Productivity and image will suffer. The question will arise automatically: why do they change so many managers?
- Bad environment. The inadequate integration of a person in a logical time to the rhythms of the company, to its culture, will inevitably cause a bad work climate. This climate of negativity will

affect both the rest of the squad and the newcomer who will not find himself comfortable to give his best.



Solution

For this project I was inspired by my past experiences, I have been in projects where the members don't belong to the team, not because they are bad in programming or something related, it is because they have other skills.

The idea of the project is to give all the possible combinations of member for each project, of course only the members who fit on each project. It could be people that fit in all of them, or cases where nobody fits in one project.

Implementation

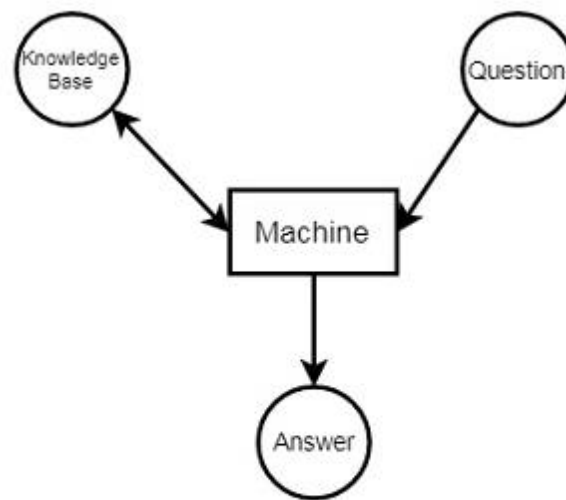
The project is programmed in Prolog, this language is a declarative logic programming language, It is an attempt to make a programming language that enables the expression of logic instead of carefully specified instructions on the computer. It is very common to be used in artificial intelligence applications.

The execution of a Prolog program is equivalent to searching possibilities and determining the objects which satisfy the given rules. There can be several answers which are true in the given circumstances. The program does not terminate as soon as the first found answer, it keeps until the entire tree of possibilities has been checked.

Logic programming is a programming paradigm that is based on logic. This means that the language has sentences that follow logic, they express facts and rules. Computation using logic programming is done by making logical inferences based on all the available data. In order for computer programs to make use of logic, there must be a base of existing logic, called predicates. Predicates are used to build atomic formulas or atoms. Which state true facts. Predicates and atoms are used to create formulas and perform queries.

Logic programming can help thought

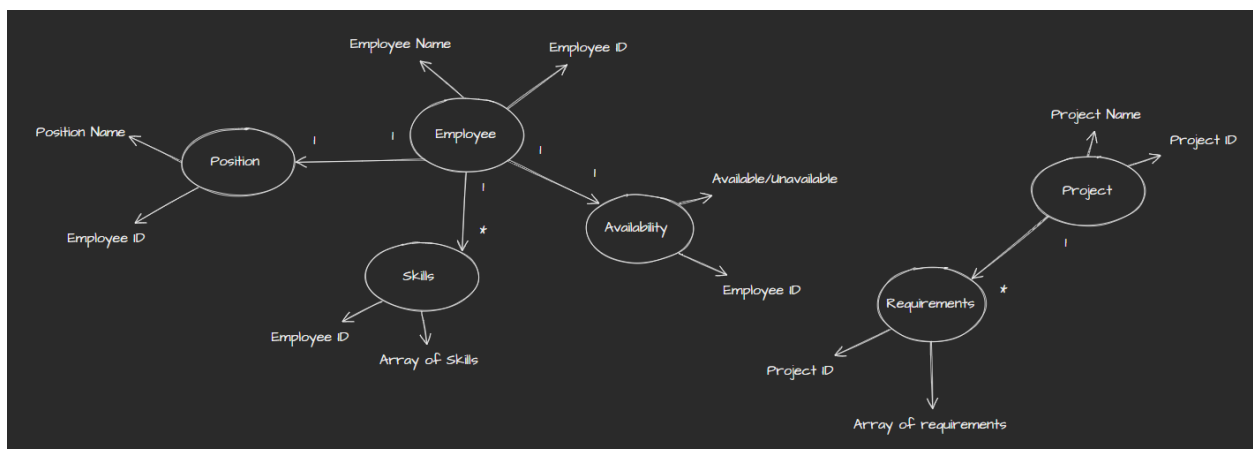
- Natural language processing: allows for better interactions between humans and computers
- Database management: can be used for creating, maintenance and querying of a NoSQL database
- Predictive analysis: with a lot of data, the language can search for inconsistencies or areas of differentiation in order to make predictions.



Logical Programming

Diagram

First, I have defined the data and the relation between them, also I defined the attributes. to simulate a relational database. the graph has circles, which are representing the entities (tables), each one has attributes.



- Each Employee has a name and ID.
- The position is related with the Employee, the relations is 1 to 1, it has the ID of the employee and the position name.

- The availability is related with Employee, the relation is 1 to 1, it has the ID of the employee and if its available or is unavailable.
- The skills are related with Employee, the relation is 1 to many, it has the ID of the employee and an array of skills.
- The project has a name and ID.
- The requirements are related to Project, the relation is 1 to many, it has the ID of the project and an array of requirements.

Business rules

To assign employees to a project the program follows these rules:

- The total number of members for each project is 3.
- At least one member must be Solution Architect.
- At least one member must be Developer Senior.
- At least one member must be Developer Junior.
- The Solution Architect must satisfy with 50% of the requirements of the project.
- The Developer Senior must satisfy with 33% of the requirements of the project.
- The Developer Junior must satisfy with 10% of the requirements of the project.
- Each member must be available.
- If no one in any position satisfy with the percentage or the requirements, the program return all the employees in this position who are available.

Establish facts and relations.

We can define the base cases and establish facts such as:

The employee has a name and id.

```
employee('Angeles Anaya',1).
```

Employee_position is related with employee with the id of the employee, and the second value is the position of the employee.

```
employee_position(1,solutionArchitect).
```

Employee_skill is related with employee with the id of the employee, and the second value is an array of the skills that each employee has.

```
employee_skill(1,[azure,cloud,java,cpp,excel,python,prolog,db,c]).
```

Employee_available is related with employee with the id of the employee, and the second value means if its available or unavailable.

```
employee_available(1,unavailable).
```

Each project has a name and an id.

```
project_id(overtimetool,00099).
```

Project_requirements are related to project with the id of the project, and the second value is an array of the requirements skills that each employee needs.

```
project_requirements(00099,[cloud,azure,excel,python]).
```

Unification.

Unification is very useful in this paradigm, by replacing certain sub-expression variables with other expressions, unification tries to identify two symbolic expressions.

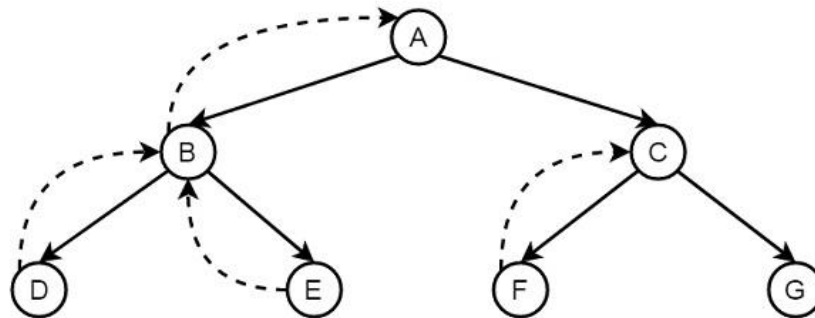
```
check_requirements(Project,[H|TA],[H|R],P):-  
    employee(_Nombre,H),  
    employee_available(H,Available),  
    available(Available),  
    project_id(Project,IdProject),  
    employee_skill(H,Skills),  
    project_requirements(IdProject,Requirements),  
    intersect(Skills,Requirements,ListIntersected),  
    list_length(ListIntersected,Size),  
    list_length(Requirements,SizeRequirements),  
    fits_criteria(SizeRequirements,Size,P),  
    !.
```

In this call we unify the variable Available as you can see.

```
Call: employee(_30680, 1)  
Exit: employee('Angeles Anaya', 1)  
Call: employee_available(1, _30552)  
Exit: employee_available(1, unavailable)
```

Backtracking.

This searches the truth value of different predicates by checking whether they are correct or not. In other word the program will go back to the previous goal, and it will try to find another way to satisfy the goal.



Recursion.

Recursion in any type or programming language are function that call itself unit the goal has been succeed.

In prolog the recursion appears when a predicate contains a goal that refers to itself.

Results.

When you open the User Interface you will find a dropdown which shows you the different projects, also you can 3 empty dropdowns, one for the Software Architects, other for Developers Seniors and another for the Developers Juniors

Imagen de la interfaz del usuario

You select the Project for which you want to assign employees and click in search for options.

Immediately the program will send the query to the prolog program and it will start to implement the different rules to search the best members for the team.

The query looks like this:

```
create_Team_Options(datawarehouse,Namesavailablearquitects,Namesavailablejuniors,Nameavailableseniors,Flags)
```

If the first parameter you send the name of the project, the the next one you send different types of variables that the program will return with different types of values.

Namesavailablearquitects,Namesavailablejuniors and Nameavailableseniors are Arrays of employees names that fits in the project

Flag you receive and array of 1s and 0s, this because if nobody fits the criteria the system will return the array depending of the position of all the employees available, and the flag will turn to 1, if there is only one employee who fits the criteria, you will see only that employee and the flag will be 0.

Example, there is at least one employee for 1 position that fits the criteria.

```
Flags = [0, 0, 0],
Nameavailableseniors = ['Renata Garcia', 'Sofia Rodriguez', 'Mateo da Silva', 'Leonardo Di Caprio', 'Emiliano Persaud', 'Miguel Smith',
'Alexander Smith'],
Namesavailablearchitects = ['Jonathan Cruz'],
Namesavailablejuniors = ['John Yakimeshi', 'Cuauhtemoc Segundo', 'Pepe Tercero', 'Guadalupe Martin', 'Antonio Rusu', 'Daniel Rossi', 'Miguel Tamm',
'Petter Ivanov', 'Liam Cuarto', 'Ian Ramirez', 'Lucas Muecas', 'Negrito Bimbo']
```

Example, there is nobody that fits the criteria.

```
Flags = [1, 1, 1],
Nameavailableseniors = ['Renata Garcia', 'Sofia Rodriguez', 'Mateo da Silva', 'Santiago Gonzalez', 'Daniel de la Hoya', 'Regina Quisque',
'Leonardo Di Caprio', 'Emiliano Persaud', 'Miguel Smith', 'Alexander Smith', 'Ali Mohammed'],
Namesavailablearchitects = ['Jonathan Cruz', 'Luis Rosado', 'Eduardo Cadena', 'Angie Contreras', 'Jose Jose', 'Gilberto Ortega'],
Namesavailablejuniors = ['John Yakimeshi', 'Cuauhtemoc Segundo', 'Pepe Tercero', 'Guadalupe Martin', 'Antonio Rusu', 'Daniel Rossi', 'Miguel Tamm',
'Petter Ivanov', 'Liam Cuarto', 'Ian Ramirez', 'Lucas Muecas', 'Negrito Bimbo']
```

```
Flags = [1, 1, 1],
Nameavailableseniors = ['Renata Garcia', 'Sofia Rodriguez', 'Mateo da Silva', 'Santiago Gonzalez', 'Daniel de la Hoya', 'Regina Quisque',
'Leonardo Di Caprio', 'Emiliano Persaud', 'Miguel Smith', 'Alexander Smith', 'Ali Mohammed'],
Namesavailablearchitects = ['Jonathan Cruz', 'Luis Rosado', 'Eduardo Cadena', 'Angie Contreras', 'Jose Jose', 'Gilberto Ortega'],
Namesavailablejuniors = ['John Yakimeshi', 'Cuauhtemoc Segundo', 'Pepe Tercero', 'Guadalupe Martin', 'Antonio Rusu', 'Daniel Rossi', 'Miguel Tamm',
'Petter Ivanov', 'Liam Cuarto', 'Ian Ramirez', 'Lucas Muecas', 'Negrito Bimbo']
```

Conclusions

Possible improvements

Set up

References

[Consecuencias de reclutar de forma errónea | HRTRENDS \(infoempleo.com\)](https://infoempleo.com)

["Asignación equivocada del personal" \(3/10\) Factores que destruyen proyectos \(systec.com.mx\)](https://systec.com.mx)

[Criterios para seleccionar a tu equipo de trabajo en un proyecto | OBS Business School](https://obsbusinessschool.com)

[PROLOG | computer language | Britannica](https://www.britannica.com)

[Learn Prolog Now! \(rug.nl\)](https://rug.nl)

Facts

As said earlier, the conditions that a Sudoku puzzle has to comply with can be represented as facts. A fact is a predicate expression that makes a declarative statement about the domain of a problem, an example in this project is shown below:

This predicate declares that every row of the Sudoku puzzle must be of length 9, where Rows, is a variable that is matched to item 9 through unification.

Rules

Rules are predicates that use logical implication to describe the relationship among facts, for this particular program the Sudoku solver takes the form of a rule and the facts that it describes are the constraints that make a grid of numbers a Sudoku puzzle. For example, the rule of the implemented program takes this form:

Backtracking

The process that takes place when one or more of the given conditions are broken, backtracking is to retrace all steps until the point where all conditions were satisfied.

Prolog

Non-procedural (declarative) programming language conceived by Alain Colmerauer in 1973 and further developed by Robert Kowalski. The language makes use of the resolution theorem-proving method of automated deduction developed in 1963 by J. A. Robinson.

Prolog is popular in AI, natural language processing and is the most widely used language for the logic programming paradigm because it allows the programmer to engage in controlled deduction, meaning that the solutions obtained rely on run-time engine, hence programmers should describe the space to be explored by the engine rather than the processes to calculate results directly.