Miodrag Strak
Slavka T. Nikolić

Implementation of Complex Projects Using Constraint Programming

Article Info:

Management Information Systems,
Vol. 7 (2012), No. 3,
pp. 011-019

Received 12 Jun 2011
Accepted 24 April 2012

UDC 007:004 ; 004.42:005.8

Summary

During the implementation of the complex projects, all planned activities and resources must be taken into account. In general, it is necessary to assign the resources to the activities, but to also avoid simultaneous engagement of resources for multiple activities. In order to solve these problems, various techniques and methods are used. Mathematic and integer programming, genetic algorithms, simulated annealing, or taboo search are just some of the techniques used for solving this problem. Constraint programming comes from artificial intelligence i.e. papers from this area that occurred in 1960s and 1970s. Constraints exist in every segment of human environment. They represent a natural medium for expressing relations that exist in the physical world. Fulfilment of constraints is used in many different areas. Problems such as scheduling, allocations etc. are typical examples of constraints problems, where the basic concept of constraint programming can be applied. This paper considered implementation of the Bor Regional Development Project. Development of constraint programming was followed by the development of appropriate tools. B-Prolog was used in this paper. Many systems, including B-Prolog, enable interface with classic object-oriented languages, such as C++ or Java. One of the greatest advantages is the possibility of simple modelling, even for beginners in planning and implementation of the project.

Keywords

implementation, complex projects, project management, constraint programming

1. Introduction

A need to make decisions regarding constraints in the allocation, number of types and quantity of different resources over time and within set activities often presents a problem during the implementation of real projects. The resources can be staff, machines, material etc. Activities are operations in certain production or service process, such as administration tasks, operating machines etc. Resource allocation is a problem of great practical significance that scientists in the area of operations research attempted to tackle for a long time. Mathematically, resource allocation represents an optimization problem, where constraints are fully known, optimization criterion is clearly and precisely defined and everything happens in predictable conditions. Development of fully automated systems for solving allocation problems is often not fully supported by the users. The reason could be constraints that are often very difficult to identify completely, decision criterion is difficult to determine, and the users are not experts in using complex mathematical expressions such as matrix algebra or the multi-criteria optimization.

This paper will discuss parts of implementation of the "Bor Regional Development Project" referring to organization of public works for the unemployed.

Due to difficult situation in Bor and Majdanpek municipalities, Republic of Serbia took a loan from the World Bank in order to implement "Bor Regional Development Project", with a goal to solve urgent ecological problems and improve socio-economic situation. The latter part of the project is primarily directed targeted at the unemployed, which increased in number after the restructuring of the local industrial giant RTB Bor. One of the most important is the Temporary Employment Program, which supports local communities, by means of organising public works.

2. Resource Allocation

Resource allocation and scheduling are the problems that often occur in practice. Good allocation of staff, equipment, money etc. enables efficient usage of necessary resources in the project implementation stage. Allocation problems are tasks and resources that need to be managed. Solving these problems is difficult due to accompanying requests, expressed through constraints, which are often opposite to each other. Resource allocation is a discrete optimization problem and belongs to NP-hard problems. Finding solutions that meet all constraints means searching a huge number of possible solutions. In the case of allocation, its size is most often exponential compared to the size of the problem.

Implementation of modern projects is burdened with extreme complexity and uncertainty. Construction of a building, road, dam, development of new airplane prototype, starting new production, fair organization etc. are just some examples of modern, large-scale projects. This type of problem is also the group of problems where resources are assigned to specific activities. This process is accompanied by a great number of constraints: duration of activity, resource availability, resource allocation etc. These constraints define the scope of allowable solutions. In order to find any solution, it is often necessary to additionally ease the problem with additional, priority constraints. These constraints are often opposed to one another. In practice, priority constraints are combined in a single function that needs to be minimized i.e. maximized, and the other way is to create appropriate heuristics. In both cases, the allocation problem is defined as a group of constraints that need to be fulfilled. Activity is the basic unit within the project that requires time and resources and needs to be planned and allocated within the whole project. The number of activities in projects varies from several to thousands. Certain algorithms that give good solutions for small problems can be completely useless with bigger problems (Le Pape, 1994).

A large number of participants of different profiles are involved in the realization of any project and techniques are developed that enabled representation of projects in a way that would be understandable to everyone involved. Methods that are especially emphasized are methods of network planning CPM and PERT, which are used for planning the resolution of a specific problem. Methods of network planning are methods based on graphic presentation of activities schedule within the project and their mutual dependencies. Such logical structure enables detailed analysis of realization duration of specific activities and the project as a whole.

Besides the above-mentioned elements, the fact that large quantities of resources are used contributes to the great complexity. Since the available resources for the realization are most often limited, it is necessary to plan them in an appropriate way. Planning project activities regarding the necessary resources includes determining certain types of resources and deadlines when the specific resources are needed. Resources in the concept of project management include various material and parts, workforce, equipment, financial assets etc. (Jovanović, 1999).

The resource planning process involves planning material, labour force and equipment, i.e. planning all specific types of resources necessary for the implementation of the project. Planning different resources is a very complex process, and these specific processes must be synchronized and included in the whole resource planning process, i.e. the whole of project implementation planning process (Jovanović, 1999; Lever, Wallace, & Richards, 1995).

Implementation of a project without taking into account the engagement of limited resources can lead to its inefficient usage and, consequently, to delays of certain activities and the project as a whole. Presentation of the project with the standard methods of network planning is good from the aspect of presenting the whole project and mutual dependencies of specific activities, as well as timeframe analysis. However, these methods are not so useful in terms of planning and allocation of activities that take into account the need for certain resources. Resource constraints define resource capacity needed for the execution of certain activities.

Constraints, in general, have several features (Barták, n.d.):

- Constraints can describe partial information i.e. they do not have to explicitly define the values of variables, for example when $X > 2$, the value of the variable X cannot be determined;
- Constraints are heterogeneous i.e. it cannot be used for describing relations between the variables whose values are in different domains;
- Constraints are not directed, for example, limitation $X = Y + 2$ with two variables can be used to get the values for both of them;
- Constraints are declarative i.e. define the relation that must exist, but not the procedure how the relation is defined;
- Constraints can be added and the sequence of constraints is not important if the final result is a conjunction of the constraints and
- Constraints are not independent because refer to the same variables.

Developing real applications for solving resource allocation problems has shown that the classic approach to constraints could not always describe the problem adequately. The reason lies in the fact that simple true/false statements cannot describe the real problems. These problems are also related to the terms of preferences, probability or uncertainty. Besides, many real problems, regardless of their proper modelling, have too many constraints that cannot be fully met at the same time. Thus, a concept of hard and soft constraints is introduced (Bistarelli, Codognet, Georget, & Rossi, 1999; Rossi, n.d.).

Hard constraints are those that have to be fully met during the problem solving. Unlike them, the number of fully met soft constraints is a measure of quality of the solution. In general, it is necessary to minimize the number of soft constraints that cannot be met.

3. Constraint Programming

In general, a constraint can be seen as a restriction related to the scope of possible solutions. Mathematical constraints are precisely defined relations between several variables, which get the values from the given domains. Constraints limit the possible values which a variable can get and represent a natural medium for formalization of rules which provide a connection between natural and physical world and its mathematical abstractions, which dates back to Euclidean geometry, if not before (The Constraint Programming Working Group, 1996).

For example, the sum of three angles in a triangle is always 180 degrees, bases that constitute DNA chain can be combined only in the precisely defined way, pillars of a bridge can hold precisely defined static and dynamic pressure...

Constraint Programming (CP) represents a computer system based on constraints. It is a result of century-old discussions on analysis and deduction in mathematics along with a several modern techniques, such as programming languages, with the aim to create and implement adequate program, modelling and problem solving in different domains. Modelling languages that use some logic constructions constitute a separate group. This group comprises the Constraint Logic Programming (CLP) and Concurrent Constraint Programming (CCP).

A big advantage of using CP programming instead of other methods is using constraints to reduce time necessary for problem solving. Constraints are not only used for checking the solution of the problem, like in conventional programming languages, but also to solve the problem through the process of inconsistency detection (Le Pape, 1994).

Success in searching the scope of possible solutions often depends on the selected model and its accompanying options. In practice, there are global constraints for modelling that secure better results of the problem solving process. Complex conditions for groups of variables can be modelled declaratively with these types of constraints (Frühwirth et al., 1993; Goltz, Küchler, & Matzke, 1998; Goltz, 2000; Goltz & Matzke, 1999; Gueret, Jussien, Boizumault, & Prins, 1995).

3.1. Cumulative Constraint

Cumulative constraint is created specifically to solve the problem of allocation and scheduling. The constraint is described as follows (Goltz, et al., 1998; Gueret et al., 1995; Marriott & Stuckey, 1998):

cumulative([S1,...,Sn],[D1,...,Dn],[R1,...,Rn],L)

where [S1,..., Sn], [D1,..., Dn], [R1,..., Rn] are non-empty series of domain of a certain variable, and L is an integer. S stands for activity starting point; D stands for a duration and R for necessary quantity of resources for each activity. L is a total number of resources. This constraint secures that the number of allocated resources cannot be higher than the set number L in every moment of activity.

For example, it is necessary to schedule three activities, first of which requires one of the total number of available resources with duration of four time units, and activities 2 and 3 use two resource units with duration of two and three time units, respectively. If the starting points of activities are set for each activity respectively to 0, 1 and 3, this constraint can be presented as in Figure 1.

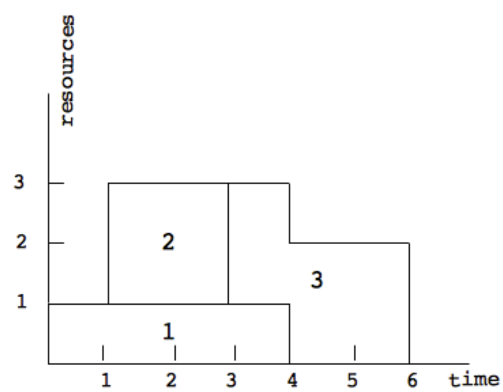

Figure 1 cumulative([1,2,4],[4,2,3],[1,2,2],3)
Source: Authors

The constraint would then be:

cumulative([1,2,4],[4,2,3],[1,2,2],3)

A total quantity of the engaged resources at every moment is less than or equals the total number.

In order to solve the problems with the resources that cannot be used by different activities, it is very important to take into account the case when the total number of resources equals one. This is especially important for solving the timetabling problem or scheduling, when, for example, a classroom or employee cannot be engaged for several tasks at the same time. The example for using cumulative constraint can be as follows:

cumulative([1,4,6],[2,1,1],[1 11],1)

Figure 2. illustrates this constraint.

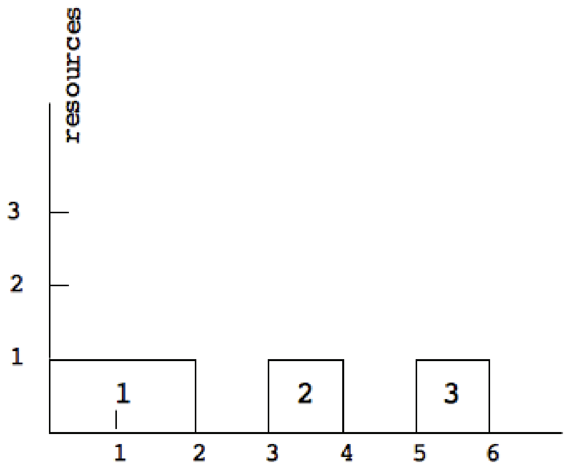

Figure 2 cumulative([1,4,6],[2,1,1],[1 11],1)
Source: Authors

3.2. Diffn Constraint

A constraint called diffn appeared for the first time within CHIP language for multi-dimensional positioning problem solving (Goltz, Küchler, & Matzke, 1998; Goltz & Matzke, 1999).

The arguments for this constraint are in the form of n-dimensional figures that are represented with n-tuples [X1,...,Xn,L1,...,Ln] where X is a start coordinate and L is a length of the figure in dimension:

diffn([X1,...,Xn,L1,...,Ln])

The constraint secures that the set figures are not overlapping in any direction.

Due to practical reasons of resource allocation problem, it is very important to discuss the case of two-dimensional rectangles, which should not overlap in any direction.

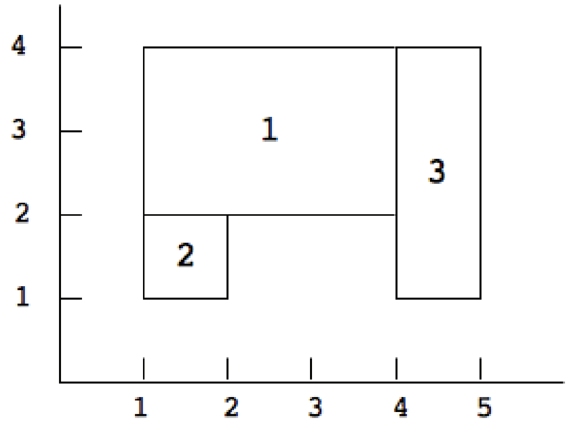

Figure 3 diffn([[1,1,1,1],[1,2,3,2],[4,1,1,3]])
Source: Authors

Rectangle market with the number 1 is shown as follows [1,2,3,2], where the first two numbers represent coordinates of the lower left angle point and the other two represent the sides' length in given directions. Similarly, rectangles 2 and 3 can be described as follows: [1,1,1,1] and [4,1,1,3], respectively.

The constraint for this example is as follows:

diffn([[1,1,1,1],[1,2,3,2],[4,1,1,3]])

4. B-Prolog

During the implementation of the complex projects, all planned activities and resources must be taken into account. In general, it is necessary to assign the resources to the activities, but to also avoid simultaneous engagement of resources for multiple activities. In order to solve these problems, various techniques and methods are used. Mathematic and integer programming, genetic algorithms, simulated annealing, or taboo search are just some of the techniques used for solving this problem. Constraint programming comes from artificial intelligence i.e. papers from this area that occurred in 1960s and 1970s. Constraints exist in every segment of human environment. They represent a natural medium for expressing relations that exist in the physical world. Fulfilment of constraints is used in many different areas. Problems such as scheduling, allocations etc. are typical examples of constraints problems, where the basic concept of constraint programming can be applied. This paper considered implementation of the Bor Regional Development Project. Development of constraint programming was followed by the development of appropriate tools. B-Prolog was used in this paper. Many systems, including B-Prolog, enable interface with classic object-oriented languages, such as

C++ or Java. One of the greatest advantages is the possibility of simple modelling, even for beginners in planning and implementation of the project.

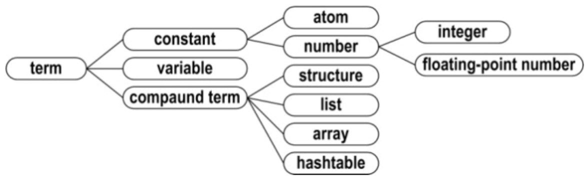

Figure 4 B-Prolog basic types
Source: Authors

Term is a basic data structure in Prolog, i.e., everything including program and data is expressed in the form of term. There are four basic types of terms in Prolog: variables, compound terms, atoms and numbers (Zhou, 2002).

A separate system called DJ (Declarative Java) represents an interpreter for Java programming language implemented in B-Prolog. With the aid of these systems, the process of creation of Java applets is significantly easier (Strak & Krčevinac, 2004).

Latest versions of B-Prolog, with limited license, can be downloaded at the following website: http://www.probp.com/.

5. Implementation of the Bor Regional Development Project

Due to difficult situation in Bor and Majdanpek municipalities, Republic of Serbia took a loan from the World Bank in order to start the Bor Regional Development Project (World Bank, 2007). The objective of the Project is to support the efforts of the Government of Serbia to revive the depressed Bor region (municipalities of Bor and Majdanpek) through:

1. dealing with urgent environmental and social legacy issues arising from mining sector restructuring and
2. fostering new sources of economic growth and job creation in the region.

The Project is financed through a combination of IDA Credit and IBRD Loan. The Bor Regional Development Project has two components:

- Component A: Environmental Management and Remediation and
- Component B: Socio-economic Regeneration Component.

The primary goals of the Component A are as follows:

- strengthening the capacity to manage the historic environmental liabilities,
- strengthening the capacity to monitor the compliance of the operator of RTB Bor mining and smelting assets with the Serbian environmental norms and
- remediation of the most pressing environmental problems remaining within state responsibility.

The primary goals of the Component B are as follows:

- to support the improved labour redeployment programs and employment services aimed at RTB Bor's redundant workers and general population of the Bor region and
- to promote the development of private sector in the region.

Within the Component B, the Labour Redeployment and Employment Support sub-component supports a range of active labour market programs targeting both the workers laid off due to decline in mining activities and the general population in the region. The restructuring of RTB Bor raised the unemployment rate in the region. In 2002, the first stage of labour downsizing resulted in the redundancy of 3.457 employees, or 27% of RTB Bor's 12.687 employees at the time. Another 2.300 surplus workers further reduced employment level in late 2006.

Labour Redeployment and Employment Support sub-component finance the following activities:

- job search assistance programs, including vocational and social information and counselling, provision of the labour market information, job club programs, vacancy and job fairs, and labour exchange and placement services,
- off-the job education and training services, including:
  - ✓ short term vocational courses, short-term technical courses for more skilled employees,
  - ✓ short-term upgrading of current skills for those who believe they can be re-employed in the traditional industry and
  - ✓ training women's groups and other initiatives to empower women in the labour market.

- on-the-job training services combining classroom training with informal apprenticeship programs at the employer's workplace,
- temporary employment program (public works) to assist local communities with environmental cleanup, refurbishment of public infrastructure, and provision of assistance and support to social agencies (e.g., schools, retirement homes, clinics) and
- strengthening the institutional capacity of regional branches of the National Employment Service through provision of training, equipment, and technical advice.

Local Implementation Unit that works directly with the unemployed is responsible for project field implementation. Local offices within the project are formed in Bor and Majdanpek. The main goal was to identify the wishes of the unemployed in terms of trainings, public works and other programs within the project. A separate database of the unemployed was created, with basic information, but also their wishes to take part in specific programs.

Most people showed interest in the public works program, despite the large number of different programs. Public works were divided into two groups: care for the elderly and public areas reconstruction in rural regions. The prescribed budget enables implementation of around 30 public works projects, within the period of 3 to 6 months, with more than 700 employees.

Public works are mainly implemented in rural local communities, which are remote. A large number of the unemployed who were interested in public works came from urban areas and regardless of that they accepted to work in remote villages. From the aspect of organization of a public work, it is very significant to engage the unemployed from the very local community in concern. The unemployed had the possibility to express their wishes in terms of location where they want to be engaged, but the contractors were making the final decision regarding the employment.

Having this in mind, the Local Unit creates initial lists of the unemployed for engagement in public works that are proposed to the contractor. The wishes of the unemployed are treated as soft constraints, and the selection of the employer and legal issues represents hard constraints.

It can be said that the creation of list of unemployed is followed by the following constraints:

01. The same person cannot be engaged in two different public works.
02. All public works must be found in the implementation plan.
03. Large numbers of public works are simultaneously performed.
04. Every public work project engages a certain number of people (from 10 to 30).
05. Public work takes place every working day in a week from 07:00 to 15:00.
06. Employees have an opportunity to express their wishes regarding the specific public work.
07. The final list should be approved by the contractor.

B-Prolog/Java application was created for the purpose of the preparation of the list from the project database of the unemployed.

Many CLP systems enable adequate interface for the languages such as C++ or Java. This enables usage of basic CP techniques in these programming environments, and their implementation is limited with possibilities of the interface. The application that handles the data is developed in Java programming language. There are many reasons for using Java as an additional program for B-Prolog. The first is a possibility to communicate with the database where the data on subjects that need to be allocated is set. The second reason is a possibility to create applets that graphically represents the solution, which enables simple presentation of the results on the Internet. The whole system is comprised of B-Prolog and Java code.

There are several criteria based which it can be determined whether once schedule is better than other possible solutions, if there are any. It is decided that the criterion for this application should be the wishes of the unemployed. Based on this criterion, a system is developed that would fully meet all hard and as many soft constraints as possible.

In the first stage, the unemployed are registered and give their wishes to the administrator in terms of specific public work project. At the moment when the list should be created, the administrator starts the execution file. The system takes over the data on the unemployed from the database. After creating adequate objects in Java, the data are forwarded to B-Prolog application. A B-Prolog program creates initial lists that can be proposed to the contractor later. In order to prepare initial list with unemployed persons, two-dimensional space is created. Public works are represented on the horizontal axis and the positions for each worker on the vertical axis (Figure 5.).

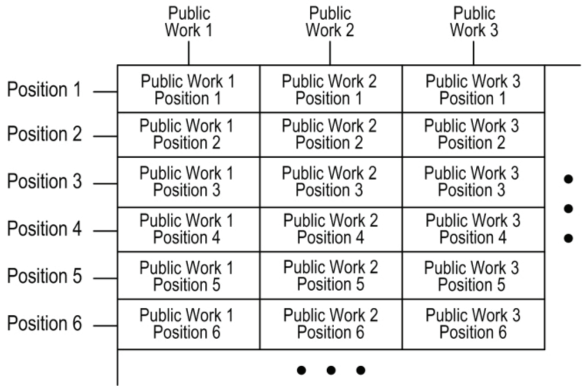

Figure 5 Public works represented in two dimensions
Source: Authors

The data from the project database about unemployed persons interested in the public works were presented in the following form:

unemployed(id, name, period, wishes, public_work, position)

where:

- id – id number,
- name – name of the person,
- period – period in the registry of the project,
- wishes – list of public works in which the unemployed person is interested,
- public_work – mark of the public work if unemployed person was already engaged,
- position – public work position.

In order to better understand the mechanism of the applied global constraints, the worker is presented as a rectangle marked with the name and id number. The example from Figure 6 represents Marko Marković, whose id number is 467. He expressed a wish to take part in a public work project in the local community "Šarbanovac" (marked with number 34). He was registered for more than two years and he was not engaged in any other program of the project (variables public_work and position are 0).

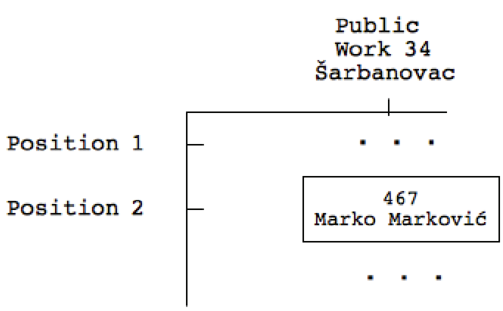

Figure 6 Example of the unempolyed
Source: Authors

In B-Prologthis is represented in the following way:

unemployed(497, 'Marko Markovic', 2, 34, 0, 0)

Period in the registry of the project is a soft constraint that needs to be met to as high extent as possible. One of the requirements is that FIFO principle is complied with as much as possible, i.e. the workers who first applied for the public works have the advantage. At the beginning of the work project, the constraints regarding the period in the registry of the Local Unit is created. People are ranked according to the period in the registry (Table 1).

Table 1 Priorities for soft constraints

Period	Rank
More than 2 years	7
From 1,75 to 2years	6
From1,5to 1,75 years	5
From1,25to 1,75 years	4
From 1 to 1,25years	3
From 0,75 to 1 year	2
From 0,5 to 0,75 years	1
Less than 0,5 year	0

Source: Authors

The wishes of the best-ranked unemployed are met first, as long as there is enough room in the particular public work. Example in the Figure 7 illustrates that the Petar Petrović cannot be engaged in the public work in Šarbanovac, which requires 20 persons, because the wishes of the better ranked people are met first.

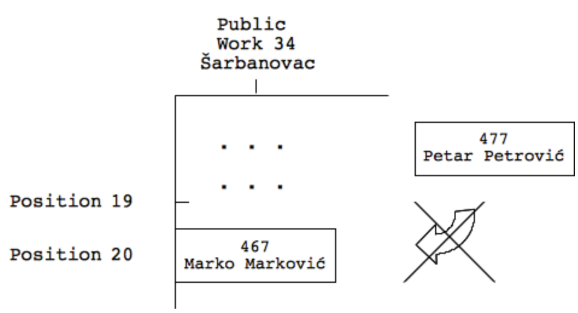

Figure 7 Example of the soft constraint that cannot be satisified
Source: Authors

The application meets the wishes of the unemployed as long as positions for certain public work are available. If the wish can be fulfilled, then public_work variable gets the value that refers to the mark of the public work and position marks the position at the public work (Figure 8.).

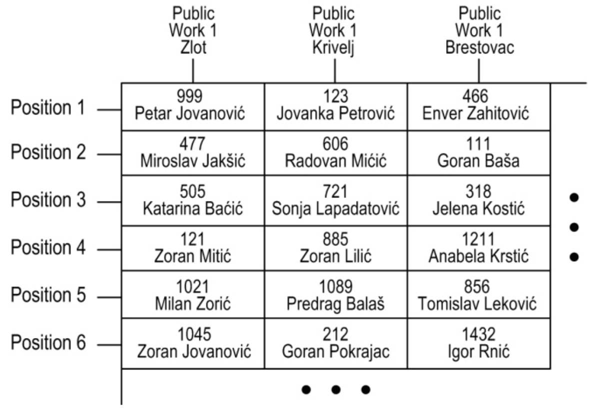

Figure 8 List of the unemployed creted by B-Prolog
Source: Authors

The diffn constraint was used to prevent two unemployed persons from being engaged in the same public work at the same position:

diffn([[X,A,1,1],[Y,B,1,1]])

where variable X and Y represent the public work, while A and B are positions in the public work for two unemployed.

The variables get the appropriate values through the Labelling process. After using the solver for the final domain, B-Prolog finds a solution and the obtained data can be returned to Java method, which could create an adequate Java applet.

6. Conclusion

One of the most important problems that need to be solved during the implementation of the real projects is resource allocation. In general, it is necessary to allocate resources to activities, but in such way that the resources do not overlap for more activities. Within the problem of staff allocation, activities have the established time of start and duration, but it is necessary to allocate adequate resources, according to the appropriate pattern that would meet some legal constraints and rules of the organization.

In order to solve these problems, numerous techniques and methods were used. Constraint programming represents an independent computer system based on constraints. It represents a formation of century-old observations, derived from analysis and deduction, into mathematic structure with a few modern additions. CP comes from artificial intelligence, more specifically, from the papers from this area that were published in the 1960s and the 1970s. In order to solve real problems of resource allocation, numerous techniques from the area of operation research were used. A significant progress was made after implementation of the method based on constraints. The advantage of using these methods, in comparison to the classic methods of operation research, is not only in the system performances. Besides that, constraint programming offers a large flexibility and transparency. Implementation of constraint programming is followed by the large number of built-in or global constraints that are implemented for different classes of the problem. This paper describes these constraints, such as cumulative and diffn, which are significant for solving the resource allocation problems.

The development of constraint programming was also followed by the development of appropriate tools. B-Prolog is a classic CLP system with all advantages and disadvantages of a Prolog language. Many CLP systems provide interfaces with classic object-oriented languages, such as C++ or Java. These languages do not have enough support for describing relations or constraints between the different entities defined by the programmer. For this reason, it is very significant for all these programming languages to have a possibility to implement CP programming method, because this provides them with necessary declarativity. A segment referring to communication with the person setting up a schedule is very significant, because that person does not have to know how to write programs. Making the program easier-to-use is one of the elements that need more attention in the future.

Solving staff allocation problems with CP method is discussed in the example of implementation of the public works program within Bor Regional Development Project.

References

Barták, R. (n.d.). *Constraint Programming – What is behind?* Retrieved January 15, 2011, from KGsePG: http://www.kgsepg.com/project-id/13421-constraint-programming-what-behind

Bistarelli, S., Codognet, P., Georget, Y., & Rossi, F. (1999). Abstracting Soft Constraints. *New Trends in Constraints*, 108-133.

Frühwirth, T., Herold, A., Küchenhoff, V., Le Provost, T., Lim, P., & Monfroy, E. (1993). *Constraint Logic Programming – An Informal Introduction. Technical Report ECRC-93-5.* München: European Computer-Industry Research Centre.

Goltz, H. J. (2000). On Methods of Constraint-Based Timetabling. *Proceedings of PACLP'2000* (pp. 167-177). Manchester: PACLP.

Goltz, H. J., & Matzke, D. (1999). University Timetabling Using Constraint Logic Programming. In G. Gupta (Eds.), *Practical Aspects of Declarative Languages, LNCS 1551* (pp. 320-334). Berlin: Springer-Verlag.

Goltz, H. J., Küchler, G., & Matzke, D. (1998). Constraint-Based Timetabling for Universities. *Proceedings INAP '98, 11th International Conference on Applications of Prolog* (pp. 75-80). Tokyo: INAP.

Gueret, C., Jussien, N., Boizumault, P., & Prins, C. (1995). Building University timetables using Constraint Logic Programming. *Proceedings of* *the First International Conference on the Practice and Theory of Automated Timetabling ICPTAT '95* (pp. 393-408). Edinburgh: Napier University.

Jovanović, P. (1999). *Upravljanje projektom – Project Management.* Beograd: Grafoslog.

Le Pape, C. (1994). Implementation of Resource Constraints in Ilog Schedule: A Library for the Development of Constraint-Based Scheduling Systems. *Intelligent Systems Engineering, 3* (2), 55-66.

Lever, J., Wallace, M., & Richards, B. (1995). Constraint logic programming for scheduling and planning. *British Telecom Technology Journal, 13* (1), 73-80.

Marriott, K., & Stuckey, P. J. (1998). *Programming with Constraints: An Introduction.* Cambridge: The MIT Press.

Rossi, F. (n.d.). *Constraint Logic Programming.* Retrieved January 15, 2011, from CiteSeerX: http://citeseer.ist.psu.edu/422446.html

Strak, M., & Krčevinac, S. (2004). Razvoj grafičkog korisničkog interfejsa uz pomoć CP metoda. *YuInfo 2004.* Beograd: Društvo za informacione sisteme i računarske mreže.

The Constraint Programming Working Group. (1996). *Constraint programming. ACM-MIT SDCR Workshop.* Cambridge: The Constraint Programming Working Group.

Whitty, S. J., & Maylor, H. (2009). And then came Complex Project Management. *International Journal of Project Management, 27* (3), 304-310.

World Bank. (2007). *Serbia - Bor Regional Development Project.* Retrieved January 15, 2011, from The World Bank Group: http://documents.worldbank.org/curated/en/2007/05/7680342/serbia-bor-regional-development-project

Zhou, N. F. (2002). *B-Prolog User's Manual – Version 6.2. Afany Software.* Retrieved January 15, 2011, from B-Prolog: http://www.probp.com/download/manual.pdf

Zhou, N. F. (n.d.). *Garbage Collection in B-Prolog.* Retrieved January 15, 2011, from CiteSeerX: http://130.203.133.150/viewdoc/summary?doi=10.1.1.32.1010

Miodrag Strak
University of Belgrade
Faculty of Organizational Sciences
Ul. Jove Ilića 154
11000 Belgrade
Serbia
Email: miodrag.strak@gmail.com

Slavka T. Nikolić
University of Novi Sad
Faculty of Technical Sciences
Trg Dositeja Obradovića 6
21000 Novi Sad
Serbia
Email: slavican11@sbb.rs