Team-building with Answer Set Programming in the Gioia-Tauro Seaport

G. Grasso[1], S. Iiritano[2], V. Lio[2], F. Ricca[1], and F. Scalise[3]

[1] Dipartimento di Matematica, Università della Calabria, 87030 Rende, Italy
{grasso,ricca}@mat.unical.it
[2] Exeura Srl, Via Pedro Alvares Cabrai - C.da Lecco 87036 Rende (CS), Italy
{salvatore.iiritano,vincenzino.lio}@exeura.com
[3] ICO BLG Logistics Automobile Italia SPA - Gioia Tauro, Italy

Abstract. The seaport of Gioia Tauro is the largest transshipment terminal of the Mediterranean Sea. A crucial management task for the companies operating in the seaport is team building: the problem of properly allocating the available personnel for serving the incoming boats. Teams have to be carefully arranged in order to meet several constraints such as: availability of employees and their skills, contract conditions (e.g., fair distribution of the working load, turnover of the heavy/dangerous roles) etc. This makes team building a hard and expensive task requiring several hours per day to be performed manually.
In this paper we present a system based on Answer Set Programming (ASP) for the automatic generation of the teams of employees in the seaport of Gioia Tauro. The system is currently exploited in the seaport by ICO BLG, a company specialized in automobile logistics.

1 Introduction

In this paper we report on a recent successful industrial application we have developed in the area of Knowledge Representation and Reasoning (KRR). We present a system based on Answer Set Programming (ASP) [1], which is currently exploited by the ICO BLG company in the international seaport of Gioia Tauro, that allows for the automatic generation of teams of employees. Roughly, the problem we deal with is computing a suitable allocation of the available personnel of the seaport such that cargo boats that every day shore the port are properly handled. To accomplish this task several constraints have to be satisfied. An appropriate number of employees, providing several different skills, is required depending on the size and the load of cargo boats. Moreover, the way an employee is selected and the role he/she will play in the team (each employee is able to cover several roles according to his/her skills) are subject to many conditions (e.g., fair distribution of the working load, turnover of the heavy/dangerous roles, etc.).

The system can either build new teams or automatically complete a partial allocation when the user manually selects some key employees. Also, the system is able to verify that a team satisfies all the requirements. It might be that case that no team exists that satisfies all requirements or a team violates some condition due to user selections. To this end, the system provides a suitable explanation of violations, allowing the user to take corrective actions. Further, the system can also be used by the managers of the company for performing simulations in order to estimate important financial aspects. For instance, by computing ahead the whole shift scheduling for the next month it is possible to evaluate how much overtime will be needed and allow/deny leave request.

In the literature there are a number of system and methods for solving various variants of the workforce management problem [2–6]. The available solutions can be mainly divided in two categories depending on whether (i) an ad-hoc algorithm or (ii) a generic method is employed. The first category has the drawback of producing solutions that are very specific which cannot be easily adapted for solving a different variants of the same problem; in practice they are employed for solving problems with unique features. Conversely, our approach belongs to the second category where, historically, both Operational Research and Artificial Intelligence methods (like e.g., Constraint Programming, Fuzzy logic, Genetic Algorithms, Local Search etc.) were employed.

In this paper, the domain knowledge is modeled by exploiting ASP and a team building system is implemented by using the ASP system DLV [7]. Desired allocations are computed by means of a set of suitably defined logic programs. The contribution of this paper is twofold: firstly, we describe how ASP can be exploited for solving a specific KRR problem (on a par with -and not in substitution of- other AI techniques); secondly we report on a successful industrially-developed ASP-based system which is currently employed in the important port of Gioia Tauro. The system, commercially-distributed by EXEURA s.r.l., a spin-off company of the University of Calabria, represents a practical demonstration that ASP is well-suited for developing real-world KRR systems. In particular, the pure declarative nature of the language allowed us for refining and tuning both problem specifications and encodings together while interacting with the stakeholders of the port. It is worth noting that the possibility of modifying (by editing text files) in a few minutes a complex reasoning task (e.g. by adding new constraints), and testing it "on-site" together with the customer was a great advantage of our approach, that might be easily applied in several contexts.

In the following, we assume the reader familiar with ASP syntax and semantics [1, 7]. The remainder of this paper is organized as follows. The Section 2 illustrates the scenario and the motivation of our application; Section 3 provides a simplified ASP encoding of the problem while Section 4 describes the whole system we have developed. Section 5 reports on the performances of the system on real-world data; Section 6 concludes the paper.

2 Scenario

In this section we illustrate the application scenario and the main motivations of this paper. The seaport of Gioia Tauro (http://www.portodigioiatauro.it) is the largest transshipment terminal of the Mediterranean Sea. Historically, container transshipments are the main activity of the port (related problems were subject of extensive research [8]); recently, Gioia Tauro has become also an automobile hub. Automobile logistics is carried out by the ICO B.L.G. Company (a subsidiary of the B.L.G. Logistics Group - http://www.blg.de). Every day several ships of different size shore the port; vehicles they transport are then handled, warehoused, processed if technically necessary and finally delivered to their final destination. A critical goal is to promptly serve arriving

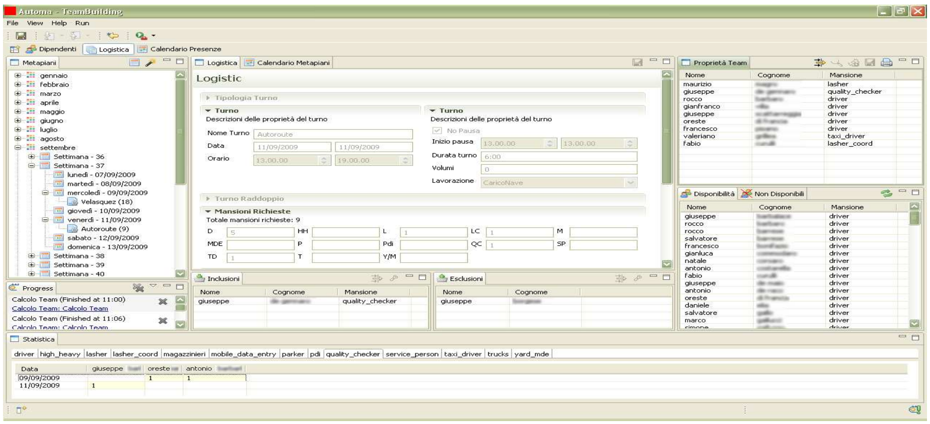

Fig. 1. The Team-builder Graphic User Interface

boats; that because permanence and processing time of any boat is regulated by contract, and any delay cause financial penalties to the company. Data regarding the shoring boats (arrival/departure date, number and kind of transported vehicles, etc.), is available at least one day in advance and suitable teams of employees have to be arranged and scheduled for the processing tasks.

Teams composition is subject to many conditions. Some constraints are imposed by the employees contract and in general are identified as legal conditions. For example, an employee cannot work more than 36 hours (plus max 12 of overtime) per week. Also, very important requirements are related to heavy/dangerous roles that must be turned over, while a fair distribution of the workload has to be guaranteed. Once the information regarding shoring boats is received, the management easily produces a *meta-plan* specifying the number of employees required for each skill; but a more difficult task is to assign the available employees to shifts and roles (recall that each employee is able to cover several roles according to his/her skills) in such a way that the above-mentioned requirements are always satisfied. The difficulty of allocating teams to incoming boats might cause delays and/or violations of the contract with shipping companies, with consequent pecuniary sanctions for B.L.G. Team building is definitively a crucial management task.

3 ASP-based team builder

In this section we provide a simplified version of the ASP program we have developed to realize the team builder system. This core program is reported below:

$$
\begin{array}{l}
(r)\; assign(Em, Sh, Sk) \vee nAssign(Em, Sh, Sk) \text{ :- } skill(Em, Sk), metaPlan(Sh, Sk, \_, D),\\
\qquad not\;\; absent(Em), not\;\; manuallyExcluded(Em),\\
\qquad workedHours(Em, Wh), Wh + D \leq 36.\\
(c_1) \text{ :- } metaPlan(Sh, Sk, EmpNum, \_), \#\texttt{count}\{Em : assign(Em, Sh, Sk)\} \neq EmpNum.\\
(c_2) \text{ :- } assign(Em, Sh, Sk1), assign(Em, Sh, Sk2), Sk1 \neq Sk2.\\
(c_3) \text{ :- } wstats(Em1, Sk, \_, LastTime1), wstats(Em2, Sk, \_, LastTime2),\\
\qquad LastTime1 > LastTime2, assign(Em1, Sh, Sk), not\;\; assign(Em2, Sh, Sk).
\end{array}
$$

$$(c_4) :- workedHours(Em1, Wh1), workedHours(Em2, Wh2), threshold(Tr),$$
$$Wh1 + Tr < Wh2, assign(Em1, Sh, Sk), not\ \ assign(Em2, Sh, Sk).$$
$$(r_{aux}) workedHours(Em, Wh) :- skill(Em, \_), Wh = \#count\{H, Em : wstats(Em, \_, H, \_)\}.$$

The inputs are: employees and their skills (predicate *skill(employee, skill*); a meta-plan specification (predicate *metaPlan(shift, skill, neededEmployees, duration)*); weekly statistics specifying for each employee both the number of worked hours per skill and the last allocation date (predicate *wstat(employee, skill, hours, lastTime)*); not present employees (predicate *absent(employee)*; and employees excluded by a management decision (predicate *manuallyExcluded(employee)*). Following the guess&check programming methodology [7], the disjunctive rule r generates the search space by guessing the assignment of a number of available employees to the shift in the appropriate roles. Absent or manually excluded employees, together with employees exceeding the maximum number of weekly working hours are automatically discarded. Then, admissible solutions are selected by means of constraints: c_1 discards assignments with an wrong number of employees in some skill; c_2 avoids that an employee is selected to cover two roles in the same shift; c_3 implement the turnover of roles; and c_4 guarantees a fair distribution of the workload. r_{aux} computes the total number of worked hours per employee. Note that, only the kernel part of the employed logic program is reported here (in a simplified form), and many other constraints were developed, tuned and tested.

4 The system

The team-building system integrates the ASP system DLV [7] and features a Graphical User Interface (GUI) developed in Java. In particular, the GUI is based on the Rich Client Platform (RCP) technology, whereas reasoning services and and data-storage features are implemented with OntoDLV [9], an ontology management and reasoning system based on DLV. The GUI combines in a single frame all the controls (see Figure 1). A tree-shaped calendar (displayed on the left) allows for browsing and scheduling working activities. Meta-plans specifications, usually identified by the name of the corresponding cargo boats (e.g. Velasquez, Autoroute), are the leafs of the tree, which can be added or removed by right-clicking on their name and selecting the proper command from a context-menu. Meta-plans information (ship arrival and departure date, available processing time and requested skills) is displayed in (and is modified by editing) the "Logistics" panel. Once a meta-plan is run, input information and personnel statistics are fed into the DLV system and the result is displayed on the top-right panel ("Team Properties"). The computed team can be also modified manually, and the system is able to verify if the manually-modified team still satisfies the constraints. In case of errors, causes are outlined and suggestion for fixing a problems proposed. Further, it is possible to enable/disable each constraint separately.

5 Some experiments on real-world data

To provide a concrete image on the behavior of the system, we report on a use case performed on real-world data. In particular, we run the system on archive data provided from ICO BLG for preparing: (r_1) a shift plan, (r_2) a daily shift plan, (r_3) a weekly shift plan, and (r_4) a monthly shift-plan. The problem specification was made of 130 employees to be arranged according to 36 meta-plans per weekwhere all the employee-allocation constraints were enabled. The system was installed in the ICO BLG workstation featuring an Intel Core 2 Duo CPU P8600 machine clocked at 2.40 GHz with 4 GB of RAM running Microsoft Windows XP and Java version 1.6.0_14. The runtime for solving the requests (r_1), (r_2), (r_3) and (r_4) was: 4.2 seconds, 25.3 seconds, 128.7 seconds, and 490.6 seconds, respectively. Basically, in a few seconds the system is able to generate a shift plan, and in some minutes the system can simulate a complete allocation covering an entire month, (this feature can be used for estimating both long-range and mid-range employees allocation needs). This is a fully satisfactory behavior.

6 Conclusion

In this paper we have presented a team building system based on Answer Set Programming. The system features a graphical user interface that gives full control on the status of the seaport-staff and allows for transparently running the DLV system for computing, checking and/or completing suitable teams of employees respecting the given constraints. The system has been developed side by side with the personnel of ICO BLG and is currently employed by the management staff of that company. The practical effectiveness of the proposed solution is confirmed by the on-the-field usage impressions reported by the ICO BLG management members: the system is able to obtain more reliable results when compared to manual team composition, and its usage reduced the time spent for team building of several hours each month.

References

1. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing **9** (1991) 365–385
2. Naveh, Y., Richter, Y., Altshuler, Y., Gresh, D.L., Connors, D.P.: Workforce optimization: Identification and assignment of professional workers using constraint programming. IBM Journal of Research and Development **51**(3/4) (2007) 263–280
3. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research **153**(1) (February 2004) 3–27
4. Tien, J.M., Kamiyama, A.: On manpower scheduling algorithms. SIAM Review **24**(3) (1982) 275–287
5. Lau, H.C.: On the complexity of manpower shift scheduling. Computers & Operations Research **23**(1) (1996) 93 – 102
6. Yang, R.: Solving a workforce management problem with constraint programming. In: In Proceedings of PACT'96. (1996) 373–387
7. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning. ACM Transactions on Computational Logic **7**(3) (July 2006) 499–562
8. Vacca, I., Bierlaire, M., Salani, M.: Optimization at container terminals: Status, trends and perspectives (2007)
9. Ricca, F., Gallucci, L., Schindlauer, R., Dell'Armi, T., Grasso, G., Leone, N.: OntoDLV: an ASP-based system for enterprise ontologies. Journal of Logic and Computation (2009)