A case study on graph-based planning for emergency evacuation

Santa Agreste[1], Pasquale De Meo[2], Massimo Marchi[3], Maria Francesca Milazzo[4], Salvatore Nunnari, Alessandro Provetti[1]

[1] DMI, University of Messina, Italy
[2] DICAM, University of Messina, Italy
[3] Network services, University of Milan, Italy
[4] DIECII, University of Messina, Italy

Abstract. We present a pilot study on the implementation of a software, based on declarative knowledge representation and logic-based automated planning, which assists the management of severe Chemical hazard events, e.g. fire or emissions that may require the evacuation of the area surrounding the affected Chemical plant. We model the geography, the road network and the population of the chemical plant and the surroundings by weighted, labeled graphs, which are updated as the hazardous situation develops. Intervening factors, e.g. the spread of toxic in the air, are represented in the graph in terms of their effects. Risk for the resident population and possible evacuation plans are evaluated and re-evaluated as the accident develops. Also evacuation plans are contingent and as conditions change may be re-evaluated from scratch; moreover, they may involve complex coordinated actions among the rescue units. Both the evaluation of the emergency scenario and the evacuation planning phases have been prototyped by means of an Answer Set Programming planner.

Keywords: Automated Planning, Applied Computational Logic, Chemical Plants Safety

1 Introduction

We present a pilot study on the implementation of the central component of a safety tool for risk evaluation and evacuation planning to be deployed in response to fire or emissions due to accidents at a large Chemical plant[5].

We have created an abstract model of the geography of the chemical plant and of its surroundings by a weighted, labeled graphs, which is updated as the hazardous situation unfolds. Evacuation from the surroundings of the plant is evaluated and planned wrt. to the graph representation. Intervening factors, e.g. the spread of toxic in the air, are represented in the graph in terms of their effects. Evacuation plans are contingent and as conditions change may be re-evaluated

[5] Due to legal reasons, at this stage we must omit the name of the plant and of the residential area for which the tool has been developed.

from scratch. Both the computation of the likely effects and the planning phase have been prototyped by an Answer Set Programming planner.

Answer Set Programming (ASP) [5] [9] (also called Stable Logic Programming (SLP) [10]), is a relatively recent but well-established style of logic programming: each solution to a problem is represented by an answer set (also called stable model), and not by answer substitutions produced in response to a query. A rich literature exists on applications of ASP in many areas, including problem solving, configuration, information integration, security analysis, agent systems, semantic web, and planning (see, among many, [2, 1, 6, 13, 4] and the references therein).

ASP is the language by which we represent all types of knowledge required to address this scenario: declarative knowledge about the surroundings, procedural knowledge about actions (escape actions, take cover actions and so on), contingencies, and planning as a domain-independent strategy to solve a motion problem. In this sense, our approach is in the same vein as the pioneer work of Zepeda and Sol [14] on evacuation as an instance of logic-based automated planning; the first complete (and delivered) instance of this approach in their *Plan Popocatpetl* project. We believe that our solution represents a marked improvement and generalization of their approach for the following main reason: we have adopted an intermediate formal representations with labeled graph for the representation of the scenario and the a priori evaluation of risk, for the declarative specification of the planning and observing part. These two intermediate representations make the modularization of the underlying ASP code possible and manageable thus enabling the higher degree of adaptivity that is required by the problem.

2 Evacuation plans

The drafting of evacuation plans in emergency requires finding a sequence of actions that lead from an initial state, representing a risk scenario, to a goal objective, representing a situation where the entire population is rescued (or generally safe). Unlike the planning scenarios that are traditionally considered in the Artificial Intelligence literature, evacuation plans specify administrative/security policies which can be hard to formulate[6]—even informally and often hard to execute even in small scenarios, i.e., those where the number of subjects, the spacial dimension and the time-scale are reduced.

Another important difference is the value to give to the 'do nothing' action. While in AI planning the so-called *nop* action is there mostly for padding fixed-length plans, in our applicative scenario they have a precise meaning which must be re-evaluated constantly: in case of chemical hazard, staying inside the building and limiting air circulation could be safest option available. Therefore we can say that automated planning with AI techniques, which is the subject of this

[6] See the norms regulating Save & rescue in Italy from `http://www.protezionecivile.gov.it/`

paper, is only one dimension of the inherent complexity of emergency evacuation management.

The other key element to the formalization of evacuation plans is the representation of the area, with a dynamic description of elements such as i) source and type of hazard ii) risk diffusion maps, which are specific to the type of risk, i.e., iii) number and localization of the population that needs to be evacuated iv) transport means and their level of mobilization v) Accident & Emergency (A & E) services with trained personnel and specialized equipment. For know risks, normally associated to Chemical/energy plants, the complexity of the task is essentially decreased by the availability of pre-compiled maps, which can specify the following two types.

First, during the emergency the danger areas extend (or contract) following the evolution of the accident, moreover such expansion/contraction is not easily characterized by simple circumferences around the site of the accident (consider, e.g., liquid chemicals in rivers, or fire under constant-direction winds). Normally, risk diffusion maps create a three-level partition of the areas in *i) impact,* i.e., areas close to the epicenter of the disaster, with high likelihood of lethality, *ii) damage,* normally external to the former, where lack of protection would cause irreversible damage to those who are contaminated, especially children and old people, and *iii) attention,* where damage is possible but not irreversible, in any case requiring medical treatment and possibly causing unrest in the population.

Second, so-called *safe areas* and their features. These areas are further detailed in i) waiting areas, ii) concentration areas for the rescuers and iii) recovery areas, which are safe places where the refugees will end up as a result of the evacuation.

3 Representation of the geography and of the escape scenarios

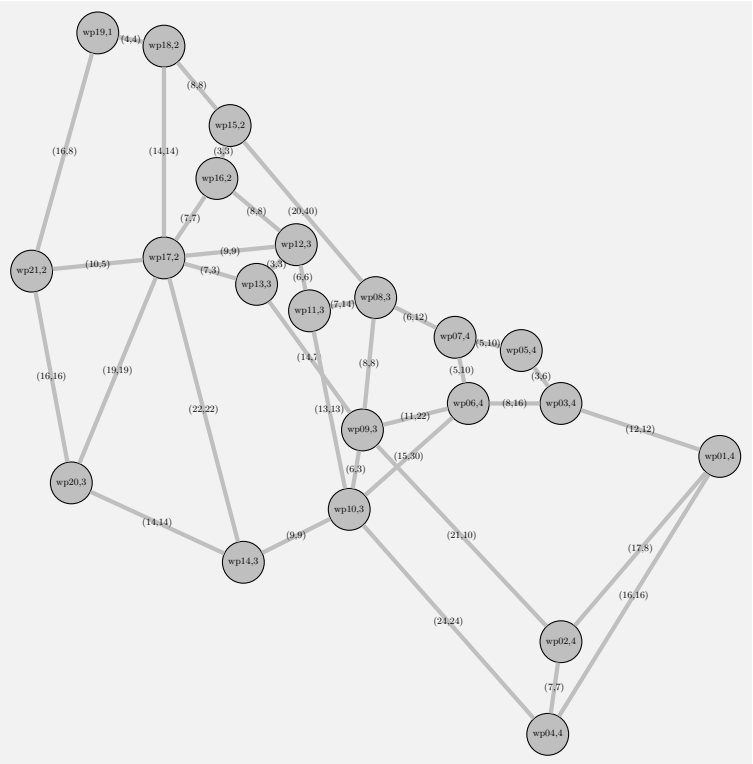
Two key aspects of the knowledge representation needed for this planning instance are the representation of the geography, namely roads and rivers, and of the level of risk assigned to areas by the domain experts. These information are synthesized by the risk graph, which is reported in Figure 1 for the first instance of problem we considered.

For comparison, we report in Figure 1 one of the annotated maps, in the standard format for Geographical Information Systems (GIS) that have been used to compile the graph in Figure 1. The twenty-one relevant area (called *waypoints)* identified by the domain experts (in this case, Fire patrol senior officers) are connected by 36 relevant routes.

As it can be noticed in Figure 1, domain experts have assigned each waypoints to one of 5 levels of risk for the population, according to the following standard risk scale.

- $RiskLevel = 1$: recovery area, destination for evacuation plans;
- $RiskLevel = 2$: low-risk area, close to recovery areas and far from the risk areas;

Fig. 1. The graph representing the geography and the risk levels of designed areas



- $RiskLevel = 3$: average-risk area, far from recovery areas but sufficiently close to the risk areas;
- $RiskLevel = 4$: high-risk area, close to high-risk areas, thus very far from recovery areas;
- $RiskLevel = 5$: high-risk area, the starting point for evacuation plans.

3.1 Description in ASP

This subsection describes the ASP predicate definitions that have been developed to capture the specific aspects of the evacuation planner. The following description does not cover the general part of the evacuation planner, which has been adopted as is from the ASP translation of the action description languages $\mathcal{L}_I$ and $\mathcal{L}_\infty$ developed in [3] and extensively described thereof. The only changes were made to embed the general rules into an answer-set program ready for interpretation by the ASP solver DLV [8, 7]; the syntax of the ASP program is thus specific to that accepted by the DLV grounder.

It should be added that some simplifying assumptions have been embodied directly in the ASP representation of the planning instance. These assumptions

Fig. 2. Annotated GIS information on the area subject to evacuation planning

are taken from the pre-compiled evacuation plan now in use, e.g., that for all evacuation actions there are some vehicles, typically buses, available to transport people to recovery areas. Another simplification is in the consideration of groups of evacuees, as opposed to single individuals. The structure of the graph in Figure 1 is embedded into the ASP program by means of the waypoint relation:

$$waypoint(Name, RiskLevel). \tag{1}$$

where variables *Name* and *RiskLevel* indicate the vertex of the graph and its assigned risk level. Communication routes, i.e., the edges of the graph are described by facts of this type:

$$route(Place1, Place2, Length, Criticality). \tag{2}$$

where variables *Place1* and *Place2* indicate the two areas that are connected, *Length* captures geographical distance and *Criticality* is a parameter representing the danger assigned to the usage of the given connection. Another important information is the representation of the evacuee groups:

$$group(Name). \tag{3}$$

where *Name* is assigned to thus-formed groups. The cardinality is not specified but as noted above we assume than one vehicle can evacuate a group. The last type of extensional predicate is for representing the position of the groups on the map, by this type of facts:

$$holds(position(Group, Place), 0). \tag{4}$$

Notice how relation $position(Group, Place)$ is reified into a fluent; variables *Group* and *Place* have their obvious meaning, whereas time-stamp 0 relates these facts to the initial state of the planning activity. Of course, we can have more than one group sitting on the same waiting area, as well as empty waiting areas. Finally, to describe actions where a certain areas become unreachable, i.e., a communication route has become nonviable (e.g., busy or disrupted or dangerous), we use these types of fact:

$$\begin{aligned} &waypoint\_blocked(WP1). \\ &route\_blocked(WP1, WP2). \end{aligned} \tag{5}$$

The predicate described above are to be added to the domain description and changed often, to adapt to the changing scenario, especially the (possible) disruption of roads, to be acquired, in the full version of this planner, from real-time GIS information.

4 Results and open issues

One of the most important problems to be solved in case of disasters is the draw and quick deployment of evacuation plans for the population. We describe a methodology based on knowledge representation and reasoning to formulate Evacuation Plans, using the intermediate graph representation and the DLV inferential engine.

Our evacuation planner considers the present situation, the type of danger, weather conditions, traffic or other modification of the zone to be evacuated, and formulates alternative evacuation plans to be face-validated on a case-by-case basis.

Studying the real case of the External Emergency Plan for a Refinery, we have implemented a planner able to generate appropriate evacuation plans, on the basis also of incomplete information derived from a possible Geographic Information System, supplying a representation of the scenario on the ground.

The results against a benchmark of 5 realistic emergency scenarios are encouraging: computation times remain within few minutes and the generated solutions were rated "excellent" by domain experts. From the point of view of computational logic, these results are entirely satisfactory, and, in our opinion, should become even more significant and widely applicable by the introduction of two further formal devices. The first device is the formal apparatus of ASP programs with weak constraints developed by Leone et al. and implemented in DLV. Even though there have been successful applications in literature, e.g. [12], at the moment, our tests indicate that weak constraints are too heavy computationally to be deployed in our platform, so we have decided to leave them out of the current implementation.

The second improvement would be a full model of context to be applied to data, i.e., to redesign the data as to capture their contextual aspects, and have the devices, e.g., local-cell emergency broadcasting, to selectively handle them, along the lines of the methodology defined Rauseo et al. [11].

References

1. Anger, C., Schaub, T., Truszczyński, M.: ASPARAGUS – the Dagstuhl Initiative. ALP Newsletter 17(3) (2004), see http://asparagus.cs.uni-potsdam.de
2. Baral, C.: Knowledge representation, reasoning and declarative problem solving. Cambridge University Press (2003)
3. Baral, C., Gelfond, M., Provetti, A.: Representing actions: Laws, observations and hypotheses. Journal of Logic Programming 31(1-3), 201–243 (1997), http://dx.doi.org/10.1016/S0743-1066(96)00141-0
4. Gelfond, M.: Answer sets. In: Handbook of Knowledge Representation, chapter 7. Elsevier (2007)
5. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICLP/SLP. pp. 1070–1080 (1988)
6. Leone, N.: Logic programming and nonmonotonic reasoning: From theory to systems and applications. In: Baral, C., Brewka, G., Schlipf, J.S. (eds.) Logic Programming and Nonmonotonic Reasoning, 9th International Conference, LPNMR 2007. p. 1 (2007)
7. Leone, N., Faber, W.: The dlv project: A tour from theory and research to applications and market. In: de la Banda, M.G., Pontelli, E. (eds.) ICLP. Lecture Notes in Computer Science, vol. 5366, pp. 53–68. Springer (2008)
8. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The dlv system for knowledge representation and reasoning. ACM Trans. Comput. Logic 7(3), 499–562 (Jul 2006), http://doi.acm.org/10.1145/1149114.1149117
9. Lifschitz, V.: Answer set planning. In: Schreye, D.D. (ed.) Logic Programming: The 1999 International Conference, Las Cruces, New Mexico, USA, November 29 - December 4, 1999. pp. 23–37. MIT Press (1999)
10. Marek, V.W., Truszczyński, M.: Stable logic programming - an alternative logic programming paradigm, pp. 375–398. Springer (1999)
11. Rauseo, A., Martinenghi, D., Tanca, L.: Context through answer set programming. In: Fletcher, G.H.L., Staworko, S. (eds.) Proceedings of the 4th International Workshop on Logic in Databases, Uppsala, Sweden, (EDBT/ICDT '10 joint conference), March 25, 2011, Proceedings. p. 58. ACM (2011), http://doi.acm.org/10.1145/1966357.1966369
12. Rauseo, A., Martinenghi, D., Tanca, L.: Contextual data tailoring using ASP. In: Schewe, K., Thalheim, B. (eds.) Semantics in Data and Knowledge Bases, 5th International Workshop, SDKB 2011, Zürich, Switzerland, July 3, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7693, pp. 99–117. Springer (2011), http://dx.doi.org/10.1007/978-3-642-36008-4_5
13. Truszczyński, M.: Logic programming for knowledge representation. In: Dahl, V., Niemelä, I. (eds.) Logic Programming, 23rd International Conference, ICLP 2007. pp. 76–88 (2007)
14. Zepeda, C., Sol, D.: Evacuation planning using answer set programming: An initial approach. Engineering Letters 15(2), 240–249 (2007)