Towards a Framework for Adaptive Constraint Propagation

Broderick Crawford[1,2,3], Ricardo Soto[1,4,5], Franklin Johnson[1,6(✉)], Eric Monfroy[7], Enrique Norero[8], and Eduardo Olguín[3]

[1] Pontificia Universidad Católica de Valparaíso, Valparaiso, Chile
{broderick.crawford,ricardo.soto}@ucv.cl
[2] Universidad Central de Chile, Santiago, Chile
[3] Universidad San Sebastián, Santiago, Chile
eduardo.olguin@uss.cl
[4] Universidad Autónoma de Chile, Santiago, Chile
[5] Universidad Científica del Sur, Lima, Perú
[6] Universidad de Playa Ancha, Valparaiso, Chile
franklin.johnson@upla.cl
[7] CNRS, LINA, University of Nantes, Nantes, France
[8] Facultad de Ingeniería, Universidad Santo Tomás, Viña El Mar, Chile
enorero@santotomas.cl

Abstract. In this paper we address a recent situation created by the explosive growth of web systems. For these reason we propose a framework to support adaptive elements in Web pages. Web pages can be accessed by different platforms with different browsers and through different devices such as laptops, tablets or cellphones. In particular we focus on adaptive menus for this different kind of devices or browsers to optimize the selection patterns and their implementations. We propose a framework using an Adaptive Constraint Programming technique to optimize the decision of developers. Constraint Programming is a programming paradigm able to find efficient resolution in optimization problems. In Constraint Programming a problem is defined in term of variables and constraints. The variables hold a domain and represent the unknowns of the problem, while the relations among them are modeled as constraints.

Keywords: Autonomous search · Constraint programming · Web system

1 Introduction

In this paper we address a recent situation created by the explosive growth of web systems, which can be accessed from different devices. For these reasons we have developed a framework to support adaptive elements in Web pages. Web pages can be accessed by different platforms with different browsers and through different devices such as laptops, tablets or cellphones. In particular we focus on adaptive

© Springer International Publishing Switzerland 2015
C. Stephanidis (Ed.): HCII 2015 Posters, Part I, CCIS 528, pp. 578–581, 2015.
DOI: 10.1007/978-3-319-21380-4_98

menus. The idea is to create a mechanism able to adapt the menus according different devices or browsers to optimize the selection patterns and their implementations. Developers are able to define rules for menu adaptation according to the features of the device and browser in use. We propose a framework using an Adaptive Constraint Programming technique to optimize the decision of developers. Constraint Programming (CP) is a programming paradigm able to find efficient resolution in optimization problems. In Constraint Programming a problem is defined in term of variables and constraints. The variables hold a domain and represent the unknowns of the problem, while the relations among them are modeled as constraints. We formulate an adaptive system for constraint propagation which aims at reducing domains by eliminating those values that do not lead to any solution. This is carried out by a enforcing a local consistency to the problem. The idea is to create a system capable of selecting the best propagator from a portfolio of propagators to solve our specific problem. This paper briefly describes the selected adaptation patterns and their implementation.

2 Constraint Programming

The pioneering works on constraints were motivated mainly by problems arising in the field of picture processing [4]. Artificial Intelligent research, concentrated on difficult combinatorial problems, it has contributed to considerable progress in constraint-based reasoning. Many powerful algorithms were designed that became a basis of current constraint satisfaction algorithms [1].

In computer science, constraint programming is a programming paradigm where in relations between variables are stated in the form of constraints. Constraints differ from the common primitives of imperative programming languages in that they do not specify a step or sequence of steps to execute, but rather the properties of a solution to be found [5]. This makes constraint programming a form of declarative programming. The constraints used in constraint programming are of various kinds: those used in constraint satisfaction problems [8], those solved by the simplex algorithm, and others strategies. Constraints are usually embedded within a programming language or provided via separate software libraries [2,7]. Constraint programming can be expressed in the form of constraint logic programming [3], which embeds constraints into a logic program. Whatever the language in which it is implemented, the problems should be modeled as a constraint satisfaction problem (SCP).

2.1 Constraint Satisfaction Problems

A CSP is a formal representation of unknowns variables , and the relations among them is defined as constraints. Formally, a CSP $\mathcal{P}$ is defined by a triple $P = \langle \mathcal{X}, D, C \rangle$ where:

- $\mathcal{X}$ is a set of variables $\mathcal{X} = (x_1, x_2, \ldots, x_n)$.
- $\mathcal{D}$ is a set of corresponding domains $\mathcal{D} = (d_1, d_2, \ldots, d_n)$ such that $x_i \in d_i$, and d_i is a set of possibles values for x_i.

– $\mathcal{C}$ is a set of constraints $\mathcal{C} = (c_1, c_2, \ldots, c_m)$, and a constraint c_j is defined as a subset of the Cartesian product of domains $d_{j_1} \times \cdots \times d_{j_{n_j}}$, for $j = \{1, \ldots, m\}$.

Then a solution to a CSP is an assignment $x_i \rightarrow a_i$ such that $a_i \in d_i$ for $i = \{1, \ldots, n\}$ and any constraint c_j is violated, for $j = \{1, \ldots, m\}$.

3 Framework for Adaptive Web Systems

The diversity of devices available has changed the way we access to web systems. At the same time, it has created a number of challenges for application developers. It has become more and more important to offer solid user experiences to an increasing number of contexts. The developers can create different copies of the same system to work with the different devices or we can create a mechanism to adapt a web system to different devices on real time. To adapt the characteristics of the browsers or devices exists a single code that is able to provide users with increasingly rich user interfaces, based on the characteristics of the platform, but also change the structure of the user interface to adapt to different form factors, in response to different platforms.

The first step is detecting the context of execution. This can be done, either on the server, or directly in the browser. Server side detection explores the user agent string sent by the browser in the HTTP requests. This supports obtaining detailed information about the browser and the device being used.

Client side detection is performed directly in the browser, and a number of JavaScript libraries are available to support it. With these libraries we can get features, such as support for HTML5 or CSS3 in the browser, touch or georeferencing capabilities in the device [6].

We propose a framework using an Adaptive Constraint Programming technique to optimize the pattern selection for an adaptive menu in a web system.

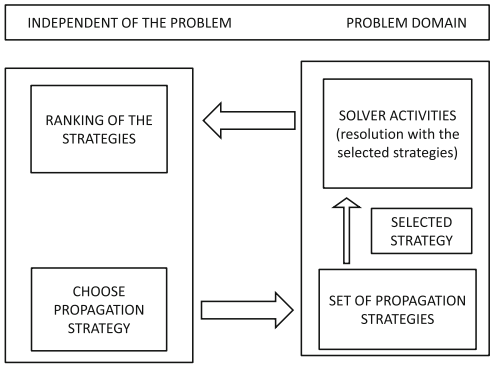


Fig. 1. Adaptive constraint propagation.

The first step is to model de pattern selection as a CSP problem, for this we propose define the variables of the problem as X=set of characteristics, the domain of the variables determines the possible patterns suitable for each characteristic. The constraints determines de possible combination of characteristics to adequate the system to the different browsers or devices.

The proposed framework implements an autonomous technique to allow the CP solver to adapt its own parameters, to selects the best propagation strategies. That is presented in Fig. 1, where the framework uses a strategy independent of the problem domain, that operates at a higher level of abstraction than the solver. It decides which strategy is applied next during the search. This decision is guided by the ranking generator which evaluates the performance of strategies. The ranking generator can be implemented using a genetic algorithm.

4 Conclusion

In conclusion we have proposed a framework able to select the suitable pattern to support adaptive elements in Web pages. The proposed framework uses an Adaptive Constraint Propagation techniques. We propose an adecuate model to adpat the pattern selction to a CSP. Also we present the scheme to representa the Adaptive Constraint Propagation framework.

Acknowledgments. Broderick Crawford is supported by Grant CONICYT/FONDECYT/REGULAR/1140897, Ricardo Soto is supported by Grant CONICYT/FONDECYT/INICIACION/11130459, Franklin Johnson is supported by Postgraduate Grant PUCV 2014.

References

1. Apt, K.: Principles of Constraint Programming. Cambridge University Press, New York (2003)
2. de la Barra, C.L., Soto, R., Crawford, B., Allendes, C., Berendsen, H., Monfroy, E.: Modeling the portfolio selection problem with constraint programming. In: Stephanidis, C. (ed.) HCII 2013, Part I. CCIS, vol. 373, pp. 645–649. Springer, Heidelberg (2013)
3. Hentenryck, P.V.: Constraint Satisfaction in Logic programming. MIT Press, Cambridge (1989)
4. Montanari, U.: Networks of constraints: fundamental properties and applications to picture processing. Inf. Sci. **7**, 95–132 (1974)
5. Oliveira, C.: Review of essentials of constraint programming by T. Fruhwirth and S. Abdennadher. SIGACT News **35**(3), 17–20 (2004). Springer-verlag
6. Perkowitz, M., Etzioni, O.: Towards adaptive web sites: conceptual framework and case study. Artif. Intell. **118**(1–2), 245–275 (2000)
7. Soto, R., Crawford, B., Monfroy, E., Paredes, F.: Syntax extensions for a constrained-object language via dynamic parser cooperation. Stud. Inform. Control **41**(1), 41–48 (2012)
8. Soto, R., Kjellerstrand, H., Duran, O., Crawford, B., Monfroy, E., Paredes, F.: Cell formation in group technology using constraint programming and boolean satisfiability. Expert Syst. Appl. **39**(13), 11423–11427 (2012)