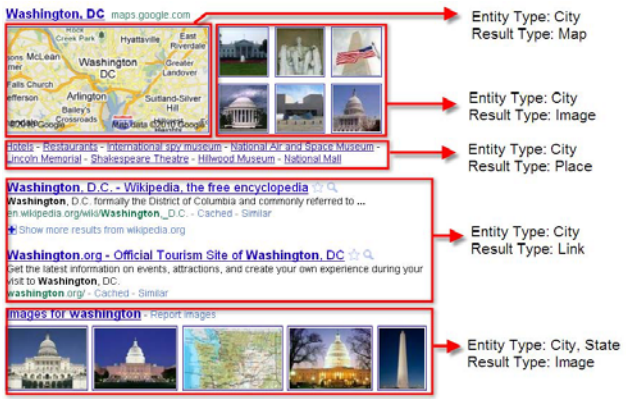
A Constraint Programming Approach to Automatic Layout Definition for Search Results

Politecnico di Milano
Department of Electronics and Information (DEI)
Piazza L. Da Vinci 32,
I-20133 Milan, Italy
{alessandro.bozzon,marco.brambilla,sara.comai}@polimi.it

Abstract. In this paper we describe a general framework based on constraint programming techniques to address the automatic layout definition problem for Web search result pages, considering heterogeneous result items types (e.g., web links, images, videos, maps, etc.). Starting from the entity type(s) specified in the search query and the result types deemed more relevant for the given entity type, we define an optimization problem and a set of constraints that grant the optimal positioning of results in the page, modeled as a grid with assigned weights depending on the visibility.

1 Introduction

Search engines represent one of the most important classes of applications that support the user information seeking process [5]. As the amount and the different kinds of information grow, also user requirements change, and search engines need to adapt accordingly. For instance, recently Web searchers started looking directly for objects of interest instead of Web pages that describe such objects. Search engines adapted to this new demand by improving the management of *named entities* specified within the queries and by providing well organized result pages that comprise different result types (e.g., maps, news, images, etc.) based on the type of entity involved in the query. The different result types may be produced by different search engines (image search engines, blog search engines, and so on), but then the results need to be aggregated in an optimal unified layout. For instance, a query involving an actor will produce a very different set of result types with respect to a query involving a city: the former case will feature videos, pictures, and blogs, while the latter will include maps, touristic reviews, and local news (see Figure 1 for an example of result page returned for the query "Washington"). Several problems arise in this scenario: how to map named entities to the most significant result types, how to select the best mix of items to show, and how to define the best layout of the result page. In this paper we propose a general framework that exploits constraint programming techniques to automatically compose a *search engine result page* (SERP): result items are positioned in a fixed page grid, based on a score function to be maximized and on



Hotels - Restaurants - International spy museum - National Air and Space Museum - Lincoln Memorial - Shakespeare Theatre - Hillwood Museum - National Mall

Washington, D.C. - Wikipedia, the free encyclopedia
Washington, D.C. formally the District of Columbia and commonly referred to ...
en.wikipedia.org/wiki/Washington,_D.C. - Cached - Similar
Show more results from wikipedia.org

Washington.org - Official Tourism Site of Washington, DC
Get the latest information on events, attractions, and create your own experience during your visit to Washington, DC.
washington.org/ - Cached - Similar

Fig. 1. Excerpt of the result presentation page for the query "Washington"

a set of structural constraints imposed by the typical search engine behaviour. We provide an overview of the behaviour of our system (Sect. 2) and the description of the automatic page layout composition approach (Sect. 3).

2 Search Engine Model

The behaviour of our system basically consists of the following phases [2]: 1) the **Query submission** phase, in which the user submits the query typically through a search form, by specifying some keywords, also including named entities; 2) the **Entity recognition** phase, in which the system first identifies the types of entity provided by the user in the query; 3) the **Result type selection** phase, in which the system defines which result types should be considered in the results; 4) the **Search Services Invocation**, in which the appropriate search services are invoked for each of the identified result types; and finally 5) the **SERP composition**, in which the results are positioned in the page. In this paper we focus on the latter phase.

3 SERP Model

We defined a conceptual model of the objects relevant for the problem, including: 1) the **page model**, defined as a grid of cells with an assigned level of importance as shown in Figure 2.a (in particular, we assigned the importance according to the "Golden Triangle" [4] pattern, that defines the most valuable area of the page as the top-left one – the importance levels are represented by shades of gray); 2) the **result type** , defined as the media type that is retrieved by a search engine and referenced in the page (e.g., image, web link, map, ...); 3) a (typed) **result block** (Figure 2.b) defined as a set of homogeneous results of a given result type, boxed within a shape of given size in terms of number of cells that it covers (e.g., a map occupies an area of 3x2 cells, while each web

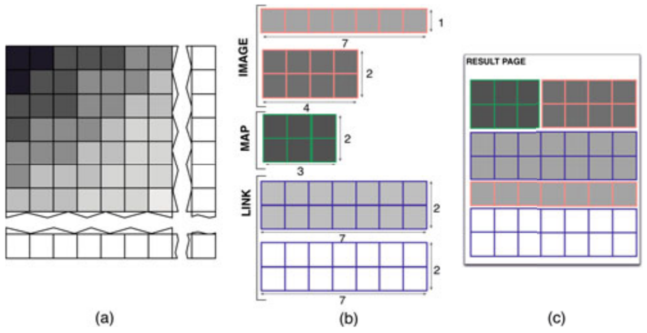


Fig. 2. Example of (a) page grid with importance levels, (b) some weighted result blocks, and (c) a possible block allocation

link occupies an area of 7x1 and blocks aggregating two items are considered). Result types are assigned with a score that depends on the entity type (e.g., for cities, maps have a higher score than pictures, while for actors maps have a very low score and pictures are extremely relevant). Result blocks are assigned a score that depends both on the result type they represent and on the result instances that are actually retrieved by the search engine for the specific query. This is exemplified in Figure 2.b, where blocks have been colored according to the corresponding result types and instances (for example, in Figure 2.b we have two image blocks, with different scores).

4 SERP Layout Composition

Given the SERP model described above, the SERP layout composition consists in an optimization problem with constraints induced by some structural rules in the positioning of blocks, specified to restrict the space that a block can occupy (e.g., the image block can be place on the top, in the middle, or at the end of the page; sponsored links must occupy a fixed position, related searches too, and so on). Examples of constraints include:

```
Constraint C1 = and(eq(SLx, 8), eq(SLy, 0)); //Sponsor link block pos. fixed at (x=8, y=0)
Constraint C2 = and(geq(L1x, 0), leq(L1y, 4)); //L1 block starts in one of the first 5 rows
Constraint C3 = distanceEQ(Ay, By, 1); //Blocks A, B must be positioned one after the other
```

The constraint satisfaction problem (CSP) describing the structural constraints for page positioning is modeled and then solved through a standard CSP solver [1]. The best layout is selected by maximizing the sum of the products between the block scores and the weight of the occupied cells:

$$max \sum_i (score_{ResultBlock(i)} * \sum_j weight_{OccupiedCells(j)})$$

This function is actually modeled by means of some heuristic constraints that position blocks with high scores in cells having high weights, so as to avoid generating layouts that are for sure suboptimal. This reduces the computation cost of determining the optimal layout among the possible ones, without increasing significantly the computation cost of the constraint solver. Once the optimal layout is produced by the constraint solver (i.e., all the coordinates of the result blocks are fixed), a concrete html page can be produced, where positioning is defined by means of appropriate CSS rules (see the example in Figure 2.c).

5 Evaluation

A preliminary analysis has been performed to evaluate to two aspects: comparison with existing solutions and added value perceived by the users. The *comparison* consists in comparing the page layouts of our approach with the ones of the major search engines (Google, Yahoo! and Bing). We assessed that our approach can deal with all the layouts produced by the various search engines without any conceptual change; the only change is in the set of constraint rules: on average, we needed to add or change 3 rules to align to Google, 6 rules for Yahoo!, and 5 rules for Bing. The *perceived value analysis* consisted in a user test aiming at comparing different constraint rule configurations. The evaluation involved 16 users, who were asked to provide their preferences among 4 versions of the result pages, for all the entity types. The analysis concluded that there is an high perceived value associated with the optimization algorithm, while different version of the constraint rules were perceived basically the same: the 3 optimized versions got around 30% of the score each, while the other got 10%.

6 Conclusions and Future Work

We discussed a framework for the automatic layout of search results, which is a more and more critical issue in industrial search engines despite being overlooked in the research community. We proposed a constraint based solution to the optimization problem of positioning the search items in the page at the purpose of maximizing the result page quality perceived by the user. As future work we plan to tackle queries involving several entity types at the same time [3] and to add additional variables to the problem, such as information coming from runtime statistics on contents, personalization and contextualization of the search task.

References

1. Choco library, `http://choco.emn.fr/`
2. Bozzon, A., Brambilla, M., Comai, S.: A Characterization of the Layout Definition Problem for Web Search Results. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 150–159. Springer, Heidelberg (2010)
3. Ceri, S., Brambilla, M. (eds.): Search Computing - Challenges and Directions. LNCS, vol. 5950, pp. 3–10. Springer, Heidelberg (2010)
4. Hearst, M.A.: Search User Interfaces, 1st edn. Cambridge University Press, Cambridge (2009)
5. Kuhlthau, C.C.: Inside the search process: Information seeking from the user's perspective. Journal of the American Society for Information Science 42(5)(5), 361–371 (1991)