Constraint-Based Matchmaking: A Personal (Interim) Perspective

Richard J. Wallace

Cork Constraint Computation Centre
University College Cork, Cork, Ireland
r.wallace.4c.ucc.ie

Abstract. The purpose of this paper is to step back and take a look at what needs to be done to build a viable constraint-based matchmaker. The original idea was that matchmaking agents are conceived as constraint-based solvers that interact with customers to select products in accordance with customer needs, which may often only become clear in the course of interacting with such a system. Constraint-based matchmakers provide potential solutions ("suggestions") to the problem as they understand it, but are able to correct their representations using feedback from the customer. It is argued that to realize this conception in practice, a system will need a product modeling capacity to guide constraint acquisition, as well as the capacity to represent product functionality in a way that can incorporate (potential) customer benefits.

1 Introduction

A few years ago, an intriguing proposal was made concerning an agent-based system for assisting customers in product selection [Fre95] [FW98] [FW02]. It was offered as an alternative to the "Deep Interview" approach, where customers answer queries about product features, basically by selecting values of these attributes, which can then guide product selection. In the proposed form of dialogue, the primary mode of interaction is a suggestion (rather than the query) made by the matchmaker to the customer. This suggestion is made in the form of a product. The customer replies with a set of corrections, to indicate how the suggested product fails to meet the customer's needs.

This was connected to a tandem proposal to implement this form of dialog as a kind of CSP. Both the Matchmaker's knowledge base and customer's needs were to be modeled as networks of constraints. Within this framework, a suggestion corresponds to a solution of a CSP, while a correction specifies customer constraints that the proposed solution violates. Repeating the cycle of suggestion and correction allows the matchmaker to modify its own CSP to better match that of the customer, until a suggestion constitutes a satisfactory solution. The problem of both acquiring and solving a CSP has been termed the "constraint acquisition and satisfaction problem" (CASP) in [Fre95].

As stated in the paper describing this work [FW02],

> The objective here is to model a situation in which customers do not enter the interaction with a fully explicit description of their needs. They may be

> unfamiliar with what is available in the marketplace. They recognize their constraints during the interaction with the Matchmaker. They cannot list all their requirements up front, but they can recognize what they do not want when they see it. ... We believe this to be a common form of customer conduct:- picture yourself browsing through a store or a catalogue, or interacting with a salesclerk.

Since that time, at least two simple demonstration systems have been built. One was written at the Constraint Computation Center in New Hampshire by a summer student working at the center; the other was written at the University of Padua by a student of Francesca Rossi. Although this represents a promising beginning, 'in the flesh' these demos are not markedly distinct from the earlier Deep Interview websites. (One of these is described in detail in Section 3.)

In the opinion of the author, there were features implied in the original proposal that may be crucial to its realization, but which were not clearly understood intially and which were not incorporated into the implementations. Here, I will try to delineate these features in order to proceed to more compelling demonstrations of constraint-based matchmaking.

An important aspect has to do with how the product features are presented. What made the original idea compelling was that it involved giving the customer a palpable sense of a distinct product, that could then be evaluated for its benefits and drawbacks. It seemed to be a promising approach to the problem of allowing a customer to comprehend a basic product and its variants in order to make an intelligent selection. What existing demos demonstrate is that this quality of presentation does not come from using CSP concepts in and of themselves. I don't think this is necessarily a shortcoming of this form of representation, although it may be in its simplest form. Instead, I think it suggests that the concepts must be used in a more creative way in order to realize the original intuitive portrayal. The purpose of this paper is to make some tentative proposals regarding the means of doing this.

The problem just stated almost certainly devolves into several problems, which include recognizing the key components of a product, understanding how a product works, and flexible handling of product functionality. Functionality may pose special difficulties since it often involves more than physical features alone. (No product is an island.)

Still another issue is communicating tradeoffs effectively. This is important, because it seems to be an especially difficult aspect of evaluating competing products, and, therefore, a place where an automated system could be of great benefit. Here is where a CSP representation might be especially helpful, because tradeoffs can be naturally represented as constraints.

A related requirement is explanation, of product features and functions. Again, a CSP representation looks like an excellent choice for this, based on earlier work in this area [FLW01]. There it was shown that, at least under restricted conditions, a constraint-based model can produce good explanations because of its well-defined inference processes.

The issues just referred to pertain to product properties. But if the goal is to sell a product, a cogent description is not sufficient, as indicated by the following quotation from a contemporary textbook in selling ([Fut99], p. 112):

> Typically, when used alone in the sales presentation, features have little persuasive power since buyers are interested in specific benefits rather than features.
>
> When discussing a product's features *alone*, imagine the customer is thinking, "So what? So your product has this shape or quality; how does it perform and how will it benefit me?" That is why you have to discuss the product's advantages as they relate to the buyer's needs.

Finally, there is the customer's view of the system itself. One aspect here is the issue of trustworthyness or 'respect'; maybe another way of saying this is whether the user will take the system seriously. Another quote from the same book on selling is pertinent here (p. 112):

> How does the prospective customer know that your claims for a product are true? Imagine a prospect thinking, "Prove it!" Be prepared to substantiate any claims you make.

Another aspect is the interesting-ness of the system. The assumption here is that, if the user finds the system sufficiently intriguing as well as useful, this may be an added incentive to use it again.

These concerns have been listed here in the spirit of taking stock. I don't think it would be a good idea to tackle all of them at once, and it may be that some of them can be finessed. The point of including all of them here is that, if these concerns are really relevant, they should at least be kept in mind.

The next section presents a brief background on constraint acquisition and satisfaction. Section 3 describes and critiques an existing demo system. Sections 4-5 discuss in more detail the problems of representing product features and functionality. Sections 6-8 describe some tentative ideas regarding how a system could be realized, and in so doing elaborate further on the system conception. Section 9 makes some brief remarks about testing. Section 10 outlines a plan of action.

2 Background: Constraint-Based Matchmaking

Before proceeding, it is worth nailing down some of the basic ideas that are critical to the present matchmaking proposal. It is assumed that the reader is familiar with constraint satisfaction problems (CSPs). So this account is restricted to the concept of a constraint acquisition and satisfaction problem (CASP) and its application to matchmaking.

In a constraint acquisition and satisfaction problem (CASP) the constraint solver must acquire information about the constraints before it can solve the problem. The situation can be conceptualized by assuming some universe of constraints, i.e., all the constraints which can possibly be part of the CASP. A certain set of constraints within this universe forms the current problem, P. The CSP solver (here the Matchmaker) knows a subset of the constraints in P at the outset, call it K, but it must in fact solve problem P. It will, therefore, have to acquire knowledge about the remaining constraints in P before it can find a satisfactory solution (cf. Figure 1).

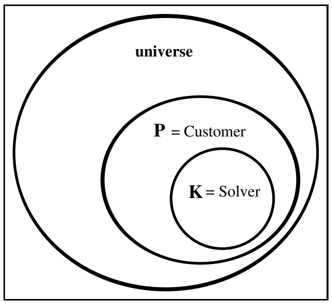

Fig. 1. The three classes of constraints that set the scene for a Matchmaking dialogue: those initially known to both solver and customer (K), those known (perhaps implicitly) to the customer (P), and the universal set of constraints, which includes all those that might have been part of the problem.

Matchmaking based on the CASP paradigm involves two agents, the CASP solver and the customer. In this situation, the customer 'knows' the problem to be solved, but not so explicitly that it can tell the solver outright. The solver elicits some of this customer knowledge by suggesting a solution based on the constraints that it (the solver) knows about. The customer then evaluates the solution to determine whether there are constraints of concern that are violated. These violations are communicated to the Solver, which incorporates this information as constraints between the variables involved in each violation. The solver then solves the new CSP and presents this solution as a new suggestion to the customer. This communication cycle is repeated until the solution is fully satisfactory to the customer, i.e., none of the latter's constraints are violated.

In the above account we have assumed that the universe of constraints is known to the solver. However, in practice it may be difficult to forsee all the variables that might be relevant to the user, and even if these are known, for any subset of variables there are numerous possible constraints. An approach to the latter problem is to allow information about a given constraint to accumulate by addition of nogoods based on customer evaluations. (For recent work on constraint acquisition see [OFO01].) Initially at least, the former problem will probably have to be evaded by restricting user responses via selections presented in the interface.

3 A Case Study: Digital Cameras

Digital cameras are a new, cutting-edge technology, that has aroused a great deal of consumer interest. A digital camera is also a rather complex product, as Figure 2 shows. So it would appear to be a good domain for constraint-based matchmaking.

Fig. 2. A typical digital camera (Sony CybershotF707 ®) .

A few years ago a simple demonstration system based on digital cameras was written by a student in our lab. This particular product was selected by the student because he was already a camera hobbyist and knew what information to include. The system was written in Java and could be fielded on the Web. Although this demo was not intended to be a full-fledged example of a constraint-based matchmaker system, the intention was to provide a basic suggestion-correction cycle in which a product was presented, and the user could then indicate which features should be changed and in what way.

The interface was very simple: a product name was given in one panel and the values of the features associated with this product were listed below. Each feature had a button beside it; clicking one of these buttons initiated a short interaction in a separate window, in which the user was allowed to choose another value for that feature. This, in turn, could alter the product displayed to match the current set of choices.

The representation of the products was, in fact, simply a database file (in html) of camera features. Each product was both represented and presented as a list of features and values, e.g. Brand: Sony, PixelWidth: 640, ZoomType: Digital, Flash: Off, On, Auto, Depth: 2.5, LCD Display: 2.5, etc.

The present writer knows very little about digital cameras, and has forgotten a lot of what he ever knew about cameras in general. As a result, when I tried to use the system, I found that I had no idea how to compare one phalanx of features and values against another. More distressing, perhaps, was that, in addition, I had no overall sense of how these features ‘fit together’ that would allow me to reach some composite judgement based on the entire set, or even some selected subset. As a consequence, the present demo would probably be of little use for this consumer in deciding which digital camera to buy.

Moreover, there are aspects to buying a digital camera that are critical for decision making but which do not involve simply examining product features. These quotes from a recent news article on digital cameras make this clear [LaG01]:

> But the leap into digital isn't always easy. The cameras can be difficult to use, you need a PC and other peripherals to take advantage of them, and the market is a confusing maze of megapixels and megabytes. Cole took the plunge only after finishing digital photography classes at a university: "It requires a computer, and I'm not very computer literate."
>
> Printing [a picture] is a whole new adventure: You've got to learn how to format a page size at, here's that bugaboo again, the correct resolution. Figure more hours if you want to learn how to manipulate the image on your PC, getting better colors or cropping out an ex-husband. Most cameras come with simple editing software ...
>
> Meanwhile, it's tough to find labs that can make prints from your memory card, a reusable device as small as a stick of gum that many models use to store photos. Online services, such as Shutterfly, and a few camera stores can do the job. ...

It is worth noting that I got a much better sense of what would be involved in buying a digital camera from this article than I did from the present matchmaker demo. More importantly, there are issues covered in the article that are critical for customer decision making, but are not even hinted at in the demo. I have to admit that, being more or less ignorant of how these cameras work, I wasn't even aware that a PC was needed until I read the article. Note also, that the description is largely in terms of costs and benefits (actually just costs - the flipside of benefits), and, therefore, connects with points made toward the end of the Introduction.

4 What's to be Done?

First let me note that, in my opinion, the idea of the "palpability" of a product representation, as adumbrated in the Introduction, is not merely perceptual; I suspect that it is to a much greater degree a phenomenon of the 'mind's eye'. Thus, while simply showing a picture of the product is a good idea, it will not be sufficient either to enable the user to comprehend the product or to make an intelligent decision about it. (In fact, I'm tempted to say that a picture of a product only gives one an illusion of palpability.)

What may be needed (in the mind of the customer) is some sort of schematic 'view' or naive understanding that is sufficiently coherent and accurate to incorporate the important features. The latter, therefore, need to be identified by the system.

In this connection, it is worth noting that there are now websites that include articles on how a product works in addition to offering products (see www.howstuffworks.com).

Assuming that this is a response to an identified or suspected want on the part of consumers, this seems to bolster the argument that some sort of schematic understanding is important in matchmaking.

Along with overall comprehension, it is important to be able to identify the values of the features that make one product different from another and to comprehend why they make it different in a potentially important way. This is related to what advertisers call "positioning" a product, although in the present situation the goal is customer comprehension rather than selling a particular item. Here, this entails highlighting the features that distinguish one product from another of the same type. It may also involve assessing the knowledgeability of the user.

It is relevant that, in the textbook referred to, there is a bridge between product features and customer benefits, which the author calls "product advantages" or "performance characteristics" [Fut99]. But for a system to be able to 'think' in these terms, it would seem to require some sort of model of how a product works.

5 What's to be Done? (Continued)

To return to the point made by the author quoted in the Introduction [Fut99]: selling a product often or always involves touting benefits for the customer. This means going beyond a coherent depiction of the product to connect it in some way with the user. So while the key word in the last section was comprehension, the key word in this section is relevance.

The distinction that this author makes between "advantages" and "benefits" would seem to entail a more extended idea of functionality than that required to describe the product itself. Specifically, it looks as if a network of relations must be extended until an effect or causal link is found that the user can 'fit (i.e. map) into his world', and which also carries positive affect. It is, therefore, a very different process from mapping features onto a preference scale and choosing on that basis. In fact, the discussion by this author suggests that the latter would be a rather unnatural procedure for most users.

On the other hand, when there are multi-dimensional objectives, scaling features may be one way to make the tradeoffs comprehensible, and, therefore, to allow the customer to arrive at a sensible, satisfactory decision. This suggests a role for "preferences" that is more specialized and technical than the way they are often depicted, but one which is nonetheless important.

Being able to describe functions that are also benefits raises thorny problems, in that this may involve a great deal of general knowledge. Consider the following whimsical example. Let's suppose our customer wants to use a digital camera as a doorstop. In this case, he presumably would like a heavier, more compact camera, so, for this customer, size and weight are the most important features. The important question for us is, what would it take for a matchmaking system to handle this situation well enough to be satisfactory to this customer?

Of course, this example may be irrelevant for our purposes. Nonetheless, if one looks at brochures for digital cameras, perhaps because this is a new and unfamiliar product, one finds that the sales pitch is often couched in a narrative that depict situations where someone might want to use a digital camera - that, while not unusual, would not come readily to mind.

6 Product Models

The most important idea in this context may be that of using a model (in whatever sense) as a basis for generating constraints. In other words, grounding the suggestions in a kind of theoretical framework. This may be a way of creating a constraint set (K in Figure 1) that can be extended in an effective manner.

It is also important to keep in mind the perspective that one is building elaborations of a set of product features, which is still the aspect that will lead to decision-making, i.e. choices of one product over others. The goal is to organize these features so that a user is better able to arrive at a decision and is more satisfifed with the result.

Possible approaches to this modeling problem may be found in the fields of configuration (for key components especially, but possibly also for functionality), in process models from the field of qualitative reasoning (especially for comprehending how a product works), and in description logics (for inter-product relations). In regard to modeling functionality, it is important to remember that all functions (causal links) occur via components and features of the product. As a result, an extended notion of configuration ports may be useful here [MF89].

Function of course involves processes, and here qualitative process models (qualitative physics) may be relevant. For instance, in the doorstop example, a concept of movement and one of blocking movement would be helpful, in addition to a representation of user wants, e.g. want($\neg$ movement), etc.

This example also brings up another important issue, namely, what is the proper scope of function of a single matchmaker agent? What I'm getting at here is that there is no reason to expect one agent to do everything required in a matchmaking session. This is a case where agent technology could be critical, if one can 'call up' experts to interpret a query or to make specialized inferences. At the same time, the system must be able to somehow integrate other agents' knowledge within the current session or dialog.

So how does all this relate to CSPs? In the first place, it's worth noting that constraint satisfaction already plays a major role both in qualitative physics and in configuration modeling. Perhaps more directly pertinent is the fact that here I am suggesting various ways in which product features should be nested, but processing is still critically dependent on these features, which are still expected to be represented as CSP variables and values.

In addition, the above suggests that, in addition to adding constraints to the problem based on user corrections, it may be possible to activate variables and constraints given pertinent information about user interests. In this case, the conditional CSP concept may be pertinent [MF90] [BB91].

7 User Models

Modeling functionality in relation to customer benefits would also seem to entail user modeling. Whether this necessarily involves a distinct ‘module’ or representation is not clear. What seems more pertinent is to identify user requirements or benefits that can be linked to product features. To be useful, user modeling must contribute to this end, making such hook-ups more likely and more efficient. An obvious idea is to generate separate graphs related, repectively, to product capacities and user concerns and then to seek to connect them.

There may be a place for some generic categorizing of user types. For example, [Fut99] distinguishes certain “buyer styles”, such as “thinker” and “intuitor”. However, it’s not clear to me that such categories are going to be related in any clearcut fashion to differences in desired benefits. On the other hand, somewhat analagous to the product models discussed in the last section, a generic user model may be useful, e.g. to represent basic notions like want or (dis)like.

8 Presentation Factors

In this connection, one thing that I would like to see is use made of the power and generality of the CSP representation to link what are normally thought of as presentation features with features that pertain to content, here the product. Some obvious examples are the size of the product and the size of a pictorial layout, the structure of the product (physical and functional) and the organization of the views on the screen (here I’m thinking of skeletal diagrams, hierarchies of views, etc), product functions and settings portrayed (a door perhaps?). Here is an area where it should be possible to develop a CSP approach that runs rings around other approaches to product presentation.

Another important issue is how to guide the user through a large number of products (such as digital cameras) as well as allowing the user to discover and comprehend tradeoffs. As already indicated, CSPs are the probably the way to go here, but it is a difficult problem. On the other hand, it appears to be a problem that is unsolved, and often unrecognized (cf. current sales websites, or product reviews in PC magazines). In other words, it seems to be largely virgin territory for research (as attested to by [FO01]).

At this point, I suspect that interesting-ness (mentioned in the Introduction) is something to be finessed, maybe by a few simple rules (however embodied) that provide variety of presentation. A more important issue is ease of use, but this is perhaps essentially a testing issue.

9 Testing

This is included as an eventuality, since it requires an extant working system. But, also prior to testing is the issue of defining the goal of the system, or better, a particular instantiation of it. There are two general goals that I can think of: informing the customer and making a sale. More specific aspects or dimensions that should be evaluated are comprehensibility, ease-of-use, and user satisfaction.

Another important part of testing is a componential analysis of the system, to determine which component or combination of components is contributing to the success of the system along any of the above dimensions.

10 What's to be Done? (Concluded)

The first problem to tackle is the development of a product model or product modeling system that generates, or is realized in some sense as a constraint network. At the same time, a preliminary implementation would be useful, in part because I have some specific ideas about the user interface that I would like to try out, related to the property of product "palpability".

After this a tentative scenario is:

- Stage two would extend the product model to handle functionality, which includes both product "advantages" and "benefits" in the sense used in [Fut99]. This may also involve looking more deeply into user modeling issues.
- Stage three would develop explanation facilities.
- Stage four would develop capacities that allow the product model and other internal information to guide the organization of the interface.
- Stage five would consider integrating multiple agents to extend the system's expertise.

Testing of the system could probably begin during stage two.

References

[BB91] J. Bowen and D. Bahler. Conditional existence of variables in generalized constraint networks. In *Proceedings, Ninth National Conference on Artificial Intelligence, AAAI-91*, pages 215–220, Menlo Park/Cambridge, 1991. AAAI Press/MIT.

[FLW01] E. C. Freuder, C. Likitvivatanavong, and R. J. Wallace. Deriving explanations and implications for constraint satisfaction problems. In T. Walsh, editor, *Principles and Practice of Constraint Programming - CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 585–589. Springer, Berlin, 2001.

[FO01] E. C. Freuder and B. O'Sullivan. Generating tradeoffs for interactive constraint-based configuration. In T. Walsh, editor, *Principles and Practice of Constraint Programming - CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 590–594. Springer, Berlin, 2001.

[Fre95] E. C. Freuder. Active learning for constraint satisfaction. In *Active Learning. AAAI-95 Fall Symposium Series, Working Notes*, pages 34–35, 1995.

[Fut99] C. M. Futrell. *Fundamentals of Selling*. Irwin McGraw-Hill, Boston, MA, 6th edition, 1999.

[FW98] E. C. Freuder and R. J. Wallace. Suggestion strategies for constraint-based matchmaker agents. In M. Maher and J.-F. Puget, editors, *Principles and Practice of Constraint Programming - CP'98*, volume 1520 of *Lecture Notes in Computer Science*, pages 192–204, Berlin, 1998. Springer.

[FW02] E. C. Freuder and R. J. Wallace. Suggestion strategies for constraint-based matchmaker agents. *International Journal on Artificial Intelligence Tools*, page (in press), 2002.

[LaG01] David LaGesse. Digital dynamite! a new crop of cameras is exploding the photographic world. *U.S. News and World Report*, (June 11):58–60, 2001.

[MF89] S. Mittal and F. Frayman. Towards a generic model of configuration tasks. In *Proceedings IJCAI-89*, pages 1395–1401, 1989.

[MF90] S. Mittal and B. Falkenhainer. Dynamic constraint satisfaction problems. In *Proceedings AAAI-90*, pages 25–32, 1990.

[OFO01] B. O'Sullivan, E. C. Freuder, and S. O'Connell. Interactive constraint acquisition. In *CP01 Workshop on User-Interaction in Constraint Satisfaction,UICS'01*, pages 73–81, 2001.