Identification of spatial template models DRAFT ONLY

Steven Woods*
Stephane Lapointe†
Marlene Gauvin‡

July 12, 1994

Abstract

This is an abstract. *Comment:* **This paper constitutes meling of the work of Woods (1993*a*) in describing hybridizations of search strategies based upon constraint satisfaction techniques that have been demonstrated to be effective in identifying spatial templates in complex, noisy situations with the representational work of Lapointe & Prouix (1994) which employs a possibilistic approach to representation of these spatial templates. Specifically the use of abstraction as a conceptual tool in both approaches is show to offer a point of contact between the two methodologies which suggests .....**

1 Spatial Template Recognition

Comment: **Marlene ... will make use of TIF references and real appli-**

*University of Waterloo, Logic Programming and Artifical Intelligence Group, sgwoods@logos.uwaterloo.ca

†Defence Research Establishment Valcartier, Combat Intelligence Automation Group, lapointe@jupiter.drev.dnd.ca

‡Defence Research Establishment Valcartier, Combat Intelligence Automation Group, gauvin@jupiter.drev.dnd.ca

cation discussion to place this paper in the context of the TIF project and as well describe the domain in a realistic fashion. To this end, it is assumed that spatial templates are described generically in this section.

2 Templates, constraint satisfaction, and search

Comment: – **STR w/ CSP : 1 approach to locating templ insts in confused situations**

Spatial templates consist of a set of slots, each an object in this particular domain. These slots have characteristic features that describe them, and participate in relationships with other slots in the set. These inter-object relationships, such as relative spatial position, may be represented quite naturally through the use of a constraint graph. In such a graph, each object is represented by a node in the graph, and inter-node arcs represent the inter-object constraints. Similarly, reflexive arcs can be seen to represent the particular characteristic values for each particular object or node. Inter-object constraints can include spatial position, or possible correspondences between other object characteristics. Reflexive slot constraints are object characteristics or features such as type, size, and orientation.

Comment: – **Templates are generalizations of some unknown set of specific instances. Situations are approximations in time, space and identity**

A template describes a particular concept. The intent is to locate *instances* of this concept in a particular *situation.* A situation is a representation of the state of the world. This world situation is composed of many situation elements, each of which may correspond to one or more template slots. Each situation element is fully or partially described in terms of its known attributes or features, and represents either a reference to information acquired about an element at a location at some time in the recent past, or a reference to a the location and attributes of an element believed exists based on inferences made from information acquired earlier. Thus, we are esstentially attempting to identify a template instance based on information that is approximate, dated, and in some cases, hypothetical. In some fashion this information needs to be "standardized" with respect to time since for the purposes of this paper we are assuming identification of template instances in which the templates have been defined such that the slots are assumed to possess relative characteristics at a single point or standard range of time : essentially a template represents the range of possible arrangements that could possibly be encountered if a snapshot were taken of the template slots at a particular time.

Comment: – **Solutions are complete or partial mappings from situation elements to template slots**

We wish to make use of our template or model and its slot features to locate sets of situation elements in the situation that map to the template, and the problem is to identify any mappings from situation elements to template slots such that all of the template slots have a corresponding situation element, and all or enough of the inter-slot and reflexive constraints are satisifed. An assignment of situation element to a template slot such that the reflexive constraints are satisifed is said to be a node-consistent assignment. Similarly, one may define arc consistency as two assignments where the inter-slot constraints are satisifed, or path consistency as the extrapolation of arc-consistency to N slots.

Problems specified in this fashion, where a solution is composed of a set of consistent element-to-slot assignments are well described in Kumar (1992), and are known as a Constraint Satisfaction Problems (CSPs). An example of a template constraint graph is shown in Figure 1 with five node slots where each slot is constrained by several attribute ranges, and 11 inter-slot spatial constraints on distance, position, and orientation.

We are interested in finding all possible instances of a particular template in a set of situation elements, and consequently we are interested in finding all solutions to a CSP modelling our template recognition problem. We have constructed a non-interactive system known as the MOdel Template Object Recognition System (MOTORS) (Woods 1993*a*) which is capable of solving CSPs in general, and spatially oriented CSPs such as template recognition in particular.

Comment: – **Hybridized Search is one possible engine with which to solve CSPs**

Initial naive slot domains are generated by assigning each slot a set of situation elements that *possibly* match the reflexive slot constraints or feature set. Consistency algorithms attempt to reduce the size of these domain sets where possible by application of template or domain constraints, and search strategies attempt to locate a satisfying set of assignments from slot to element

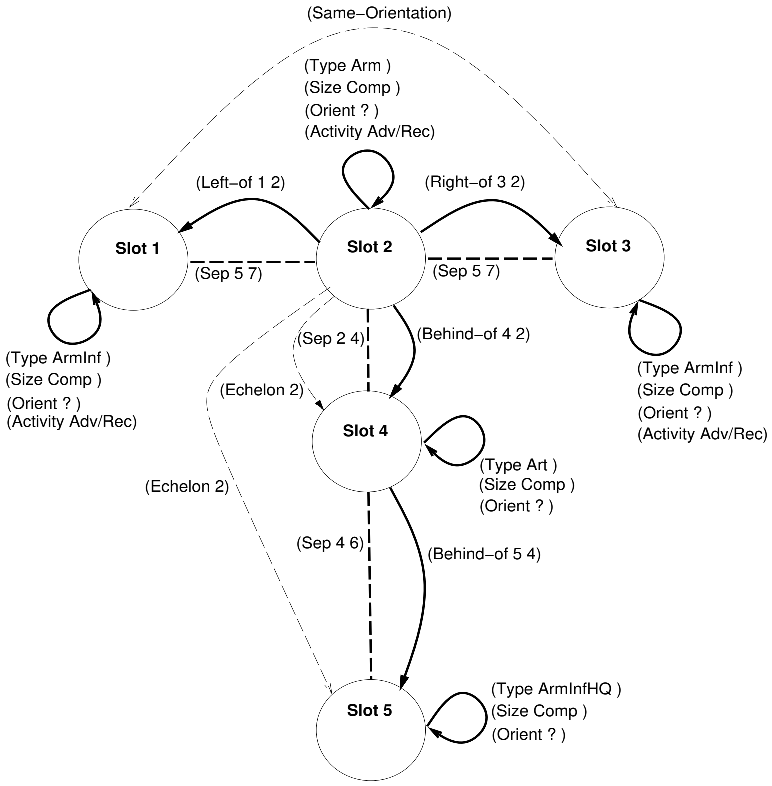

Figure 1: Whole CSP 5 variables, 11 arc constraints.

such that all constraints are satisfied. CSPs are typically solved either via some straightforward search strategy, search coupled with domain-dependent heuristics, or via application of a graph-reduction method which attempts to propagate consistency throughout the problem graph thus reducing the domain size of some graph slots prior to performing search. More recent work has focused on hybrid approachs which interleave search and propagation. Recent experimental results presented in Nadel (1989) show that hybrids which perform either some degree of constraint propagation *before* or *during* search tend to have better performance than either individual approach. It has been shown that search strategies with "look-ahead" properties can be thought of as augmenting search with some degree of during-search constraint propagation. By defining search strategies in this way the amount of constraint propagation adopted during search can be more easily identified and adapted, and various strategies more easily compared empirically.

Comment: – **We look at BackTracking, BT with Forward Checking and Dynamic Rearrang, BackMarking, BackJumping and assume complete Node Consistency before commencing search. Other test cases apply the AC3 arc consistency algorithm in advance of search and essentially reduce the problem to a trivial search problem, however, the amount of work used by AC-3 is greater than FC/DR and some other strategies in most problem instances ... suggesting that we don't know how much in-search propagation we SHOULD do, or even how much before-search reduction ... these are problem-dependent open questions.**

Although Nadel (1989) indicates which type of hybrid strategy is best for some small toy domains, published results solving interesting problems using this strategy have been difficult to find. Some recent work which has furthered this research with some novel hybrid algorithms including a melding of BackMarking and BackJumping is presented in Prosser (1993). The spatial template recognition problem provides an excellent example of a real-world problem which is of sufficient complexity to be interesting and yet still be susceptible to solution within a time period that would be potentially useful to experts in the domain.

Comment: – **So basically what did we do for experiments ?**

In Woods (1993*a*), experiments are detailed in which one satisfying instance of a particular spatial template is placed in a situation in which there are between 100 to 1000 additional situation elements or "noise". The template itself consists of 5 objects or slots each constrained for 4 object attributes, and 11 inter-object or arc constraints on spatial orientation, and relative location. Attempts were made to identify all[1] instances of this template in each of 10 different random instances at each "noise" level using several well-known search strategies: BackTracking (BT), BT with Forward Checking (BT/FC), BT/FC with Dynamic Rearrangement, BackJumping, and BackMarking. Additionally, each case was attempted a second time with the pre-search phase consisting of the AC3[2] arc-consistency algorithm in addition to node-consistency reduction.

Comment: – **And what did these experiments tell us ? BT was bad (no surprise) both with respect to median and variance ; BC/DR was very good with little variance between situations ; BJ was median-good with a sometimes high variance depending on situation ; BM was median-good also but with a VERY high variance ; AC3 was median-good also and with a reasonable variance - little search, but we also see that we have traded off during-search work for pre-search work.**

From the experiments and as shown in the partial results shown in Table 1[3] it was discovered that BackTracking performed poorly in all cases, and further had a high variance between different same-size problem instances. BackJumping improved over BackTracking as would be expected, however, also demonstrated a high variance over problem instances. BackMarking performed comparably to BackJumping, however exhibited a very high sensitivity to problem instance. The FC/DR strategy was clearly the superior choice, and had the lowest variance between problem instances of all approaches tested. The application of AC3 prior to search reduced the object domain sizes to the point that very little search was required following reduction, and resulted in a low variance across problem instances, although more constraint checks were required. The forward checking action of FC/DR is

[1]Since only one instance was created, one might expect identification of a single template, however, random element additions result in ghost or false-positive solutions. In no case tested were there more than 2 such solutions.

[2]AC3 is described in detail in Mackworth (1977).

[3]Note that the results shown were obtained on a Sparc10 processor with the application written in compiled Allegro Common Lisp. The *Constraints Applied* measure indicates how many constraint checks had to be performed during solution and does not include the preliminary node consistency checks as the template slot domains are generated, *Nodes Expanded:Generated* indicates the size of the search space, the Preprocess time indicates the amount of time spent performing consistency checking

Strategy and Case	Contraints Applied	Nodes Expand:Generate	Preprocess Time	Search Time	Total Time
FCDR Average	2269	36:37	0s	28s	28s
FCDR, Best	871	22:23	0s	8s	8s
FCDR, Worst	3403	52:54	0s	48s	48s
AC3, Average	3583	4:5	98s	0s	98s
AC3, Best	2752	4:5	80s	0s	80s
AC3, Worst	5294	4:5	133s	0s	133s
BJ, Average	4107	182:3472	0s	89s	89s
BJ, Best	780	42:684	0s	14s	14s
BJ, Worst	14558	700:12331	0s	444s	444
BM, Average	6223	736:21610	0s	172s	172s
BM, Best	780	42:684	0s	14s	14s
BM, Worst	16716	1049:44036	0s	600s	600s
BT, Average	10796	287:9104	0s	319s	319s
BT, Best	780	42:684	0s	12s	12s
BT, Worst	20064	408:16329	0s	600s	600s

Table 1: Experimental results over 10 situations with 1000 element noise.

a very controlled and systematic form of in-search constraint propagation in which a chosen assignment of situation element to a template object results in an immmediate propagation or reduction of the domain of the next-to-be-instantiated template object.

Consequently we would like to determine not only how much in-advance constraint propagation would result in the fewest constraint applications in this domain, but also how much in-search propagation should be utilized. We know that forward checking improved dramatically over non-forward checking search, but possibly a hybrid with further in-search checking would improve even more. This is an area in which more work is required.

Comment: – **Domain dependent heuristics offer one promising form of controlling application of domain knowledge directly to problem solution. In a real sense these are not heuristics but rather known methods in which real experts deal with the complexity of these problems and still manage to form solutions in reasonable lengths of time. Spatial co-**

hesion corresponds to several similar heuristics identified in discussion with intel analysts.

Another area in which great performance improvements may be made for this domain ins in the application of domain heuristics. One such heuristic that is effective in spatial problems can be formulated from the concept of spatial locality amongst situation elements that are members of a solution set for a template. Essentially, the identification of each element-to-slot assignment can be used to limit the scope of each of the objects remaining to be assigned a situation element, and thus dramatically reduce search.

Comment: – **STR as an interactive or collaborative task between analyst and assistant**

While what has been discussed to this point describes a "stand-alone" template recognition system, of greater interest in practical problems is a system which is capable of acting as an intelligent assistant with a suggest-and-refine mode of operation. A model of interaction based upon abstracting both the model and the search strategy outlined above is discussed in Woods (1993*b*).

Comment: **Abstraction provides a way of dealing with partial solutions .... in a way that is familiar to the expert .... key elements, addnl evidence...**

An abstracted template model is essentially a specification of the model that includes a specification of which aspects of the model represent the "key" or minimal template aspects that should be recognized, along with a detailed description of which additional aspects represent supporting evidence for a successful match. As evidence for a successful match accumulates through more feature identification, an interactive strategy would allow the expert interacting with the recognizer to accept the partially instantiated model as a successful match at any point in the search. From this partial match of situation elements to template slots, the expert might wish to make some nonmonotonic inferences about the existence of other as yet unseen situation elements according to the structure and constraints inherent in the accepted template. These inferred but unseen elements are referred to in Williams & Woods (1993) as *Expectations.*

Comment: – **more work in learning improvements of model, etc. ...**

As described, the templates used as the basis for search are approximations constructed by experts based upon domain knowledge and previous experience. The ongoing refinement of these models to provide a more accurate matching process would be of great benefit. The ramifications of attempting to learn template refinements with abstracted template models is introduced in Woods (1994).

Comment: – **vs implementation as standard constraint logic programming languages ...**

Standard "AI" approaches to solving CSPs have been rooted in a melding of search strategies with constraint propagation algorithms to form hybridized search strategies that make use of domain independent and domain dependent heuristics during search. More expressive representation languages for CSPs are useful since additional problem descriptiveness may be exploited in controlling search refinement. While it would be attractive to attempt implement recognition systems based upon existing constraint specification technology such as provided by current constraint logic programming implementations, they are too rigid and inflexible in both representation and control issues to be considered as a *sufficient* toolset for the CSP application developer (Woods 1994).

Essentially, the tools provided are tied too tightly together to allow sufficient flexibility to take advantage of specific problem characteristics to best effect, and what is really needed is a toolset consisting of a strong representation language, search and consistency tools that may be merged to any degree, and support for dynamic changing of control strategy throughout the CSP search space.

3 A possibilistic representation strategy

Comment: **Stephane ... will briefly describe his strategy for representation.**

This work is expanded in Lapointe & Prouix (1994).

4 Combining search and possibilistic representations

Comment: **Steven and Stephane ... merge the two methodologies - with Stephane's representational semantics using hybridized search ... OR compare and contrast the two strategies.**

5 Conclusions

Comment: **Hm ... basically what we have learned and how the implementation improves on others or on either the possibilistic or csp representation alone.**

Acknowledgments

This work has been carried out under the auspices of the Department of National Defence, Defence Research Establishment Valcartier as part of the Tactical Information Fusion prototype design project. Steven Woods would also like to acknowledge support from the University of Waterloo, and the National Sciences and Engineering Research Council.

References

Kumar, V. (1992), 'Algorithms for constraint-satisfaction problems', *AI Magazine* pp. 32–44.

Lapointe, S. & Prouix, R. (1994), Fuzzy geometric relations to represent hierarchical spatial information, *in* 'Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence'. to appear.

Mackworth, A. (1977), 'Consistency in networks of relations', *Artificial Intelligence* **8**, 99–118.

Nadel, B. A. (1989), 'Constraint satisfaction algorithms', *Computational Intelligence* **5**, 188–224.

Prosser, P. (1993), 'Hybrid algorithms for the constraint satisfaction problem', *Computational Intelligence* **9**(3), 268–299.

Williams, G. & Woods, S. (1993), Representing expectations in spatial information systems: A case study, *in* 'Proceedings of the 3rd International Conference on Large Spatial Databases'.

Woods, S. (1993*a*), Interactive recognition of spatially defined model deployment templates, Research memorandum, Combat Intelligence Automation Group, Defence Research Establishment Valcartier, Courcelette, Quebec CANADA. unclassified.

Woods, S. (1993*b*), A method of interactive recognition of spatially defined model deployment templates using abstraction, *in* 'Proceedings of the 1993 DND Workshop on Advanced Technologies in Knowledge Based Systems and Robotics'.

Woods, S. (1994), Topics in spatial template recognition: Learning, complexity and implementation, Technical report, University of Waterloo. advance copy.