Althea: Recalled Process For Planning and Execution

Jonathan Handler, Paul Cooper, Jeremiah Faries

Intelligent Perception and Action Laboratory
Institute for the Learning Sciences
Northwestern University
1890 Maple Avenue
Evanston, IL 60201

Phone: (847) 491-7979
Fax: (847) 491-5258
E-mail: jhandler@ils.nwu.edu
cooper@ils.nwu.edu
faries@ils.nwu.edu

Abstract: We address two central problems causing a breakdown between planning and plan execution in robots. The first is the difficulty in reliably maintaining an accurate world model over time. Especially hard is the problem of perceptually segmenting objects and then tracking those objects. We suggest a representation language for plans that is based on perceptual data, not objects. The second major problem we address is the brittleness of plan operators. The execution of plan operators on actual robots has revealed the myriad of ways that the assumptions built into the construction of those operators can break. We suggest interleaving the execution of plans that contain sequences of motor *and* perceptual primitives, including operator failure. Having dramatically increased the size of each plan, we suggest a case-based representation as a means of keeping the search space tractable.

Introduction

Robust execution of plans by robots will require that the agent be able to respond to its environment. Because the world is constantly changing, plans have limited useful lifetimes. It is likely that at some point, some assumption that went into constructing the plan will fail. An agent will always be faced with the problem of selecting an action appropriate to its current circumstance (Agre and Chapman, 1990). This paper explores two central breaks in the assumptions of classical planners — world modeling and primitive-operator execution. The issues faced in these subproblems are illustrative of the wider issues of correct action selection.

The automatic generation of a veridical and exhaustive first-order logic representation of the world is a major foundation of classical planners (Fikes and Nilsson, 1971; Sacerdoti, 1977; Brooks, 1991). Unfortunately, it has proven an impossible task. Object segmentation, especially based on vision, is not reliable. Even when segmentation is possible, the information generated is almost guaranteed to change without the planner's knowing. For this reason, symbols in a classical plan that refer to objects in the world are often left dangling with no existing referent.

In order to address the unreliability of perceptual segmentation of objects, we need planners that are embedded in autonomous agents to plan based on perceptual features, not objects. It is a tautology that a call to read the gripper's width on a robot will yield the result that it does. It is an inference to conclude that there is an object in the gripper. This inference is not necessarily reliable. For instance a 66mm grip width may mean that there's a Duplo in the gripper. Then again, the gripper might just be stuck open. A system that is limited to the predicate Gripping(Duplo-54) takes for granted that there is

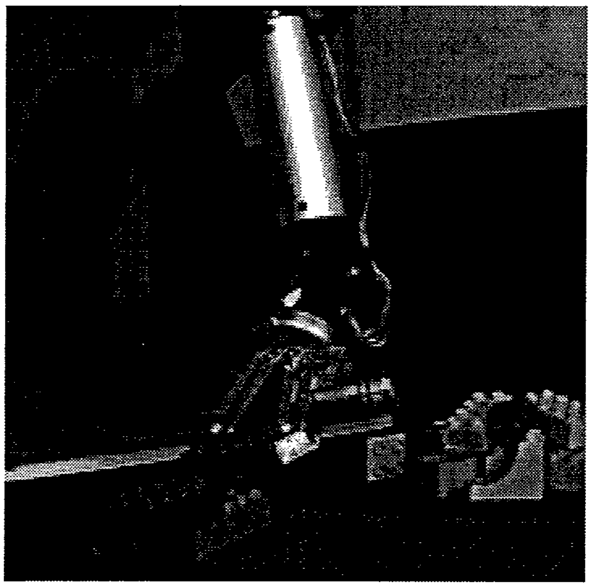

Figure 1: Althea's task: use the pile of bricks to build a wall.

only one way to interpret that grip width. It is better to make the inference explicit in the plan by representing directly the perceptual information that supports that inference.

Errors in the execution of plans are inevitable. Attempts to execute plan operators on real robots have shown that even simple operators are brittle, breaking often and in unpredictable ways. In order to provide the flexibility needed for robust execution in the face of these brittle operators, plans must have interspersed sensing operations in their strings of "blind", primitive operators. There are some current efforts in this direction. Pryor (1994) treats perception as something to be planned for. Reactive systems (Firby, 1989) employ perceptually-oriented success conditions to insure the success of their operators. Behavior-based systems (Brooks, 1991; Horswill, 1993) allow sensing directly to drive the selection of the agent's actions. We are working on a way to include the perceptual data in the plan itself.

This paper describes work in progress on a hand-eye system, Althea, being developed at Northwestern University. Althea's task is to build a wall out of Duplo™ bricks (Figure 1). It will employ a case-based method of planning and execution (Reisbeck and Schank, 1989; Schank, 1982; Hammond, 1989). Althea's cases will be grounded in perceptual data and will also contain representations of strings of perceptual, motor and inference actions. We seek to add perception to plans by defining plans as cases that contain both actions and perceptual primitives. We seek robust execution through timely case retrieval.

Case-Based Planning and Execution

The essential insight of case-based systems is that it is easier to do something the way you did it before than to figure out how to do it again from scratch. In working out this idea, typical case-based planners, e.g., Hammond's CHEF (Hammond, 1989), have worked from a description of the desired result, generating a matching plan/case from a case-base. Once a close match is generated, it must be tweaked to fit the input situation exactly and then executed. CHEF does its retrieval once, essentially ignoring the problem of the execution of its plan's operations.

One problem with applying this paradigm to embodied agents is that we cannot assume that a single plan can be reliably executed. Where CHEF retrieves only one case, then executes it, Althea will retrieve a case whenever the current case no longer fits the situation. The most common mismatch is likely to be a datum representing the failure of some operation. Another common failure, explored below, is likely to be the reception of a datum that clashes with the current plan's expectations, as represented by a datum in the case. For instance, the next datum in the current case might be "33mm grip width", while a reading of the grip width might have yielded 66mm.

Althea's cases will represent at least two kinds of information — operators and data (see below). The type of computation performed at execution time by these operators is currently unlimited. We expect that useful classes of operations will include motor actions, perceptual actions and inference actions. The data represented in the cases will not be raw perceptual results, e.g. images. Instead, some useful abstraction over classes of raw data must be discovered empirically.

We hope that interleaving the execution of disparate cases will provide extra flexibility where it is needed. Althea will execute steps from its current case until it receives a perceptual mismatch, gross error, or the end of that case. At that time, it will have a new case as the best match for the current situation. It will need to reify the new case with the actual execution trace. This will generate a next action that is based on the first element of the retrieved case that could not be matched to the execution trace.

This fluid application of the cases in the case-base means that the failure of primitive operators can be handled by case switching. Some cases will have a representation of the operator and then its failure explicitly represented as a datum. One of those cases will match to the point of the failure datum. Subsequent operations in the selected case can constitute the recovery from the error. When execution begins in the case, the recovery will commence.

Althea will attempt to skirt the issue of building or rebuilding hierarchical plans by retrieving them instead. Good results will depend crucially on our definition of "case that best fits its current situation." Below, we describe a similarity metric that is sensitive both to surface features and to hierarchical goal information. There are obvious analogies between this metric and goal trees that lead us to believe that its application will be feasible. We note here that in general, sense data is most meaningful when it is seen in the context of the operation that generated it. It may or may not be meaningful that the current gripper width is 33mm. It is *certainly* meaningful when the last operation was a "close the gripper" operation. It is *most* meaningful in the full context of a sequence of sensing, moving, and inference actions that are part of a case to pick up a Duplo.

Two Examples from the Wall-Building Task

In this section, we highlight two parts of Althea's task as representative of the problems faced in planning for and executing actions by embodied agents. The first example, acquiring a Duplo from a pile of Duplos, underscores the intractability of the segmentation problem. It shows the benefits of refusing to do visual segmentation when the task allows for simpler answers. The second example, orienting a Duplo, treats flexible execution in the face of uncertainty. It shows how case switching can result in robust execution.

Consider the planning problem for acquiring an object from

Figure 2: Camera's-eye views (left to right). (a) End-on view of a Duplo. Is it 2x2 or 2x4? (b) Pick operation for 2x2 Duplo (c) Pick operation for 2x4 Duplo.

the environment. First the plan must specify operators to find the object using at least one perceptual modality. Failing this, it must already know where such an object is. The plan must specify a means of moving to the object's location and then gripping. What plan can we make in cases in which the location of the object is unknown? A good example of this type of situation is a perceptually complicated, single image of a pile of Duplo bricks (see for example the pile in the background of Figure 1). It is difficult to imagine any generic method succeeding at segmenting a Duplo from this image and determining its location in space.

Since the system lacks the ability reliably to segment an object, the planner must plan with something different than objects. A good plan to acquire a Duplo might be to figure out the center of the pile of Duplos (using more reliable visual routines like visual center of mass), drive the arm into the pile, and close the gripper. Detection of success will be based on perception. The resulting width of the gripper will provide evidence as to whether or not something was gripped. Further evidence can be provided through looking at the gripper, expecting to find a Duplo. This is a good plan for this situation since it does not require the system to locate a Duplo nor to commit to the identity of a Duplo.

This example provides challenges for traditional robotic systems as well as for traditional planning systems. Traditional robotic systems plan paths as sequences of joint velocities and accelerations given the desired destination of the effector in three-space. But, a pile of Duplos is an extremely amorphous destination. The desired destination of the effector is better defined by a tactile result of moving the arm — the pile is located where the system runs into something. Because the destination of the effector is not known, path planning seems like a poor choice.

The challenge to traditional planners is to create a plan to acquire a Duplo from a pile of Duplos without reasoning about every individual Duplo in the pile. It is certainly the case that such a plan can be constructed. However, it could not be based on symbols that represented individual Duplos, since their visual segmentation is nigh impossible. As with the path-planning problem, the best results are based in perceptual primitives like running into something. However, each additional operator (and its result) makes the search space worse by an exponential factor. Here a case-based planner is a superior alternative to exponential search. In the worst case, search through a case-base has linear time complexity. We believe that we will be able to create a case-base that is not exponentially large and that still covers the necessary actions of the system.

Continuing with the example, one consequence of blind gripping is that the result is unknown. Althea's cases will need perceptual operations to recognize the object in the gripper. Its task currently allows for two different sizes of Duplo. Smaller Duplos have a square top face with four raised studs (2x2 Duplos). Larger Duplos have a rectangular top face that is twice as long as wide, with eight raised studs (2x4 Duplos). Importantly, both sizes of Duplo have the same width. So, the size of the Duplo, and hence the width of the gripper when it is gripping a Duplo, is diagnostic, but only some of the time.

The situation depicted in Figure 2a is a good example of how to exploit the flexibility of deferring object identification. The gripper is in fact holding a 2x4 Duplo, gripped along the long edge, and seen end-on. Perceptual data implies that the system is holding a Duplo — the 33mm grip width and the blue rectangle are both consistent with this judgment. However, there are no perceptual data to indicate which size Duplo is actually in the gripper. Knowing the size of the Duplo is critical for insuring that the system grips it so that its fingers are not in the way when it adds the Duplo to the wall.

Althea will be able to take the appropriate action in this situation since the character of most cases when their execution reaches this point is likely to be similar. That is, most cases in the case-base will likely be written with an eye to doing something at this juncture that best serves some longer-term goal. Here, most cases will probably specify putting the Duplo on the table, since the image in Figure 2a can never indicate that there is a Duplo gripped correctly to be placed on the wall. All cases will be written to reorient the Duplo and putting it down is the first step for any reorientation. Thus, most cases, when execution

reaches this point will specify putting the Duplo down. That means that whichever case is retrieved, putting the Duplo down is likely to be the next, unmatched operation.
It is important to note that a main reason that the cases will have similar local character is that all 2x2 and 2x4 Duplos are equivalent as far as the task is concerned. In fact, the ability to exploit these kinds of task-specific equivalences will likely be a significant strength of Althea.
Althea continues executing its current case, placing the Duplo on the table and regripping it. To do this, the case calls for placing the Duplo down, changing the arm's rotation, and picking up the Duplo. On the regrip operation, the grip width returned will be 66mm (Figures 2b and 2c). This uniquely identifies the Duplo as being 2x4 and causes the system to change cases.
These examples show that Althea is able to avoid symbols that represent individual objects whose state must be reliably kept up to date all the time and that it can be responsive to expectation failure. It is not important as far as the task is concerned that the system segment and return a single Duplo object from a perceptually confusing pile of Duplos. A more generic method of grabbing at the pile suffices to deliver the desired result. Althea is also responsive to new information. It is able to replan with fairly low cost when the situation warrants. Having seen something that fails to be diagnostic, the system can continue with its current case. When information arrives that indicates that the current case is wrong, Althea is able to select a new case. The next section describes a low-cost matching method that allows Althea to retrieve cases at less than exponential cost.

Retrieval: How?

The major work that lies ahead is the development of a representation of cases whose execution can support the robustness and flexibility described. There are two primary issues, design of operators and representation of cases. We can take inspiration for the representation from the examples. In the examples, concrete operators, especially as paired with the result of applying those operators, signaled that the current case continued to be the correct one. The *actions* and the *perceptual data* generated by those actions continued to match the case. Action and data elements will thus be the foundation of the representation.
The design of operators is based on selecting task-relevant visual, motor, or inference actions (cf. Ullman, 1984). The point is to decompose the wall-building task enough to identify pieces of action that are large enough to perform some useful part of the computation. These operators will be most useful when they generate diagnostic perceptual data. Their representation will simply be a string tag. In particular, there will be no attempt to represent the consequences or even the type of these actions. We are relying on the design of the case-base to insure that the actions are placed appropriately in their cases.
The data to be represented will be the results of performing these perceptual or motor actions. When the system is running, data will be instantiated by actual data generated as a result of performing the actions in the case. At run time, the data will exist as program objects. However, it would be undesirable to store program objects (or even the raw data) in the case-base. We are currently exploring ways to generate useful abstractions, guided by the demands of the task, that will result in categories of data that can be easily matched on the fly.
We speculate that it will be useful for cases to represent the goal tree as well as the sequence of operators and data. We hope that we will be able to exploit structural properties of cases to make the matching more robust. In prior work (Handler, 1992; Handler and Cooper, 1993), we showed how a fixed-size tree could be matched in parallel to another, same-sized tree with an empirically observed time complexity of the simulation algorithm of $O(n)$ in the total number of nodes in the system. That system, SMERF, used three coupled Markov Random Fields to represent two trees and the cross-product of the possible matches between nodes of those trees at the same levels of the hierarchy. Cross-level connections in the matching network provided a conduit for abstract information to flow top down, influencing choices about concrete primitives. The same connections allowed a bottom-up movement of information that could prompt a change in the high-level match in response to enough low-level tension.
Several elements of that system should carry over to Althea. The matching between the current situation and the stored case base will be a similar tree-match. In Althea, the case-base is a forest of trees, each representing a plan. The current situation is represented by a trace of the primitive actions and data and a current best-matching plan. The primitive trace, along with the current case will be matched by a third, hierarchical matching network.
SMERF was built to model data from psychology on the nature of similarity. Markman and Gentner (1992) showed that relational information is able to overcome feature matches during judgments of similarity. In that experiment, an object's relations to other objects in its conceptual environment were more important in determining human match judgments than the concrete features of that object. We suspect that case retrieval will be more robust when high-order information is included in the matching process. Structure mapping theory (Gentner, 1983), and especially the Structure Mapping Engine (Falkenhainer, Forbus and Gentner, 1989) provides a means of mapping trees that includes the top-down flow of information. Since our cases will be trees, we should be able to apply this method directly. Further work in psychology on problem solving leads us to believe that this method is consistent with human problem solving. Experimentation with both experts and novices indicates that people use abstract information to classify problems and solve them (Faries and Reiser, 1988; Faries, 1991; Chi, Feltovich and Glaser, 1981).

Related Work

There is a long history in AI research of using trees in problem solving and visual and motor tasks. An early system (Ambler, et al., 1975) used a part-whole decomposition quite successfully in performing a construction task. Shapiro and her colleagues (Shapiro, 1980; Shapiro and Haralick, 1985) present a metric for inexact graph matching of part graphs. Ferguson (1994) employed structure in the generation of symmetry judgments. Cooper (1990) used structure as a partial basis for disambiguation in object recognition.

The RAPs system (Firby, 1989; Firby, 1994) is quite close in spirit to Althea. Major similarities include the prespecification of task nets as analogous to cases. However, we think the differences are quite striking. RAPs does not and is not able to reason about which primitive operator to execute in the context of the entire goal tree. We hope Althea will take both its currently-executing plan and the rest of the database of cases into account. One of the strengths of Althea is the possibility of constant re-appraisal of its current case by comparing the execution trace to the case base. A second major difference is the presence of data elements in the case itself, compared with a (albeit secondary) world model in RAPs. A final important difference is the fact that RAPs contain satisfaction conditions built into them. Althea manages this desirable property through responding to failure by essentially reinterpreting its situation.

Conclusions

We have described a case-based system, Althea, that will perform a significant perceptual-motor task with a camera and robot arm. In order to overcome the difficulty of reliably maintaining an accurate world model, Althea's plans will represent perceptually simple primitives. In order to insure flexible execution in the face of uncertainty, Althea will interleave the execution of cases from its case-base, essentially replanning when failure occurs. We hope that we will be able to design a case-base that is hierarchical, and based on a model of human problem solving.

In the end, it is the cases that provide the baseline solution to the problems described. The presence of perceptual primitives in cases will keep the plans in appropriate correspondence to the state of the world. Having access to a limited number of plans will keep the time and space complexity tractable. Finally, the context provided by the representation of strings of perceptual and motor primitives in the cases should make matching to the present situation robust and insure correct selection of the next action. In future research, we hope to show how these solutions can scale to more general problems of action selection.

Acknowledgments

This work received funding under NSF grant #IRI-9521483. Ian Horswill and Penni Sibun were instrumental in bringing this document into being. Thanks to Larry Birnbaum and Ron Ferguson for comments on an early draft and for continuing good discussions.

References

Agre, P. and Chapman, D. 1990. What are plans for? *Robotics and Autonomous Systems.* 6:17–34

Ambler, A. Barrow, H. Brown, C. Burstall, R. and Popplestone, R. 1975. A versatile system for computer-controlled assembly. *Artificial Intelligence.* 6:129–156

Brooks, R. 1991. Intelligence without representation. *Artificial Intelligence.* 47:139–160.

Canny, J. F. 1987. *The Complexity of Robot Motion Planning.* The MIT Press.

Chi, M. T. H., Feltovich, P.J., and Glaser, R. 1981, Categorization and representation of physics problems by novices and experts, *Cognitive Science,* 5(2):121-152

Cooper, P. R. 1990, Parallel structure recognition with uncertainty: coupled segmentation and matching, In *Proceedings of the International Conference on Computer Vision, ICCV-3,* Osaka, Japan, IEEE Conference Press

Falkenhainer, B., Forbus, K. and Gentner, D. 1989. The structure-mapping engine: algorithm and examples. *Artificial Intelligence.* 41:1–63

Faries, J. and Reiser, B. 1988. Access and use of previous solutions in a problem-solving situation. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.* pp.433-439. Lawrence Erlbaum Associates, Hillsdale, NJ.

Faries, J. 1991, Reasoning-Based Retrieval of Analogies, Doctoral dissertation, Princeton University

Ferguson, R. 1994. MAGI: Analogical encoding using symmetry and regularity. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.* Atlanta, GA.

Fikes, R. and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence.* 2:189–203.

Firby, R. J. 1989. *Adaptive execution in complex dynamic worlds.* Doctoral Dissertation. Yale University Department of Computer Science. YALEU/CSD/RR #672.

Firby, R. J. 1994, Task networks for controlling continuous processes, *Proceedings of the Second International Conference on AI Planning Systems*, Chicago IL
Gentner, D., Structure-Mapping: A Theoretical Framework for Analogy, *Cognitive Science*, *7*, pp. 155-170, 1983
Hammond, K. 1989. *Case-Based Planning: Viewing Planning as a Memory Task.* San Diego, Ca. Academic Press.
Handler, J. 1992. *Analogical Similarity: Performing Structure Alignment in a Connectionist Network.* Master's Thesis. Northwestern University, Evanston, IL.
Handler, J. and Cooper, P. 1993. Analogical similarity: performing structure alignment in a connectionist network. *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society.* pp.522–527. Lawrence Erlbaum Associates, Hillsdale, NJ.
Horswill, I. 1993. Polly: A vision-based artificial agent. In *Proceedings of the Eleventh National Conference on Artificial Intelligence.* MIT Press.
Markman, A.B. and Gentner, D. 1992. Evidence for structural alignment during similarity judgments. Cognitive Science Technical Report, UIUC-BI-CS-92-01. University of Illinois, Urbana. IL.
Pryor, L. 1994. *Opportunities and Planning in an Unpredictable World.* Doctoral Dissertation. Technical Report #53. The Institute for the Learning Sciences. Evanston, IL.
Riesbeck, C. and Schank, R. 1989. *Inside Case-based Reasoning*. Lawrence Erlbaum Associates. Hillsdale, NJ.
Sacerdoti, E. 1977. *A Structure for Plans and Behavior.* American Elsevier. New York, NY.
Schank, Roger C. 1982. *Dynamic Memory.* Cambridge University Press.
Shapiro, L. 1980. A structural model of shape. In *IEEE transactions on pattern analysis and machine intelligence.* PAMI-2[2]:111–126.
Shapiro, L. and Haralick, R. 1985. A metric for comparing relational descriptions. In *IEEE transactions on pattern analysis and machine intelligence.* PAMI-7[1]: 90–94.
Ullman. S. 1984. Visual routines. *Cognition.* 18:97–159.