

Analogical Localization: Flexible Plan Execution in Open Worlds

Scott Friedman, Mark Burstein, Jeffrey Rye, and Ugur Kuter
{friedman, burstein, rye, ukuter}@sift.net

SIFT, LLC
Minneapolis, MN, 55401, USA

Abstract

Cognitive systems face the challenge of pursuing changing goals in an open world with unpredictable collaborators and adversaries. Considerable work has focused on automated planning in dynamic worlds, and even re-planning and plan repair due to unexpected changes. Less work explores how humans and computers can negotiate to define shared goals and collaborate over the fulfillment of those goals. Our work takes a domain-general approach to *plan localization*, the problem of establishing the set of steps within the plan that are candidates (potentially after some adaptive repair actions) for next actions given the world's unforeseen changes. We use analogical mapping to help agents determine the nearest states in a diverse plan relative to the current world state, identifying both the maximal *satisfied* states that the world conforms to presently, and the closest *desired* states adjacent to satisfied states that are both achievable by an action and makes progress toward the goal. These are demonstrated in a system called CLiC. The system's overall purpose is to engage in symmetric dialog with human users about goals and recommended actions to achieve those goals. Both the human and the system may choose to take those actions, or describe them to the other party. They may not always do what they are told. Preliminary results indicate that our approach suits collaborative situated agents with flexible goals in open worlds.

1 Introduction

Sometimes plans fail. The issues of uncertainty and exogenous events dominate, and require plan adaptations during execution of the plans. Progress in AI automated planning research has developed different approaches to plan adaptation [12, 10, 1, 14], but despite these advances, the problem of *analogically identifying* the most relevant goals and states in plans to achieve when a failure occurs in the world is still under-explored. This paper describes an approach to *plan localization* that addresses unexpected changes in the world, primarily due to well-intentioned collaborators.

Our approach is different than most work on reactive plan monitoring and repair. For instance, our system does not track whether the last action was precisely matched, since it exists in an open world without full knowledge of actions and paths to success. As other agents act in the world, it examines the world and reasons about the progress that has been made, if any. Our system identifies the plan states that are most *similar*

to the world state, and uses analogical inferences to resume (or re-enter) the plan at the most preferred state. Our approach (1) computes a plan *locale*, (2) opportunistically identifies a *plan reentry state* if the locale is not a recognized future plan state, and (3) selects actions to enter to that re-entry state from the observed world state. For instance, in a block-building domain, the collaborator may have placed two or three blocks at once in an unexpected configuration, putting the world in an unexpected state, but the world state may analogically similar to a more desirable planned state (e.g., with slightly different block configuration), and we can change the world to satisfy that desirable state given some small action. Our approach enables the system to take opportunistic advantage of these situations and also take action despite setbacks.

By using analogical similarity matching to compare the world against future states in a diverse plan tree we can compute two different states of interest:

1. The best *satisfied state*: the state in the plan that is satisfied and closest to the goal, where distance ties are broken by structural similarity.
2. The nearest similar *desired state*: the plan state following a satisfied state that is most structurally similar to the current world state.

The satisfied and desired states comprise the *locale*, which is where the agent should focus its development of a next action to execute. Given the desired state found by this localization process, the system attempts to improvise actions— computed online via analogical inference— that will transition the world *back into the plan* by transitioning to the desired state. We call this the opportunistic *plan reentry*. Localization and reentry flexibly allow the agent to react to setbacks in the world as well as exploit unexpected world changes to more quickly achieve goals. This does not require a *complete* plan of all possible world states, but the more diverse and complete the plan is, the more resilient the localization behavior will be.

We have implemented this approach in CLiC, a dialogue-oriented agent that collaboratively builds block-based structures in a visual environment shared with a human collaborator. The high-level plan localization approach is illustrated in Figure 1 with a working example. The human collaborator suggests a goal structure in language, and CLiC uses its conceptual knowledge to envision that goal and any mentioned constraints. It then generates a diverse plan via regression to the current (empty table) state. CLiC both issues linguistic directives so that the user can move blocks, or alternatively, it responds to user directives by taking actions to move blocks on the table to incrementally achieve the shared goal. At each step, if the user responds unpredictably, e.g., by placing two blocks instead of one, or not placing them where directed, CLiC uses analogical plan localization to re-establish its perceived position in the plan by matching against the shared world state. In this process it re-aligns which real blocks correspond to which planned blocks, and then proceeds to improvise new directives to move the plan forward, e.g., “*push block 8 together with block 6*,”.

We continue with a description of CLiC’s planning, plan revision, and the task of collaborative building in Section 2. In Section 4, we describe our domain-general analogical plan localization approach, which we empirically support in Section 5 with multiple scenarios of CLiC utilizing this approach to flexibly mitigate and exploit unexpected changes in the world. We close with a brief discussion of related work (Section 6) and future work (Section 7).

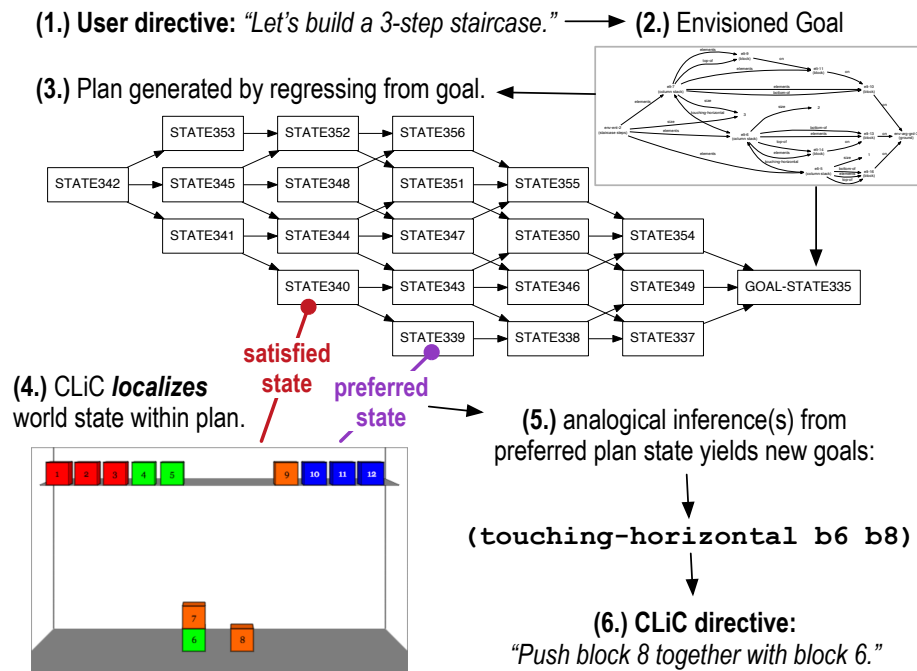


Fig. 1. CLiC localizes the unexpected world state within its plan via analogical mapping, and then selects an action via analogical inference. This allows it to re-enter the plan at a desirable state.

2 Background: CLiC & Collaborative Building

We are developing CLiC for DARPA's Communicating with Computers program, a program about contextually-grounded multi-modal communication with machines. We are presently working in two collaboration-oriented test domains: (1) collaborative goal selection and construction of structures using blocks and (2) a system for discussions with biologists about modeling and evaluating biochemical pathways. The basic CLiC reasoning infrastructure is shared across these domains. CLiC is a subsystem of a larger integration with the TRIPS agent-based dialog system architecture [4]. CLiC provides the domain reasoning for the dialogs occurring in both domains, and does planning and action selection, as well as responses to the user. Agents from the TRIPS system do the natural language understanding and maintain the state of the dialog as a high level goal tree, issuing high-level collaborative goals to CLiC.

CLiC reactively pursues collaborative goals by directly acting in the world or issuing linguistic directives to the user [11], and sometimes asking questions when more information is required to formulate a reasonable objective. In the biocuration domain, CLiC coordinates domain-specific biological modeling and reasoning agents, developed by other collaborators, to achieve collaborative goals with biologists. The rest of the paper focuses on the collaborative block-building setting, since CLiC's biocuration setting does not presently demand advance planning.

In the blocks world domain, the human collaborator suggests a named structure to build, e.g., row, stack, wall, staircase, with additional specifiers for size. CLiC uses conceptual knowledge to envision (as a set of logical and spatial relations over blocks) the form to be constructed. This is shown in Figure 2(a). When the user issues subsequent revisions to the goal, such as specifying sizes or colors— some of which may involve conflicting constraints— CLiC transforms these revisions into rules representing the specified constraints and then runs these rules to revise the goal. The result is illustrated in Figure 2(b), which is the same goal after being told “*the blocks should be green*” and then “*the tops should be red.*”

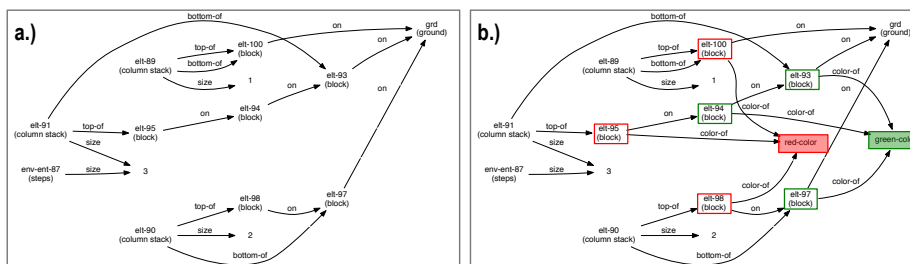


Fig. 2. Simplified semantic graph of a to-be-built staircase goal envisioned by CLiC, including a sequence of imaginary stacks with imaginary blocks: (a) the initial envisionment; (b) the envisionment after CLiC is told “the blocks should be green” and then “the tops should be red.”

When CLiC receives a new goal, or when a goal is modified, it performs a simple plan regression search to generate a diverse plan from the solution state to the initial (empty table) state. One such plan graph CLiC generates to build a three-step staircase is shown in Figure 1. Unless specified by the user, the plan states are agnostic as to which *specific* blocks are used, what color they are, and the orientation of the structure (e.g., whether the staircase ascends to the right or left).

After agreeing on a goal— and in absence of additional directives and questions from the user— the rest of the block-building proceeds by (1) plan localization, (2) action selection, (3) action execution, and (4) awaiting collaborator actions before looping again. Before describing the plan localization approach that is our focus here, we briefly review the theory and implementation of CLiC’s analogical reasoning.

3 Background: Analogy & Similarity

CLiC uses a domain-general implementation of the Structure-Mapping Theory [8] with a greedy algorithm similar to the Structure-Mapping Engine [5].

Structure-mapping takes two relational semantic graphs as input— a *base* and a *target*— and greedily computes a best *mapping* (i.e., nodes that correspond between base and target). Structure-mapping is a *maximal common subgraph* (MCS) problem, with the following hard constraints:

- **One-to-one**: a node in the base can match to at most one on the target, and the reverse. (This follows from the definition of MCS and isomorphism.)
- **Parallel connectivity**: two relational nodes can only correspond if their arguments also correspond.
- **Identity**: relations (or attributes) can only correspond if their predicates (or categories) correspond.

Structure-mapping specifies an additional soft constraint:

- **Systematicity**: higher-order structures (e.g., relations over other relations or functions) are preferred over independent facts.

Structure-mapping computes a similarity score σ_b^t between the base b and target t , which is a weighted sum of nodes put into correspondence (higher-order relational nodes are weighted higher to implement systematicity). Intuitively, σ_b^t increases with the number of nodes in the mapping and with the systematicity of the nodes in the mapping, all else being equal. This is the objective function to maximize in the MCS solution. We use it to rank similarity: for a base b , target t_1 is more similar than target t_2 if $\sigma_b^{t_1} > \sigma_b^{t_2}$.

Finally, structure-mapping produces *analogical inferences* between base and target graphs. These analogical inferences are relations and attributes that are excluded from the mapping (i.e., they have no match in the MCS), that describe elements that *are* in the mapping. Analogical inferences can be projected from base-to-target or target-to-base. For example, suppose blocks in the world **world-b1** and **world-b2** correspond to blocks in a plan state **plan-b1** and **plan-b2**, respectively, and the relational statement (**touching-horizontal plan-b1 plan-b2**) is asserted in the plan, but the corresponding statement is *not* asserted in the world. Structure-mapping will produce the analogical inference (**touching-horizontal world-b1 world-b2**) as a projection in the world graph. Analogical inferences are relations or attributes projected whenever symbols correspond across graphs (e.g., $\langle \text{world-b1, plan-b1} \rangle$ and $\langle \text{world-b2, plan-b2} \rangle$) and one graph lacks a relation or attribute over the corresponding elements (e.g., (**touching-horizontal world-b1 world-b2**) is not asserted, so it is inferred). These inferences are not provably sound, but as we discuss next, they can be used very practically for improvising actions in the world.

4 Approach

CLiC’s analogical plan localization runs whenever the world changes. For the purpose of illustration, suppose the task is to build a three-step staircase out of cubed blocks (comprised of six blocks in total) and CLiC is directing the human user how to build the structure. Suppose also that there is a single block **b6** on the table, and CLiC selects a planned action and tells the user, “Put block B6 on the table, and push B6 together with B8.” This directive— should the human user obey it— would traverse a single edge in the plan graph and result in a transition to a planned state.

Now suppose that instead of putting **b6** next to **b8**, the user put **b6** on the table *apart* from **b8** and then immediately put **b7** on top of **b6**, as shown in Figure 1, bottom left. CLiC’s plan localization runs after this unexpected (and undirected) world change.

The plan localization algorithm is given the world state w and sets its current state c to the initial state (e.g., **STATE342** in Figure 1), and then performs a best-first search:

1. If c is the goal state, the world satisfies the goal. Return success.
2. Otherwise, compute N as c 's next immediate states in the plan graph.
3. Compute next state c' with highest similarity to the world: $c' = \max_{n \in N} \sigma_w^n$.
4. If the w to c' mapping has no analogical inferences to w , w satisfies state c' then set $c = c'$, loop to step 1.
5. Otherwise, return c as best satisfied state s_{sat} and return c' as desired state s_{des} .

This best-first search uses structural similarity as a guide through the state space of the plan to orient the agent within the plan. The agent has now approximated the best state in the plan whose conditions have been satisfied s_{sat} and a structurally similar state that might be opportunistically re-entered s_{des} .

The algorithm then computes actions that will change the world to satisfy s_{des} by computing analogical inferences in the mapping from w to s_{des} : all analogical inferences from s_{des} into w are treated as requirements to satisfy s_{des} . In the case described above (where **b6** is not touching **b8**), the system can force the world into **STATE339** in Figure 1 by achieving the analogical inference (**touching-horizontal b6 b8**). This action will allow CLiC to jump from its previous locale with one block (i.e., **STATE341**) two steps ahead (i.e., to **STATE339**) by exploiting the unexpected change.

5 Experiments

We present four scenarios where CLiC localizes itself and achieves its goal, despite unexpected changes in the world and unexpected starting conditions.

5.1 Change spatial relations to reenter plan

In Section 4, we used the scenario in Figure 1 (on page 3) to outline the approach. In this scenario, the goal is to build a staircase. With a single block on the table, CLiC directs the human to put another block next to it; however, the user instead stacks two blocks on the table apart from the first. CLiC uses analogy to find the similar, desired plan state with a stack of two blocks and another block touching it on the table. The analogical inference is that the **B6** and **B8** should be touching, so it directs the user to push them together. This improvisation allows CLiC to reenter and complete the plan.

5.2 Utilize unexpected structure toward the goal

In this scenario, shown in Figure 3, the human suggests building a three-block stack. CLiC suggests stacking a second block on **B1**, but the human instead stacks **B10 B11** apart from **B1**. CLiC then localizes this state to the penultimate state using the new **B10/B11** entities, and refocuses effort on the **B10/B11** stack instead of the initially-suggested **B1** stack, with the directive “*How about you put B12 on B11?*”

User-specified goal: “let’s build a stack with three blocks.”

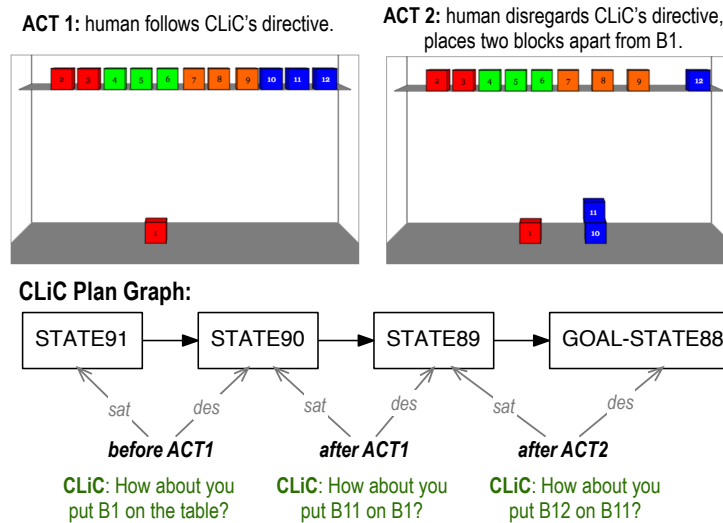


Fig. 3. CLiC exploits an unexpected change to refocus its effort toward the goal.

5.3 Reframe a symmetric goal with analogy

Staircases can be built in either direction, unless otherwise specified. In the Figure 4 scenario, CLiC and the human reach a state where a row of three blocks are on the table, and a fourth block is on top of the middle. From here, the staircase could still be built in either direction. CLiC suggests building in one direction, but the human disregards and puts a block on the other side.

CLiC reacts to this unexpected change by still localizing the world into the penultimate state and directing the user to put **B3** on **B2** to complete the staircase in the direction the collaborator determined.

5.4 Exploit starting conditions to enter a plan near the goal

In the scenario shown in Figure 5, the human places a four-block row on the table, with a fifth block on top, before specifying the staircase goal via dialogue. CLiC reactively localizes the already-developed world into its plan, identifying penultimate state **STATE1988** as the desired state s_{des} . The fourth block **B4** on the table corresponds to a second-row block in the planned state, so the analogical inferences include (**touching-horizontal b10 b4**) and (**on b4 b3**). CLiC achieves these spatial relations by stacking **B4** on **B3** and then subsequently achieves the goal state.

This illustrates that analogical plan localization is useful for orienting the agent in unexpected starting states in addition to reconciling unexpected changes in the world.

User-specified goal: "let's build a staircase with three steps."

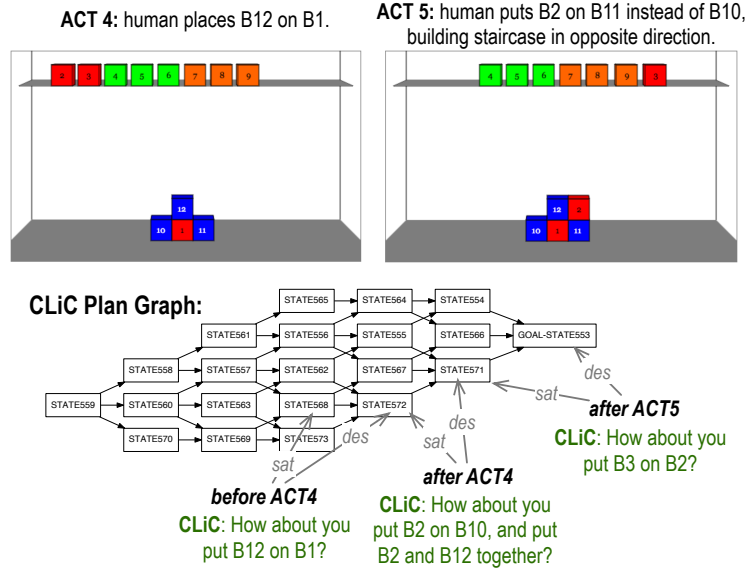


Fig. 4. CLiC reflects its goal staircase in the opposite direction to accommodate the collaborator.

User-specified goal: "build stairs with three steps."

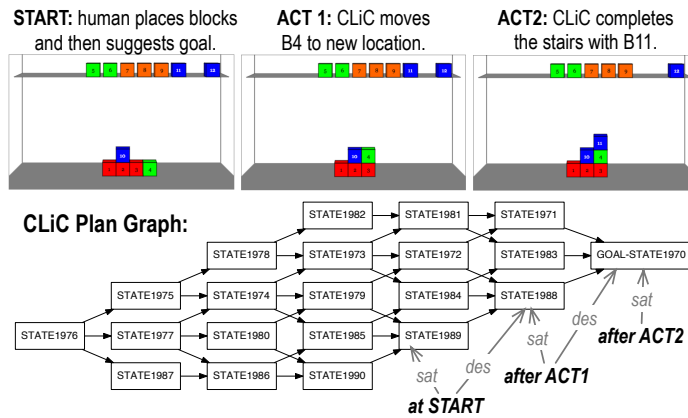


Fig. 5. CLiC adapts an existing scene via analogical inference to enter a plan near the goal state.

6 Related Work

Many AI planning systems have addressed re-planning, and plan repair. Decades of work on reactive planning (e.g., [12]) have investigated insertive and destructive approaches to reconciling the world state with plan states. The approach described in this paper differs from these AI reactive replanning and plan repair methods in that our approach uses structurally similar states to opportunistically repair plans, rather than just replanning from scratch given the current situation. As such it is somewhat similar to a case-based adaptation approach, though the adaptations are not made on prior cases, but on the initially developed plans for the goal.

Approaches for replanning from scratch [3] are fundamentally different than the approach described in this paper, since replanning generates new plans from the current state to the end of goals of a planning problems; conversely, our approach identifies opportunities for the agent to re-enter the plan, and then the existing plan can be reused.

AI plan repair typically focuses on locally repairing hierarchical plans when the system identifies a discrepancy during execution. For instance, replanning approaches like HOTRIDE [1] and SHOPLifter [10] are closely related to the case-based plan adaptation techniques proposed in the RepairSHOP system [14].

Approaches to diverse planning (e.g., [2, 13, 9]) have generated measurably diverse plans to cover more space of possible outcomes and explore possible state discrepancies in plans *a priori*, before execution. Among these are Coman & Munoz-Avila's work describing how to use case-based similarity/analogy methods in order to generate semantically-different plans. Unlike these approaches that use analogy and similarity to identify differences among diverse plans, our approach uses analogical reasoning to compare the world to the plan, orient the agent, and select actions.

7 Discussion & Future Work

We presented an approach to analogical plan localization that allows agents to flexibly recompute their locale of execution in a plan after drastic or unexpected world changes. In this setting, the plan is not considered as a strict policy for execution in the world; rather, it provides an ordering over partial world states that can be opportunistically reentered and traversed.

Our choice of analogy is particularly useful in a block-building domain. For instance, if our plan states that a blue block should go on a red block, it does not matter which *specific* blue block we choose. This means that analogical mapping can flexibly re-frame which *actual* blue block corresponds to the *planned* blue block in order to accommodate other entities and spatial relations in the analogical mapping. Other domains that allow substitution of entities will likewise benefit from this approach, whereas highly-specific goals are less amenable, e.g., if the goal in a logistics domain is for a specific truck to arrive with specific cargo at a specific location.

Our approach of localization via analogy is not limited to plans; we are using similar approaches to build systems that read articles and then localize extracted information within large models [7], akin to event recognition and information fusion. In this setting, extracted material— such as an abstract description of an event— may localize against

many concrete events in a large model, so we use a constrained similarity-based retrieval model similar to MAC/FAC [6].

Near-term future work includes validating this approach on other planning domains, as we believe that PDDL will support analogy. Other considerations include scaling to situations with larger branching factor, where CLiC's exhaustive regression planning is not tractable. In these cases, we could utilize HTN planning with diversity (e.g., [9]) to cover a subset of the plan space and support plan localization.

8 Acknowledgments

This work was supported in part by Contract W911NF-15-C-0238 with the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO). Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

1. Ayan, F., Kuter, U., Yaman, F., Goldman, R.P.: HOTRiDE: Hierarchical Ordered Task Replanning in Dynamic Environments. In: ICAPS-07 Workshop on Planning and Plan Execution for Real-World Systems (2007)
2. Coman, A., Muñoz-Avila, H.: Generating diverse plans using quantitative and qualitative plan distance metrics. In: Proceedings AAAI (2011)
3. Cushing, W., J.Benton, Kambhampati, S.: Replanning as deliberative re-selection of objectives. Tech. rep., Computer Science Department (2008)
4. Ferguson, G., Allen, J.: A cognitive model for collaborative agents. In: Proceedings of the AAAI 2011 Fall Symposium on Advances in Cognitive Systems. AAAI (2011)
5. Forbus, K.D., Ferguson, R.W., Lovett, A., Gentner, D.: Extending SME to handle large-scale cognitive modeling. *Cognitive Science* (2016)
6. Forbus, K.D., Gentner, D., Law, K.: MAC/FAC: A model of similarity-based retrieval. *Cognitive science* 19(2), 141–205 (1995)
7. Friedman, S., Burstein, M., McDonald, D., Paullada, A., Plotnick, A., Bobrow, R., Cochran, B., Pustejovsky, J., Anick, P.: Learning by reading: Extending and localizing against a model. In: Proceedings of the 4th Annual Conference on Advances in Cognitive Systems (2016)
8. Gentner, D.: Structure-mapping: A theoretical framework for analogy. *Cognitive science* 7(2), 155–170 (1983)
9. Goldman, R.P., Kuter, U.: Measuring plan diversity: Pathologies in existing approaches and a new plan distance metric. In: AAAI. pp. 3275–3282 (2015)
10. Kuter, U.: Dynamics of behavior and acting in dynamic environments: Forethought, reaction, and plan repair. Tech. Rep. 2012-1, SIFT (2012)
11. McDonald, D.D., Greenbacker, C.F.: 'If you've heard it, you can say it': towards an account of expressibility. In: Proceedings of 6th International NLG Conference. pp. 185–189 (2010)
12. Musliner, D.J., Durfee, E.H., Shin, K.G.: Execution monitoring and recovery planning with time. In: Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications. vol. 1, pp. 385–388. IEEE (1991)
13. Myers, K.L., Lee, T.J.: Generating qualitatively different plans through metatheoretic biases. In: AAAI. pp. 570–576. AAAI/MIT Press, Menlo Park, Cal. (July 1999)
14. Warfield, I., Hogg, C., Lee-Urban, S., Munoz-Avila, H.: Adaptation of hierarchical task network plans. In: FLAIRS-2007 (2007)