The Why UI: Using Goal Networks to Improve User Interfaces

Dustin A. Smith
MIT Media Lab
20 Ames St; Cambridge, MA 02139
dustin@media.mit.edu

Henry Lieberman
MIT Media Lab
20 Ames St; Cambridge, MA 02139
lieber@media.mit.edu

ABSTRACT
People interact with interfaces to accomplish goals, and knowledge about human goals can be useful for building intelligent user interfaces. We suggest that modeling high, *human-level* goals like "repair my credit score", is especially useful for coordinating workflows between interfaces, automated planning, and building introspective applications.

We analyzed data from 43Things.com, a website where users share and discuss goals and plans in natural language, and constructed a goal network that relates *what* goals people have with *how* people solve them. We then label goals with specific details, such as *where* the goal typically is met and *how long* it takes to achieve, facilitating plan and goal recognition. Lastly, we demonstrate a simple application of goal networks, deploying it in a mobile, location-aware to-do list application, ToDoGo, which uses goal networks to help users plan where and when to accomplish their desired goals.

Author Keywords
Learning Goal Networks, Plan Recognition, To-Do Lists

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces—*Natural Language*; H.4.1 Information Systems Applications: Office Automation—*Workflow Management*

General Terms
Design, Human Factors, Theory

INTRODUCTION

People interact with interfaces to solve problems to accomplish goals, and knowledge about human goals can be useful for building intelligent user interfaces. In previous work, goal knowledge has used in recommendation systems, disambiguating natural language queries in search engines [8] and in the more general problem of building **plan recognition** systems, which try to infer the user's goals from an incomplete and mixed sequence of observed behaviors.

IUI'10, February 7–10, 2010, Hong Kong, China.

Goals are desired world states and people can describe them in natural language: "[I want to] click the 'okay' button", "start the movie", and "entertain my date". As you can see, goals can span different temporal granularities and their English descriptions can abstract away many important details: of those three descriptions, each contained roughly the same number of words. A goal library is required to map between the *interface-level* goals, about clicking buttons and performing actions in the interface, to the *human-level* goals, the kind that people regularly communicate with each other.

To date, work on modeling user goals has either involved costly knowledge engineering efforts to manually encode the plan knowledge, or automated efforts which learn single plans at a time from user behaviors. Here, we investigate a third option: processing a natural language plan corpus, constructed by a community of people. This approach has the potential of acquiring broad coverage of "human level" goals automatically, through the automated construction of a machine interpretable goal network. In this paper, we explain our approach for deriving a goal network from text and present an example application that uses this goal knowledge. In addition to kind of applications where this goal knowledge is immediately useful, our method of recognizing English description of sub-goals can be used to attach goal networks to descriptions of interface level goals.

GOALS IN USER INTERFACES

There are many uses of goal knowledge in user interfaces. When the application is aware of the user's goal, knowledge about the relationships between goals is useful for automated planning, to coordinate workflows that span multiple interfaces and to build introspective applications.

Automated Planning

Goals can either be inferred by the system (plan recognition), or communicated explicitly by the user. Plan recognition systems attempt to infer the user's active goals and plans based on some partial observation of his or her actions. If the system infers correctly, it tries to help the user by automating the task when it can. See [1] for a review of plan recognition in IUIs. An example of explicit communication through natural language is Roadie [7], a consumer electronics interface where users speak their intentions, such as "I want to watch the news", and the system attempts to understand the goal in order to create and execute a plan.

Modeling tasks between interfaces or within interfaces

Imagine pursuing the goal of "buying a home": In between searching for the perfect house, you may spend time calling realtors, repairing your credit score, applying for mortgages, etc. Each of these tasks involves a variety of subtasks, different tools and interfaces (web browsers, cell phones, financial software, etc), which are all oblivious to your larger workflow. Similarly, when a single application can do many different tasks, like a smart phone or a word processing program, it can be useful to predict which feature subset the user is interested in, depending on his or her goals. A broad library relating human goals to interface actions will be essential for this kind of interface behavior.

Introspective applications

When people make their goals explicit and detailed, it facilitates accomplishing them; however, paradoxically people commonly fail to formulate explicit goals for the domains they most value and formulate more specific goals for the domains they value less [10]. One could imagine a "self-reflective" interface that helps the user to self-regulate (verifying that one's behavior aligns with one's goals), or allows them to reflect upon a longitudinal perspective of how their goals change over time. Again, to achieve this, it will be important to know relations between goals and actions.

ACQUIRING AND USING PLAN LIBRARIES

All plan recognition systems require a hypothesis space, a library of plans; and, designing and acquiring a plan library is an arduous task. Generally, plans are handcrafted by knowledge engineers or learned by observing the user's behavior. TaskTracer [9] for example, monitors a user's interactions with Microsoft WinXP and Office, asks the user to label a few tasks and then uses these labels to classify the user's later actions. *End-user programming* is when the system learns a specific plan from the user's explicit behavior, with the goal of automatically executing that plan in the future. Typically, such plan libraries are small and confined to the tasks the user has taught the system. An interesting exception is the CoScripter learning by example project, where users can explore and exchange web scripts in an online script repository [2]. Such script repositories still require annotating scripts with the goals it helps to accomplish.

Learning plans from a natural language corpus

Our goal is to create a plan library that covers the breadth of everyday human goals and actions. Fortunately, in recent years several websites have appeared which focus on codifying (in English) and exchanging descriptions of procedural problem-solving knowledge: 43things.com, eHow.com, wikiHow.com, hassleme.co.uk; and more targeted knowledge acquisition efforts like OMICS [6]. The knowledge in these resources is encoded in natural languages and requires challenging transformations before they can be used in interface applications. We focused on the knowledge from 43Things.com, a free online community of users who list, compare and discuss each other's goals, with the objective of improving time management and problem solving skills.

It may seem at first that the high-level or frivolous goals mentioned in 43Things might have little to do with the concrete details of a user interface. That might indeed be the case for some entries; but, there are many applications where users indeed record life goals, such as calendars, to-do lists, and project management software, even if the software has no operations for directly achieving those goals. Second, a large goal like "buy a house" might have subgoals like "find a real estate agent" that are directly satisfiable by software operations. Third, we might imagine creating corpora that are more focused on lower level goals, such as "get bibliography into IUI format", which might also encounter the same problems of incompletely described or missing steps, etc. The same sort of analysis techniques presented here will prove relevant.

MINING A GOAL NETWORK FROM TEXT

We downloaded the 43things.com corpus, which contains data about a wide range of *goal statements* including "buy a house" and "travel to Hong Kong". Some statements have "how I did it" stories by people who have achieved the goal and describe how they did, how long it took, and the way achieving the goal made him or her feel. We called these stories *plan statements*. At the time of acquiring a snapshot of the database (Fall 2008) it contained over 2.5 million goal statement entries and $22,528$ plan statements about $9,304$ distinct goals.

The 43Things corpus does not contain explicit relations between goal statements, so the global graph structure must be inferred. Our approach was to look within the text of each plan statement to detect other goal statements. Consider this plan statement by someone who accomplished the goal "eat better":

> i hate eating healthy! i do, i do. i plan to **lose 50 pounds** though, and that is the inspiration you need. everyone needs something to look forward to...

Here, the goal to ''lose 50 pounds" matched another goal in the database. Our procedure looks at the text and recognizes these embedded goals. We looked for a list of subgoals available from 43thing goal statements as well as a filtered list of public todo-lists from Tadalist.com. Because the statements were contributed by thousands of different authors, our mining approach must handle noisy text. Our approach[1] is:

Preprocessing the English Plans First, we corrected misspellings, stripped non-words, segmented phrases and sentences using the Punkt tokenizer, labeled each token with its part of speech and stemmed word root.

Building a directed weighted goal graph For each plan statement, each step of $p \in P_i$ for goal statement i is compared with each stemmed goal statement j to see if the tokens are a *longest common subsequence*[2]If the goal occurs as a subsequence, a directed edge (i, j) is added to the goal graph, given a *support* weight equal to the ratio of plans

[1]Code available from `http://web.media.mit.edu/~dustin/goals`

[2]A **subsequence** is like a **substring** that permits gaps in between the successive elements (comparing only tokens, not POS tags).

that goal occurred in $w(i,j) = \frac{\sum_{p \in P_i} hasSeq(p,j)}{|P_i|}$.

Divisive graph clustering The last step produced a large directed weighted graph with 3335 nodes and 6406 edges and 69 components. We consolidated some of the nodes and removed some of the outliers. Graph clustering looks at the relational structure (between goals), while regular object clustering looks at internal features describing the goals (*e.g.*, the overlapping words between two plans, the duration of the goals, etc). We clustered this large data set using a stochastic flow clustering algorithm [4].

Extracting verb-phrases and locations with rules We filtered the tagged plan descriptions with hand-authored Regular Expression Chunking rules to extract verb phrases and prepositional phrases that were indicative of a location. The rules matched syntactic tags with particular tokens, but yielded many false positives due to the problems of the tagger.

ANALYZING A CORPUS OF HUMAN GOALS

The GOALS themselves contain meta-data specifying the number of users pursuing the goal and the number of people who voted on the goal as 'worth doing' or 'not worth doing'. We analyzed this and our learned graph structure to answer some questions:

Which goals are most popular?

The number of people who have picked a particular goal follows a power-law distribution with its characteristic thick tail: of the 2,524,563 unique goal strings, 88.8% of them are listed by 1 or no people, yet each goal is pursued by 2.6 people on average. The top goals: *lose weight* (32,504); *stop procrastinating* (24,100); *write a book* (22,342); *fall in love* (22,091); and *be happy* (19,498).

Which goals are most controversial?

Users of 43Things.com also have the option to vote on each other's goals, indicating whether they view the goal as "Worth it" or "Not worth it". We used this voting data to try and capture the notion of goals that were the most polarizing. We approximated this using **entropy**, which measures uncertainty—in our case, measuring the community's uncertainty about whether or not a goal is "worth it". Here are some controversial goals:

GOAL	WORTH IT	NOT WORTH IT
don't shave for a few days	149	55
try soy milk or rice milk	122	45
gamble in Vegas	38	14
lose the game	239	89
be a better girlfriend	30	11
own a house	248	12

Table 1. Listed are the top five most controversial goals (highest entropy of worth it ratings). "own a house" was added for an ongoing comparison.

Which goals in the goal network are most central?

After clustering the goals into a goal network, we looked for nodes that had an important role in the structure of the graph. One metric is the **closeness centrality**, measuring the mean shortest path of one node (goal) to all other nodes, which generalizes easily to weighted graphs. The most central goals in our inferred goal network are: *beat depression* (0.212), *be more confident* (0.200), *be happy with myself* (0.196), *have better sex* (0.182), and *find my soulmate* (0.17931). These goals are at the center of influence of the network, and–assuming goals lead to each other–should have the most significance in "opening the most doors" to other goals in the graph.

EXAMPLE INTERFACE

We put the goal knowledge to work in a mobile, location-aware to-do list application, ToDoGo.

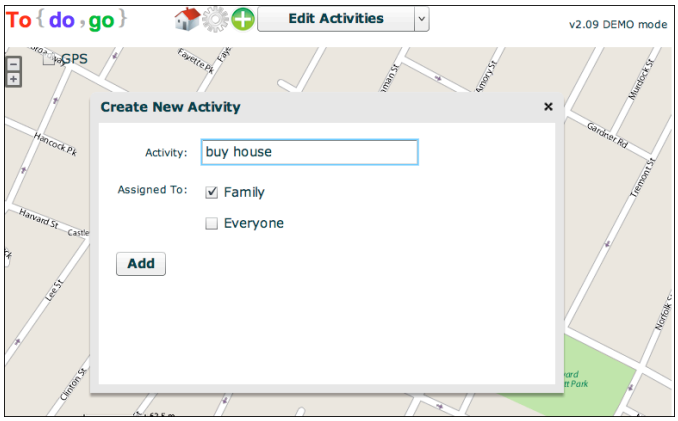

Figure 2. The user describes their goal, "buy house" to the to-do list program (side-stepping the problem of plan recognition).

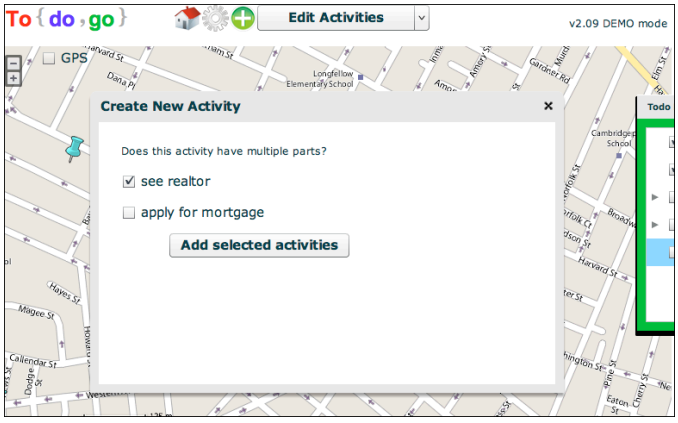

Figure 3. Using the background goal knowledge, the system infers plausible sub-goals for the goal "buy house", including "see relator" and "apply for mortgage". The user selects the appropriate tasks.

To-do lists make plan recognition trivial because the user is already explicitly declaring his or her goal. Several recent projects have worked to automate to-do lists [3]. They

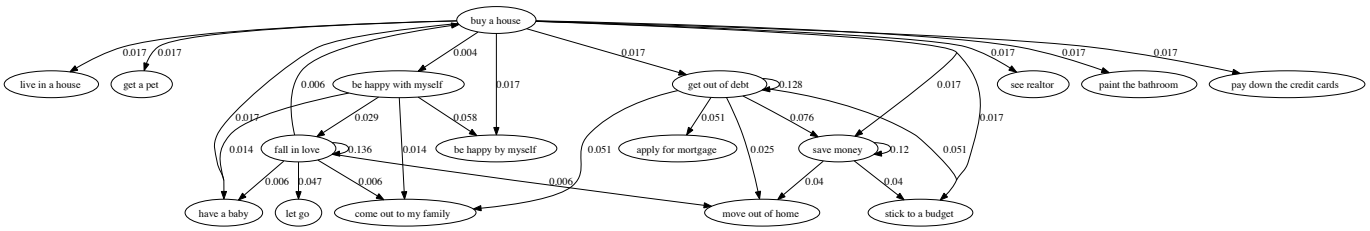

Figure 1. This subgraph shows nodes related to the goal *buy a house*. The weight in the directed arrow indicates the ratio of the parent goal's plan statement the embedded goal appeared.

work by mapping the parsed to-do list entry into actions that can be solved by an automated agent. This is challenging because many to-do list entires are abbreviated and not all entries are tasks that can be automated [5].

Plan and goal knowledge could be put to use in a straightforward manner: Returning to our original scenario, you communicate your goal "buy house" to ToDoGo . Using a corpus of goal knowledge, ToDoGo suggests sub-goals that may be relevant , looks up surrounding points of interests and plots them on a local map.

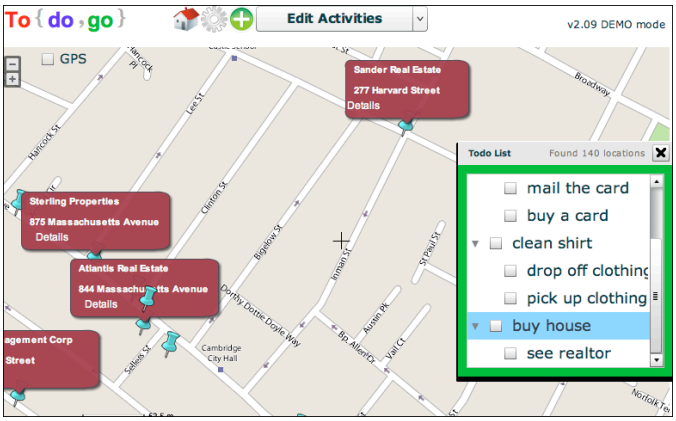

Figure 4. ToDoGo uses default knowledge about events, knowing that "see relator" takes place at real estate office, it queries local points of interests by category and presents them on a map

CONCLUSION

This paper argues for building programs that are sensitive to a user's goals. We first analyzed a corpus of plans and goals. Then we showed how we used natural language processing techniques to transform collaboratively collected semi-structured text into a goal graph to be used for plan recognition. Finally we showed how these could be useful in a mobile to-do list application.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their helpful comments and suggestions on the first draft, and Josh Petersen from 43things.com.

REFERENCES

1. M. G. Armentano and A. Amandi. Plan recognition for interface agents. *Artificial Intelligence Review*, 28(2):131–162, Aug 2007.
2. C. Bogart, M. Burnett, A. Cypher, and C. Scaffidi. End-user programming in the wild: A field study of coscripter scripts. *IEEE Symposium on Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008*, pages 39–46, 2008.
3. K. Conley and J. Carpenter. Towel: Towards an intelligent to-do list. In *Proceedings of the AAAI Spring Symposium on Interaction Challenges for Artificial Assistants*, 2007.
4. S. Dongen. Graph clustering by flow simulation. *Ph.D. Thesis*, Jan 2000.
5. Y. Gil and V. Ratnakar. Automating to-do lists for users: Interpretation of to-dos for selecting and tasking agents. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, page 7, Apr 2008.
6. R. Gupta and K. Hennacy. Commonsense reasoning about task instructions. *AAAI-05 Workshop on modular construction of human-like intelligence. Pittsburgh, PA, July*, 10:05–08, 2005.
7. H. Lieberman and J. Espinosa. A goal-oriented interface to consumer electronics using planning and commonsense reasoning. *Knowledge-Based Systems*, 20(6):592–606, 2007.
8. M. Strohmaier, M. Kröll, and C. Körner. Intentional query suggestion: making user goals more explicit during search. *Proceedings of the 2009 workshop on Web Search Click Data*, pages 68–74, 2009.
9. S. Stumpf, X. Bao, A. Dragunov, T. Dietterich, J. Herlocker, K. Johnsrude, L. Li, and J. Shen. Predicting user tasks: I know what you're doing. In *20th National Conference on Artificial Intelligence (AAAI-05), Workshop on Human Comprehensible Machine Learning*, 2005.
10. P. Whitmore. *The Maieutics of Goal Articulation: Motivating the choices of the highest import*. PhD thesis, Stanford University, March 2000.