AFRL-AFOSR-JP-TR-2016-0049

Understanding how to build long-lived learning collaborators

Kenneth Forbus
NORTHWESTERN UNIVERSITY

03/22/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ IOA
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
23-06-2016	Final	01 Sep 2010 – 1 Mar 2016

4. TITLE AND SUBTITLE

Understanding how to build long-lived learning collaborators

5a. CONTRACT NUMBER
FA2386-10-1-4128

5b. GRANT NUMBER
Grant AOARD-104128

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Prof. Kenneth Forbus

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northwestern University
Ford 3-320, 2133 Sheridan Road
Evanston, IL 60208
United States

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD-104128

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A: Approved for public release. Distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This project conducted basic research aimed at creating software systems that can collaborate naturally with people over extended periods of time. This involved investigating how to make a habitable combination of natural language and sketch understanding that supports interactive learning of complex domains, including giving advice, learning by reading, and learning by demonstration. We developed the notion of type-level qualitative representations that significantly improve expressive power and compactness, both of which improve reasoning and learning, while also providing a simpler path for learning qualitative models from natural language. We also made progress on using qualitative representations for strategic thinking, where continuous processes and causal knowledge about quantities provide a higher level of description, within which specific planning goals arise. This includes expressing goals in terms of maximizing/minimizing quantities, recognizing and analyzing tradeoffs, and encoding broader-scale strategies in terms of continuous processes. We explored how to extend the Companion cognitive architecture to incorporate more self-learning, including automatic detection of near-miss examples to improve discrimination in learning, and dynamic encoding strategies to improve visual encoding for learning via analogical generalization. We showed that spatial concepts can be learned via analogical generalization. Moreover, we explored learning sketched concepts via analogy at a larger scale than ever before, using a 20,000 sketch corpus to examine the tradeoffs involved in visual representation and analogical generalization.

15. SUBJECT TERMS
Artificial Intelligence, Computer Science, Autonomous Agents and Multi-Agent Systems, Machine Learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Brian Lutz, Lt Col, USAF
U	U	U	SAR	25	**19b. TELEPHONE NUMBER** *(Include area code)* +81-3-5410-4409

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Final Report for AOARD Grant FA2386-10-1-4128

Towards Long-Lived Learning Software Collaborators

March 16, 2016

Name of Principal Investigators (PI and Co-PIs): Kenneth D. Forbus, Thomas R. Hinrichs

- e-mail address : forbus@northwestern.edu
- Institution : Northwestern University
- Mailing Address : 2133 Sheridan Road, Evanston, IL, 60208
- Phone : 847.491.7699
- Fax : 847.491.4455

Period of Performance: Sept 29 2010-March 28 2016

Abstract

This project conducted basic research aimed at creating software systems that can collaborate naturally with people over extended periods of time. This involved investigating how to make a habitable combination of natural language and sketch understanding that supports interactive learning of complex domains, including giving advice, learning by reading, and learning by demonstration. We developed the notion of *type-level qualitative representations* that significantly improve expressive power and compactness, both of which improve reasoning and learning, while also providing a simpler path for learning qualitative models from natural language. We also made progress on using *qualitative representations for strategic thinking*, where continuous processes and causal knowledge about quantities provide a higher level of description, within which specific planning goals arise. This includes expressing goals in terms of maximizing/minimizing quantities, recognizing and analyzing tradeoffs, and encoding broader-scale strategies in terms of continuous processes. We explored how to extend the Companion cognitive architecture to incorporate more self-learning, including *automatic detection of near-miss examples to improve discrimination in learning*, and *dynamic encoding strategies to improve visual encoding for learning* via analogical generalization. We showed that *spatial concepts can be learned via analogical generalization*. Moreover, we explored learning sketched concepts via analogy at a larger scale than ever before, using a 20,000 sketch corpus to examine the tradeoffs involved in visual representation and analogical generalization.

1. Introduction

The specific aims of this research have been to explore how to create *software social organisms* (Forbus, in press) that can collaborate with people using natural modalities, working as apprentices to build competence and trust, while maintaining and adapting themselves over time. This is in stark contrast to today's model of intelligent system as tool or specialized, single-purpose system. For example, both Watson and AlphaGo are stunning achievements in terms of their capabilities on specific tasks (factoid question-answering and playing Go, respectively). However, neither system can do what the other does. And both systems were the result of large teams of experts, manually tweaking and changing their internals, retraining, and modifying as needed until the required standard of performance – as measured by the team of experts, not the system itself – was met. For systems that must operate in rapidly changing tasks and environments, and learn new tasks on the fly, such large support staff and manual fiddling with their internal structure does not scale. Instead, we are trying to learn how to create AI systems that are organisms, i.e. capable of autonomy, maintaining themselves and improving themselves, with interaction only in terms of the natural interaction modalities that we would use with a human collaborator. This places a larger burden on communication, i.e. being able to communicate complex ideas through language that otherwise might take hundreds or even thousands of examples for a system to learn on its own.

Creating software social organisms is an extraordinarily difficult problem, and while we made important progress, much work lies ahead. In fact, Forbus (in press) argues that human-level AI will simply be sufficiently smart software social organisms, which indicates how ambitious the extreme version of this goal is. Importantly, though, we believe that there will be multiple intermediate points that lead to useful applications along the way.

Our specific objectives, and major results concerning them, were:

1. Scale up analogical processing to enable learning substantial bodies of knowledge. We showed that spatial concepts from a strategy game could be learned via automatically constructed representations generated from sketching over a game's map. We further showed that analogical learning is promising for learning spatial concepts at a larger scale, using an independently developed corpus of 20,000 sketches. This is radically larger, in both number of examples and complexity of examples, than have ever been tackled with analogical learning before. While we have been only partly successful on that corpus to date, this has led to several important insights about encoding at larger scales that we believe are domain-general, as well as leading us to broaden out our visual representation vocabulary in ways that provide stepping stones to learning from camera inputs in the future.
2. Investigate how to make a habitable combination of natural language and sketch understanding that supports interactive learning of complex domains. This has involved developing new representations for expressing the dynamics of complex domains. We view our development of *type-level qualitative representations* as a breakthrough, in that they enable constructing qualitative models for larger problems, including tracking what is happening in worlds that are too complex to record everything, reasoning about larger-scale systems, and simplifing learning qualitative models from natural language. We have also shown that qualitative models can be used to express goals concerning continuous properties (e.g. maximizing income, territorial expansion, minimizing expenses), detecting and analyzing tradeoffs, and expressing larger-scale strategic ideas (e.g. first expand to control the most territory that you can, and smoothly switch over to developing your economy, is a common strategy in many games).
3. Extend our Companion cognitive architecture to incorporate self-learning, including encoding strategies and memory organization policies. The lack of a cluster for several years left this aspect on back-burner, although we did make progress on dynamic encoding strategies for sketches. Moreover, we built out the architecture itself with *worker agents* for offline learning, and developed experiment plans where a Companion can basically organize its own workload, given a number of nodes to work with, to carry out larger-scale experiments.

We view this research as having several significant benefits for the Department of Defense. First, software collaborators would be a fundamental advance in intelligent systems, providing a substantial enhancement in autonomy and reducing the need for expert technical support staff to adapt or re-engineer a system for changing environments. Second, incorporating human-like reasoning and learning, via qualitative representations and analogical reasoning, as described below, should make software more effective and trusted collaborators. Their result should be close enough to our own ways of reasoning that we will understand (and thus trust) the concepts that they learn and the explanations that they give for their advice or actions. However, we may be able to deliberately engineer out some known human weaknesses (e.g. confirmation bias, working memory limitations) to provide complementary strengths. Third, the apprenticeship model for training intelligent systems should, again, reduce the need for technical support staff, and build trust in intelligent systems via experience, working alongside them.

2. Approach

Our research used the Companion cognitive architecture (Forbus et al., 2009), which is being built with the goal of creating software social organisms. The hypotheses that we are exploring in this architecture include

- Rich, domain-independent relational representations are essential for rapid learning and flexible performance. Of special importance are qualitative representations, which are used to symbolically represent and reason about continuous quantities, processes, shape, and space.
- Analogical reasoning and learning is central in human cognition.
- Natural language and sketching are important modalities for communicating naturally with collaborators and trainers.
- An important part of an organism's mental life is formulating new learning goals and adapting its processes to perform more effectively and efficiently.

The rest of this section discusses the ideas concerning relational representations, especially qualitative representations, analogy, and the testbed we are using to explore these ideas.

2.1. Rich, domain-independent relational representations.

There has been a surprising de-evolution of representational sophistication in AI, especially in machine learning. The mathematical tractability of feature vectors, and computational support through GPUs, has seduced many researchers into exploring vector representations even in situations where there is strong evidence that human cognition involves relational representations. We believe that this is the reason why, for example, deep learning systems require massive amounts of data to operate, far more than people ever see in a lifetime. We think an important job of AI is to explicate what representations are needed to carry out robust, flexible intelligence. This includes, for example, being able to represent and reason about causality, lines of argumentation, evidence, planning, and other constructs which lie outside the expressiveness that vectors can provide. In our work we use the Cyc knowledge base contents as a starting point, specifically ResearchCyc, since it is freely available for research and incorporates many more axioms and mappings between KB concept and natural language than any other knowledge base available. We do end up extending it as needed, and in some areas substantially. The single biggest area of expansion is in qualitative representations, which we view as so important that we focus on them next.

2.2. Qualitative Representations

Many aspects of the physical, social, and mental realms are continuous. Physical examples include quantities like temperature, pressure, and land area. Social examples include degree of blame, affinity, and depth of friendship. Mental examples include difficulty of a task, capability for solving particular kinds of problems, and available mental energy. While numerical values are sometimes available for some of these quantities (e.g., for physical quantities), much of what we know about them is more abstract. We may be able to provide estimates of relative blame in a situation, for example, while not being able to confidently give a numerical value for responsibility to each those involved. We know causal relationships involving such abstract quantities, e.g. we know that the more tired we are, the harder a problem will seem, even if we cannot even specify units for these quantities in any sensible way. Thus something beyond traditional mathematics is needed to capture such human concepts. Qualitative representations were developed for exactly this purpose. A number of qualitative representations for quantity have been developed, for example, including sign values and ordinal relations, which capture important properties of reasoning about continuous parameters (e.g. continuity, partial information) with much less information. We use Qualitative Process (QP) theory (Forbus, 1984), which also provides a causal, qualitative mathematics and a notion of continuous process that serves as ontological mechanism underlying causality. QP theory has been used to model a wide range of phenomena, both physical and social (Forbus, in preparation). In this project we extend both the ideas of QP theory to provide type-level representations, and show

how they can be used in strategic thinking.

Another hypothesis that we are building on is that *QP theory provides an inferential semantics for natural language.* Decades of research, by us and by others, suggests that QP theory is sufficient to capture a wide range of novice and expert reasoning about continuous dynamical systems, including simple physics and chemistry, various engineering domains, and even aspects of social phenomena (e.g. blame assignment, moral decision-making). Moreover, our prior work, and our current work on a parallel project, have shown that constituents of QP theory can be identified with particular syntactic patterns, and mapped consistently onto FrameNet semantics (Kuehne & Forbus, 2004; McFate et al. 2014; McFate & Forbus, 2015). For example, the QP theory notion of an indirect influence (aka qualitative proportionality) can be captured by a QP Frame whose core frame elements link two quantity frames, e.g.

- Core Frame elements: Constrained, Constrainer, Sign
- Ex: As the temperature of the steam rises, the pressure in the boiler increases.

We built on this hypothesis in this project, showing how QP knowledge could be extracted from natural language advice and from reading the Freeciv manual.

2.3. Analogy

Gentner's (1983) structure-mapping theory proposed that analogy consists of finding correspondences between structured, relational representations. These correspondences are also used in finding differences – candidate inferences that suggest ways to project information from one description to another. Gentner (2003) further proposes, and we concur, that analogy is a core operation in human cognition more broadly. There is evidence that the same laws of structure-mapping govern comparisons involved in high-level vision, similarity judgments, reasoning, problem-solving, and conceptual change. Viewed from an AI perspective, this model of analogy has several important advantages. First, similarity is not an arbitrary term that can be defined any way one likes – people behave concerning similarity judgments in ways predicted by structure-mapping theory. This means that similarity models based on feature vectors, for example, will tend to have problems when compared closely with human performance, making their results less likely to be trusted. By contrast, structure-mapping models have been used to both successfully explain existing phenomena and to successfully predict new phenomena (Forbus et al., in press). Second, they have been successfully used in AI performance systems, including some implemented by others (e.g. IBM's Watson used a specialized version of SME as one of its methods of checking candidate answers). Each model corresponds to a specific process involved in analogy:

- SME (Falkenhainer et al., 1989; Forbus et al., in press) performs analogical matching. It produces one or more mappings, each of which consists of correspondences indicating what statements or entities in one description align with the other, candidate inferences that indicate how non-mapped information can be projected from one description to the other, and a numerical similarity score that indicates how well the two descriptions align.
- MAC/FAC (Forbus et al., 1995) performs analogical retrieval. Given a structured representation as the probe, it first computes a simple vector version to compare, conceptually in parallel, with vectors corresponding to all of the cases stored in a case library. Up to three cases are returned, and their structured representations are then compared with the probe via SME, again in parallel. The case with the best mapping, or another one or two if very close in score, is returned as the retrieved case.
- SAGE performs analogical generalization. Each concept to be learned by SAGE has a generalization context that is a case library containing both generalizations and unassimilated examples. When a new example is added, MAC/FAC is used to retrieve the closest generalization or example. If sufficiently close, the example is assimilated into the generalization, if that is what was retrieved, or the overlap between the new and old example are used to create a new generalization, otherwise. Non-identical entities are replaced with *generalized entities*, which are still concrete but unidentified constants – SAGE does not introduce logical variables. Generalizations are probabilistic, in that frequency information is tracked for each aligned statement in a generalization. Low-frequency information does not influence matches, and eventually "wears away" over time. SAGE can handle disjunctive concepts, via maintaining multiple generalizations, and handle outliers, via

storing unassimilated examples.

Since these models rely upon each other, we consider them an *analogy stack*, the start of a new technology for analogical reasoning and learning grounded solidly in cognitive science. Understanding the properties of analogical processing, and how to build systems that reason and learn via analogy at scale, is one of the central research challenges we are tackling. This fits synergistically with our focus on rich, relational representations – such representations are perfect grist for analogy, and analogical processing can provide flexible ways to use such representations, to complement traditional first-principles logical and abductive inference.

2.4. Freeciv: Strategy Game as Testbed

For exploring interactive and offline learning, it is useful to have a complex simulated world, where the complexities have reasonable analogs to real-life systems and situations. We used the open-source strategy game, Freeciv [1] as our testbed. The Civilization line of games enable players to control an entire civilization, from stone-age routes to the space age. Initially the world map is unknown, requiring exploration both on land and sea. Players create cities and improvements, transportation networks, and establish trade and diplomatic relationships with other players (which may be people or bots). There is, naturally, warfare, requiring that players master military tactics, handle guns/butter tradeoffs, and do strategic planning.

From an AI research perspective, Freeciv has several important advantages. First, playing it well requires mastering a large set of spatial concepts and problems, including types of terrain, city placement, and construction of transportation networks. It includes complex dynamics, e.g. units and improvements take time and resources to build. Military technologies change considerably over the span of history, requiring adaptation of tactics and concepts (e.g. from archers and chariots to aircraft and nuclear weapons).

We note that others have used strategy games like Freeciv, albeit with different goals in mind. For example, (Branavan et al., 2012) worked on learning using Freeciv as a domain. Their approach was to "read" the Civilization 2 manual to find mappings between words and game features in their simulation. This was used to tune a heuristic evaluation function, rather than to construct a model of the game. They restricted their model to a much smaller map than is standard, and halted the game after 75 turns, which factors out most of the complex dynamics of the game. While their system was able to perform well on this scaled down game, it required many trials to learn the game, required using the game engine to do lookahead search, and it cannot explain its models nor the reasons for its actions. By contrast, we used, and plan to continue using, Freeciv as a platform for learning more human-like representations, reasoning, and learning strategies.

3. Results and Discussion

We made progress on four fronts in this project: qualitative modeling, natural interaction, interactive learning, and self-directed learning. We discuss each in turn.

3.1. Qualitative Modeling

We have two ideas that we think are breakthroughs in this area, *type-level qualitative representations*, and *qualitative models for strategic thinking*. We discuss each in turn. The extraction of qualitative models from natural language is discussed under Natural Interaction.

[1] Freeciv.org

Type-level qualitative representations: Traditionally, qualitative reasoning has been performed by instantiating logically quantified *model fragments* from domain theories to produce propositional representations of all of the causal and constraining relationships between the entities of a situation. This works well in many scientific and engineering domains, where the starting point of an analysis is the equivalent of a blueprint of a system, something that stays fixed through an analysis. But in dynamic worlds (like strategy games and our own world), objects come and go, and are created and destroyed by us and by others. Reasoning about the properties of things that do not yet exist is a necessity. The size of these worlds means that explicitly modeling every relationship between every entity does not scale well. Consequently, we started exploring higher-order qualitative reasoning, using type-level qualitative representations.

The key ideas of our approach are the following:

1. Avoid propositionalization whenever possible. By constructing explicit type-level models, we can reuse the same model over different parts of a situation over and over again as needed.
2. Use type-level predicates as an expressive formalism for planning, learning, and natural language semantics.
3. Integrate discrete actions more fully with influences and continuous processes. Previous integrations of actions and processes (e.g. Forbus, 1989; Drabble, 1990) used non-durative actions.
4. Provide underspecified causal representations suitable for learning and language understanding.

Here is an example of how type-level qualitative predicates are defined in terms of instance-level predicates:

```
(qprop+TypeType <constrained> <constrainer> <constrained-type>
                <constrainer-type> <reln>)
 ≡ (forall ?x
    (forall ?y
      (implies (and (<constrained-type> ?x)(<constrainer-type> ?y)(<reln> ?y ?x))
               (qprop+ (<constrained> ?x)(<constrainer> ?y)))))
```

How much of a savings can type-level representations provide? The table below shows the number of type-level influences versus propositional influences in a typical Freeciv game after 75 turns. There is a factor of 20 fewer type-level inferences, and the effect will be even stronger deeper into the game.

Type-level influences		Propositional Influences	
i+TypeType	4	i+	36
i-TypeType	0	i-	0
qprop+TypeType	10	qprop	210
qprop-TypeType	7	qprop-	179
Total:	21	Total:	425

Connecting actions to qualitative models: Using a qualitative model to drive behavior requires explicitly representing the effects of primitive actions on continuous quantity fluents. We introduced new vocabulary to express these instantaneous positive or negative effects, for example:

```
(actionPositivelyAffectsQuantity
  ((MeasurableQuantityFn cardinalityOf) FreeCiv-City)
  (doBuildCity ?settler21 ?city21))
```

which says that the `doBuildCity` action increases the number of cities. With this sort of information, an agent can infer how to manipulate independent variables to transitively influence goal quantities. Moreover, because these connections are simple declarative facts, they are far easier to learn from observation than complex plans. More in-depth discussion of these ideas can be found in

(Hinrichs & Forbus, 2012b).

Qualitative models for strategic thinking: Complex and dynamic problems cannot be planned in detail from an initial state to some ultimate goal state. Rather than focus on learning an inscrutable evaluation function to guide an agent through such a state space, we instead explored representations and techniques for planning with explicit strategies. Strategies have the benefit of being communicable and instructable, providing concise explanations of motivations, and being general, or reusable, across domains.

Our key contributions in strategic thinking are:

1. An enumeration and analysis of types of goal tradeoffs that propose and constrain strategies,
2. A formulation of strategy as the relative prioritization, decomposition, and scheduling of competing goals, and
3. The application of qualitative process modeling to the activation of strategies and prioritization of goals.

We found that the type-level representation of goals allowed for a new kind of goal decomposition. Because type-level goals are implicitly quantified over sets of entities, such as cities or locations, it is possible to decompose a goal by subdividing its scope. This lets different entities have different or shifting priorities for the same goal type, yielding intuitive kinds of strategies, such as functionally specializing entities.

The conception of strategies as policies for resolving goal tradeoffs, rather than as search control heuristics, led to the treatment of goal activation (or priority) as a kind of quantity amenable to being influenced through processes. This is an economy of mechanism that treats meta reasoning much the same as object-level reasoning. Moreover, because processes are quantity-conditioned on the qualitative state of the domain environment (in this case, a game state), strategies themselves are dynamically activated and deactivated as the environment changes. For example, here is a type-level process representation of a strategy for shifting the priority of goals based on the game state being in the ExpansionPhase.:

```
(isa BuildGrow StrategyType)
;; roles and types of participants:
(participantType BuildGrow doneBy Player)
(participantType BuildGrow initialGoal Goal)
(participantType BuildGrow deferredGoal Goal)
(associatedRoleList BuildGrow (TheList doneBy initialGoal))
;; relations between participants:
(participantConstraint BuildGrow
  (goalTradeoff initialGoal deferredGoal PartialProgressiveTradeoff))
(participantConstraint BuildGrow
  (activeMF (MFInstanceFn ExpansionPhase (TheList doneBy))))
;; influences on goal activation:
(consequenceOf-TypeType BuildGrow
  (c+ ((MeasurableQuantityFn goalActivation) initialGoal)
     (Percent 100)))
(consequenceOf-TypeType BuildGrow
  (c+ ((MeasurableQuantityFn goalActivation) deferredGoal)
     (Percent 0)))
```

Such a strategy can be used to resolve a progressive tradeoff between expanding a Freeciv civilization by building cities and then focusing on growing the individual cities. The qualitative representation of strategies is described in more detail in Hinrichs & Forbus (2015).

Qualitative process representations serve another objective in our research: to support long-term learning by breaking the learning problem down to small pieces that can be learned and communicated independently. We have pursued this in the context of learning by demonstration, where the agent focuses on learning a qualitative model of the game initially, rather than on playing

well per se.

Learning at this level lets an agent acquire independent facts piecemeal that combine together to form a model:

- *Qualitative influences* are relations between quantities in the game. The hypothesis space between any two game quantities is sufficiently small that it usually does not take many trials to resolve their qualitative relation.
- *Parametric decision points* are parameters that are revised periodically, as triggered by events or conditions in the game, for example, setting the production queue whenever a city is first built. Learning the event triggers for decision points is a small association that is critical for learning to play the game.
- *Events* are explicit changes in state. To the extent that the environment (game) signals explicit events, these often correspond to limit points in qualitative processes. This expectation reduces the search for activating conditions on processes.
- *Domain-specific operationalizations* are task expansions that can be learned by demonstration. Most commonly, these are hierarchical task network methods for achieving the preconditions of a primitive action. These are learned, like macros, by searching the execution trace backwards from the executed primitive, applying relevance heuristics, and lifting the immediately prior action sequence to make a new indexed plan. This sort of learning is limited to tasks that are oriented around very concrete primitive actions. Other problems are not so easily addressed by acquiring simple facts or task definitions.

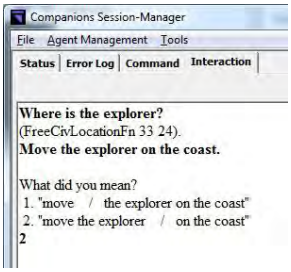
Consider, for example, the problem of defense. In experiments involving resource allocation strategies, most of the trials performed as predicted, but some did very poorly. In the few that did poorly, we realized that the enemy civilizations were particularly aggressive, and because Companions weren't defending themselves, their civilizations were wiped out before turn 100, when measurements were made. Defense is an interesting example of how strategic thinking deviates from traditional AI notions of planning. Defense is not an action. No single primitive or durative action suffices to define defense. Defense isn't a state to achieve, it's more about preventing bad events and states. If we think in terms of defense as reducing vulnerability, a quantity associated with anything that can be attacked, then type-level model fragments can be defined to express causal constraints on vulnerability. For instance, here is a type-level model fragment that expresses the notion of Defending, with comments explaining the meaning of these statements in English:

```
(isa Defending ModelFragmentType)
;; Defending is a model fragment, i.e. something used to assemble models of specific situations.
(genls Defending ProtectingSomething)
;; Defending is a subclass of the Cyc concept ProtectingSomething.
(participantType Defending protector-Agentive FreeCiv-MilitaryUnit)
(participantType Defending objectProtected FreeCiv-Actor)
;; The kinds of participants are military units and actors within the FreeCiv portion of the ontology we
;; developed.  The 2nd arguments are the relationships that indicate which participant is which.
(associatedRoleList Defending (TheList protector-Agentive objectProtected))
;; This indicates the roles above are the complete set of participants.
(participantConstraint Defending
  (and (objectFoundInLocation protector-Agentive objectProtected)
       (different protector-Agentive objectProtected)))
;; The protector must be in the same location as the protected, and the defender is different from that defended.
(consequenceOf-TypeType Defending
  (qprop- ((QPQuantityFn Vulnerability) objectProtected)
          (DefensiveStrengthFn protector-Agentive FreeCiv-MilitaryUnit)))
;; Defending causes the vulnerability of the defended to be lower, as a function of the defensive strength of the
;; defender.  The function QPQuantityFn coerces a Cyc quantity type to a function whose range are fluents of
;; that quantity type.
```

Similar model fragments are used to capture the effects of city walls on reducing the vulnerability of a city, and the effect of treaties on reducing `HostilityLevel`, which again reduces vulnerability. Incorporating these model fragments into the Companion's qualitative model of the domain enabled it

to build defending units and walls, and accept treaties, when appropriate, leading to survival in the scenarios mentioned earlier. This is of course just a starting point: Strategies are hierarchical, e.g. when deciding to conquer the continent its civilization started on, a Companion will take actions that will increase hostility levels, something the qualitative model should be able to predict, and include in its plans actions that will bolster its defenses in advance.

3.2. **Natural Interaction**



Multimodal deictic reference. Teaching an AI system about a simulated world is greatly facilitated by being able to refer to objects in a shared world. Mappings between game-specific meanings of words (e.g. "explorer", "city") were added to the knowledge base manually, since we wanted to focus on learning higher-level concepts. One of the first ways we used language in this project was in tasking Companions to take actions in the game world. Early on, we extended dialogue management in Companions to provide feedback when it could not completely disambiguate something on its own, as shown on the right. As the abductive reasoning in Companions has been improved, most disambiguation now happens automatically.

Multimodal references are also needed to identify regions corresponding to spatial concepts. For this we introduced the idea of an *interaction glyph*, something drawn in a CogSketch layer overlaid on the simulation's map. CogSketch interprets any selected glyph as an interaction glyph, and interprets linguistic references to "this" or "that" when such a glyph is available as referring to either an entity within that glyph, or the region itself, depending on context. (Learning spatial concepts, summarized below, used this capability.)

An important kind of tasking is directing a Companion's attention. In teaching a Companion aspects of combat by demonstration, both successful and unsuccessful attacks are demonstrated. A simple "That was bad" causes a Companion to look for events in the recent past (here, one of its units being destroyed), which causes it to both learn a new goal (i.e. preventing such events), and a rule for detecting when such goals fail, which can in turn lead to posting new learning goals.

```
Interpretations produced for Sentence-3595165610-2191:
 "That was bad." =
 ((hasEvaluativeQuantity that2197 (MediumToVeryHighAmountFn Badness-Generic)))

Learned new goal: (goalName (GoalFn 35) (PreventFn
                                    (negativeOutcomeForActor
                                     (GenericInstanceFn
                                      (CollectionSubsetFn DefenderDestroyedEvent
                                       (TheSetOf ?evt-var
                                        (and (lookupOnly (isa ?evt-var DefenderDestroyedEvent))
                                             (doneBy ?evt-var ?var1) (maleficiary ?evt-var ?var1)
                                             (unitOwner ?var1 (IndexicalFn currentPlayer))))))
                                     (IndexicalFn currentPlayer))))

Learned new rule: ((<==
                   (eventTriggersResponse ?evt-var (respondToGoalFailure ?mt ?role% ?evt-var (GoalFn 35)))
                   (executionContext ?mt) (currentPlayer ?role%)
                   (ist-Information ?mt (lookupOnly (isa ?evt-var DefenderDestroyedEvent)))
                   (ist-Information ?mt (doneBy ?evt-var ?var1))
                   (ist-Information ?mt (maleficiary ?evt-var ?var1))
                   (ist-Information ?mt (unitOwner ?var1 ?role%)))
User statement registered: (hasEvaluativeQuantity that2197 (MediumToVeryHighAmountFn Badness-Generic))
```

3.3. **Extracting qualitative models from natural language**

As mentioned above, one of our hypotheses is that qualitative representations play a role in human natural language semantics. We extended the formalism of QP Frames to include type-level frames as well as proposition-level frames. Here is an example of how such frames are generated incrementally from language:

"A city produces food points." leads to the following frame being generated:

```
Process9707
       isa: TypeLevelProcessFrame
       processType: Production-Generic
       referingEvent: produce9510 ;;The discourse variable for "produces"
       participantType: doneBy [Freeciv-City]
                        outputsCreated [(AmountFn Food)]
       consequenceOf: (I+ (AmountFn Food)
                      (RateFn Production-Generic)
```

"As the population of the city increases, the food production of the city increases." leads to adding

```
       (qprop (RateFn Production-Generic) cityPopulation)
```

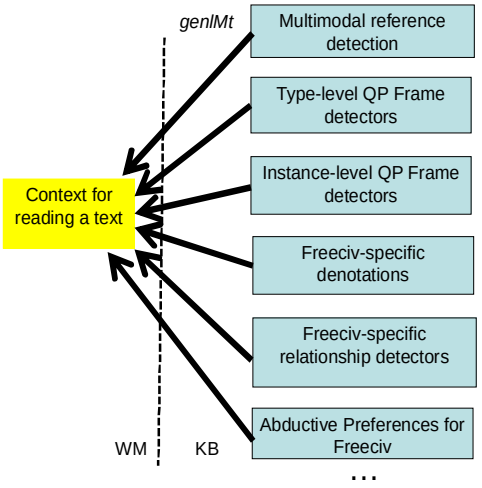
to the frame above.

"A citizen consumes food from the city." produces a second type-level QP frame, describing a destruction event, based on its interpretation for "consumes":

```
Process10083
  isa: TypeLevelProcessFrame
  processType DestructionEvent
  participantType: inputsDestroyed [(AmountFn Food)]
                   from-UnderspecifiedLocation
                                      [Freeciv-City]
                   doneBy [FC-Unit-Citizens]
  consequenceOf: (I- (AmountFn Food)
                  (RateFn DestructionEvent))
  referingEvent: consume9987
```

Type-level qualitative information provides an excellent medium for providing advice during instruction. For example, adding just six pieces of advice, such as "Adding a university in a city increases its science output." can significantly improve a Companion's performance in the early expansion phase of the game (McFate et al. 2014).

Narrative Function provides context-sensitive semantic interpretation. There is an important tradeoff in research on natural language processing between breadth of coverage and depth of understanding. Most statistical NLP, such as bag-of-word systems, word2vec systems, and LSA, focus on breadth at the expense of depth. That is, they can operate efficiently on large corpora, but cannot provide high-precision answers. On the other end of the spectrum, work on deep understanding systems, such as semantic grammars, focus on depth at the expense of breadth. This breadth reduction can be extreme: For example the Geoquery data set[2] is an oft-used testbed, consisting of queries about geography facts from a restricted database. It consists of just six predicates, which would expand into 23 binary predicates when represented in a manner more useful for incremental learning and reasoning. That is far smaller than the number of predicates needed for a reasonable coverage of English semantics even for the game world of Freeciv.



How context drives semantic interpretation in Companions. Microtheories containing narrative function detection rules can be combined dynamically, and domain-specific abductive preferences can be specified declaratively to help guide the semantic interpretation process.

We have been exploring a new approach to achieving both breadth and depth. For breadth, we use a domain-independent grammar which is mapped to

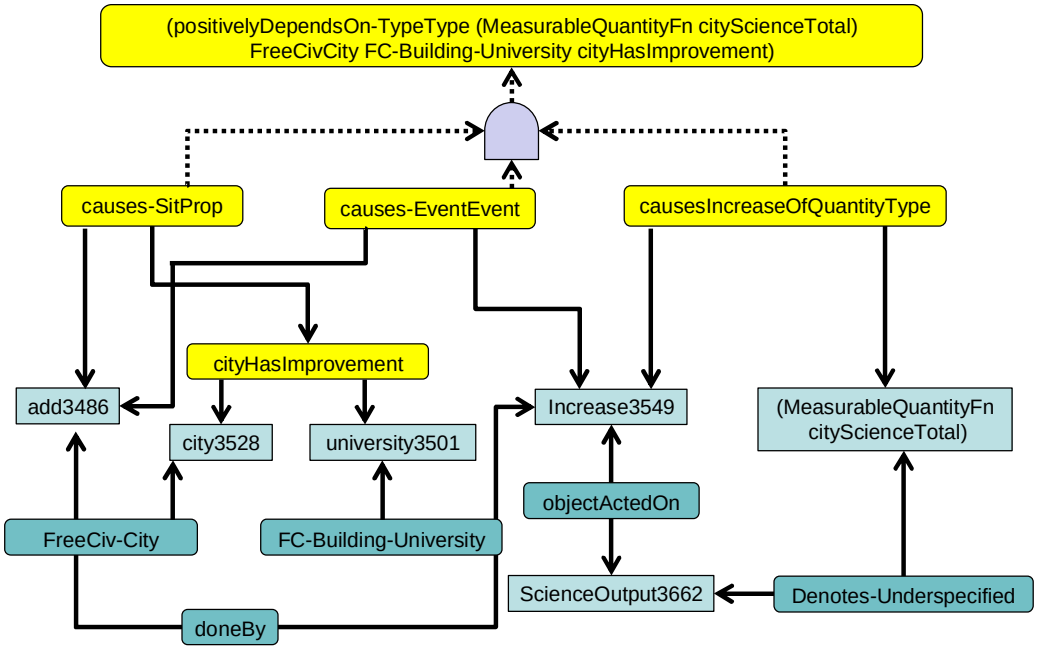
[2] https://www.cs.utexas.edu/users/ml/nldata/geoquery.html

broad, general-purpose representations (i.e. the Cyc ontology and ResearchCyc knowledge base contents), augmented by Discourse Representation Theory (Kamp & Reyle, 1993), which provides a general framework for handling conditionals, logical and numerical quantification, and counterfactuals. For depth, we augment these with *narrative functions* (Tomai & Forbus, 2009), which ascribe purpose to particular sentences. Our initial work with narrative functions focused on traditional narrative structures in fables, e.g. introducing a character. We have found that the same concept can be extended to detect QP content in texts, e.g., introducing a qualitative proportionality (McFate et al., 2014).

Conceptually, we think of narrative function rules as detectors, looking for specific kinds of information. The queries that invoke narrative function rules are dynamically assembled from the current context by retrieving them from the knowledge base, and sorting them based on priorities expressed in the queries itself. The priorities enable interleaving for efficiency, e.g. quantity frames are detected before qualitative proportionality frames and other frames that link them. The queries are run antecedently on the analysis of each sentence as it comes in, and make abductive hypotheses to constrain choices of word sense and parsing choices based on which provide useful interpretations.

This approach enables declarative knowledge to be used to provide top-down guidance, rather than having to generate a new grammar for each new domain, as semantic grammars do. Our grammar and parser have limitations, hence we still rely on simplified English syntax. We view this as an effective interim strategy, since people often use simplified syntax when conversing with children and non-native language speakers, and so it provides a more natural way to communicate with AI systems than using predicate calculus.

For example, to extract qualitative information from a sentence like "Adding a university in a city increases its science output." there are two levels of interpretation, shown (partially) below. The first, shown in green below, is the initial language-level descriptions. Abductive reasoning based on Freeciv preferences is used to interpret "city" as FreeCiv-City, "university" as that type of Freeciv building, and so on. The relationships in yellow are causal relationships extracted by rules reasoning over the NLU output, which in turn provide the basis for extracting the type-level inference at the top of the diagram.



Another way we have been testing the effectiveness of narrative function for extracting QP models from text is that we simplified four chapters from the Freeciv manual, to enable a Companion to read them. The statistics on the simplification and the number of QP frames extracted are shown in the table below. As expected, the economics chapter provides the most QP frames, since it concerns the causal effects of different units and properties on the economics of a civilization. Similarly, the units chapter provides the least quantity frames, since it is mostly a long list of specific properties of units.

Chapter	# Sentences	# Simplified	# QP Frames
Economics	78	105	65
Cities	125	61	10
Combat	67	51	8
Units (subset)	21	31	5

We view these as promising initial results. However, detailed analyses of the results suggest that we are missing some QP frames that we ought to get (i.e. lower than desired recall). Consequently, we are currently experimenting with using semantic information provided by FrameNet (Baker et al., 1998) to improve our QP frame detectors.

Our experiments to date have led us to the conjecture that there are a medium number of commonsense theories that provide general knowledge to support disambiguation. The most common example is typology, i.e. type constraints on predicate arguments or syntactic constraints on verb arguments provide a means of filtering possible semantic choices. QP theory's notion of causality provides another, although we suspect we will need to add explicit temporal patterns to what we use already (QP theory's *encapsulated histories*, which have been used to provide temporal schemas and representations for equations where time is an explicit parameter in other research). Some of these temporal schemas are interlinked, e.g. the idea of a balance, combined with surplus and deficit. In Freeciv, a surplus of food is accumulated and leads ultimately to city growth, whereas a deficit leads to starvation. Similarly, a surplus of production in Freeciv accumulates, leading to a city's current project being finished, while a deficit leads to it being (temporarily at least) abandoned. Identifying a common set of such schemas and formally representing them in a domain-independent way, so that they can be recognized in language and their implications applied to the system's knowledge in specific domains, is something that we plan to pursue. There are other sorts of schemas worth investigating as well, such as mereology (i.e. the study of part/whole relationships). A sentence like "Cities consume food." does not specify where the food comes from. In Freeciv, it comes from the cities themselves, whereas gold to pay for maintenance comes from the civilization of which it is a part. An analysis of FrameNet semantics, from an inferential perspective, should provide a starting point for identifying a set of such theories.

3.4. **Interactive Learning**

An important aspect of social collaboration is learning interactively from a teacher or mentor, either deliberately in the context of explicit instruction, or opportunistically through collaborative problem solving. We have explored the former through *learning by demonstration*, and the latter through simple *learning from advice.*

In learning by demonstration, the learner is mostly passive and observes a teacher's actions and their effects on the task environment. Typically, this is used to teach procedures as a sort of powerful extension of recorder macros. Our emphasis, however, has been learning a *qualitative model* by observing how quantities change in response to actions and how those changes propagate through quantities. To do this requires 1) reconstructing and recording the teacher's action sequence from the accessible percepts (by no means a given in Freeciv), 2) tracking quantity and propositional changes using an add-list/delete-list buffer, 3) explaining pairwise changes in terms of hypothesized influences and effects, 4) pruning those hypotheses in a manner similar to version spaces, and 5) normalizing the resulting influence graph to omit chords (shortcut influence paths). The resulting model provides a way for a Companion to pursue quantitative goals by searching the graph for actions that transitively

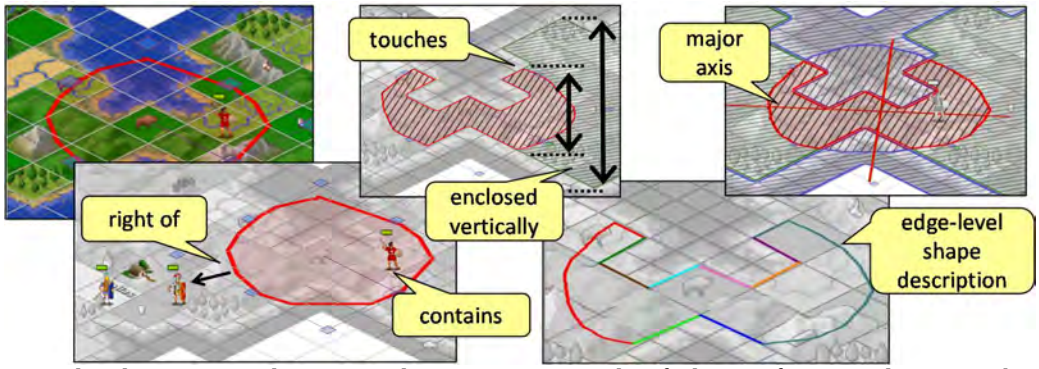
influence the goal quantity in the desired direction.

In addition to learning qualitative influences, a Companion also learns decision points and achievement tasks via demonstration. As described above, a *decision task* is a parametric decision that is periodically revisited in response to events. These are learned through demonstration by detecting parameter setting actions that consistently follow related events (events with the same object being acted on). An *achievement task* lifts and rolls up an observed action sequence that culminates in either making a primitive action legal (i.e., a task for achieving preconditions) or attaining a known active goal proposition. During learning by demonstration, the Companion is not completely passive: it posts explicit learning goals in response to ambiguities and knowledge gaps, and in some cases it tries to reduce ambiguity by directly asking the teacher simple yes/no questions. Learning qualitative models by demonstration is described in more detail in (Hinrichs & Forbus, 2012a).

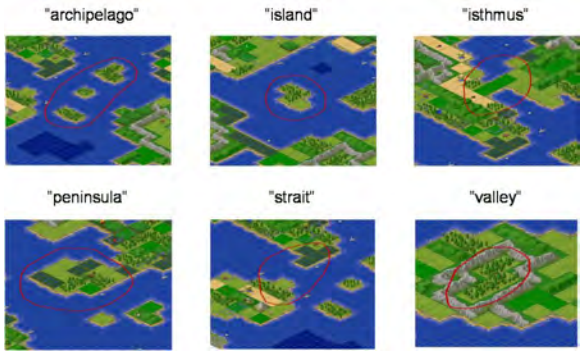
Learning from advice is a kind of learning by being told. Here, the Companion plays a more active role in playing the game and the teacher occasionally provides feedback about decisions and events. The example presented above of "that was bad" feedback leads to the construction of a new performance goal and recognition rule.

3.5. Analogical Learning of Spatial Concepts

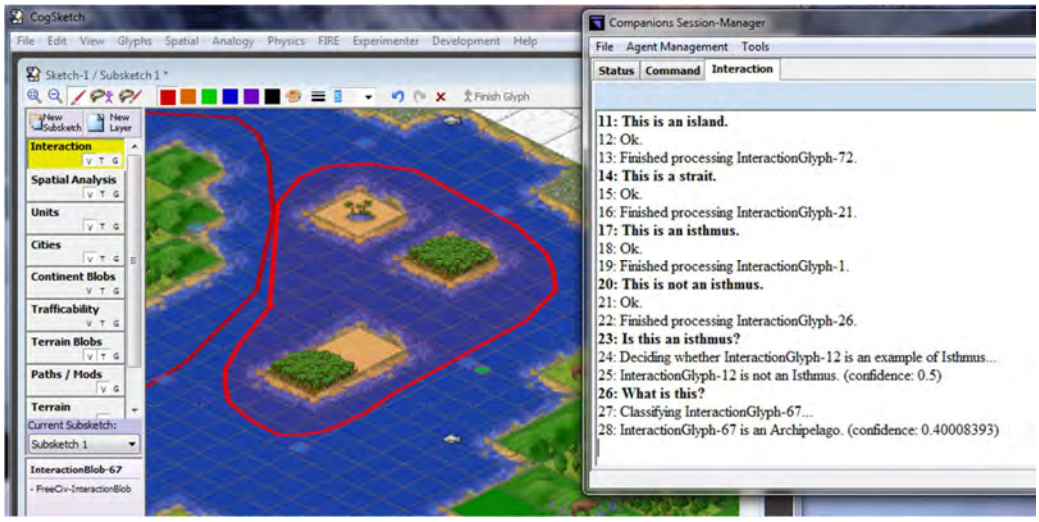
We developed a variety of qualitative spatial representations to support spatial reasoning and learning. Qualitative spatial representations are quantizations of space where one or more properties remain constant, making them useful to distinguish from other places. For example, in Freeciv (and military operations more generally), distinguishing land from water, types of terrain (e.g. forests, deserts), and trafficability (e.g. how well vehicles can traverse it) are useful distinctions. More fine-grained distinctions are often useful, e.g. valleys, islands, peninsula. To perform spatial analyses, we extended our sketch understanding system, CogSketch (Forbus et al., 2011) to enable sketching to be done over a Freeciv map. The underlying spatial model in Freeciv is an array of tiles, but we deliberately abstract away from that for many spatial computations in order to ensure generality of results. Thus CogSketch is used to automatically segment the map into a variety of basic blobs (based on land/sea, terrain type, and trafficability), which can then be analyzed by CogSketch's normal visual operations, as shown below.



To explore learning spatial concepts relevant to maps, we identified a set of six spatial concepts that are of strategic importance, illustrated below.



How should cases be encoded for analogical learning? This is an important, fundamental question. Attention is a scarce resource: No system can pay attention to everything in a sufficiently complex world. Moreover, while larger descriptions may make more distinctions, if those distinctions are not useful, they can distract and even swamp analogical processing. Encoding decisions, for a domain well-understood in advance, could be hand-tuned by experts. But we want Companions to take responsibility for their own encoding processes, hence we experimented with adaptive encoding. Maps were created with 10 examples of each concept, and these examples were entered once, interactively, as shown below.

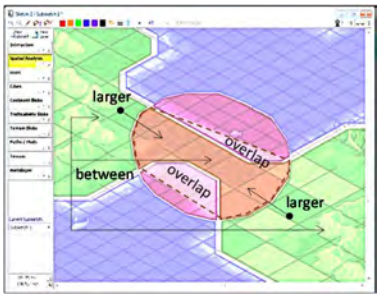


Our first experiment used a library of possible encoding strategies, consisting of intersecting or overlapping the interaction glyph with terrain, trafficability, or continent blobs. Companions started out using three schemes from this library, in parallel, to encode examples. Classification accuracy was used to incrementally update which encoding schemes were used, and unlabeled examples were classified based on voting from answers produced by current encodings. Initial results achieved 67% accuracy with only 8 training examples per concept (Chance = 17%).

While encouraging, our examination of the results suggested that the representations were not capturing enough information to discriminate between these concepts. Consequently, we extended CogSketch to compute medial axis transforms (Blum, 1967), a commonly used technique in computer vision to produce reasonable skeletons for shapes. The medial axis is the set of points that have more than one closest point on the boundary. Each point on the axis induces a radius function, i.e. the distance between a medial axis point and its closest points along the exterior. Changes in radius function induce further qualitative representations of medial axes, e.g. segments where the radius is constant, increasing, or decreasing. Junctions in these axes can be described as sources or sinks,

depending on the direction(s) in which the radius function is narrowing. This representation is called a shock graph (Siddiqi et al. 1999). Medial axis transforms for three terrain types are show below.

Another distinction we explored is the idea of *severed blob encodings*. These are based on the observation that important relationships may be between parts of a blob – specifically, the parts of the blob produced by segmenting it with the interaction glyph - so carving up its interior and exteriors and computing relationships between them could prove important. The image below shows an example of a severed blob representation.



In this case, the piece of the land blob that lies on the interior of the interaction glyph is smaller than the adjacent pieces of land on the exterior, and its convex hull overlaps two separate interior pieces of water blobs.

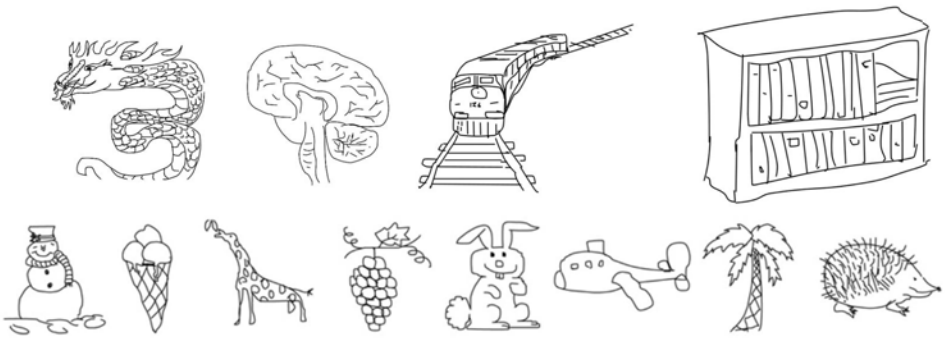
This suggested reorganizing the library of encoding strategies into a 3x3 table, with continent/terrain/and trafficability as one set of choices, and whole blobs, severed blobs, and blob skeletons as the other set of choices. A simple decision-tree can then be used to detect which strategy is likely to be more reasonable, e.g. if an interaction glyph overlaps multiple continent blobs, then a strategy based on them is most appropriate. This provided an improvement of 10% to 77% accuracy, using 10-fold cross validation, one example per fold (McLure & Forbus, 2012). Empirically, only four of the possible strategies were chosen – all three continent strategies and the whole/trafficability strategy. In a domain where maps have a wider range of terrain properties, we expect the terrain-based encoding strategies would see more use.

While a definite improvement, there were several aspects of this approach that suggest further improvements. First, while decision trees can be learned from data, it would take a substantial amount of experimentation to generate enough data to construct such a tree automatically. That does not seem to be a reasonable approach for learning to encode in a rapidly changing world. Another issue is that, while analogical generalization helps provide better transfer than simply analogizing to concrete examples, there is nothing in SAGE that promotes discrimination. Finally, a minor matter, further examination of the kinds of terrain actually appearing in Freeciv maps was that valleys were quite rare, whereas bays are quite common, so we replaced valleys in our data set with 10 examples of bays.

Our next experiment was to extend SAGE to handle near-misses, to improve discrimination. The original concept of a near-miss is due to Winston (1970), where he showed that a teacher-supplied negative example could improve learning of a structured description. This is an intriguing idea, but as formulated, it assumes that the system has just one representation for a concept, that the teacher knows this internal representation, and that the teacher can find/construct an example with a single important difference. By contrast, McLure et al. (2010) figured out that SAGE can be used to find its own near-miss examples, as follows. Suppose the teacher provides an example with a positive label. Suppose further that MAC/FAC is used, not just over the case library for the label, but also for the case libraries with contradictory labels. Then an example or generalization retrieved from a different label must constitute a near-miss for that example. In such cases SAGE then formulates both positive hypotheses (i.e. statements that must be true of any example) and negative hypotheses (i.e. statements that must not be true of any example). In the case of a simple Blocks World arch, for example, the top block must be a block, not a trapezoid, and the two supports cannot directly touch each other.

An experiment in McLure et al. (2015a) demonstrated the effectiveness of near-miss learning on the Freeciv geo-spatial dataset. Using 60 examples and a 10 fold cross-validation test, SAGE with near-misses achieved 77% classification accuracy, whereas SAGE without near misses achieved only 62%. (Again, chance is 17%.) Using near misses showed a 15% improvement in accuracy over setting SAGE to always generalize, and a 24% improvement in accuracy over using pure nearest-neighbor, based on similarity via MAC/FAC (both $p < 0.05$, using a one-tailed paired t-test).

While this is encouraging, the small number of spatial concepts, and their relative simplicity compared to general hand-drawn sketches suggested that we needed to broaden our investigations, to ensure that our techniques are robust and can scale. Consequently, we started experimenting with the Eitz et al. (2012) sketch corpus. This corpus consists of 20,000 sketches, divided into 80 sketches for each of 250 concepts. Below is an example of some of the sketches from this corpus.

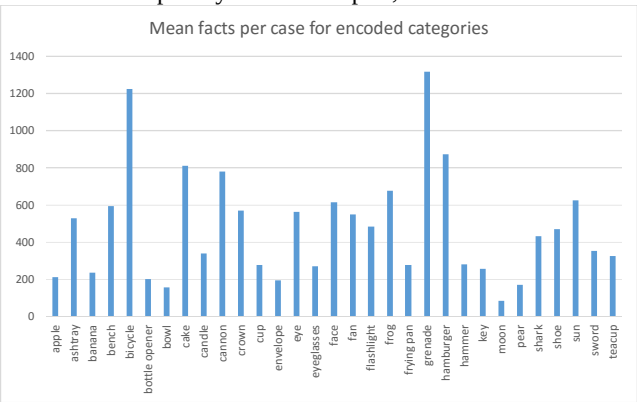


6

This is an extremely challenging corpus. Using pixel-level features and standard machine learning techniques, typical performance on this corpus is 56%. Human performance is only 73%, indicating that these answers are by no means universally agreed upon. (One team claimed a result higher than human agreement, but they used timing information involving strokes rather than just visual properties, so we ignore their results here.) Using an initial set of multi-level encodings (i.e. edges, edge-cycles (McLure et al., 2011), and objects) on a subset of 10 concepts, using 8 fold cross-validation, SAGE was able to perform with equivalent accuracy to the ML baseline results. While we were encouraged, this only worked on visually simple concepts: Visually complex concepts caused heap blow-outs[3].

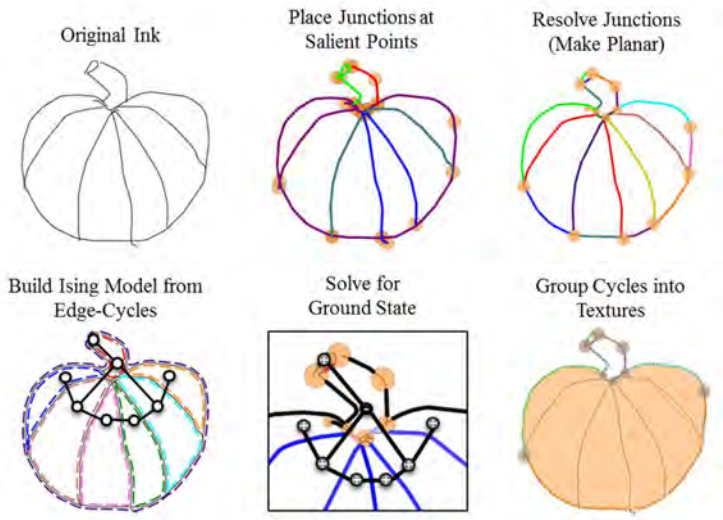
[3] Initial attempts to engineer our way out of heap bloats led to some significant improvements in the Companion architecture, including an agent restart capability that enables worker agents to restart when heaps get too large, which has enabled Companions to run for days at a time on larger experiments.

A close analysis revealed several problems with our initial encoding. First, the redundancy of the multi-level encodings was hurting, not helping. Second, the encodings themselves were massive. Part of this is due to the visual complexity of the examples, as shown below.



Note that while SME operates in polynomial time, the size of the initial match hypothesis forest grows worst-case as the square of the number of facts in the two cases increases. Matching two bicycles, for example, could lead to 1.4 million match hypotheses. That is far beyond what SME was intended to handle.

The message is clear: One constraint on encoding strategies should be that the number of facts that they can produce must be capped. And this size limitation must be systematic, i.e. similar examples should be re-represented in similar ways, if they are going to match. This suggests looking again to human vision, and how it might handle such problems. One technique that people use is detecting textures – repeating patterns of similar things should themselves be abstracted away, into more concise descriptions. For example, much of the complexity of the sketches of bicycles, grenades, and turtles can be captured concisely by representing repeating structure as textures. In computer vision, Ising models are commonly used to group pixels into similar regions. In McLure et al. (2015b), we applied Ising models, but over edge-cycles (closed regions) in the planar network of edges computed via CogSketch, as illustrated below, with an example of a pumpkin.



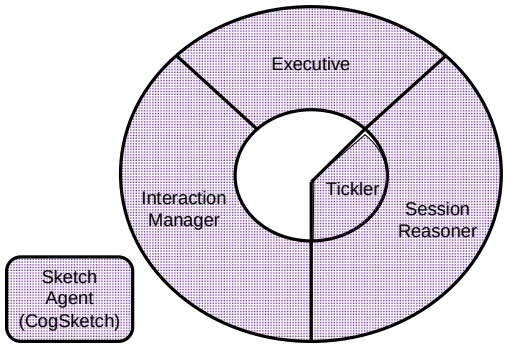
Ising models produce substantial reductions in number of facts (5% to 44%, depending on concept) and number of entities (11% to 58%, depending on concept). We took 10 simple concepts and 10 complex concepts, the latter which couldn't even be encoded previously with our prior methods. On

the simple concepts, we obtained results that were comparable with our prior accuracy, although with the more complex concepts accuracy topped out at 50%. Consequently, we are exploring additional techniques to detect and further compress relational structure in meaningful ways. Our hypothesis is that, with the appropriate visual representation techniques, we can achieve human-level performance on this dataset using visual information alone.

3.6. **Self-guided learning**

The efforts on learning by demonstration and learning by instruction both involve Companions formulating their own learning goals, and using learned qualitative models to formulate goals to drive subsequent behavior. This new model-based approach to formulating strategies leads to formulating goals based on the qualitative model, using type-level representations, and indexicalize them so that they can be reused across a dynamic world (Hinrichs & Forbus, 2013).

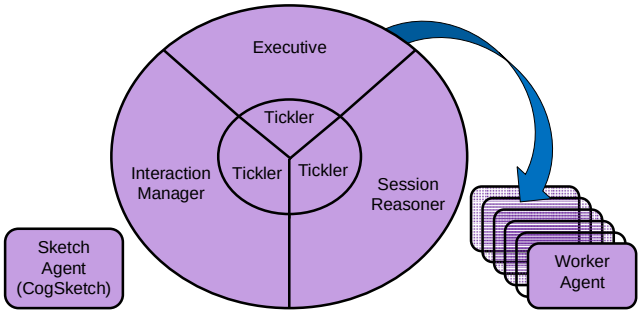
We made several architectural improvements to Companions to support them running their own experiments in the future, a necessary prelude to more extensive experiments on self-guided learning. First, we generalized the long-term memory agents in Companions so that analogical reasoning and learning can be applied more broadly. Here is what the Companions architecture looked like before these extensions:



As an engineering matter, Companions are implemented via a distributed agent architecture, whose agents communicate via the KQML protocol, over sockets (when running across multiple machines) or directly (when running on a single computer). This allows small Companions to be run on a decent laptop, but running across many cluster nodes when required for larger-scale experiments. Here is a breakdown of the roles of the agents:

- The Interaction Manager handles multimodal communication with users. This includes natural language understanding, simple phrase level generation, and interactions via sketching. The sketch agent provides a wrapper around CogSketch to integrate it into the architecture, via messages corresponding to events.
- The Session Reasoner performs domain reasoning. This is useful to split off because it may be engaged in heavy processing while the Interaction Manager is handling user interactions.
- The Tickler provides a long-term memory system for the Session Reasoner, which provides analogical retrieval and generalization services for domain learning.
- The Executive monitors the other agents, and can pause or even reboot them, if they have gone awry.

The diagram below shows the extensions we made to the architecture in the course of this project.



The key changes are:

- Ticklers can now be associated with other types of agents, not just the Session Reasoner. This is intended to support experiments on analogical learning of interaction strategies and analogical learning about a Companion's own internal processing, e.g. how long different tasks take, failure rates of particular approaches, and other internal data.
- Worker agents can be given tasks to be conducted in parallel on other machines, thereby harnessing more parallelism. Worker agents are spawned by the Executive, by monitoring the agendas of the Session Reasoner and Interaction Manager, identifying jobs which can be handed off to a pool of workers.

These changes have enabled us to formulate new plans, e.g. a method for K-fold cross-validation experiments has been developed into a parameterized plan that Companions will be able to use for conducting their own learning experiments. This plan has been amply tested by the experiments described earlier in this report, whose design were specified by us.

4. References

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet project. *Proceedings of the 17th International Conference on Computational Linguistics*, 1, 86-90.

Blum, H. (1967). A transformation for extracting new descriptors of form. *Models for the Perception of Speech and Visual Form*, 362-380.

Branavan, S. R. K., Silver, D., & Barzilay, R. (2012). Learning to win by reading manuals in a Monte-Carlo framework. *Journal of Artificial Intelligence Research*, 661-704.

Drabble, B. (1990). Planning and reasoning with processes. *Expert Planning Systems, 1991., First International Conference on IET*, 191-195.

Eitz, M., Hays, J., and Alexa, M. (2012). How do humans sketch objects? *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 31, 4.

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

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence*, 24, 85-168.

Forbus, K. (1989). Introducing actions into qualitative simulation. *Proceedings of the 11th International Joint Conference on Artificial Intelligence,* 2, 1273-1278.

Forbus, K., Gentner, D., and Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19, 141-205.

Forbus, K., Klenk, M. and Hinrichs, T. (2009). Companion cognitive systems: Design goals and lessons learned so far. *IEEE Intelligent Systems*, 24(4), 36-46.

Forbus, K. D., Usher, J., Lovett, A., Lockwood, K. and Wetzel, J. (2011). CogSketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science,* 3, 648-666. 43.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*, 155-170.

Gentner, D. (2003). Psychology of analogical reasoning. In *Encyclopedia of Cognitive Science* (Vol 1, pp. 106-112). London: Nature Publishing Group.

Kamp, H., & Reyle, U. (2013). *From discourse to logic: Introduction to Model Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory* (Vol. 42). Springer Science & Business Media.

Kuehne, S. and Forbus, K. (2004). Capturing QP-relevant information from natural language text. *Proceedings of the 18th International Qualitative Reasoning Workshop*, Evanston, Illinois, August.

McFate, C.J. and Forbus, K. (2015). Frame Semantics of Continuous Processes. *Proceedings of the 28th International Workshop on Qualitative Reasoning (QR2015)*. Minneapolis, MN.

McLure, M., Friedman, S., & Forbus, K. (2010). Learning concepts from sketches via analogical generalization and near-misses. *Proceedings of the 32nd Annual Conference of the Cognitive Science Society (CogSci)*.

Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., and Zucker, S. W. (1999). Shock graphs and shape matching. *International Journal of Computer Vision*, 30, 1-24.

Tomai, E. and Forbus, K. (2009). EA NLU: Practical Language Understanding for Cognitive Modeling. *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference.* Sanibel Island, Florida.

Winston, P. H. (1970). *Learning structural descriptions from examples* (Doctoral dissertation). MIT, Cambridge, MA.

5. Publications and Significant Collaborations

Peer-reviewed Journals

1. Hinrichs, T. & Forbus, K. (in press) Qualitative Models for Strategic Planning. *Advances in Cognitive Systems*.
2. Forbus, K. (in press) Software Social Organisms: Implications for Measuring AI Progress. *AI Magazine.*

Peer-reviewed Conference Proceedings

1. McLure, M.D., Friedman, S.E., Lovett, A., & Forbus, K. (2011), Edge-cycles: A qualitative sketch representation to support recognition. *Proceedings of the 25th International Workshop on Qualitative Reasoning.* Barcelona, Spain.
2. Hinrichs, T.R. and Forbus, K.D. (2012a), Learning qualitative models by demonstration. *Proceedings of the 26th AAAI Conference on Artificial Intelligence,*207-213.
3. Hinrichs, T.R. and Forbus, K.D. (2012b), Toward higher-order qualitative representations. *Proceedings of the 26th International Workshop on Qualitative Reasoning.* Los Angeles, CA.
4. McLure, M. and Forbus, K.D. (2012), Encoding strategies for learning geographical concepts via analogy. *Proceedings of the 26th International Workshop on Qualitative Reasoning.* Los Angeles, CA.
5. McFate, C., Forbus, K. & Hinrichs, T. (2013), Using narrative function to extract qualitative information from natural language texts: A preliminary report. *Proceedings of the 27th International Workshop on Qualitative Reasoning*. Bremen, Germany.
6. Hinrichs, T. & Forbus, K. (2013), Beyond the rational player: Amortizing type-level goal hierarchies. *Workshop on Goal Reasoning, 2nd Annual Conference on Advances in Cognitive Systems.*
7. McFate, C., Forbus, K., & Hinrichs, T. (2014), Using narrative function to extract qualitative information from natural language texts. *Proceedings of the 28th AAAI Conference on Artificial Intelligence.* Quebec City, Quebec, Canada.
8. McLure, M.D., Friedman S.E. and Forbus, K.D. (2015a). Extending analogical generalization with near-misses. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin, Texas.
9. McLure, M.D., Kandaswamy, S. and Forbus, K.D. (2015b). Finding textures in sketches using planar Ising models. *Proceedings of the 28th International Workshop on Qualitative Reasoning (QR2015)*. Minneapolis, MN.
10. Hinrichs, T., & Forbus, K. (2015), Qualitative models for strategic planning. *Proceeding of the 3rd Annual Conference on Advances in Cognitive Systems*.

Manuscripts not yet published

1. Forbus, K. (in preparation). *Qualitative Representations: How People Reason and Learn about the Continuous World*. Book in progress, expected to go to publisher in 2016.
2. Forbus, K., Ferguson, R., Lovett, A., & Gentner, D. (in press). Extending SME to handle large-scale cognitive modeling. *Cognitive Science*.
3. Forbus, K., McFate, C., & Hinrichs, T. (in preparation). Narrative function for extracting knowledge in learning by reading.

Significant Collaborations

- Google is funding internal experiments to use a version of SME with one of their natural language systems, to see if a hybrid symbolic/statistical system can provide new capabilities for language processing tasks.
- IBM's T.J. Watson research center is discussing with us how to incorporate both analogical processing and qualitative models in their next-generation Watson research.
- Our lab is a founding academic partner of the Microsoft Research psi consortium, an as-yet unannounced effort to develop open-source tools for new natural interaction technologies.