Qualitative Reasoning for Decision-Making: A Preliminary Report

Kenneth D. Forbus & Thomas Hinrichs
Qualitative Reasoning Group, Northwestern University
2133 Sheridan Road, Evanston, IL, 60201, USA
forbus@northwestern.edu

Abstract

Agents often have to make decisions with incomplete knowledge and few computational resources. We argue that qualitative representations and reasoning, especially combined with analogy, provide a natural approach to performing decision-making in situations with little data, incomplete models, and under tight computational constraints. Moreover, qualitative models provide a means of recognizing and framing decision problems. This paper describes our progress in exploring these ideas to date, using examples from experiments with a system that learns to play Freeciv, an open-source strategy game.

1 Introduction

Any agent operating in a world must make decisions. Many decisions are immediate, e.g. which way to turn during navigation or searching a physical environment. Reinforcement learning (Sutton & Barto, 2017) has often been used as a model for how to make such decisions. The tasks to be performed are fixed outside the learning mechanism, and the appropriate notion of state has been settled by the agent's designer (or evolution). Many other decisions require more analysis, such as how to optimize a supply chain, and the mathematical tools of decision theory are often brought to bear on these. Such tools require accurate mathematical models of the system to be optimized and accurate probabilities to handle uncertainty. Yet other kinds of decisions involve design: How should an efficient transportation network be built economically, or how much should be invested in what kinds of military units to provide an effective defense? These decisions involve organizational policies and values more directly, as well as having components of optimization, but with much less agreement on what quantitative models should be used, if any. What is common about all three kinds of decisions is that the formalisms used to address them do not incorporate the problem of framing the decision problems themselves. What should an agent pay attention to, what choices does it have to make, and how should it evaluate its progress? We believe that qualitative representations and reasoning can be used to capture aspects of decision-making that tend to be left implicit, just as they have been used to capture tacit knowledge in science and engineering. There has already been productive work on using qualitative representation in decision-support systems to be used by people (e.g. Agell et al. 2006; Benaroch & Dhar, 1995; Rosello et al. 2010;Zitek et al. 2009), but our focus here is on using QR within autonomous agents. The goal of reinforcement learning is to make good local decisions. The equivalent goal in qualitative decision-making is to make sensible decisions, and especially to avoid repeated blunders. We propose to do this by using analogy-based episodic memory, combined with qualitative representations, to detect problematic situations, build up models of their properties, and modify the agent's decision-making to avoid them in the future. The goal of traditional decision-theory is optimization. The equivalent goal in qualitative decision-making is to ensure that resource allocation is in alignment with the agent's priorites, as expresed by activations of goals. In keeping with our aim of formalizing more of the strategic thinking process itself, we use qualitative models to express strategies that are used to achieve goals (Hinrichs & Forbus, 2015). This includes synthesis goals, where setting up the means of production and handling investments is part of the problem.

This paper summarizes our work to date on using qualitative reasoning in decision-making. Most of this work has been done in a strategy game domain, Freeciv, so we start by briefly reviewing it. Then we discuss the use of qualitative representations to encode strategies, followed by a discussion of qualitative reasoning about resources. The problem of enabling agents to formulate their own evaluation metrics is discussed next, which leads naturally into how an agent can design systems in a domain that serve its strategic goals. Learning from experience via episodic memory is discussed next. We close with future work.

1.1 Freeciv in a nutshell

Freeciv[1] is a turn-based strategy game, played on a grid of tiles (see Figure 1). Players start with a few units – settlers can found cities, and workers can improve terrain, to make

[1] http://freeciv.org/

it more productive. Most of the map is unknown, and must be explored, by an explorer unit or a military unit. Unlike chess or go, new entities can be created by a player's cities. This includes new settlers, once the population grows enough, thereby fueling an expansion (and thus the need to find more terrain to build upon). Cities can be linked by

Figure 1: Freeciv

transportation networks, which again are constructed by the player using workers. Cities produce resources (food, production, luxury goods) that go into feeding citizens, expanding the population, and producing new units or buildings. Buildings enhance properties of a city: City walls improve its defensive capabilities, and a Library improves its science output. There are 40 types of city improvements and 51 types of units that can be built in Freeciv version 2.2.4, depending on which of 87 available technologies a player has achieved via research. Games are typically played on a 4,000 tile grid, and can last for hundreds of turns. Thus the sheer size contributes to the game's complexity.

There are several additional sources of complexity from the game's dynamics. There is finding good places to put cities, since some terrain has advantages, just like placing a city on a river or bay has advantages in real life. The kinds of units that can be built depends on what technologies the player has, which in turn depends on which technologies they choose to research. Some technologies directly allow the construction of new units: Catapults become possible once Mathematics is understood, for instance. Other technologies enhance properties of a civilization: Democracy, for example, enhances economic productivity but makes for citizen unrest if war is declared. Research investments trade off against production and food, leading to classic short-term spending versus long-term investment choices. And of course there are competing civilizations, with a simple diplomacy system and warfare, with units ranging from warriors to musketeers to nuclear weapons, depending on what technologies a civilization has gained (or stolen from others). Unlike some games, there can be serious disparities in technological advancement, based on a civilization's decisions: Archers trying to defend a city against tanks is a good lesson in the drawbacks of under-investing in research[2]. There are two ways to win: Either wipe out all other civilizations, or send the first starship to colonize Alpha Centari, which requires considerable research and economic prowess. Thus civilization-style games are extremely complex, much more so than chess or go.

Since Freeciv is open-source, and there is an active player community, quantaitive models are possible. Importantly, most players neither develop nor use them. Qualitative causal models of the game dynamics, combined with a sense of relative magnitudes and some spatial reasoning, suffice to play well in our experience. This makes it an excellent testbed for exploring qualitaitve reasoning in decision-making by autonomous agents.

Freeciv has been used by other AI researchers as well. Branavan et al. (2012) explored using Monte Carlo simulation and text analytics to construct a heuristic evaluation function. While it played well on a small subset of the game (smaller map, games ended at 75 turns), it required many trials to learn the game and used the game engine to do lookahead search while playing, a tactic which is not available for most domains. It also did not construct an inspectable model of causality in the domain, unlike our learned qualitative models. Ulam et al. (2008) investigated combining metareasoning and reinforcement learning for the subtask of city defense in Freeciv. While it uses model-based reasoning, the quantitative model it uses is constructed by hand, by contrast with our automatically learned qualitative models.

2 Strategic Planning as Qualitative Reasoning

We argue that continuous processes provide a representation for strategies (Hinrichs & Forbus, 2015). Consider the gap between a strategy, e.g. expand the cities in one's civilization, versus the actions actually available to carry out this strategy, e.g. build settlers, find terrain, send settlers out to establish new cities, and so on. An agent's individual actions are discrete, i.e. move one of it's units to a new tile (and thereby reveal the contents of adjacent squares, if not already revealed), build a city by using a settler (which is consumed in the process). Some actions are durative, e.g. irrigating a tile or building a road takes multiple turns. Formulating a crisp specific end goal would be very complex: A large continent can support a dozen cities and similar numbers of units. Exploration takes time, and so locations can only be planned as terrain becomes revealed. But in the meantime, other civilizations are building as well – there is a race for territory. So a strategy of doing a phase of data gathering followed by designing an optimal solution will be thwarted. Instead of this discrete, planning oriented model, we think instead of strategies as continuous processes that the agent implements by its choices of actions. Exploration is a process that increases the size of the pool of known tiles. Expansion is a continuous process that increases the size of the pool of a civilization's cities.

[2] We note that in Freeciv, research always succeeds and the benefits are accurately known in advance.

```
(isa Defending ModelFragmentType)
(genls Defending
      ProtectingSomething)
(participantType Defending
    protector-Agentive
    FreeCiv-MilitaryUnit)
(participantType Defending
    objectProtected FreeCiv-Actor)
(associatedRoleList Defending
   (TheList protector-Agentive
            objectProtected))
(participantConstraint Defending
  (and (objectFoundInLocation
          protector-Agentive
          objectProtected)
       (different protector-Agentive
                  objectProtected)))
(consequenceOf-TypeType Defending
  (qprop-
    ((QPQuantityFn Vulnerability)
          objectProtected)
    (DefensiveStrengthFn
      protector-Agentive
      FreeCiv-MilitaryUnit)))
```

Figure 2: Defense as a model fragment

Actions can be planned and evaluated based on whether they will ultimately contribute to implementing the processes that represent the agent's current strategy. This approach supports incrementality, an important property for dynamic worlds. The last few cities built, for instance, are typically created by settlers who were built in cities that did not even exist at the start of the game. Constructing extremely detailed long-range plans makes little sense in an adversarial situation, when units or terrain that are assumed turn out to no longer be available[3].

A concrete example will make this clearer. Consider the concept of defense. Defense isn't an action: Attacking an attacker is an action taken in the course of defending a city (or the unit itself), but is not the same thing. Defense isn't a state to be achieved, it is more about ensuing that an undesirable state (i.e. destruction or conquest) is prevented. Figure 2 illustrates a qualitaitve model of using a unit to defend a city or another unit. All concepts not in QP theory (Forbus, 1984) are from the OpenCyc ontology or our extensions of it for Freeciv. The key point is that the vulnerability of the object protected is reduced by the defensive strength of the protector. By expressing the defense of a civilization in terms of a sufficiently low vulnerability (discussed below), a limit point can be constructed for the process of adding defenses that adds defenses when the civilization becomes more vulnerable and stops building them when it estimates that it is sufficiently protected.

[3] As military commanders sometimes say, "The enemy has a vote."

3 Resources

A central concept in decision-making is the idea of resources. Some resources are the inputs to production or carrying out events: In Freeciv, there are several such resources. Gold provides a notion of money. Light bulbs (i.e. ideas) must be generated and accumulated to achieve a new technology. Shields are a unit of production, which is used in building new units or buildings in a city. Food is needed to keep a city alive, and when there is a surplus for a long enough period, the city's population grows. Cities produce these resources, based on where they are, how their citizens are put to work, tax rate settings, and what buildings have been created to improve a city. These trade off against each other, and the agent can exploit these tradeoffs in subtle ways. For example, cities along a hostile frontier might invest more in production, to create city walls and military units, while cities safely inside the civilization's borders might focus on research or economic advancement. These resources are also fungible: Gold can be spent to finish something a city is producing, e.g. city walls if there are barbarians approaching.

While these resources are represented in the game as integers, they can be effectively reasoned about as continuous quantities. Qualitative models describing the dynamics of such quanitites can be learned via demonstration, where an agent watches a human player (Hinrichs & Forbus, 2012) and by natural language instruction (McFate et al. 2014). These learning methods complement each other, and our agent uses a qualitative model of domain dynamics that combines knowledge learned by these methods. This model can be used to express overall goals of the game: Winning by military conquest is driving the number of enemy civilizations to zero, for instance.

One issue that arises with a complex set of goals is identifying tradeoffs. We automatically construct tradeoffs for type-level goals via a static analysis of the learned qualitative model (Hinrichs & Forbus, 2015). Goal tradeoffs can be characterized in terms of two dimensions: (1) Total versus Partial determines whether or not all instances of the goal must be adjusted in lockstep. For instance, how taxes are spent is determined at the level of the civilization, not individual cities, so that is a total tradeoff, whereas what is produced in cities can vary with the city. (2) Abrupt versus Progressive concerns whether the change in goals is instantaneous or can be gradually changed over time. Setting a tax rate is an abrupt action, whereas reducing emphasis on producing new settlers as a continent is filling is a progressive change in the relative priority of goals. These distinctions are independent, and hence there are different strategies for each of the four possible cases.

The nature of a *constructive* domain is that resources can be used to build new means of production (e.g. cities in Freeciv) or improve existing means (e.g. build a Factory in a city in Freeciv). While such resources are discrete, we find it useful to express goals about them in terms of continous properties. The cardinality of sets of some type of entity, such as number of cities, is a useful measure of progress in expansion. Sums across a civilization are another type of useful quantity for decision-making, e.g. overall research capacity, military strength, which can be defined by using a compositional sum (C+, from QP theory) over the appropriate types of entities (Hinrichs & Forbus, 2013).

Two other important resources that hold for almost any domain are space and time. In Freeciv, like today's planet, cities can only be built on land. For each new game, a new map is randomly generated. If a player is lucky enough to start on a large continent, they can focus on expansion and technologies for land-based units, leaving seafaring technologies for later, when their civilization is more advanced. If their continent is small (or even an island of a single tile), then their research priorities should instead focus on seafaring. Making this tradeoff requires taking information from exploration into account. Civilizations can span multiple continents, but this involves building sea units to transport other units (e.g. settlers, military units for protection) and coordinating such transportation. A landlocked civilization on a small continent is in a dismal place indeed, and may need to resort to warfare to expand. Such a strategy would involve first shifting production to military units, and then shifting back to settlers to grab new territory (and defend conquered cities). Thus the relative value of resources can shift drastically depending on the nature of the environment.

Time is perhaps the most subtle of resources. Adversarial domains often involve some sort of race, so the effective use of time becomes important. A qualitative model that stratifies durations of actions can be surprisingly useful in planning. Consider a city which is under threat by an enemy unit. It could switch production from what it is currently building, to create either city walls or a warrior (in the early game). Or a military unit can be moved from a neighboring city could be moved in to protect the city under threat. Switching production has a cost – which is moot if the city is conquored or destroyed, naturally – so avoiding that if possible would be good. Is the neighboring unit sufficiently close that it can make it in time? This depends in part on what transportion networks are available to the two units, their relative distance to the city, and how fast they can move (i.e. how many movement points per turn). In the early game, production is sufficiently slow that producing a defender in response to a perceived enemy threat is usually too late. This temporal consideration suggests a strategy of pre-positioning defenses and defenders before they are needed. Note that units and city improvements have upkeep costs, so this strategy (like all strategies) is not without drawbacks. Trying out alternate strategies, and keeping track of how well they succeed or fail, is a way of gathering data about the distribution of these relative intervals in a way that is directly relevant for decision-making. We plan to explore this by using analogical generalization (McLure et al. 2015), setting up a strategy, executing on it, and recording what happened afterwards, to learn which strategies work. One subtlety with dynamic worlds, of course, is that things change – in late-game strong civilizations, new technological advances may take only a turn or two, so researching a new technology and then building a needed unit based on that becomes a more viable strategy, whereas it is a recipe for defeat in the early game.

4 Formulating Evaluation Metrics

One of the key tasks of an agent in making decisions is deciding how to evaluate its alternatives. Rather than assuming a built-in evaluation function (as reinforcement learning does) or learning an evaluation for a single task (as inverse reinforcement learning does[4]), we believe that agents should formulate their own evaluation functions based on broad world knowledge as well as experience. An agent can be making many decisions at once, affecting a large set of ongoing strategies. Since one of the jobs of qualitative reasoning is framing problems, we view constructing evaluation metrics as one of the important tasks of QR for decision-making. We take *evaluation dimensions* to be parameters that can be approximated as continuous parameters. Every resource described above can be treated as an evaluation dimension, using either a continuous perspective on an integer quantity (e.g. gold, light bulbs) or integer quantities defined across sets (e.g. cardinality, totals). We denote the cost of an action or plan by the logical function `CostFn`. This function has two arguments: The plan itself and an evaluation dimension. Thus each evaluation dimension potentially provides a different way to look at the cost of an action or plan, and thereby enables tradeoffs to be explored. For instance, a plan ?p1 to reinforce a city under threat by moving a defender to it would have, as part of the constraints on any plan involving motion,

```
(qprop+ (CostFn ?p1 Time)
        (TravelTimeFn ?p1))
```

By contrast, a plan ?p2 to buy city walls in a city ?c would incur a cost in gold, which depends on how much effort had already been invested in building them:

```
(qprop+ (CostFn ?p2 Gold)
        (- (ProductionCostFn CityWalls
                                 Shields)
           (ProductionSoFarFn CityWalls
                                  ?c)))
```

The underlying game engine provides numerical values for some of these parameters (e.g. Shields) but not others (e.g. travel time). Such parameters are used by our systems to learn qualitative models from experimentation (e.g. Hinrichs & Forbus, 2007), but we do not use them for constructing

[4] We note that inverse reinforcement learning assumes that the expert traces it is observing are optimal – something that is not consistent with human decision making in complex domains (Kahaneman, 2011).

exact quantitative cost functions, because for constructive adversarial games in general, accurate mathematical models of the underlying domain are not available. Instead, we use experimentation to learn decision trees based on accumulating information from direct measurements. For example, a city built on grassland with wheat is much more productive than a city built on a desert. A learned decision tree that evaluates locations for city placement is used in the planning process for selecting expansion sites. Another method we plan to explore is using learned estimates of ordinal relationships to decide among alternatives. A rough estimate of the arrival time of the enemy would be enough to determine what alternatives (if any) are actually feasible.

5 Strategic Design

Some decisions are about how to build up new entities and systems to serve an agent's goals. In Freeciv, for example, building up a civilization entails creating a number of cities (as many as a dozen or more), improving the terrain around them, and linking them with roads (or railroads, once that technology is discovered). Such systems can be partially characterized by parameters whose settings should be learned by the agent, based on experience.

For example, how far apart should cities be? Claiming territory is useful, since it provides a buffer against enemies and gives an agent's civilization room to grow. On the other hand, unless each city has capable defenses, sending defenders to reinforce a city becomes more expensive. That suggests making the mean distance between cities smaller rather than larger. This consideration can be expressed qualitatively as follows: Consider ?p to be a generic plan involving travel between two cities, which can be approximated by the mean travel time:

```
(c+ (CostFn ?p Time) (TravelTimeFn ?p))
(qprop+ (TravelTimeFn ?p)
        (MeanCityDistFn ?civ))
```

implies

```
(qprop+ (CostFn ?p Time)
        (MeanCityDistFn ?civ))
```

On the other hand, trade routes are more valuable when two cities are far apart – if ?p is establishing a trade route between two cities, then

```
(qprop+ (ValueFn ?p Trade)
        (DistanceFn ?city1 ?city2))
```

Then taken across the entire civilization,

```
(qprop+ (ValueFn ?civ Trade)
        (MeanCityDistFn ?civ))
```

How do these qualitative models help in decision-making? They tell an agent about what relative likelihoods it needs to estimate. If warfare is likely to be common and trade is less important, keeping cities tightly clustered would be a better strategy. If bolstering trade is more important, then larger spacing might be a reasonable strategy. Building up models of what is likely in a game, via analogical generalization over episodic memories, could provide a way to estimate such likelihoods.

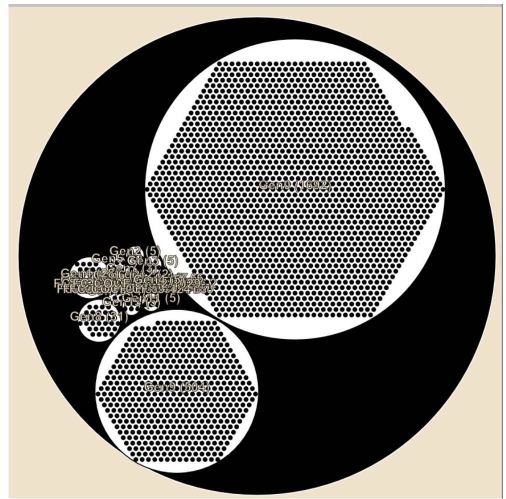

Figure 3: The generalization pool for doMove. The white circles are generalizations, the black dots within them represent particular experiences used in building that generalization.

6 Analogical Learning from Experience

Incomplete and incorrect models are the norm for agents operating in complex domains, especially when adversaries are involved, because other agents are often less predicable than domain physics. For example, a city can be weakened without being directly attacked by placing enemy units on the tiles immediately surrounding it, which prevents them from being worked and can cause starvation – an emergent behavior which is effectively a siege. A naïve agent can make suboptimal decisions, such as leaving military units out in the middle of nowhere, neither providing early warnings of approaching enemies nor defending anything. Trying to build settlers before a city has size 2 is impossible, because one citizen goes into the construction of a settler. All of these are things that human players figure out by watching their own behavior and learning how to improve it. We propose that analogical generaliztion over episodic memories provides a distillation of experience that can be used for such learning. This provides the rapid retrieval of either something to do, or something to avoid, a mechanism for the kind of human decision making that Kline describes in his recognition-primed decision model (Kline 1999).

For example, consider learning the immediate effects of actions. Most actions are fairly boring, either there is the same kind of change (i.e. changing production or research changes what a city is producing or the civilization is researching) or nothing happens for a while, if it is a durative action (e.g. irrigation has no immediate effect except for the worker no longer being idle). Movement is typically similarly boring, with one exception: Entering a hut. Huts on tiles can lead to multiple outcomes – gold or a new technology might be found, the unit might be killed by the inhabitants, or a new unit or city might be added to the civilization that made contact. Figure 3 illustrates a SAGE generalization pool for the primitive action doMove. Generalization pools accumulate examples of a concept incrementally, merging them into generalizations when they are suffiicently similar. Here the largest two generalizations are the typical outcomes of movement, with the different outcomes of entering a hut corresponding to smaller generalizations. Since SAGE constructs probabilities for each of the statements in every generalization pool, based on experience, the agent can compile a table of probabilities for the outcome of entering a hut (Table 1). Such experience-based probabilities are very useful for decision-making: Entering a hut can be seen here to be a good idea, overall, although given the chance of the unit being wiped out, diverting a settler on its way to found a new city to enter a hut is probably unwise.

Outcome of entering a hut	P
Gold Found	0.38
Technology Found	0.23
Unit joins your civ	0.23
City joins your civ	0.08
Killed by Barbarians	0.08

Table 1: Probability of outcomes for entering a hut, as calculated from SAGE's summaries of experience

How should a system know to build such a table? We have formulated a metric for surprise based on novelty concerning an experienced concept. That is, given an example E of a command C, whose analogical model consists of gpool(C), we define the novelty of E with respect to gpool(C) as

$$1\text{-NSIM}_B(\text{BestMapping}(\text{SME}(E,\text{MACFAC}(E,\text{gpool}(C)))))$$

That is, the base-normalized similarity score of the best mapping for the closest item retrieved from the generalization pool. If there is a case in the generalization pool that is identical to E (isomorphic up to entity renaming), then the numerical similarity will be 1, and E will have zero novelty. If nothing is retrieved, the numerical similarity is taken to be zero, and hence the novelty of E would be at its maximum, 1.0.

Not all novelty matters. SAGE provides a natural definition for novelty, since that can be taken as the dual of the decision that a new example is close enough for assiilation. That is, every generalization pool has an *assimilation threshold* A_t that ranges from 0 to 1. To respect this threshold, if an example E would be assimilated under the current threshold, then the novelty will be zero. The other factor which must be taken into account is how much experience the system has with the concept. We incorporate this factor by taking the product of the novelty with the following rate equation:

$$1-e^{-n/r}$$

Where n is the number of examples that have been added to the generalization pool so far, and r is a rate parameter, controlling how fast this asymptotes to 1. In the case of the doMove action, each time a new kind of outcome occurs when a hut is entered it signals a surprise, which can enable a system to keep track of that subset of actions as interesting.

Immediate effects of actions are just one kind of experience that should be routinely stored for subsequent analysis by an agent. Building up a model of the time that durative actions take can be done by taking before/after snapshots of the locale where such an action is taking place, and including the duration as part of the episodic memory. Statistics over those durations can then provide a robust way of estimating time costs for actions. In general, when decisions are made about an aspect of a domain that is not well understood, constructing episodic memories that capture what happened and how successful it was can be useful (e.g. worker assignments in Hinrichs & Forbus, 2007).

We see two other important functions of episodic memory. In adversarial domains, it is important to learn from what is being done to you, as well as what you do. Qualitative representaitons help lift descriptions to a level that is easier to compare, and hence to learn from. For example, the approach of an enemy unit to a city can be described as one interval using the Qualitative Trajectory Calculus (Van de Weghe et al. 2005) along with the duration of that activity, factoring out the specifics of the tiles traversed. Similarly, recognizing that a unit was lost because it was attacked by another unit is a very simple form of persective-taking. The other function of episodic memory is helping to set strategic parameters, e.g. what should the relative priorities of goals be, and what should limit points for strategic processes be? This, we suspect, is best done via a retrospective analysis of longer periods of play, abstracting out the specific events into statistics about global properties. For example, if a game was lost because an agent's cities were wiped out, then one potential solution is to increase the sensitivity to vulnerability, so that it pre-positions more defense resources and is more careful in future games.

7 Conclusions and Future Work

We believe that qualitative representations and reasoning can provide valuable services in formalizing the decision-making of agents in complex, dynamic adversarial worlds. The techniques outlined here complement traditional decision theory and reinforcement learning, since they are concerned with framing and formulating decision problems and using qualitative, causal models for both understanding the broad properties of domain dynamics and to express strategic concepts.

We plan to continue exploring these ideas in several ways. First, we plan to implement the other forms of episodic memory as outlined above, and explore their properties. Second, we plan to implement a reasoner that can formulate and articulate decision problems, criteria, and alternatives in a domain, so that agents can participate in joint problem solving involving strategic problems and learn more from natural langauge instruction, beyond the advice and domain-level causal models we have used language-based instruction for previously. Finally, we plan to explore cross-domain transfer: Tell a Companion stories about our world and ask what they imply about its strategies, and vice versa. Understanding when things will work in both (e.g. blockades) and when they won't (e.g. airlifts work in our world but not in Freeciv) is an important test of strategic thinking and transfer.

Acknowledgments

This research was supported by the Air Force Office of Scientific Research.

References

[Agell et al. 2006] Agell, N., Rovira, X., Sanchez, M., and Prats, F. (2006). Using Orders of Magnitude in Multi-Attribute Decision-Making. *Proceedings of QR 2006*.

[Benaroch & Dhar, 1995] Benaroch, M., & Dhar, V. (1995). Controlling the complexity of investment decisions using qualitative reasoning techniques. *Decision Support Systems*, 15(2):115-131.

[Branavan et al. 2012] Branavan, S., Silver, D., & Barzilay, R. (2012). Learning to win by reading manuals in a Monte-Carlo framework. *Journal of Artificial Intelligence Research*, 43:661-704.

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

[Forbus & Hinrichs, 2018] Forbus, K.D. & Hinrichs, T. (2018). Analogy and Relational Representations in the Companion Cognitive Architecture. *AI Magazine*.

[Hinrichs & Forbus, 2012] Hinrichs, T. & Forbus, K. (2012). Learning Qualitative Models by Demonstration*,* *Proceedings of AAAI 2012*, pp.207-213.

[Hinrichs & Forbus, 2013] Hinrichs, T., & Forbus, K. (2013). Beyond the Rational Player: Amortizing Type-Level Goal Hierarchies.

[Hinrichs & Forbus, 2015] Hinrichs, T., and Forbus, K. (2015). Qualitative models for strategic planning. *Proceedings of the 3rd Annual Conference on Advances in Cognitive Systems*.

[Kahneman, 2011] Kahneman, D. (2011) *Thinking Fast and Slow*. Macmillan.

[Klein, 1999] Klein, G. (1999) *Sources of Power: How People Make Decisions*. MIT Press: Cambridge, MA.

[McFate et al. 2014] McFate, C., Forbus, K., & Hinrichs, T. (2014). Using Narrative Function to Extract Qualitative Information from Natural Language Texts. *Proceedings of AAAI 2014*.

[McLure et al. 2015] McLure, M., Friedman S. and Forbus, K. (2015). Extending Analogical Generalization with Near-Misses. *Proceedings of AAAI 2015.*

[Rosello et al. 2010] Rosello, L., Prats, F., Agell, N., & Sanchez, M. (2010). Measuring consensus in group decisions by means of qualitative reasoning. *International Journal of Approximate Reasoning*, 51:441-452.

[Sutton & Barto, 2017] Sutton, R. & Barto, A. (2017) *Reinforcement Learning: An Introduction*. 2nd Edition, MIT Press.

[Ulam et al. 2008] Ulam, P., Jones, J., & Goel, A. (2008) Combining Model-Based Meta-Reasoning and Reinforcement Learning for Adapting Game Playing Agents. *Proceedings of AIIDE 2008*.

[Van de Weghe et al. 2005] Van de Weghe, N., Cohn, A., De Tre, B., & De Maeyer, P. (2005) A Qualitative Trajectory Calculus as a basis for representing moving objects in Geographical Information Systems. *Control and Cybernetics*, 35(1):97-120.

[Zitek et al. 2009] Zitek, A., Schmutz, S., Preis, S., Salles, P., Bredeweg, B, & Muhar, S. (2009). Evaluating the potential of qualitative reasoning models to contribute to sustainable catchment management. *Ecological Informatics*, 4:381-395.