How Experts Explain Strategic Behavior During Real-Time Strategy Games

Christoph Neumann, Aaron Schnabel, Jonathan Dodge, Ronald A. Metoyer, Simone Stumpf
Department of Electrical Engineering and Computer Science
Oregon State University
Corvallis, Oregon 97331
{neumann,schnabel,dodge,metoyer,stumpf}@eecs.oregonstate.edu

Real-time strategy games, such as *Wargus*, are examples of complex learning and planning domains that present unique challenges to AI and machine learning. These games usually comprise a large number of states, actions, resources, and decisions that a player needs to take into consideration, while, at the same time, the current game situation is influenced and modified by opponents. With the drive to acquire planning knowledge from ever fewer examples, learning and planning in this complex, dynamic environment is even more challenging.

Some headway could be made by providing notations in which an expert can annotate examples to help derive additional knowledge. Previous research in the machine learning and artificial intelligence communities has focused mainly on identifying a language for representing problems and solutions in real-time strategy games (Fern, Yoon, & Givan 2004; Geffner 2004; Khardon 1999; Ponsen & Spronck 2004; W.Aha, Molineaux, & Ponsen 2005). Rather than finding a representation most suitable for *machines*, we are interested in finding a language or representation most suitable for *people* to state their intended behavior.

Describing behavior can be problematic if there is a significant mismatch between the notation and the user's conceptualization of their behavior. We seek a natural way for users to express strategic behavior in game-like worlds. To that end, we conducted a formative study with expert players of a real-time strategy game to determine the structure of the language used by the experts to describe strategy.

In this paper, we describe a user-centered approach for capturing and analyzing the language of players of the real-time strategy game *Wargus*, the open source reimplementation of the popular *Warcraft II* (Blizzard 1995). Our formative study followed a think-aloud design, in which we paired five experts with five novices to help elicit explanations of strategies. The experts were asked to play the game while "thinking aloud" about what they were doing. The novices were instructed to ask the experts about any detail they did not understand. Each expert game-playing session lasted approximately 35 minutes, during which two games were played by the expert. We recorded a video of the game play as well as the interaction between the expert and novice. Afterward, all participants were asked to fill out a post-session questionnaire based on NASA TLX questions (Hart & Staveland 1988).

In order to understand and explain the participants' behavior, we analyzed their verbal interactions with a coding scheme we developed from the bottom up using a grounded theory approach (Corbin & Strauss 1990; Glaser 1967). We validated our coding scheme with calculation of agreement measures and then applied the codes to segmented transcripts. Statement segments were determined by breaking sentences at coordinating conjunctions (and, or) and subordinating conjunctions (because, since) but leaving correlative conjunctions (both. . .and, either. . .or, if. . .then) intact. We achieved 83.4% pairwise raw agreement between three raters (with discussion) over a set of 306 statments drawn from random blocks over all the transcripts. The code frequencies appear in Table 1. To our knowledge, we present the first attempt at a coding scheme for real-time strategy games, which may be useful to other researchers in order to investigate these domains.

The following are the final codes and their definitions:

Fact: A statement of general or situational truth, or an opinion about such a truth.

Dependency: A statement that reflects a constraint, or lack of a constraint, on a course of action.

Do: A statement giving prescriptive instructions on how to behave–often with reference to some game action, such as "build," "move," "train," or "attack."

Goal: A non-prescriptive statement of intent, purpose, desired achievement, or focus.

Mistake: A statement that criticizes what has already been done.

History: A statement that describes something that has already occurred in the game, but *without* making any value judgment about it.

Question: A statement which poses a question about the game-playing.

UI: Any statement about how to interact with the software itself ("clicking," automation devices, etc), but not *Facts* deduced from the interaction.

Extraneous: A statement that is uninterpretable or does not contain any useful information.

For consistent code application, we considered each code in the following order for each statement: *Extraneous*, *Question*, *UI*, *Mistake*, *History*, *Depend*, *Do* or *Goal*, *Fact*.

Table 1: Code frequency over the full set of transcripts (as a percent.)

	Fact	Depend	Do	Goal	History	Mistake	Question	UI	Extran
Code Frequency	33.2	12.0	12.1	7.9	7.6	1.9	9.2	7.4	8.5

Strategies are high-level plans that a user follows in order to win the game. Strategies constrain and motivate the way a player sequences actions in the game and manipulates the game elements. *Goals* were a high-level way experts could express strategy, but to our surprise, only 7.9% of all statements were *Goals*. However, 12.1% of statements were *Do* statements. It appears that experts preferred to explain their strategy by using a finer granularity, in which they made detailed reference to what to do in the context of the game.

Do and *Goal* appear to represent a spectrum of detail in which to explain strategy, that, when combined, make up $20.0 \pm 2.5\%$ of statements regardless of transcript. However, experts exhibited a preference for choosing a certain level of granularity in which to express the strategy. Experts that chose high-level strategy explanations tended to provide fewer detailed, fine-grained statements, and vice versa.

A reasonable but naive assumption would be that plans and strategies are stated at the beginning and then enacted in the remainder of a game. In our study, strategy explanations (*Do* or *Goal* statements) were found interspersed throughout both games–even for the second game in a session, in which participants could have omitted strategy explanations since they had already been covered previously.

Facts and *Depends* are found constantly throughout all games in all transcripts and comprise a total of 45.2% of all statements. *Facts* appear to draw attention to certain things in the game that are important and matter. *Depends* appear to draw out constraints that need to be met. Therefore, *Facts* and *Depends* provide the constraining context information in which strategies are enacted. Furthermore, continual modification and clarification of the context appears to be needed throughout the entire game.

Questions provide an *explicit* indication that an insufficient explanation was provided and more details are needed. Thus, we paid particular attention to questions that novices asked experts, since they indicate a gap in the novice's understanding. Similar frequencies (8.3%-13.0%) of *Questions* were recorded after every code, indicating gaps in all kinds of information, except *Do* prompted frequent questions (20.3%) and *Mistake* prompted very few (3.3%.) So, it appears that *Do*'s tended to require more detail.

Interestingly, when asked for more detail, experts did not provide answers in terms of strategy–*Goal* and *Do* answers were infrequent (1.9% and 6.5% respectively.) In contrast, 44.9% of statements that followed *Questions* were *Facts* and 13.1% were *Depends*. As discussed previously, *Facts* and *Depends* explain the context in which the strategy is situated. It appears that answers often focused on the applicability and constraints to consider when applying the strategy.

Mistake statements were infrequent (1.9%) and tended to appear when the expert was losing. 7.6% of statements were *History* statements, but 43.2% of all *History* codes are concentrated right after a game. Within the game, *History* statements seemed to draw out important aspects of the situation that needed to be attended to at the time–sometimes this was related to a mistake, but sometimes as part of a rationalization for the present course of action.

Conclusion

In designing an annotation language for strategy, one should consider the following. Experts exhibited preference for different levels of granularity in their explanations. Furthermore, experts typically did not state strategy explicitly, but provided it through a significant amount of prescriptive and intentional explanations throughout the game. Within the context of a game, experts tend to prefer to state what to do in concrete game terms rather than high level terms. Additionally, experts used statements of fact and dependency to provide an applicable and constraining context for their explanations. Finally, experts require a means for pointing out substantiation or errors of strategy or action already expressed or demonstrated.

References

Blizzard. 1995. Warcraft II:Tides of Darkness. Blizzard Entertainment. http://www.battle.net/war2/, Last accessed March 2007.

Corbin, J., and Strauss, A. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology* 13(1):3–21.

Fern, A.; Yoon, S.; and Givan, R. 2004. Approximate policy iteration with a policy language bias. *Advances in Neural Information Processing Systems* 16(3):847–854.

Geffner, H. 2004. Planning graphs and knowledge compilation. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*.

Glaser, B. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter.

Hart, S. G., and Staveland, L. E. 1988. Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In Hancock, P. A., and Meshkati, N., eds., *Human mental workload*. Amsterdam, The Netherlands: Elsevier. 139–183.

Khardon, R. 1999. Learning action strategies for planning domains. *Artificial Intelligence* 113(1-2):125–148.

Ponsen, M., and Spronck, P. 2004. Improving adaptive game ai with evolutionary learning. Master's thesis, Delft University of Technology.

W.Aha, D.; Molineaux, M.; and Ponsen, M. 2005. Learning to win: Case-based plan selection in a real-time strategy game. *Case-Based Reasoning Research and Development* 5–20.